Estimating and Generating Human Motions from Interactions

Jinkun Cao

CMU-RI-TR-25-89

July 15, 2025

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Kris Kitani, *chair*
Deva Ramanan
Shubham Tulsiani
Siyu Tang, *ETH Zurich*

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Robotics.

Dedicated to my family and friends.

Abstract

Modeling human motion is a fundamental topic in computer vision and robotics. Humans interact with the 3D physical world in complex ways, involving both changes in position (global motion) and body deformation and articulation (local motion). This thesis explores human motion in interactions with other humans, environments, and manipulated objects. We focus on the tasks of estimating and generating human motions, emphasizing the integration of diverse knowledge sources such as video, motion capture, and physics simulation.

We begin by examining human-human interactions. Using widely available video data, we study implicit interactions where individuals navigate toward goals while avoiding collisions. Initially, we address multi-object tracking and then progress to trajectory generation, exploring both estimation and generation perspectives. For tracking, we start with learning-based methods and revisit classic parametric filtering. To generate socially aware trajectories, we combine parametric priors with generative models to leverage inductive biases from data.

The second part of the thesis investigates human-scene interactions. As people frequently bend and articulate their bodies for daily tasks, we examine both local and global body motion. We utilize motion capture data to ensure visual realism in motion generation and employ physics simulation to enforce physical realism. We begin by validating the use of physics-based imitators for diverse motions. Subsequently, we place a human agent in a static scene and develop a reinforcement learning policy to generate physically grounded interactions guided by language instructions.

In the third part, we extend our study to human motion during object manipulation in dynamic environments. Due to limited human-object motion capture data, we focus on generating static hand-object grasps that generalize to a wide range of object shapes using large-scale object shape datasets. These grasps then guide a reinforcement learning policy that enables full-body motion for transporting an object in hand within a simulation.

Building on insights from earlier chapters, we observe the effectiveness and flexibility of generative models for both motion estimation and generation. This motivates us to explore a unified model. We propose a diffusion model for human motion, where conditioning the denoising process allows the model to perform estimation as well. When conditioned on video, the model achieves motion estimation performance comparable to specialized estimation models.

Acknowledgments

Throughout my Ph.D. journey, I have been profoundly grateful for the unwavering support of my advisor, collaborators, family, and friends. Their encouragement, love, and wisdom have not only made this journey a joyous one but have also been instrumental in the completion of this thesis.

First and foremost, I wish to extend my heartfelt gratitude to my advisor, Kris Kitani. Over the years, Kris has been a pillar of patience and support. His passion, guidance, and meticulous attention to detail have been pivotal in my academic and personal growth. Under his insightful mentorship, I have refined my research and presentation skills, learning invaluable lessons in both academia and life. Kris is the best advisor that I could have imagined,

I am also deeply appreciative of my thesis committee members: Deva Ramanan, Shubham Tulsiani, and Siyu Tang. Their invaluable insights and advice have greatly enriched my thesis and research endeavors. Deva's pioneering work in computer vision has been a constant source of inspiration. Shubham's exceptional contributions to 3D computer vision and hand-related studies have continually sparked new ideas in my own research. I am grateful to Siyu for her willingness to join my committee; her groundbreaking work has significantly influenced my studies on human motion and interaction modeling. Discussion with them has always inspired my research and improved the thesis.

I owe a debt of gratitude to the mentors who have guided me from the inception of my journey in computer vision. Cewu Lu ignited my passion for this field by introducing me to my first computer vision paper and online course. Xiaoyong Shen and Yu-Wing Tai provided exemplary mentorship during my internship at Tencent, where I achieved the milestone of publishing my first first-authored academic paper. Trevor Darrell and Fisher Yu afforded me the invaluable opportunity to observe the workings of top-tier research laboratories worldwide. Yang Gao, who hosted me before my Ph.D. journey began, has been both a mentor and a cherished friend along the years.

My friends and peers at CMU have been a constant source of support and inspiration. Zhengyi Luo is the best friend and collaborator I could have imagined and I can't believe how much I have learned from him while sharing the office over the years and in every project we collaborated on. Xinshuo Weng and Rawal Khirodkar were instrumental in launching my research at CMU, and I have greatly benefited from their expertise. Yuda Song has expanded my knowledge in unfamiliar areas, and our discussions have always been a delight on basically everything. I am also grateful for the support and camaraderie of Yu-Jhe Li, David Park, Shun Iwase, Erica Weng, Jenny Nan, Rohan Choudhury, Navyata Sanghvi, Ryosuke Hori, Jyun-Ting Song, and many other peers at CMU.

I have been fortunate to collaborate with some of the best collaborators outside CMU: Hao-Shu Fang, Jiefeng Li, Hao Wu, Jiangmiao Pang, Runsen Xu, Zeqi Xiao, Tai Wang, Jingbo Wang, Jingyuan Liu, Haotian Zhang, Davis Rempe, Yuke Zhu, and many others. I learned a lot from my internships at Meta, Adobe and Nvidia. My mentors, Zhe Cao, Shih-En Wei, Weipeng Xu, Yi Zhou, Ye Yuan, and Umar Iqbal, have taught me so much that I have been benefiting.

Ph.D. journey would have been far less vibrant without the companionship of my friends outside my academic life. Among the friends I have shared so many happy memories with, I would like especially extend my thanks to Jinhao Zhu, Muyang Li, Haoyue Dai, and Chen Bao, with whom I have shared a home at 1209 Shady Ave over the years.

Lastly, I am profoundly grateful to have Wen Shao by my side, offering unwavering support whenever needed. I am deeply thankful to my parents, Weilin Lai and Jingming Cao, whose unconditional love and support have been the foundation of my life. Without them, I would not be the person I am today.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

From autonomous driving and surveillance to motion capture and robotics, many tasks in computer vision and robotics rely on accurate modeling of human motion. Typically, human motion can be divided into two categories: (1) *Global motion*, which refers to changes in a person's position (translation) within a given environment; and (2) *Local motion*, which involves the articulation and deformation of the human body as a non-rigid structure.

Research in human motion modeling generally focuses on one or both of the following aspects: (1) *Motion estimation*, which involves recovering a human's global position or local body movements from sensory data such as videos or point clouds; and (2) *Motion generation*, which aims to synthesize global or local motion sequences that appear both visually natural and physically plausible.

Previous work often isolates these aspects. For example, in motion estimation, multi-object tracking and person re-identification concentrate on tracking a person's position over time using keypoints or bounding boxes, without modeling body deformation. In contrast, video-based human pose estimation captures local motion by predicting skeletal or surface representations over time. On the generation side, trajectory prediction usually focuses on global motion, while imitation learning and motion synthesis also consider detailed joint movements and body surface deformations.

In this thesis, we address a variety of tasks across this broader spectrum and explore how shared representations and learned knowledge can connect previously disjoint areas of human motion understanding and generation. We tackle the problem of multi-object tracking from the vision side to estimate the human motion when implicitly interacting with other humans. Then, we use the generation tools for the human motion patterns in interacting with humans, objects or environments. We extend the research from vision techniques on videos to motion capture data in 3D and further physics simulation for a more comprehensive study of the topic of estimating and generating human motions.

1.2 Challenges and Contributions

To investigate human motion patterns, we began with video-based motion estimation, focusing on multi-human tracking from monocular videos. At this stage, we set aside local body motion and concentrated on accurately estimating individuals' positions when multiple people are moving simultaneously, leading to occlusions and identity switches. This is a challenging task: when bounding boxes are used for localization, pixels from overlapping subjects can significantly degrade the performance of conventional appearance-based matching methods.

We first tackled this problem by designing new architectures for appearance feature extraction, processing, and matching. These improvements enhanced tracking performance in crowded scenes but still struggled when individuals had visually similar appearances. Motivated by this limitation, we turned to parametric motion modeling using filtering techniques. Building on the classical Kalman filter [173] and its application in multi-object tracking [28], we identified typical failure cases in crowded scenes and proposed a new filter-based model. This approach improved tracking robustness while maintaining computational speed and online inference capability.

Following this, we extended our research to trajectory generation. We developed a normalizing flow-based method that enables diverse and controllable trajectory synthesis. A key component of this approach is a data-driven model of motion intent: by fitting a mixture of Gaussians to training data that captures common motion goals (e.g., walking straight, turning), we enhanced the effectiveness and expressiveness of the flow model without compromising generalizability. The explicit modeling of motion intentions also allows for behavior-conditioned generation of plausible future trajectories.

Throughout the above work, we implicitly considered human-human interaction in the context of multi-human motion and intention modeling. In the next stage of the thesis, we began to model human-scene interaction explicitly—examining how the environment shapes and constrains possible motions. Here, we addressed both global position changes and local body articulation. While generative models trained on motion capture data can produce visually appealing motion, they often lack physical realism. To overcome this, we introduced physics-based constraints. By training a physically plausible human controller, we demonstrated the ability to generate a wide variety of motions that are both visually convincing and physically grounded. Our method achieved 100

Building on this progress, we explored human motion in more realistic indoor environments. We proposed a hybrid method that combines a large language model (LLM)-based planner with a reinforcement learning controller. This system enables the generation of seamless, physically valid human motion in simulated scenes based on natural language instructions.

In the third part of this thesis, we investigated human-object interaction. To generate whole-body motions that are both visually natural and physically plausible when interacting with objects, we began with the generation of realistic hand-object grasp configurations. By learning from large-scale 3D object datasets and a smaller hand grasp motion capture dataset, we generated plausible grasps across a wide variety of object shapes and sizes. These grasps serve as constraints for full-body motion synthesis. We then trained a human agent in a physics simulator to transport objects along predefined trajectories. By combining motion capture data, generative priors, and reinforcement learning with physical constraints, we developed a generalizable framework capable of handling a diverse set of objects with different geometries and physical properties.

Throughout the research described above, we followed the prevailing practice of separating motion estimation and motion generation. However, the flexibility and power of modern generative models revealed the opportunity to unify these tasks within a single framework. To that end, we developed a conditional diffusion model. Trained on both 2D and 3D motion sequences, this model leverages motion capture and video data. Moreover, by using video as an optional conditioning input, the model can generate motion that aligns closely with visual evidence—thus performing estimation as well. This unified design shows promise in bridging the gap between estimation and generation.

1.3 Thesis Overview

This thesis is divided into four parts. Each part addresses a different dimension of human motion modeling, and together they form an integrated framework that spans estimation, generation, and their unification.

1.3.1 Part I: Human Motion from Human-Human Interaction

DanceTrack: Multi-Object Tracking in Uniform Appearance In Chapter 2, we propose DanceTrack, a new video dataset for multi-object (multi-human) tracking. The dataset was motivated by the recognition of bias and limitations in existing multi-object tracking benchmarks. Our goal is to study tracking performance in scenarios where humans move in close proximity, wear similar clothing, and follow complex motion trajectories. We benchmark existing tracking methods on DanceTrack and observe that many of them fail significantly. This dataset reveals important gaps in previous multi-object tracking research and provides a platform for more challenging and realistic evaluation.

Appearance Matching for Multi-Human Tracking in Crowds With the rise of deep learning and modern pixel feature descriptors, we present two methods for appearance-based multi-object tracking in Chapter 3 and Chapter 4. In Chapter 3, we combine spatio-temporal coordinates with visual features to create a hybrid identification representation, which reduces the ambiguity of pure appearance-based matching, especially under occlusion or visual similarity. In Chapter 4, we develop a hierarchical, attention-based feature processing model. Compared to standard CNN or transformer-based visual backbones, our model enhances the distinguishability of target features while suppressing background noise and features from other targets. Both methods significantly improve the accuracy of appearance-based tracking on challenging benchmarks, including our DanceTrack dataset.

Parametric Linear Filtering for Multi-Human Tracking in Crowds Recognizing the inherent limitations of appearance-based methods, in Chapter 5, we revisit classical filtering techniques. We use the Kalman filter to track people in crowded videos based solely on their location. We enhance robustness against imperfect detections and occlusion by introducing a heuristic re-update mechanism that mitigates accumulated parameter errors when no detections are available. This re-update occurs only when a track is re-established after being temporarily lost. Importantly, our method remains fast and online. It has inspired follow-up work—such as [238]—that combines parametric filtering with appearance-based matching to advance the state of the art in this field.

Mixed Gaussian Prior for Human Trajectory Generation In Chapter 6, we transition from motion estimation to motion generation, specifically in the task of human trajectory prediction. We are inspired by the simple yet powerful observation that humans often move with specific intentions, such as walking straight or turning, and that trajectories with similar intent tend to form clusters. This insight provides an intuitive and effective inductive bias for learning more diverse and controllable generative models. Based on this idea, we construct a mixed Gaussian prior by clustering trajectories according to their intent. By replacing the commonly used unimodal Gaussian prior with this data-driven mixture prior in our normalizing flow model, we achieve notable improvements in motion diversity, interpretability, and controllability.

1.3.2 Part II: Human Motion from Human-Scene Interaction

Physics-Based Human Motion Imitation In Chapter 7, we address the challenges of modeling more complex interactions between humans and their environments. Unlike earlier settings, we now need to model both global motion and local body articulation and deformation. Physical plausibility becomes a major concern. Rather than relying solely on generative models trained on motion capture datasets, we integrate reinforcement learning within a physics simulator to ensure physically grounded motion. We train policies to imitate large-scale motion capture data, ensuring both realism and physical feasibility. Given the diversity and complexity of these datasets, we propose a multi-step imitation framework, using specialized submodules for different motion genres. Our method achieves 100

Language-Guided Human Motion Generation in Simulated Scenes After establishing a capable motion imitator, we proceed in Chapter 8 to generate motion in furnished indoor scenes with complex layouts. Compared to imitation in empty spaces, moving in a cluttered environment requires higher precision and clearer motion intent. To address this, we frame the problem in a language-guided setting. We describe human-scene interaction as a sequence of body-scene contact pairs—referred to as a chain-of-contacts. Given a language instruction, we use an LLM-based planner to decompose the instruction into step-by-step motion intentions. A reinforcement learning-trained controller then executes these steps by bending, moving, and articulating the human body to complete the contact sequence.

1.3.3 Part III: Human Motion from Human-Object Interaction

Static Hand-Object Grasp Generation In Chapter 9, we focus on generating plausible hand-object grasps. While generative models such as diffusion and VAEs have been applied to this task, their performance is often constrained by limited dataset scale and object diversity. To improve generalization, we develop a multi-modal diffusion model trained on large-scale object shape datasets like ShapeNet. Our model learns an inclusive and generalizable object latent representation. By unifying the object and hand pose latents into a shared space, we train a single model for both conditional and unconditional grasp generation. This approach improves performance across a broader range of object geometries.

Whole-Body Motion Generation in Physics with Object Interaction In Chapter 10, we extend from static grasp generation to whole-body motion for object interaction. While prior work provided us with a strong motion imitation model [233] and a rich latent representation [232], directly training a whole-body policy for object manipulation is inefficient due to the complexity of hand articulation. To resolve this, we use grasp poses from the previous model as pre-grasp guidance. This reduces the burden of hand articulation. We then introduce a set of task-specific rewards to guide reaching, grasping, and transporting objects. Our final method, OmniGrasp, enables a humanoid agent to perform physically realistic object manipulation along target trajectories.

1.3.4 Part IV: Unified Human Motion Estimation and Generation

In Chapter 11, we present a conditional diffusion framework for unifying human motion estimation and generation. Beyond combining these two tasks in one model, we also incorporate a variety of input modalities, such as 2D/3D skeletons, music, camera viewpoints, and more. When video data is used as input, the model learns diverse, in-the-wild motion patterns for generation. Trained with motion capture data, the model provides temporally consistent and visually plausible estimation, even under occlusion and blur. Extensive experiments show that the proposed approach achieves strong performance across both estimation and generation tasks.

1.4 Bibliographical Remarks

This thesis includes works where the author is either the primary contributor or a key collaborator, as some collaborative projects form essential parts of the overall narrative—for example, the projects presented in Chapter 7 and Chapter 8. Each chapter is based on original publications resulting from collaborations between the author and other researchers.

Specifically, Chapter 2 is based on joint work with Peize Sun, Yi Jiang, Zehuan Yuan, Song Bai, Kris Kitani, and Ping Luo. Chapter 3 is based on joint work with Hao Wu and Kris Kitani. Chapter 4 is based on joint work with Jiangmiao Pang and Kris Kitani. Chapter 5 is based on joint work with Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris Kitani. Chapter 6 is based on joint work with Jiahe Chen, Jiangmiao Pang, and Kris Kitani. Chapter 7 is based on a project led by Zhengyi Luo, in collaboration with Alexander Winkler, Kris Kitani, and Weipeng Xu. Chapter 8 is based on a project led by Zeqi Xiao, involving collaboration with a larger team. Chapter 9 is based on joint work with Jingyuan Liu, Kris Kitani, and Yi Zhou. Chapter 10 is based on joint work with Zhengyi Luo, Sammy Christen, Alexander Winkler, Kris Kitani, and Weipeng Xu. Chapter 11 is based on a joint work with Jiefeng Li, Haotian Zhang, Davis Rempe, Jan Kautz, Umar Iqbal, and Ye

Yuan.

1.5 Excluded Research

Some of the author's other research works, while related to the topics discussed in this thesis, have been excluded for clarity and conciseness. They are listed here to provide a complete record of the author's Ph.D. research:

1. Combining parametric filters and appearance matching for multi-object tracking [238].
2. Cooperative human-object interaction with manipulated objects [107].
3. Universal humanoid motion representations for physics-based control [232].
4. A dataset for multi-human interaction with close body contact [177].
5. A dataset of humanoid motion collected from simulated environments [228].

Part I

Human Motion from Human-Human Interaction

Chapter 2

DanceTrack: Multi-Object Tracking in Uniform Appearance

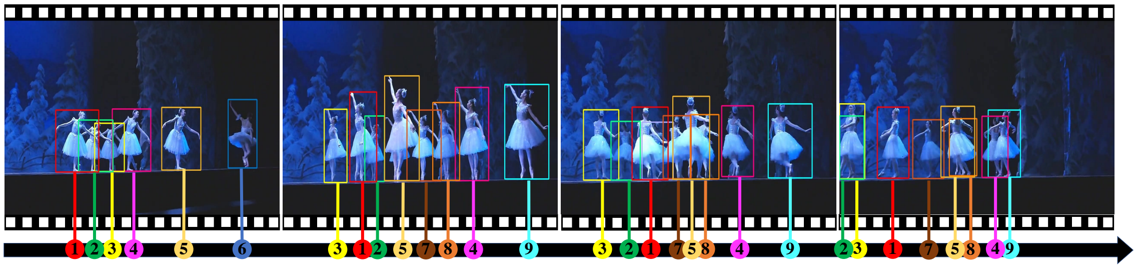


Figure 2.1: Sample images from a video in DanceTrack. The shown images are 1, 66, 307 and 327 frames in DanceTrack0027 video. The emphasized properties of this dataset are (1) *uniform appearance*: humans are in highly similar and almost undistinguished appearance. (2) *diverse motion*: they are in complicated motion pattern and interaction. The numbers below show their identification which experiences frequent relative position switches and occlusion as well. We expect the combination of uniform appearance and complicated motion pattern makes DanceTrack a platform to encourage more comprehensive and intelligent multi-object tracking algorithms.

2.1 Introduction

Object tracking has been long studied and can be beneficial to applications such as autonomous driving, video analysis, and robot planning [44, 305, 452]. Multi-object tracking aims to localize and associate objects of interest over time. Interestingly, we observe recent developments in multi-object tracking heavily rely on a paradigm of detection followed by re-ID, where mostly appearance cues are used to associate objects. This trend in algorithmic development makes existing solutions fail catastrophically in situations where objects share very similar appearance and inspires us to propose a platform to encourage more comprehensive solutions by taking other cues into modeling, such as object motion patterns and temporal dynamics.

As with many other areas of computer vision, the development of multi-object tracking is influenced by benchmark datasets. Based on specified datasets [86, 112, 255, 454], data-driven methods are sometimes argued to be biased to certain data distributions. In this work, we recognize the limitation of existing multi-object tracking datasets and observe that many objects have distinct appearance and the motion patterns of objects are very regular or even linear. Motivated by these dataset properties, most recently developed multi-object tracking algorithms [23, 28, 270, 409, 415, 482, 483] highly rely on appearance matching to associate detected objects while considering little other cues. The dominant paradigm will fail in situations out of the biased distribution. This phenomenon is not what we expect if we aim to build more general and intelligent tracking algorithms.

We also observe that appearance matching is not reliable when objects have similar appearances or heavy occlusion. These properties cause catastrophic degradation of current state-of-the-art multi-object tracking algorithms. To provide a new platform for more comprehensive multi-object tracking studies, we propose a new dataset in this paper. Because it mostly contains group dancing videos, we name it "DanceTrack". The dataset contains over 100K image frames (almost 10× the MOT17 dataset). As shown in Figure 2.1, the emphasized properties of this dataset are (1) **uniform appearance**: people in videos wear very similar or even the same clothes, making their visual features hard to be distinguished by the re-ID model and (2) **diverse motion**: people usually have very large-range motion and complex body gesture variation, proposing higher requirements for motion modeling. The second property also brings occlusion and crossover as a side-effect that the human body has a large ratio of overlap with each other and their relative position exchanges frequently.

With the proposed dataset, we build a new benchmark including existing popular multi-object tracking methods. The results prove that current state-of-the-art algorithms fail to make satisfactory performance when they simply use appearance matching or linear motion models to associate objects across frames. Considering the cases focused on in this dataset happen frequently in our real life, we believe it shows the limitations of existing multi-object tracking algorithms on practical applications. To provide potential guidelines for further research, we analyze a range of choices in associating objects and achieve some beneficial conclusions: (1) fine-grained representations of objects, e.g., segmentation and pose, exhibit better ability than coarse bounding box; (2) depth information shows positive influence on associating objects, though we are solving a 2D tracking task; (3) motion modeling of temporal dynamics is important.

To conclude, the key contributions of our work to the object tracking community are as follows:

1. We build a new large-scale multi-object tracking dataset, DanceTrack, covering the scenarios where tracking suffers from low distinguishability of object appearance and diverse non-linear motion patterns.
2. We benchmark baseline methods on this newly built dataset with various evaluation metrics, showing the limitation of existing multi-object tracking algorithms.
3. We provide a comprehensive analysis to discover more cues for developing multi-object trackers that are more robust in complicated real-life situations.

2.2 Related Works

Multi-object tracking datasets.. Many multi-object tracking datasets have been proposed focusing on different scenarios. Similar to our proposed dataset, many existing datasets focus on human tracking. PETS [100] dataset is one of the earliest in this area. And the more recent MOT15 [191] dataset and the following MOT17 [255] and MOT20 [86] datasets are all popular in this community. These datasets are limited in some aspects we care about. For example, MOT contains only handful of videos and scenarios. Even MOT20 increases the density of objects and emphasis the occlusion among them, the movements of objects are very regular, and they still have very distinguishable appearance. Association by pure appearance matching [270] also makes success and we will show that given the perfect detector, the tracking problem can be solved by a very naive association strategy on these datasets.

Besides, many other datasets are proposed for diverse objectives, e.g., WILDTRACK [58] for multi-camera tracking and association, Youtube-VIS [428] and MOTS [382] for pixel-wise tracking (Video Instance Segmentation). With the increasing attraction of autonomous driving, some datasets are built focusing on it specifically. KITTI [112] is one of the earliest large-scale multi-object tracking datasets for driving scenarios where the objects of interest are vehicles and pedestrians. More recently, BDD100K [454], Waymo [403] and KITTI360 [209] are made available to the public, still focusing on autonomous driving problem but providing much larger scale data than KITTI. The motion patterns of objects in these datasets are even more regular than those focusing on only moving people with the limitation of lanes and traffic rules. There are many datasets focusing on more diverse object categories than person and vehicles. The ImageNet-Vid [88] benchmark provides trajectory annotations for 30 object categories in over 1000 videos and TAO [83] annotates even 833 object categories to study object tracking on long-tailed distribution.

Tracking by matching appearance.. Compared to tracking-by-detection, recent developments in multi-object tracking focus more on the joint-detection-and-tracking genre where object localization and association are conducted at the same time. And appearance similarity serves as the dominant cue in many popular multi-object tracking methods. For example, QuasiDense (QDTrack) [270] designs a pairwise training paradigm and dense localization for object detection and uses highly sensitive appearance comparison to match objects across frames. JDE [402] and fairmot [482] learn object localization and appearance embedding using a shared backbone which is for better appearance representation. More recently, with the new focus of applying transformers [380] in vision tasks, transtrack [353], TrackFormer [250] and motr [466] made attempts to leverage the attention mechanism in tracking objects in videos. In these works, the features of previous tracklets are passed to the following frames as the query to associate the same objects across frames. The appearance information contained in the query is also critical to keep tracklet consistency. Although the rise of deep-learning model brings much powerful visual representations than ever before making appearance matching more robust, we still witness the failure of matching appearance in many real-world situations which are expected to be improved by taking other cues into account.

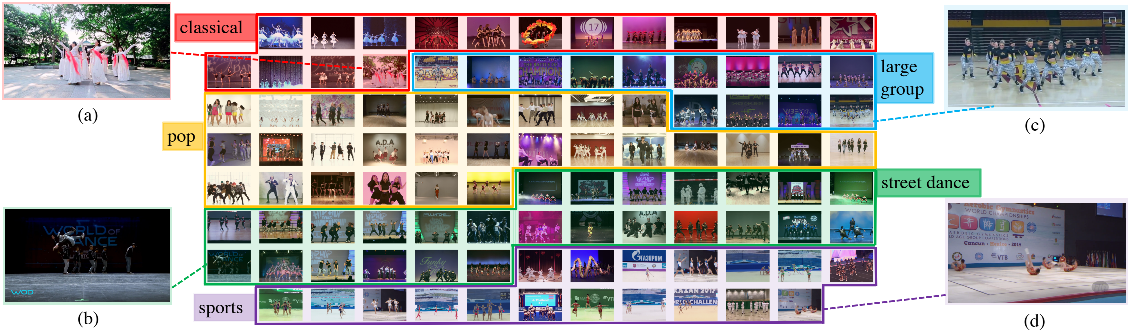


Figure 2.2: Some sampled scenes from the proposed DanceTrack dataset. (a) outdoor scenes; (b) low-lighting and distant camera scenes; (c) large group of dancing people; (d) gymnastics scene where the motion is usually even more diverse and people have more aggressive deformation.

Motion analysis in object tracking.. The displacement of objects-of-interest provides important cues for object tracking. Tracking objects by estimating their motion is thus a natural and intuitive idea and has inspired a line of researches. These tracking algorithms mainly follow the tracking-by-detection paradigm. Sequential analysis tools such as Particle filter [127, 168] and Kalman filter [173] are found efficient in such applications. SORT [28] is developed on the Kalman motion model and marks a milestone in using motion models for object tracking. Furthermore, as deep networks bring the revolutionary ability to extract high-quality visual features, DeepSORT [409] tries to combine deep visual features and motion models and gains great success. Since then, motion-based object tracker has shown weak competitiveness and many focuses are towards appearance cues. Even though motion analysis has been used in object tracking for long [402, 482, 483], all these mentioned methods can only handle simple linear motion pattern and provide limited help to multi-object tracking in more complicated situations we focus on in this work. These factors induce appearance-based tracking dominance in multi-object tracking. However, we argue that a more comprehensive and intelligent tracking algorithm should pay more attention to motion analysis since appearance is not always reliable.

2.3 DanceTrack

DanceTrack is a benchmark for multi-object tracking for estimating the locations and identities of objects in videos. The objective of proposing this dataset is to provide scenes where objects have a uniform appearance and diverse motion.

2.3.1 Dataset Construction

Dataset design.. We focus on the scenarios where objects have similar or even the same appearance and diverse motion patterns, including frequent crossover, occlusion, and body deformation. The first property makes tracking by purely comparing object appearance invalid because the extracted visual features are no longer distinguishable for different objects. The second property further requires clues rather than appearance in tracking, such as motion analysis and temporal dynamics.

We argue that focusing on "crowd" by simply increasing the density of objects of interest is not what we expect. For example, MOT20 [86] contains videos where the groups of pedestrians are very crowded. But as the pedestrians' movement is very regular and the relative position and occlusion area keep consistent, such "crowd" is not building an obstacle for appearance matching. Therefore, we focus on situations where multiple objects are moving in a "relatively" large range. The dynamically changing occluded area and even crossover are what we are interested in. Such cases are common in the real world but naive linear motion models can not handle them anymore.

Video collection.. To achieve the design goals described above, we collected videos including mostly group dancing from the Internet. As shown in Figure 2.2, the dancers usually wear very similar or even the same clothes. They make a large-range motion in the target situations. And their poses and relative positions change very frequently. All these properties greatly satisfy our motivation to propose a new multi-object tracking dataset. We collect the videos from different search engines with query keywords like "street dance", "hip-pop dance", "cheerleading dance", "rhythmic gymnastics" and so on. The collection is only for publicly available videos and under the permit of fair use of video resources.

Annotation.. We use a commercial tool to annotate the collected videos. The

Table 2.1: The comparison of dataset meta-information between DanceTrack and its closest benchmark for multi-human tracking, MOT17 and MOT20. DanceTrack contains much more videos and images than MOT datasets.

Dataset	MOT17 [255]	MOT20 [86]	DanceTrack
Videos	14	8	**100**
Avg. tracks	96	**432**	9
Total tracks	1342	**3456**	990
Avg. len. (s)	35.4	**66.8**	52.9
Total len. (s)	463	535	**5292**
FPS	**30**	25	20
Total images	11,235	13,410	**105,855**

annotated labels include bounding boxes and identifiers of each object. For a partly-occluded object, a full-body box is annotated. For a fully-occluded object, we do not annotate it; when it re-appears in the future frame, its identifier is kept as the same as in the previous frame when it is visible.

To facilitate the annotation process, our tool can automatically propagate the annotated boxes from the previous frame to the current frame, and the annotator only needs to refine the boxes in the current frame. To build a high-quality dataset, the annotations have been checked by another group of people and errors are reported back to the annotators for re-annotation.

2.3.2 Dataset Statistic

We provide some analytical information of DanceTrack dataset and compare it with existing multi-object tracking datasets. The statistical information helps to understand the uniqueness of the proposed dataset and how we built it to make a platform as we describe in the previous parts.

Dataset split.. We collected in total 100 videos in DanceTrack dataset, by default using 40 videos as the training set, 25 as the validation set, and 35 as the test set. During splitting, we keep the distribution of subsets close in terms of average length, average bounding box number, included scenes and motion diversity. We make the annotation of the training set and validation set public while keeping the testing set annotation private for competition use. Some basic information of DanceTrack is shown in Table 2.1. Compared with MOT datasets, DanceTrack has a much larger volume (10x more images and 10x more videos). MOT20 focuses on very crowded scenes, so it has more tracks but as the appearance of objects inside is very distinguishable and their motion is regular, as a consequence, the association on MOT20 still requires little motion estimation when reliable detection results are given.

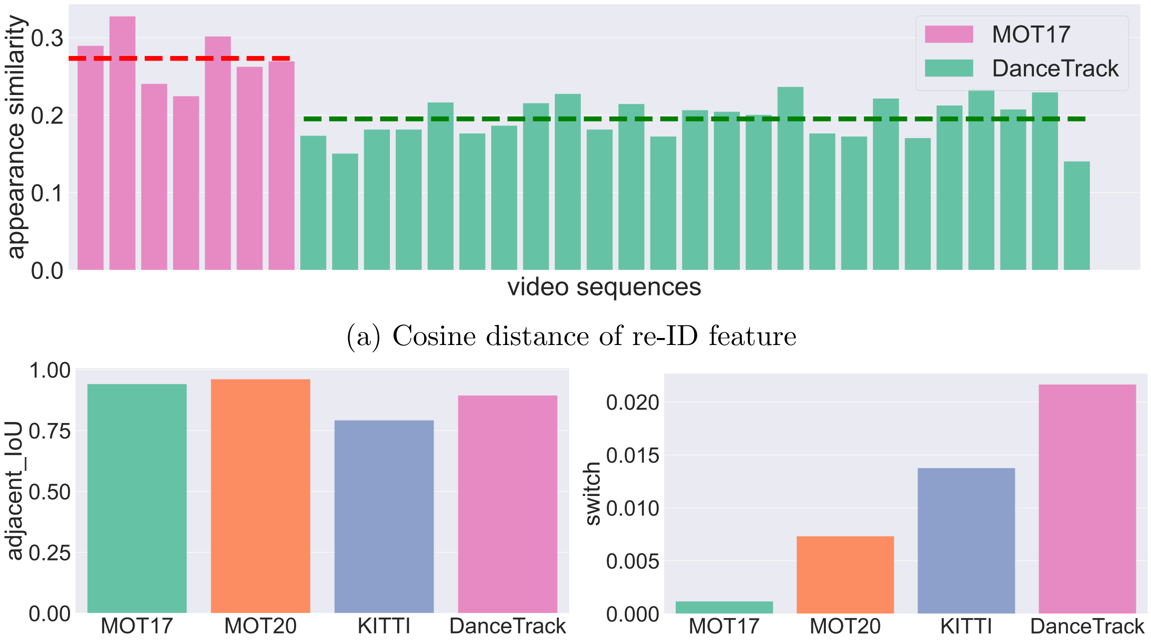
Scene diversity.. DanceTrack contains very diverse scenes. Some samples are provided in Figure 2.2. One main common point for all videos is that the instances of people in a video usually have very similar appearances. This is designed on purpose to avoid the shortcut of tracking by pure appearance matching. DanceTrack contains multiple genres of dance, such as street dance, pop dance, classical dance (ballet, tango, etc.), and large groups of people' dancing. It also contains some sports scenarios such as gymnastics, Chinese Kung Fu and cheerleader dancing. Figure 2.2(a) shows outdoor scenes though most included videos are indoor. Figure 2.2(b) shows some especially hard cases, such as low lighting and distant camera. Figure 2.2(c) and (d) show a large group of people dancing, including at most 40 people, and gymnastics where people show extremely diverse body gestures, frequent pose variation and complicated motion pattern.

Appearance similarity.. We make a quantitative analysis about how appearance-only matching is not reliable anymore on DanceTrack. We will prove this by measuring the appearance similarity among objects. To be precise, we use a pre-trained re-ID model [279] to extract the appearance features $F(B_i^t)$ of the object B_i on a frame t, then we compute the sum of cosine distance of the re-ID features among objects in the video as

$$V = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N_t^2}\sum_{i}^{N_t}\sum_{j\neq i}^{N_t}(1 - \texttt{cos} < F(B_i^t), F(B_j^t) >), \tag{2.1}$$

where T is the number of frames in the video sequence, N_t is the number of objects on the frame t and $<\cdot>$ is the angle between two vectors.

We compare the object appearance similarity in DanceTrack to that in MOT17 dataset, as shown in Figure 2.3(a), each bin represents one video sequence. It is obvious that the cosine distance of re-ID features of DanceTrack is lower than that



(a) Cosine distance of re-ID feature

(b) IoU on adjacent frames

(c) Frequency of relative position switch

Figure 2.3: (a) The cosine distance of re-ID features of DanceTrack is lower than that of MOT17, in other words, the appearance similarity between different objects is higher. The dashed lines are for the average cosine distance similarity. (b) Compared to MOT17 and MOT20, DanceTrack has a similar score. It means that the frame rate and object motion speed are still reasonable in DanceTrack. (c) This metric measures the frequency of crossover and is highly related to the occlusion between objects. DanceTrack has much more frequent relative position switches than other pedestrian tracking datasets.

of MOT17, in other words, the appearance similarity among co-existing objects is higher. This quantitative analysis shows the challenge of DanceTrack to current popular trackers using appearance matching for association.

Motion pattern.. We introduce two metrics to analyze the motion pattern in DanceTrack dataset and compare that to other multi-object tracking datasets.

IoU on adjacent frames: a natural measurement of object movement range is its bounding-box-IoU (Intersection-over-Union) on two adjacent frames. A low IoU indicates fast-moving objects or the low frame rate of videos. Given a video with N objects and T frames, we denote the i-th object's box on the t-th frame as B_i^t, then the averaged IoU on adjacent frames for this video is

$$U = \frac{1}{N(T-1)} \sum_{i}^{N} \sum_{t=1}^{T-1} IoU(B_i^t, B_i^{t+1}). \tag{2.2}$$

Frequency of Relative Position Switch: a metric to measure the diversity of objects' motion in a global view is the frequency for two objects to switch their relative position. This could happen between leftward and rightward or between upward and downward. On the contrary, movement with consistent velocity tends to cause a lower chance of relative position switch. Given a video, the average frequency of relative position switch is defined as

$$S = \frac{\sum_i^N \sum_{j \neq i}^N \sum_{t=1}^{T-1} sw(B_i^t, B_j^t, B_i^{t+1}, B_j^{t+1})}{2N(T-1)(N-1)}, \tag{2.3}$$

where sw is an indicator function, where $sw(\cdot)$=1 if the two objects swap their left-right relative position or top-down relative position on the adjacent frames, $sw(\cdot)$=0 if there is no swap. To be precise, we measure their relative position by comparing their bounding box center locations. And considering that such crossover causes potential trouble only when the objects have overlap, we only take the objects whose bounding boxes have overlap into the calculation.

From the results shown in Figure 2.3(b), we could find that DanceTrack and MOT datasets have close average IoU on adjacent frames. This indicates that DanceTrack is considered harder than MOT datasets not because of lower frame rate or unreasonably fast object movement.

Table 2.2: Oracle analysis of different association models on MOT17 and DanceTrack validation set, respectively. The detection boxes are ground-truth boxes. The result comparison shows the evident increased difficulty of performing multi-object tracking on DanceTrack than MOT17 dataset.

Appr.	IoU	Motion	MOT17					DanceTrack (Proposed Dataset)				
			HOTA	DetA	AssA	MOTA	IDF1	HOTA	DetA	AssA	MOTA	IDF1
	✓		**98.1**	98.9	**97.3**	98.0	97.8	**72.8**	**98.9**	53.6	98.7	63.5
	✓	✓	96.4	97.1	95.8	**99.7**	98.1	69.4	87.9	**54.8**	**99.4**	**71.3**
✓	✓	✓	95.0	94.7	95.4	99.3	**98.8**	59.7	82.5	43.2	97.2	60.5
✓			93.3	**99.0**	87.9	98.9	90.9	68.0	97.7	47.4	97.9	58.7

On the other hand, from Figure 2.3(c) we could find that DanceTrack has much more frequent relative position switches than other datasets such as KITTI, MOT17 and MOT20. The frequent relative position switches are caused by highly non-linear motion pattern and result in frequent crossover and inter-object occlusion. This result shows that the challenge of DanceTrack comes from the diversity of motion.

2.3.3 Evaluation Metrics

For a long time, the multi-object tracking community used MOTA as the main metric for evaluation. However, recently, the community realized that MOTA focuses too much on detection quality instead of association quality. Thus, Higher Order Tracking Accuracy (HOTA) [225] is proposed to correct this historical bias since then. HOTA has been used for the main metrics to evaluate tracking quality on multiple popular benchmarks such as BDD100K [454] and KITTI [112]. We follow this setting for evaluation metrics of DanceTrack.

In our protocol, the main metric is HOTA. We also use AssA and IDF1 score to measure association performance and DetA and MOTA for detection quality.

For the detailed definitions of these metrics, we refer to [25, 225, 317]. To make it convenient to run for fine-grained analysis, the evaluation tools also provide previously widely-used statistics, such as False Positive (FP), False Negative (FN) and ID switch (IDs).

2.3.4 Limitation

In this part, we discuss some recognized limitations of this proposed DanceTrack dataset. We emphasize again that we propose this dataset to provide a platform for more comprehensive multi-object tracking studies beyond the currently popular genre of combining detector and re-ID. However, the proposed dataset still has some limitations. First, given the mentioned motivation and the proposed dataset, we do not provide an algorithm that highly outperforms previous multi-object tracking algorithms but keep this as an open question for future study. Besides, we believe, for the cases, we emphasize in this work, the annotation of human pose or segmentation mask should be important for more fine-grained study. But limited by time and resources, we only provide the annotation of bounding boxes in this version.

2.4 Experiments

2.4.1 Experiment Setup

Dataset configurations. We compare DanceTrack with its closest dataset, MOT17. For MOT17, because the test server is not available easily, we follow the train-val splitting provided in CenterTrack [498] to evaluate on the validation subset. For DanceTrack, we follow the default splitting described in the previous section to train on the training subset and evaluate on the test subset.

Model configuration. Unless specified otherwise, we inherit the default training settings of the investigated algorithms provided in the original papers or the officially released codebases. For MOT17 and DanceTrack, algorithms use shared configurations and hyperparameter settings.

2.4.2 Oracle Analysis

To decompose the analysis over object localization and association, we perform oracle analysis here. We use the ground truth bounding boxes with different association algorithms to achieve expected upper-bound performance.

This analysis can help us to understand what is the true bottleneck of tracking on different datasets. To be precise, we try to use IoU matching or motion modeling

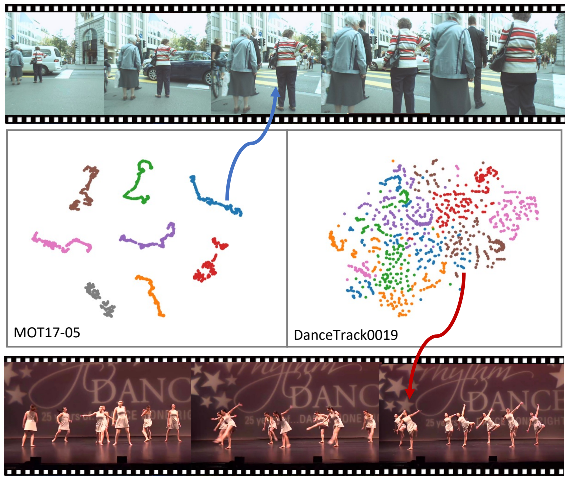


Figure 2.4: Visualization of re-ID feature from sampled video in MOT17 and DanceTrack dataset using t-SNE [379]. The same object is coded by the same color. For better visualization, we only select first 200 frames in each video sequence. The results show that object appearance is much distinguishable on MOT17 than that on DanceTrack. It brings a shortcut for tracking on MOT17 by even only appearance matching.

and appearance similarity for the association. We have experiments on MOT17 and DanceTrack respectively. The results are shown in Table 2.2. We use a pre-trained Re-ID model [279] for appearance matching and a Kalman Filter [173] for motion modeling under linear motion assumption. IoU matching is simply performed by calculating the IoU of objects' bounding boxes in adjacent frames. From the results, the tracking output is close to perfect in terms of all metrics on MOT17. And, interestingly, using only IoU matching achieves the best performance, which proves that MOT17 contains objects with simple and regular motion patterns and the bottleneck does not lie in association in most cases.

On the other hand, using only IoU matching on DanceTrack gives a much lower performance than on MOT17. Given DetA and MOTA scores are already close to 100, the bottleneck is obviously in the association part. All association metric scores in all cases experience a dramatic drop compared with that on MOT17. Besides, the best performance lies in only IoU matching, even combining a linear motion model or additional appearance information does not help. When using appearance similarity, all metrics are worse than not using any appearance cue. This is because

Motion	HOTA	DetA	AssA	MOTA	IDF1
None(IoU)	34.9	**68.2**	18.0	77.0	31.7
Kalman filter[28]	37.2	62.4	22.3	77.4	**39.9**
LSTM[54]	**38.8**	67.8	**22.4**	**78.7**	38.1

Table 2.3: Tracking performance of investigated algorithms on MOT17 and DanceTrack test set respectively. The result comparison shows the evident increased difficulty of performing multi-object tracking on DanceTrack than MOT17 dataset. To be precise, DanceTrack makes detection easier (higher MOTA and DetA scoers) but still brings significant tracking performance drop compared to MOT17 (lower HOTA, AssA and IDF1 scores). This phenomenon reveals the bottleneck of multi-object tracking on DanceTrack is on the association part.

the objects in DanceTrack videos usually have indistinguishable appearance so simply using appearance matching makes negative effects in some cases. In Figure 2.4, we visualize the appearance feature of objects extracted from DanceTrack and MOT17 videos respectively. We can observe that the appearance features of different objects are very distinguishable in the feature space on MOT17 while highly entangled on DanceTrack. This qualitatively provides evidence for the high similar appearance of objects in the proposed DanceTrack dataset.

Given the results shown in the analysis with oracle object localization, we can reach a clear conclusion that existing datasets have a heavy bias that focuses more on the detection quality only and the involved simple trajectory patterns limit the study in this area. On the contrary, DanceTrack is proposing a much higher requirement to develop multi-object trackers with improvement in association ability. Considering the scenarios included in DanceTrack are what we experience in real life, we believe it is meaningful to provide such a platform.

2.4.3 Benchmark Results

We benchmark the current state-of-the-art multi-object tracking algorithms on MOT17 and DanceTrack. The evaluation is performed in the "private setting" that the algorithm should do both detection and association. The benchmark results are reported in Table 2.3. In terms of the tracking quality measured by HOTA, IDF1 and AssA, all algorithms show a significant performance gap from MOT17 to DanceTrack. For all investigated methods, their performance on DanceTrack is far from satisfactory.

Table 2.4: Comparison of different association algorithms on DanceTrack validation set. The detection results are output by YOLOX [111], trained on the DanceTrack training set.

Association	HOTA	DetA	AssA	MOTA	IDF1
IoU	44.7	**79.6**	25.3	87.3	36.8
SORT[28]	**47.8**	74.0	31.0	**88.2**	48.3
DeepSORT[409]	45.8	70.9	29.7	87.1	46.8
MOTDT[60]	39.2	68.8	22.5	84.3	39.6
BYTE[483]	47.1	70.5	**31.5**	**88.2**	**51.9**

On the other hand, the detection quality metrics, MOTA and DetA, of all algorithms are in fact higher on DanceTrack than on MOT17. This suggests that detection is not the bottleneck to have good tracking performance on DanceTrack and continues to highlight the drop of association. The challenge on the proposed dataset is to make robust associations against the uniform appearance and the diverse motion of objects.

2.4.4 Association Strategy

In the previous section, most methods entangle the detection and tracking modules. To have an independent study on association algorithms, we use the most recently developed YOLOX [111] detector for object detection on DanceTrack and conduct different object association algorithms following that. The results are shown in Table 2.4.

SORT [28] uses Kalman Filter to model the object trajectory and DeepSORT [409] adds appearance matching. Compared to SORT, DeepSORT shows no performance boost but worse performance instead, suggesting the negative gain due to appearance matching. On the other hand, MOTDT [60] uses the tracking result to help detect bounding boxes. But in fact, detection performance can be really good on the DanceTrack dataset and the exact bottleneck is the association part, so MOTDT shows even worse performance on both detection quality and association quality with its design. Lastly, BYTE [483] uses a high-tolerance strategy to select detection results into the association stage. The design aims to decrease tracklet fragmentation in tracking. With such a strategy, BYTE shows the best association performance in terms of IDF1 and AssA metrics. This also reveals that DanceTrack is not a

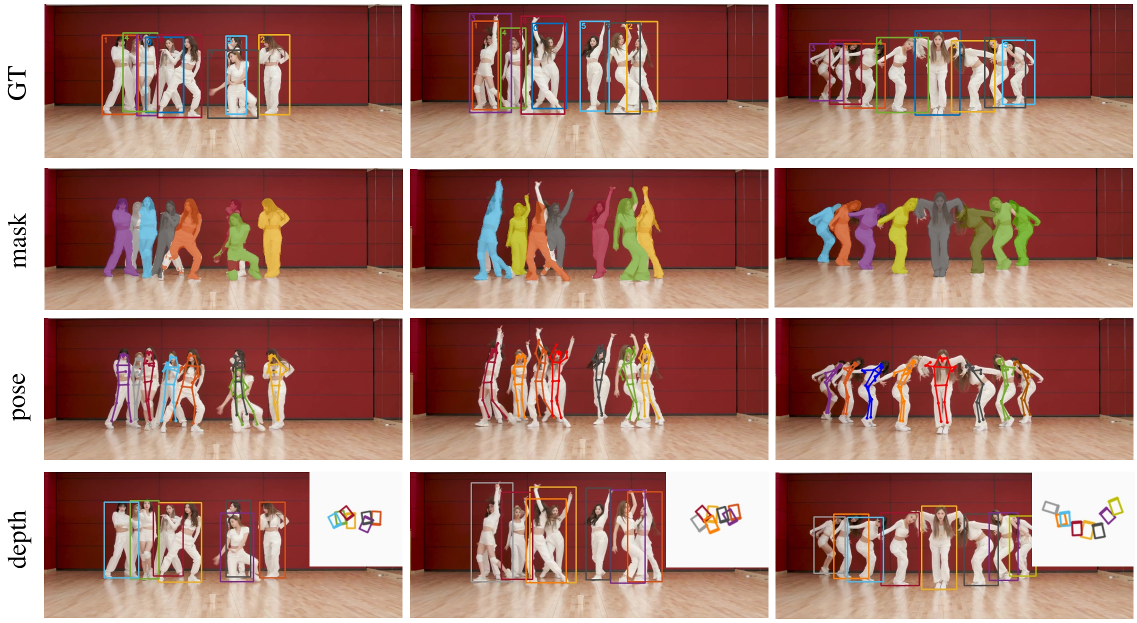


Figure 2.5: Visualization of adding more information beyond bounding box on DanceTrack. Tracks are coded by color. The 1st, 2nd and 3rd column are frame20, 120 and 200 of DanceTrack0007 video sequence, respectively. The 1st row is ground-truth boxes and identifies.

Table 2.5: Ablation study on adding more information beyond bounding box on DanceTrack validation set. All experiments are based on CenterNet [498] model and BYTE [483] association. (a) Segmentation mask improves the tracking performance on DanceTrack. (b) Pose information boosts the tracking performance with an even larger gap than the segmentation mask. (c) Though adding depth information into association shows a slightly positive influence, the results still blame the domain shift between KITTI and DanceTrack.

Data	Ass.	HOTA	DetA	AssA	MOTA	IDF1
DanceTrack	box	36.9	63.6	21.6	78.8	39.2
+ COCOmask [212]	box	38.1 (+**1.2**)	64.5 (+**0.9**)	22.6 (+**1.0**)	80.6 (+**1.8**)	40.3 (+**1.1**)
+ COCOmask	+ mask	39.2 (+**1.1**)	64.9 (+**0.4**)	23.9 (+**1.3**)	80.7 (+**0.1**)	41.6 (+**0.3**)
DanceTrack	box	36.9	63.6	21.6	78.8	39.2
+ COCOpose [212]	box	40.6 (+**3.7**)	65.5 (+**1.9**)	25.3 (+**3.7**)	82.9 (+**4.1**)	42.9 (+**3.7**)
+ COCOpose	+ pose	41.0 (+**0.4**)	65.9 (+**0.4**)	25.6 (+**0.3**)	83.1 (+**0.3**)	43.9 (+**1.0**)
DanceTrack	box	36.9	63.6	21.6	78.8	39.2
+ KITTI [112]	box	34.4 (- **2.5**)	57.8 (- **5.8**)	20.7 (- **0.9**)	72.9 (- **5.9**)	38.5 (- **0.7**)
+ KITTI	+ depth	35.1 (+**0.7**)	57.3 (- **0.5**)	21.6 (+**0.9**)	72.8 (- **0.1**)	40.2 (+**1.7**)

strict challenge for modern deep object detectors, the true challenge is in the object association part instead.

2.4.5 Analysis of More Modalities

Considering high scores of MOTA and DetA on DanceTrack, the limited performance on DanceTrack is an exact failure of trackers instead of detectors. To boost performance, a straightforward strategy is to add more cues other than bounding box. Since DanceTrack contains bounding boxes and identities annotations, we propose to use joint-training technology with other datasets, *e.g.*, COCO [212] and KITTI [112], to enable the model output more modalities including segmentation mask, pose and depth, All models are based on CenterNet [498]. If additional modal is used other than bounding box, we add a corresponding head following the backbone network.

Does fine-grained representation help ?. We investigate the influence of adding the segmentation mask into the model. The training data is a combination of the DanceTrack training set and COCO mask [212]. If the input image is from DanceTrack, we set its mask loss as 0. During inference, the matching metric is the weighted sum of bounding box IoU and mask IoU. From the results in Table 2.5, we find a performance boost by using the segmentation mask. We believe this can be explained by two reasons. First, the introduction of more fine-grained annotation makes the training more robust just as what is observed in multi-task learning. On the other hand, for crowded and occluded situations, the segmentation mask is a more reliable information form than bounding boxes. From the segmentation mask, we can surely expect to extract more accurate object identification information for the association task.

Besides the mask, another modality is human pose information. The training data is a combination of DanceTrack training set and COCO human pose [212]. If the input image is from DanceTrack, we set its pose loss as 0. During inference, the matching metric is the weighted sum of bounding box IoU and Object Keypoint Similarity(OKS) [212]. The results are shown in Table 2.5. Adding additional pose information in training better boosts the model performance on DanceTrack, and using the output pose in association further helps to achieve better tracking results. A potential reason is when most of the area of a human body is occluded already, segmentation model usually can not provide reliable output while the pose estimation

Table 2.6: Comparison of different motion models on DanceTrack validation set. The detection results are output by CenterNet [498], trained on the DanceTrack training set.

Methods	MOT17					DanceTrack (Proposed Dataset)				
	HOTA	DetA	AssA	MOTA	IDF1	HOTA	DetA	AssA	MOTA	IDF1
CenterTrack [500]	52.2	53.8	51.0	67.8	64.7	41.8	**78.1**	22.6	86.8	35.7
FairMOT [482]	59.3	60.9	58.0	73.7	72.3	39.7	66.7	23.8	82.2	40.8
QDTrack [270]	53.9	55.6	52.7	68.7	66.3	45.7	72.1	29.2	83.0	44.8
TransTrack [353]	54.1	61.6	47.9	75.2	63.5	45.5	75.9	27.5	88.4	45.2
TraDes [415]	52.7	55.2	50.8	69.1	63.9	43.3	74.5	25.4	86.2	41.2
MOTR [466]	55.1	56.2	54.2	67.4	67.0	**48.4**	71.8	**32.7**	79.2	46.1
ByteTrack[483]	**63.1**	**64.5**	**62.0**	**80.3**	**77.3**	47.7	71.0	32.1	**89.6**	**53.9**

model focusing on certain human body key-points usually shows higher robustness.

Does depth information help ?. We try to use additional depth information to help tracking on DanceTrack. The training data is a combination of DanceTrack training set and KITTI [112] 3D box. If the input image is from DanceTrack, we set all losses related to the 3D box as 0. During inference, we directly use the camera parameters in KITTI dataset, and the matching metric is the weighted sum of bounding box IoU and depth similarity. The results are shown in Table 2.5. In contrast to the COCO segmentation mask and human pose, depth information learned from KITTI dataset does not increase the performance on DanceTrack. We explain that COCO segmentation and pose estimation datasets contain humans as the main category, while KITTI mainly contains vehicle instances. Thus, the object and scene prior in DanceTrack and KITTI change, and this domain shift degenerates the model. Nevertheless, depth information indeed helps association performance if we regard the baseline as the model trained on joint-dataset of DanceTrack and KITTI. However, limited by the available resources of depth-annotated data, this is the best we could try for now. We expect more study on the influence of depth information to associate objects with uniform appearance and diverse motion.

Does temporal dynamics help ?. As shown in Table 2.6, we use different motion models to introduce temporal dynamics in the tracking process to facilitate better association. Both Kalman filter [28] and LSTM [54] outperform naive IoU association (without temporal dynamics) by a large margin, indicating the great potential of motion models in tracking objects, especially when appearance cues are not reliable. With the relatively slow progress of object model motion, we expect to see more advanced motion models in the field of multi-object tracking.

From the study above, we know that more modalities could help boost the performance of tracking on DanceTrack, especially those from similar data distributions [98, 161, 455]. Given the limitation discussed in section 2.3.4 that DanceTrack only provides bounding box annotation, for now, there would be several interesting future works: (1) extending its annotation modalities, (2) using weakly-supervised learning [374, 387, 497] to estimate other modalities, (3) using transfer learning and domain adaptation [11, 43, 206] to transfer knowledge of other modalities from other data domain to our benchmark.

2.5 Conclusion

In this paper, we propose a new multi-object tracking dataset called DanceTrack. The objects have uniform appearance and diverse motion pattern in DanceTrack, preventing being hacked by Re-ID algorithms. The motivation behind it is to reveal the bias in existing datasets that tend to emphasize detection quality and matching appearance only. This makes other cues to associate objects underrepresented. We believe that the ability to analyze complex motion patterns is necessary for building a more comprehensive and intelligent tracker. DanceTrack provides such a platform to encourage future works on this line.

Chapter 3

Multi-human Tracking by Fusing Position Encoding and Appearance

3.1 Introduction

The transformer [51, 380] has introduced a new powerful paradigm for processing sequential data. Among the innovations by transformers, positional encoding is an essential addition to the transformer. It provides information of token position for 1D text sequences. However, compared to its success in language models, positional encoding plays a relatively minor role in many vision tasks, such as multi-object tracking. When applying transformers in multi-object tracking, popular methods [251, 353, 502] still mostly rely on appearance matching to associate targets across multiple time steps.

Typically, the positional encoding is added to the tokens in the transformer to provide information about the relative order of the input tokens. It has properties such as being consistent for token pairs with the same relative distance, making it ideal for processing 1D sequences of text tokens. However, when using positional encodings in vision tasks, the previously defined position encoding is less well-formed to preserve position information in images (2D) and video tubes (3D). Consequently, many transformer-based methods have found positional encoding ineffective, especially in target tracking [353, 502] tasks, and have stuck to applying appearance similarity as the cue to associate targets.

However, we believe position information should play a more critical role in multi-object tracking. Moreover, by recognizing the flaws of directly migrating positional encoding from language processing to vision tasks, we find that the key is to keep spatial and temporal information lossless in the positional encoding. Motivated by such analysis, we propose a new paradigm of applying the positional encoding earlier, on 2D CNN feature maps, rather than later, on projected feature vectors. We could preserve pixel order and positional information now. Furthermore, we take advantage of the natural Fourier properties of our proposed positional encoding. By approximating the underlying Fourier and maintaining its linearity, we can achieve a uniform position encoding form for detections and trajectories. This enables the model to associate (1) among the detections and (2) between the detections and the trajectories in the same way. As the proposed positional encoding spans every pixel densely and can represent the pixel position evolution over time, we name it Dense Spatio-Temporal position encoding or DST encoding. We also propose using an attention mask for more accurate pixel-wise feature extraction and to avoid noise from background pixels. The attention mask can be computed from either segmentation masks, saliency discovery maps, or other coarse pixel-wise maps.

With the proposed DST encoding, we build a transformer-based method achieving state-of-the-art performance on multi-object tracking and multi-object tracking and segmentation benchmarks. We also provide an analysis of the shortcomings of classic positional encoding and how our DST encoding improves upon it as a new baseline for future works.

3.2 Related Works

3.2.1 Tasks for Tracking Targets in Videos

Topics related to Target tracking in videos include multi-object tracking (MOT), multi-object tracking and segmentation (MOTS), video object/instance segmentation (VOS/VIS), and segmenting and tracking every pixel (STEP). We choose MOT [256, 355] and MOTS [383] to evaluate our proposed method because there are multiple targets in the video and they show long-range movement, making them suitable tasks to verify the effectiveness of our proposed method. On the contrary, VOS/VIS datasets, such as DAVIS [292] and Youtube-VOS [427], contain foreground objects of very different appearances or even categories and they usually have simple and slow movement. On the other hand, STEP [404] is based on MOTS but adds static objects to consider, such as buildings, road lanes, and trees. These objects are static and easy to track by linear motion models and are not suitable for showcasing the advantages of our DST position encoding.

3.2.2 Positional Encoding as a Representation

The currently widely used positional encoding is introduced by the transformer [380] for language models and then extended to vision tasks [51]. Positional encoding or its variants with different names has been studied for a long time as a form of representation. An early work [301] has studied random Fourier features to approximate an arbitrary stationary kernel using Bochner's theorem. It is close to the use of positional encoding in the transformer. In computer vision, coordinate-MLPs provide a way to encode objects' positions as weights and are related to the study of positional encoding [365, 493]. More recently, Zheng et al. [491] also suggest a study of positional encoding beyond a Fourier lens. They show that non-Fourier embeddings can also serve as positional encoding and, in the perspective of coordinate-MLPs, the performance is determined by a trade-off between embedding matrix stable rank and the distance preservation of coordinates. However, all these explorations have not suggested an efficient form of positional encoding for vision tasks to preserve the spatial transformation of a series of positions.

3.2.3 Multi-objec Tracking Algorithms

Early works on multi-object tracking mainly focus on motion analysis on the target trajectory, where the Kalman Filter is a classic solution [27]. Later, the rise of deep learning brings the powerful deep visual representations and related algorithms follow two paradigms: tracking-by-detection and joint-detection-and-tracking methods. Both of these paradigms involve an association stage, where they mostly focus on appearance matching [269, 485], i.e., re-identification, without using the motion information. More recently, transformer [380] is introduced into the area of multi-object tracking [251, 353, 466] to take advantage of its parallel processing power.

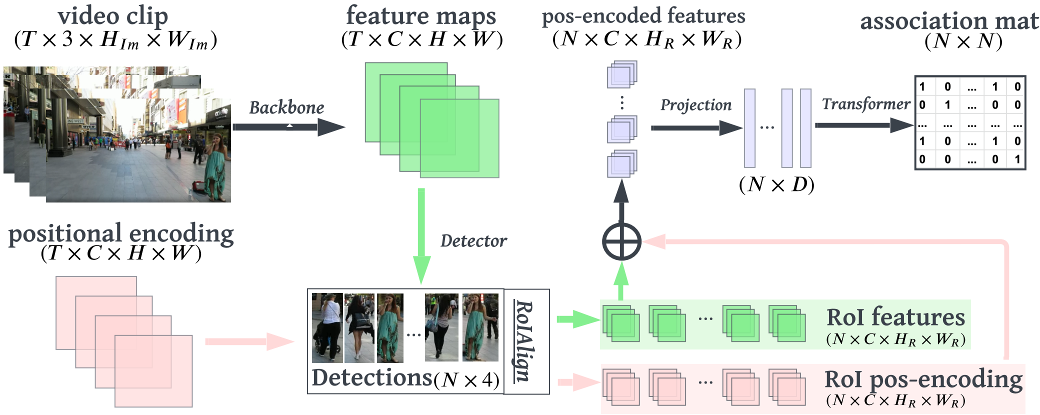


Figure 3.1: The illustration of associating targets within a video clip. For general cases, we use bounding boxes to represent the targets of interest while we can further replace the detector with a segmentation model to do the attention in a more fine-grained mask area. Positional encoding is added to CNN feature maps to encode position information.

However, existing methods still neglect the information motion information, with the exception of MOTR [466] which has attempted to model motion implicitly using a query iteration mechanism. GTR [502] shows that using position encoding decreases transformer performance on MOT tasks. All the evidence suggests that the existing ways for leveraging motion and position information in transformer trackers are ineffective, which motivates the explorations in this paper.

3.3 Method

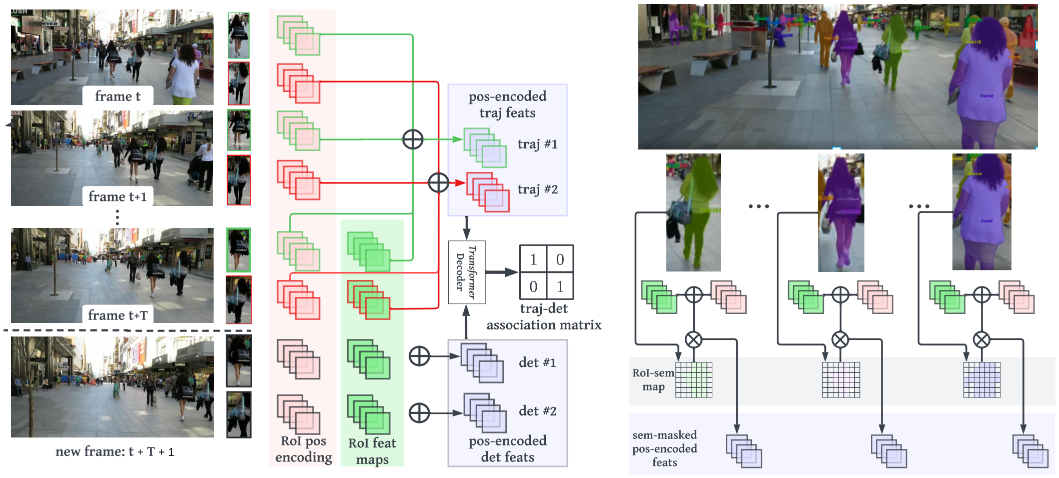
In this section, we first provide an overview of the architecture of our method and then detail its components: the design of the Dense Spatio-Temporal (DST) encoding, the attention mask, and the training and inference configurations.

3.3.1 Overview

The proposed method can make associations at two levels: between detections in a video clip or between detections and existing trajectories.

Association of detections in a video clip.. For the association of objects, we follow the "global association" scheme widely adopted by transformer-based methods [400, 466, 502], as shown in Figure 3.1. With the images of T frames as input, we first use a backbone network to extract the feature maps. Then, a detector head is used to localize N objects of interest inside these images with optional segmentation to gain more fine-grained feature representation. Given the localized objects, we extract their RoI features on both CNN features maps and DST encoding, which are of the same shape $T \times C \times H_R \times W_R$, where (H_R, W_R) is the preset size of RoI, e.g. 7×7. Finally, we add both features and project them to feature embeddings of size $N \times D$, which we then forward into a transformer decoder to compute the attention score matrix of size $N \times N$. Considering that there should be no association between objects from the same frame, we perform softmax on each frame respectively to ensure a well-formed association matrix.

Association between detections and trajectories.. The proposed method can also perform the association between the detections on a new-coming video frame and existing trajectories for online tracking during inference. During the online inference, we perform tracking frame by frame by using a sliding window on the video with a stride of 1. We align the representation of trajectories in the same shape and form as detections to enable this process to share the same model for the detection-detection association. To represent the position of detections on a single frame, we apply RoI to extract the corresponding area from the DST map. However, to represent the trajectory, we now have to use the accumulated DST encoding to record the positional evolution of the track. In this fashion, the representation of a trajectory is designed to be the element-wise addition of accumulated DST encoding of historical object positions and the CNN features of the object snapshot at the last frame. The process of associating detections and trajectories is explained in Figure 3.2a. We perform softmax over the dimension of detections and the dimension of trajectories, respectively, to output the final association matrix. We use the Hungarians algorithm to ensure an one-to-one mapping between detections and trajectories. If a detection's attention score with all trajectories is lower than a threshold β or all available trajectories are already associated, this remaining detection will give birth to a new trajectory.



(a) (b)

Figure 3.2: A deeper look at the component in our method. (a): how we generate the feature representations for both trajectories and single-frame objects. The representation of a trajectory is the accumulated positional encoding of all contained historical locations and the appearance feature of the last snapshot of the object. (b): for the video tracking and segmentation task, we use the semantic occupancy map onto object RoI to obtain more fine-grained RoI features where both position and semantics are encoded.

3.3.2 Dense Spatio-Temporal Position Encoding

Re-identification-based tracking methods associate targets across frames by comparing the appearance similarity of targets, neglecting the location information. However, we believe that location cues can significantly help associate targets because objects usually follow certain motion patterns in the real world. To present the location of each object, a navie way is to append the bounding box coordinates to the object's feature vector. However, this operation can not scale up to trajectories of arbitrary length. Recently, the transformer has been adopted in multi-object tracking with the one-dimensional sinusoidal positional encoding [380] added to token vectors. But it is not as effective [353, 466, 502] as in the language process tasks [380]. We argue that to scale the 1D positional encoding up to 2D images or 3D videos, we need to avoid the loss of spatial information during feature projection. We will demonstrate that our proposed DST encoding can solve this problem to provide a better structure of location information in representing object trajectories.

Encoding of single-frame locations.. Given the channel number of feature maps is C, for a pixel at position (x, y) in the image (or feature maps) whose size is $W \times H$, its positional encoding value at the i-th channel is

$$P(x, y, i) = \begin{cases} -cos\left[(\frac{x}{W} + \frac{y}{WH})\pi + \frac{2i\pi}{C}\right], i = 2k+1 \\ cos\left[(\frac{y}{H} + \frac{x}{WH})\pi + \frac{2i\pi}{C}\right], i = 2k \end{cases} \quad k \in \mathbb{Z} \cap [0, \frac{C}{2}). \tag{3.1}$$

Such an encoding has a few desirable properties. First, it injectively maps from the pixel position to a value on all channels of the feature maps. Second, it keeps the encoding zero-centered spanning the image area which is friendly to the model training. Finally, the term $\frac{2i\pi}{C}$ keeps the encoding fairly sensitive to location variance within the whole area of the image. Without this term, the encoding value changes more sensitively around the image center while less sensitively near the image boundary. This is easy to prove by checking the first derivative of the sinusoidal function.

What we use in the final DST encoding is the resized encoding from the RoI area of objects only. This helps the model to have an encoding of the fixed shape and focus on the object area. If the shape of RoI is $W_R \times H_R$ and the bounding box coordinates of an object on the raw image is $(u, v, u + w, v + h)$, on the cropped and resized RoI feature maps, the positional encoding becomes

$$P_R(x',y',i) = \begin{cases} -cos\left[(\frac{w}{WW_R}x' + \frac{h}{WHH_R}y')\pi + (\frac{u}{W} + \frac{v}{WH})\pi + \frac{2i\pi}{C} \right], i = 2k+1 \\ cos\left[(\frac{h}{HH_R}y' + \frac{w}{WHW_R}x')\pi + (\frac{v}{H} + \frac{u}{WH})\pi + \frac{2i\pi}{C} \right], i = 2k \end{cases} \quad k \in \mathbb{Z} \cap [0, \frac{C}{2}), \tag{3.2}$$

where $x' \in [0, W_R]$ and $y' \in [0, H_R]$, only now extending in the boundary box area. Here, the period of this encoding function changes in terms of the ratio of object size and RoI size. Therefore, this operation also implicitly encodes the target shapes instead of just the position.

Encoding of trajectory.. On two time steps t_1 and t_2, we note the bounding boxes of a target object as $\mathbf{b}_1 = (u_1, v_1, u_1 + w_1, v_1 + h_1)$ and $\mathbf{b}_2 = (u_2, v_2, u_2 + w_2, v_2 + h_2)$. Now, by adding the positional encoding in the RoI area, we have the trajectory encoding of every pixel in the two bounding boxes as

$$P_R^{\mathbf{b}_2|\mathbf{b}_1}(x',y',i) = P_R^{\mathbf{b}_1}(x',y',i) + P_R^{\mathbf{b}_2}(x',y',i). \tag{3.3}$$

Because the period of function $P_R^{\mathbf{b}_2|\mathbf{b}_1}$ is still longer than W_R and H_R on the direction of width and height, it can still represent the trajectory from $\mathbf{b}_1$ to $\mathbf{b}_2$ injectively. Furthermore, we can extend this trajectory encoding to longer video clips as

$$P_R^{\mathbf{b}_T|\ldots|\mathbf{b}_1}(x',y',i) = \sum_{t=1}^{T} \alpha_t P_R^{\mathbf{b}_t}(x',y',i), \tag{3.4}$$

where α_t is the weighting factor on the t-th frame. As for each frame, we have the dense position encoding on each pixel in the object area in the form of trigonometric functions; the trajectory encoding is well represented in a Fourier series now. We choose a linear combination of frame-wise encoding to take advantage of the linearity of Fourier series that is $\mathcal{F}(\sum_{i=1}^{K} \sigma_i f_i) = \sum_{i=1}^{K} \sigma_i \mathcal{F}(f_i)$, where $\mathcal{F}$ is the Fourier transform and σ_i is the weighting factor for function f_i. This property ensures the sanity to extend trajectory encoding by linearly adding the position encoding on the new coming frame. To show this, we note $\mathcal{T}^{\mathbf{b}_T|\ldots|\mathbf{b}_1}$ the underlying function that we aim to approximate to represent a trajectory along the bounding boxes $(\mathbf{b}_1, \ldots, \mathbf{b}_T)$. Then, if we have a function $\mathcal{L}$ that maintains the linearity, we have $\mathcal{F}(\mathcal{T}^{\mathbf{b}_T|\ldots|\mathbf{b}_1}) = \mathcal{L}(P_R^{\mathbf{b}_T|\ldots|\mathbf{b}_1})$.

Therefore, extending the trajectory to the next position $\mathbf{b}_{T+1}$ keeps the form of the positional encoding for the trajectory the same:

$$\mathcal{L}(P_R^{\mathbf{b}_{T+1}|\mathbf{b}_T|\ldots|\mathbf{b}_1}) = \mathcal{L}(P_R^{\mathbf{b}_T|\ldots|\mathbf{b}_1}) + \mathcal{L}(P_R^{\mathbf{b}_{T+1}}) = \mathcal{F}(\mathcal{T}^{\mathbf{b}_T|\ldots|\mathbf{b}_1}) + \mathcal{F}(\mathcal{T}^{\mathbf{b}_{T+1}}) = \mathcal{F}(\mathcal{T}^{\mathbf{b}_{T+1}|\mathbf{b}_T|\ldots|\mathbf{b}_1}). \tag{3.5}$$

Now, we have shown that the proposed DST encoding can preserve the position information in a spatio-temporal occupancy tube densely and at arbitrary length. On each encoding channel, the value is variant to both the absolute position of the corresponding pixel and the position difference across frames. On the other hand, the traditional positional encoding in the transformer maintains the same encoding for any tokens of the same position difference. Also, since the full period (2π) spans on the feature channel dimension (C), it can always map the same relative position shift of two pixels to the same value on different channels. In practice, we use an MLP without non-linear activation to model the function $\mathcal{L}$ along the dimension of encoding channels. If a target moves smoothly along the width and height directions, the encoding of its previous trajectory and its encoding on a new-coming frame will output a high similarity by attention.

Compared to the classic vector positional encoding, DST encoding has three main advantages: (1) preserving the object location information; (2) encoding pixel-wise dense information; (3) unifying representation for single-frame objects and trajectories across multiple frames. These properties provide additional knowledge to associate targets across frames.

3.3.3 Dense Spatio-Temporal Attention

As both visual features and location encoding are dense on every pixel, we can do the association in a pixel-wise dense fashion now. But in fact, the target objects often change their pose in the bounding box and the bounding box includes background area as noise, especially when the targets are non-rigid such as the human body in pedestrian tracking. But when the video frame rate is high, the relative movement of the object body inside the bounding box is minor, dense attention is still very useful. Moreover, we perform attention to the RoI elements instead of the raw image pixels. Each pixel in RoI is already a conclusion of multiple pixels on the raw images. It makes dense attention more robust. For the association of detections in a video clip, the features already integrated with positional encodings are noted as $F \in \mathbb{R}^{N\times C\times H_R\times W_R}$ for N objects. Then, we apply attention mask $M \in \mathbb{R}^{N\times H_R\times W_R}$ determining which "pixel" in the RoI areas should be attended to. In practice, the attention mask M can be the segmentation mask (Figure 3.2b) if that is available or an attention map without using segmentation supervision. We copy the feature along the channel dimension to scale it to $M' \in \mathbb{R}^{N\times C\times H_R\times W_R}$. Next, we apply an MLP to transform the features into 1-d feature vectors, the operation noted as $g(\cdot)$. Given all the preparation, we get the encoded feature vector as $g(M'F)$, which would later be transformed to K and Q by linear layers in self-attention. Finally, we predict the attention matrix as $S = \text{softmax}(\frac{Q\times K^T}{\sqrt{D}})$. This also works in the case of cross-attention for associating trajectories and detections. For a trajectory, M is the attention mask on its last frame. We will apply Hungarians algorithm to ensure the validity of the final binary association matrix from the attention matrix.

3.3.4 Training and Inference

Training.. During training, we draw N high-confidence detections from a detector after NMS, noted as $\mathcal{D} = \{D_1, ..., D_N\}$. The features with positional encoding added are noted as $\{F_1, ..., F_N\}$. From the self-attention-based association of objects within the video clip, we can output its association matrix as $\hat{S} \in \mathbb{R}^{N\times N}$. With the ground truth association matrix $S \in \mathbb{R}^{N\times N}$, we can derive the MSE loss for in-clip object association as

$$l_{clip}(S, \hat{S}) = \frac{1}{N^2}\sum_{i,j}(S_{i,j} - \hat{S}_{i,j})^2. \tag{3.6}$$

In addition to this, we can train the association in the detection-trajectory pairs. Similarly, in the video clip we draw, we have ground truth trajectories as $\mathcal{T} = \{\tau_1, ..., \tau_k\}$. Then, for each frame t, we would remove the footage on and after this frame from these trajectories. It results in a new set on each frame as $\mathcal{T}^t = \{\emptyset\}\bigcup\{\tau_1^t, ..., \tau_{k^t}^t\}$ where $\emptyset$ is an empty trajectory. At the same time, we note the detections on the frame t as $\mathcal{D}^t = \{D_1^t, ..., D_{m^t}^t\}$. We then output the detection-trajectory association matrix by the introduced cross-attention. With the ground association matrix noted as S^t and the estimated association matrix from softmax as

$\hat{S}^t$. The loss is formulated by logistic as

$$l_{det\_traj}(\mathcal{D}, \mathcal{T}) = -\sum_{t=1}^{N} \sum_{\tau_i^t \in \mathcal{T}^t} \sum_{j=1}^{m^t} S^t(D_j^t, \tau_i^t) \log(\hat{S}^t(D_j^t, \tau_i^t)), \tag{3.7}$$

where an object can also be associated with an "empty trajectory" which means it has no corresponding existence on other frames. Finally the overall association loss is the combination of these two terms as $l_{asso} = l_{clip} + l_{det\_traj}$. For the localization stage, we can use a pretrained detection or segmentation model and freeze it or train it at the same time as training the association module.

Inference.. During inference, we use an 1-stride sliding window to move from the first video clip of length T to the last. In the first clip, we use the association of detections to initialize trajectories. Then, for the following steps, we do detection-trajectory and detection-detection associations at the same time. Then we use their average likelihood of association to determine the final association matrix between new-coming detections and existing trajectories. Because only one frame is new at each step of the window sliding, it is averaging the score of associating detections on the T-th frame and previous $T-1$ frames. The later ones have been assigned to a trajectory already. If the average association score is lower than 0.3, we start a new trajectory from the detection. In this process, we use the Hungarians algorithm to ensure the validity of the association matrix between detections and trajectories.

3.4 Experiments

3.4.1 Setup

Datasets and metrics.. We choose two MOT datasets (MOT17 [256] and Dancetrack [355]) and a MOTS dataset (MOTS20 [383]) as the experiment platforms. For evaluation, we use HOTA [226] as the main metric, as it has a reasonable balance between localization and association quality and evaluates association quality at a trajectory level. We also emphasize AssA as it purely measures the video-level association quality. However, on the MOTS20 test set, the HOTA evaluation protocol is not reported. So we also take IDF1 as a secondary metric to compare the quality

Table 3.1: Results on MOTS20 test set. We include only single-model methods here.

Method	sMOTSA ↑	IDF1↑	MOTSA ↑	FP ↓	FN ↓	ID Sw. ↓	Frag ↓
Track R-CNN [383]	40.6	42.4	55.2	1,261	12,641	567	868
TraDes [415]	50.8	58.7	65.5	1,474	9,169	492	-
TrackFormer [251]	54.9	63.6	-	2,233	**7,195**	278	-
SORTS [4]	55.0	57.3	68.3	1,076	8,598	552	**577**
Ours	**60.0**	**68.3**	**71.7**	**634**	8,229	**275**	714

of the association. But we still note that IDF1 is calculated at a single-frame level and cannot accurately measure the quality of association at a video level.

Implementation.. We use ResNet-50 [139] as the backbone network and BiFPN [364] for upsampling of feature maps. We use RoIAlign [141] to extract RoI of size 7 × 7. For a fair comparison, we follow CenterNet [499] for detection and keep it as-is from the pretraining on CrowdHuman [333]. For training, the image size is 1280 × 1280 and we use $T = 16$ to draw video clips. We use AdamW [222] optimizer to finetune the association module for 12K (MOT17, MOTS20) or 20K (Dancetrack) iterations with the starting learning rate of 1e-3. For segmentation, we adopt the MaskRCNN head [141] upon detection and train the head with an additional mask-rcnn loss added to the association loss. We adopt two "linear-ReLU" layers to project the features in the transformer. As for the evaluation of MOTS, each pixel is allowed to be assigned to at most one object; we exclusively assign pixels to at most one object per their confidence scores on MOTS20. Our implementation is based on Detectron2 [417]. We also refer to mmtracking [76] for the implementation details.

3.4.2 Benchmark Results

On the MOTS20 test set (Table 3.1), we evaluate IDF1 as the main metric. Here we only show the results from single-model methods for fairness so some others such as ReMOTS [437] are not listed here. Our results show that the proposed method can consistently outperform existing single-model methods. In addition to MOTS, we also benchmark our method on MOT benchmarks of MOT17 (Table 3.2) and DanceTrack (Table 3.3). On the MOT17 test set, among transformer-based methods, our proposed method obtains the highest HOTA and AssA scores, showing its superior association performance. Moreover, compared to GTR [502], which uses the same detection network as ours but no position information during association, we could see the source of our method's outperforming is the use of spatio-temporal position encoding. On the DanceTrack test set, our method also achieves the highest HOTA and AssA scores among transformer-based methods.

Table 3.2: Results on MOT17 test set. Best results among transformer methods are underlined.

Tracker	Transformer	HOTA ↑	AssA ↑	MOTA↑	IDF1 ↑	ID Sw. ↓	FP ↓	FN ↓
FairMOT [485]		59.3	58.0	73.7	72.3	3,303	27,507	117,477
PermaTrack [375]		55.5	53.1	73.8	68.9	3,699	28,998	115,104
TraDes [415]		52.7	50.8	69.1	63.9	3,555	20,892	150,060
TubeTK [268]		48.0	45.1	63.0	58.6	4,137	27,060	177,483
ByteTrack [484]		63.1	62.0	**80.3**	77.3	2,196	25,491	**83,721**
OC-SORT [45]		**63.2**	**63.4**	78.0	**77.5**	**1,950**	**15,129**	107,055
TransTrk[353]	✓	54.1	47.9	75.2	63.5	4,614	50,157	86,442
TransCenter [431]	✓	54.5	49.7	73.2	62.2	3,663	23,112	123,738
TrackFormer [251]	✓	-	-	65.0	63.9	3,258	70,443	123,552
MOTR [466]	✓	-	-	67.4	67.0	1,992	32,355	149,400
GTR [502]	✓	59.1	61.6	75.3	71.5	2,859	26,793	109,854
MeMOT [40]	✓	56.9	55.2	72.5	69.0	2,724	37,221	115,248
Ours	✓	60.1	62.1	75.2	72.3	2,729	24,227	109,912

Table 3.3: Results on DanceTrack test set. Best transformer-based results are underlined.

Tracker	Transformer	HOTA ↑	DetA ↑	AssA ↑	MOTA↑	IDF1 ↑
CenterTrack [501]		41.8	78.1	22.6	86.8	35.7
FairMOT [485]		39.7	66.7	23.8	82.2	40.8
SORT [27] + YOLOX [110]		47.9	72.0	31.2	**91.8**	50.8
DeepSORT [408] + YOLOX [110]		45.6	71.0	29.7	87.8	47.9
ByteTrack [484] + YOLOX [110]		47.3	71.6	31.4	89.5	52.5
OC-SORT [45] + YOLOX [110]		**55.1**	**80.3**	**38.0**	89.4	**54.2**
TransTrk[353]	✓	45.5	75.9	27.5	88.4	45.2
MOTR [466]	✓	48.4	71.8	32.7	79.2	46.1
GTR [502]	✓	48.0	72.5	31.9	84.7	50.3
Ours	✓	51.9	72.3	34.6	84.9	51.0

Our results on diverse datasets have shown the effectiveness of our proposed method compared to other transformer-based methods. We believe that emphasizing position information during attention and association allows the DST position encoding to outperform other methods. We will continue to further prove this through an ablation study.

3.4.3 Ablation Study

Some design choices may contribute to the performance of our proposed method. To fully validate these choices, we need segmentation annotation, but the MOTS20 evaluation server has strict access restrictions, so we have to follow the common practice [501] on MOT17 [256] to split each video in MOTS20 with the first half for training and the later half for validation in the ablation study.

To have a deeper understanding of the proposed method, the first to come is the role of DST position encoding. To verify its effectiveness, we compare it with the same architecture but without positional encoding or using classic vector positional encoding [380] in Table 3.4. The results clearly suggest the effectiveness of our proposed DST position encoding. Moreover, the classic positional encoding hurts the association performance, which is aligned with the observations by Zhou et al. [502].

Furthermore, we compare the performance with and without the attention mask from segmentation on MOTS20-val. The results are reported in Table 3.5. It also shows the clear advantage of using such a mask when gathering and processing the features. It agrees with the intuition that such a mask eliminates the noise from the background and potential secondary subjects in bounding boxes from the representation features.

Table 3.4: The ablation study of **positional encoding** on MOTS20-val.

pos-encode	HOTA ↑	IDF1 ↑	DetA ↑	AssA ↑	sMOTA↑	MOTSA ↑	ID Sw.↓
w/o pos-encoding	64.4	72.5	72.5	58.0	71.6	82.8	150
classic pos-encoding [380]	64.1	72.5	69.7	59.3	67.8	79.6	162
DST pos-encoding	67.1	74.9	72.8	62.3	71.7	83.0	135

Table 3.5: The ablation study of **attention mask** on MOTS20-val.

	HOTA ↑	IDF1 ↑	DetA ↑	AssA ↑	sMOTA ↑	MOTSA ↑	ID Sw. ↓
w/o mask	64.6	71.3	72.5	58.1	71.3	82.6	156
w/ mask	67.1	74.9	72.8	62.3	71.7	83.0	135

The ablation studies demonstrate the effectiveness of the proposed DST position encoding as the main contribution of this work. Also, the attention mask to more accurately conclude the representation of objects is proven useful when necessary mask information is given. We note that without a segmentation mask, we can use a pretrained segmentation model or a saliency detection model to generate such masks. But this would introduce an unfair advantage, so we decide not to include it on the benchmark of MOT datasets.

3.5 Conclusion

In this work, we propose a novel dense spatio-temporal (DST) position encoding to incorporate target position information into the transformer for multi-object tracking. DST encoding leverages the property of the Fourier transform to make a uniform form of position representation for both single-frame objects and trajectories across multiple frames. It shows good effectiveness in the task of multi-object tracking. While multiple previous works have failed in boosting performance with classic positional encoding, our work provides a novel and efficient paradigm for future works to do object tracking beyond just appearance matching.

Chapter 4

Multi-Object Tracking by Hierarchical Visual Representations

4.1 Introduction

Discriminative visual representations can help avoid mismatches between different targets in appearance-based association for multi-object tracking. We propose a new visual representation paradigm by fusing visual information from different spatial regions in a hierarchy. We argue that, compared to the common paradigm of only using features from bounding boxes, the proposed hierarchical visual representation is more discriminative and no extra annotations are required.

In modern computer vision, we typically use bounding boxes or instance masks to define the area of an object of interest. Because the enclosed pixel area is bonded with a certain object category, such a representation is usually considered as *semantic*. However, we find that not just the *semantic* cues can make informative representations for visual recognition. We can generate more discriminative visual representations from the other two perspectives to define the existence of an object: *compositional* and *contextual*. Compositional cues describe how the parts of a target look like and contrast cues describe how a target looks different from others. For example, as shown in Figure 4.1, multiple flamingo individuals are almost indistinguishable in appearance to us. But by focusing on the distinguishable parts of certain individuals, such as the shape of the wing red mark, we can easily spot the individual (*compositional*). We can also be more confident in distinguishing instances if we can compare all individuals across timesteps (*contrast*).

We thus build discriminative visual representations from three perspectives: *compositional*, *semantic*, and *contextual*. The *semantic* level, such as a tight bounding box or instance segmentation mask, defines the occupancy area of the object with certain visual existence and semantic concept. The compositional level suggests the salient visual regions of an object instance, with which, ideally, we can track it even without seeing its full body. The *contextual* information helps to highlight a subject via contrast with background pixels and other instances. For example, we often have a hard time determining whether two object instances are the same one. However, it is typically easier to determine whether one instance is more likely to be the same one than another. Motivated by the insight, we propose to represent an object by a three-level hierarchy, i.e., *Compositional*, *Semantic*, and *Contextual*.

We adopt the proposed visual hierarchy in video multi-object tracking to avoid the mismatch among different targets. We find that it is crucial how the representations from levels are leveraged together. The naive way of stacking or concatenating them does not show a significant performance advantage. Instead, we propose an attention-based module called CSC-Attention to fuse the features. The core idea of CSC-Attention is to leverage the attention-based mechanism to attend to the salient areas on the target subject body by contrasting to the background pixels close to it. Discriminating targets by the fused features, the multi-object tracker we construct is named CSC-Tracker. It leverages a global association by a transformer to effectively track objects over time. Through experiments on multiple multi-object tracking datasets, CSC-Tracker achieves state-of-the-art accuracy among transformer-based methods with better robustness to noise, better time efficiency, and more economic computation requirements.

Our contributions are three-fold. First, we propose a visual hierarchy for more discriminative visual representations without additional annotations. Second, we propose an attention-based module to leverage the hierarchical features. Last, we build a transformer-based tracker with these two innovations and demonstrated its superior accuracy and time efficiency in a pure appearance-based fashion for multi-object tracking.

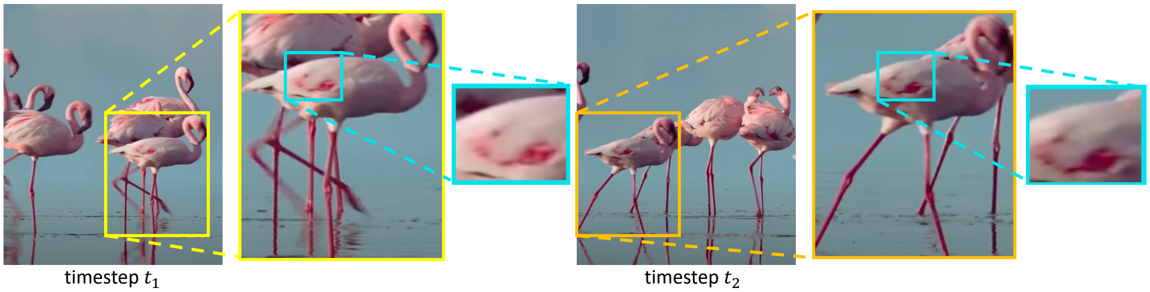


Figure 4.1: With a close look at distinct compositional visual regions, we can recognize certain individuals much more easily.

4.2 Related Works

Deep Visual Representation. We typically use a backbone network to extract features from a certain area, such as bounding boxes, as a visual representation for visual perception. However, the bounding box is noisy as it always contains pixels from the background or other object instances. For a more fine-grained visual representation, a common way is to use pre-defined regions, such as human head [333, 357] or human joints [9, 421]. However, these choices require additional data annotations and specified perception modules. Without requiring additional annotations, multi-region CNN [115] proposes to stack the features from bounding box bins to build a compositional visual representation. However, this paradigm can not generate instance-level discriminative representation though it shows effectiveness in semantic-level recognition. Moreover, simply stacking features can't emphasize the discriminative visual regions.

Hierarchy Visual Representations. The term "hierarchical visual representations" has been used indiscriminately for (1) features fused from different resolutions of the same area, such as CNN feature pyramid [213, 236] and (2) features fused from different pixel areas. Our proposed hierarchical visual representations lie in the second genre. Our idea is inspired by David Marr's hierarchical modeling of the human body [247] (*computational*, *algorithmic*, and *implementational*) and the visual cognitive hierarchy [101] (*semantic*, *syntactic*, *physical*). Compared to the two visual hierarchies, the three-level hierarchy we propose (*compositional*, *semantic*, *contextual*) is focused on building discriminative visual representations for multi-object tracking. Also, in the area of re-identification, some previous works leverage part-based hierarchical features to build visual representation. But most of them

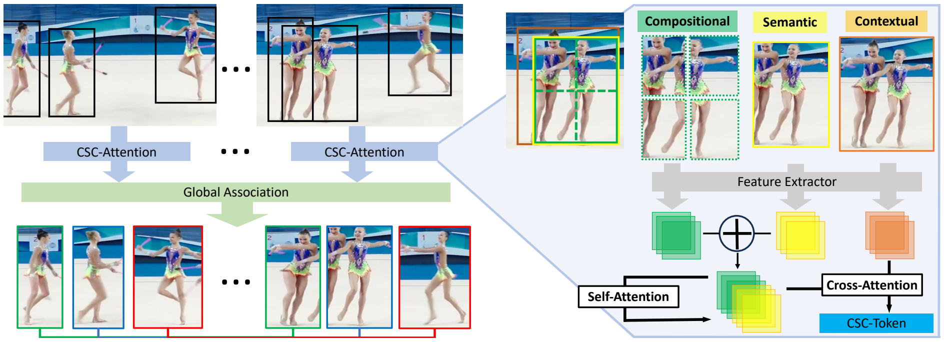


Figure 4.2: The architecture of CSC-Tracker. The left half illustrates the overall architecture. The right half is the zoomed-in CSC-Attention module. Our contributions are (1) the visual hierarchy for feature extraction and (2) the CSC-Attention module for feature fusion.

typically require additional annotations for body parts [344]. The way they fuse the features from different regions [103] is not effective in multi-object tracking cases where the background noise in the target bounding box area is usually more severe with fast-moving targets and non-static cameras.

Query-based Multi-Object Tracking. Transformer [380] is introduced to visual perception [50] after its original application in natural language processing. Later, query-based multi-object tracking methods were proposed. The early methods [249, 353] associate objects locally on adjacent time steps. Some recent methods associate targets globally in a video clip [467, 502]. GTR [502] removes secondary modules such as positional encoding, making a clean baseline to evaluate feature discriminativeness. Most recent methods improve performance by gathering information over a long period [41, 467]. However, a downside is the high requirement of computation resources, e.g., 8xA100 GPUs [41]. Instead, the improvement of our method comes from the proposed hierarchical representation. We demonstrate its state-of-the-art effectiveness and efficiency among query-based methods.

4.3 Method

In this section, we first introduce the overall architecture of CSC-Tracker. Then we describe the proposed CSC-Attention module to fuse the features from the visual hierarchy. Finally, we elaborate on the training and inference of CSC-Tracker.

4.3.1 Overall Architecture

We follow the spatio-temporal global association paradigm [400, 502] to build CSC-Tracker, whose pipeline is shown in Figure 4.2. Now, we explain the three stages of it. Notations are conditional to a generic time step t, which is the last time step where the tracks have been finalized.

Detection and Feature Extraction. Given a video clip of T frames, i.e.,$\mathcal{T} = \{t+1, ..., t+T\}$, we have the corresponding images $\mathcal{I} = \{I^{t+1}, ..., I^{t+T}\}$. Given a detector, we could derive the detections of the objects of interest on all frames in parallel, noted as $\mathcal{O} = \{O_1, ..., O_{N_t}\}$. N_t is the number of detections and $t_i \in \mathcal{T}$ $(1 \leq i \leq N_t)$ is the time step where the i-th detection, i.e.,O_i, is detected. Then, we extract the features of each detected object by a backbone network.

Token Generation by CSC-Attention. We propose CSC-Attention (to be detailed in the following section) to generate feature tokens. By CSC-Attention, we will have the object CSC-tokens $\mathcal{Q}_t^{\text{det}} \in \mathbb{R}^{N_t \times D}$, where D is the feature dimension. If we aim to associate the new-coming detections with existing trajectories, we also need the tokens to represent the existing M_t trajectories, i.e.,$\mathbf{T}_t^{\text{traj}} = \{Tk_1^{\text{traj}}, Tk_2^{\text{traj}}, ..., Tk_{M_t}^{\text{traj}}\}$. Instead of the resource-intensive iterative query passing [467] or long-time feature buffering [41], we leverage the CSC-tokens of objects on a trajectory to represent it. Within a horizon H, we represent a trajectory, Tk_j^{traj}, with the token $Q_j^{\text{traj}} \in \mathbb{R}^{H \times D}$ by combining the historical detection CSC-tokens. And all trajectory tokens are $\mathcal{Q}_t^{\text{traj}} = \{Q_1^{\text{traj}}, ..., Q_{M_t}^{\text{traj}}\}$.

Global Association. By cross-attention, we could get the association score between the set of detections and a trajectory, i.e. Tk_j^{traj}, as $S(Q_j^{\text{traj}}, \mathcal{Q}_t^{\text{det}}) \in \mathbb{R}^{H \times N_t}$. In practice, because we aim to associate between all M_t trajectories and N_t detections, we perform the cross-attention on all object queries and track queries at the same time, namely $S(\mathcal{Q}_t^{\text{traj}}, \mathcal{Q}_t^{\text{det}}) \in \mathbb{R}^{HM_t \times N_t}$. By averaging the score on the H steps in the horizon, we get the global association score $\mathbf{S}^t \in \mathbb{R}^{M_t \times N_t}$. Then, we normalize the association scores between a trajectory and objects from the same time step by softmax:

$$P(\mathbf{M}_{j,i}^t = 1 | \mathcal{Q}_t^{\text{det}}, \mathcal{Q}_t^{\text{traj}}) = \frac{\exp(\mathbf{S}_{j,i}^t)}{\sum_{k \in \{1,2,...,N_t\}} \mathbf{1}_{[t_k = t_i]} \exp(\mathbf{S}_{j,k}^t)}, \tag{4.1}$$

where the binary indicator function $\mathbf{1}_{[t_k=t_i]}$ indicates whether the i-th detection and the k-th detection are on the same time step. $\mathbf{M}^t \in \mathbb{R}^{(M_t+1)\times N_t}$ is the final global association matrix. Its dimension is of $(M_t + 1) \times N_t$ because each detection can be associated with an "empty trajectory" to start a new track. The query of the "empty trajectory" is represented by a token randomly drawn from a previous unassociated object. Also, after the association, unassociated trajectories will be considered absent on the corresponding frames. In such a fashion, we can train over a large set of detections and trajectories in parallel and conduct inference online by a sliding window. We use a uniform form for queries to represent both objects and trajectories. Thus, the global association can happen either among detections or between detections and trajectories. These two schemes of associations thus are implemented as the same and share all model modules. For online inference, we associate detections from the new-coming time step ($T = 1$) and existing trajectories.

4.3.2 CSC-Attention

Now, we explain the attention mechanism to fuse the features from the ***C****ompositional-****S****emantic-****C****ontextual* visual hierarchy. We name it CSC-Attention (right-half of Fig. 4.2).

Hierarchy Construction. There are different choices for constructing the hierarchy. To have a fair comparison with a close baseline [115], we use bounding box bins to represent object parts. Given a detection O, we divide the bounding box into 2×2 bins (to fit in GPU memory), making a set of body parts as $\mathcal{P} = \{p_1, p_2, p_3, p_4\}$. On the other hand, from a global scope, there are other targets interacting with O which are highly likely to be mismatched in the association stage. We crop the union area enclosing O and all other targets having overlap with it. We note the union area as U. Till now, we have derived the triplet $\{\mathcal{P}, O, U\}$ as the raw material for the visual hierarchy.

Feature Fusion. Among the three levels, semantic information is necessary to define a visual boundary. Compositional and contextual cues serve as the enhancement to the final representation's discriminativeness. With the extracted regions $\{\mathcal{P}, O, U\}$, we use a shared feature extractor to get their features, i.e. compositional, semantic, and contextual features. To fuse the features, we first concatenate the compositional and semantic features. Then a self-attention module is applied to help attend to the discriminative regions. Finally, the contextual features and the self-attention output are processed by a cross-attention module to get the final CSC-tokens. Before being forwarded to the global association, the tokens would be projected to a uniform dimension of D.

4.3.3 Training and Inference

Training. We train the association module by maximizing the likelihood of associating detections belonging to the same trajectory as in Eq. 4.1. We calculate the association score on all T frames of the sampled video clip simultaneously and globally. The objective thus turns to

$$\max \prod_{q=t+1}^{t+T} P(\mathbf{M}^t_{j,\tau^j_q} = 1 | \mathcal{Q}^{\text{det}}_t, \mathcal{Q}^{\text{traj}}_t), \tag{4.2}$$

where τ^j_q is the ground truth index of the detection to be associated with the j-th trajectory on the q-th time step. By applying the objective to all trajectories, the training loss is

$$L_{\text{asso}} = -\sum_{j=1}^{M_t+1} \sum_{q=t+1}^{t+T} \log P(\mathbf{M}^t_{j,\tau^j_q} = 1 | \mathcal{Q}^{\text{det}}_t, \mathcal{Q}^{\text{traj}}_t). \tag{4.3}$$

On the other hand, trajectories can also be absent on some time steps because of occlusion or target disappearance. Therefore, Eq. 4.3 has included the situation of associating a trajectory with no detection, i.e. "empty". The token for an empty detection is an arbitrary negative sample. We also have a triplet loss to pull away the feature distance between negative pairs compared to that between positive pairs:

$$\begin{aligned} L_{\text{feat}} = \max(0, \min_{u=1}^{N_P} ||\text{Att}(f(F_{p_u}), f(F_O)) - f(F_O)||^2 - \\ ||\text{Att}(f(F_O), f(F_U^{bg})) - f(F_O)||^2 + \alpha), \end{aligned} \tag{4.4}$$

where $f(\cdot)$ is the shared layers to project CNN features and $N_{\mathcal{P}}$ is the number of part patches ($N_{\mathcal{P}} = 4$ in our default setting). $\text{Att}(\cdot, \cdot)$ is the operation of cross attention. α is the margin to control the distance between positive and negative pairs. F_O and F_{p_u}

$(1 \leq u \leq N_{\mathcal{P}})$ are the semantic and compositional features. F_U^{bg} is the features of the background area in the union area U. We obtain the background features by setting the pixels of O in the area of U to 0 and forward the masked union area into the shared feature encoder $f(\cdot)$. We design Eq. 4.4 to encourage (1) the feature encoder to pay more attention to the salient and distinct area on targets while less attention to the background area and (2) the features of the background area in the union box to be discriminative from the foreground object. Finally, the training objective is

$$L = L_{\text{asso}} + L_{\text{feat}} + L_{\text{det}}, \tag{4.5}$$

where L_{det} is an optional detection loss.

Inference. We realize online inference by traversing the video with a sliding window of stride 1. On the first frame, each detection initializes a trajectory. By averaging the detection-detection association score alongside a trajectory, we get the detection-trajectory association scores, whose negative value serves as the entries in the cost matrix for the association assignment. We adopt Hungarian matching to ensure one-to-one mapping. Only when the association score is higher than $\beta = 0.3$, the pair can be associated. All unassociated detections on the new-coming frames will start new tracks.

4.4 Experiments

4.4.1 Experiment Setups

Datasets. We focus on pedestrian tracking in this paper as it is the most popular scenario and a line of previous works is available for comparison of association accuracy. On some other tracking datasets, such as TAO [82], tracking faces main difficulties at the detection stage instead of association. This causes uncontrollable noise to evaluate how discriminative the features are. For valid evaluation of visual representation distinguishness, we select three datasets, i.e., MOT17 [256], MOT20 [87] and DanceTrack [354]. DanceTrack has the largest data scale and provides an official validation set. DanceTrack contains targets mostly in the foreground but with heavy occlusion, complex motion patterns, and similar appearances. On DanceTrack,

Table 4.1: Results on MOT17 and MOT20 test sets with the private detections (FP and FN reported by ×10^4).

Tracker	HOTA↑	AssA↑	MOTA↑	IDF1↑	FP↓	FN↓	IDs↓
MOT-17 Test							
FairMOT [485]	59.3	58.0	73.7	72.3	2.75	11.7	3,303
Semi-TCL [205]	59.8	59.4	73.3	73.2	2.29	12.5	2,790
CSTrack [208]	59.3	57.9	74.9	72.6	2.38	11.4	3,567
GRTU [394]	62.0	62.1	74.9	75.0	3.20	10.8	1,812
QDTrack [269]	53.9	52.7	68.7	66.3	2.66	14.7	3,378
MAA [347]	62.0	60.2	79.4	75.9	3.73	7.77	1,452
ReMOT [438]	59.7	57.1	77.0	72.0	3.32	9.36	2,853
PermaTr [375]	55.5	53.1	73.8	68.9	2.90	11.5	3,699
ByteTrack [483]	63.1	62.0	80.3	77.3	2.55	8.37	2,196
DST-Tracker [47]	60.1	62.1	75.2	72.3	2.42	11.0	2,729
UniCorn [434]	61.7	-	77.2	75.5	5.01	7.33	5,379
OC-SORT [46]	63.2	63.2	78.0	77.5	1.51	10.8	1,950
Deep OC-SORT [238]	64.9	65.9	79.4	80.6	1.66	9.88	1,023
MotionTrack [297]	65.1	65.1	81.1	80.1	2.38	8.17	1,140
SUSHI [53]	66.5	67.8	81.1	83.1	3.23	7.32	1,149
TransCt [430]	54.5	49.7	73.2	62.2	2.31	12.4	4,614
TransTrk [353]	54.1	47.9	75.2	63.5	5.02	**8.6**	3,603
MOTR [467]	57.2	55.8	71.9	68.4	**2.1**	13.6	**2,115**
TrackFormer [249]	-	-	65.0	63.9	7.44	12.4	3,528
GTR [502]	59.1	57.0	75.3	75.1	2.68	10.9	2,859
MeMOT [41]	56.9	55.2	72.5	69.0	3,72	11.5	2,724
CSC-Tracker	**60.8**	**60.7**	**75.4**	**75.7**	2,45	10,8	2,879
MOT-20 Test							
FairMOT [485]	54.6	54.7	61.8	67.3	10.3	8.89	5,243
CSTrack [208]	54.0	54.0	66.6	68.6	2.54	14.4	3,196
GSDT [397]	53.6	52.7	67.1	67.5	3.19	13.5	3,131
RelationT [453]	56.5	55.8	67.2	70.5	6.11	10.5	4,243
MAA [347]	57.3	55.1	73.9	71.2	2.49	10.9	1,331
ByteTrack [483]	61.3	59.6	77.8	75.2	2.62	8.76	1,223
OC-SORT [46]	62.1	62.0	75.5	75.9	1.80	10.8	913
Deep OC-SORT [238]	63.9	65.7	75.6	79.2	1.69	10.8	779
MotionTrack [297]	62.8	61.8	78.0	76.5	2.86	8.42	1,165
TransCt [430]	43.5	37.0	58.5	49.6	6.42	14.6	4,695
TransTrk [353]	48.5	45.2	65.0	59.4	**2.7**	15.0	3,608
MeMOT [41]	**54.1**	**55.0**	63.7	**66.1**	4,79	13.8	**1,938**
CSC-Tracker	53.0	51.1	**65.8**	64.4	3.64	**13.**	3,948

detection is not considered as the bottleneck and the model ability of appearance discrimination becomes the key for tracking.

Evaluation Metrics. The CLEAR evaluation protocol [24] is popular for multi-object tracking evaluation but is biased to single-frame association quality [226]. MOTA is the main metric of CLEAR [24] protocol. But it is also biased to the detection quality. To provide a more accurate sense of association accuracy, we emphasize the recent HOTA [226] metric set where the metric is calculated upon the video-level association between ground truth and predictions (by default in the form of bounding boxes). In the set of metrics, AssA emphasizes the association performance, and DetA stresses the detection quality. HOTA is the main metric by considering both detection and association quality. For the result tables, we use underlined numbers to indicate the overall best value and **bold** numbers for the best query-based methods. All query-based methods are listed in blue .

Implementation. We use ResNet-50 [139] as the backbone network, which is pretrained on Crowdhuman [333] dataset first. Though advanced detector [483] is demonstrated as a key to boosting tracking performance, we want our contribution to be more from the improvement of the association stage. Therefore, on MOT17, we align the implementation with the practice of GTR [502] to use the classic CenterNet [498, 500] as the detector to make a fair comparison. The CenterNet detector is pretrained together with the backbone on Crowdhuman. For the fine-tuning of association modules on MOT17, we use a 1:1 mixture of MOT17-train and Crowdhuman. We fine-tune with only the MOT20-train for evaluation on MOT20. For DanceTrack, we use its official training set as the only training set during finetuning. The image size is set to be 1280 × 1280 during training. The image size is 1560 for the longer edge during the test. During finetuning, the detector head is also finetuned. The training iterations are set to be 20k on MOT17/MOT20 and 80k on DanceTrack. We use BiFPN [364] for the feature upsampling. For the implementation of the transformer, we use a stack of two layers of “Linear + ReLU” as the projection layers and one-layer encoders and decoders. We use AdamW [223] optimizer for training whose base learning rate is set to be 5e-5. The length of the video clip is $T = 8$ for training and $T = 24$ for inference in a sliding window for a fair comparison with GTR [502]. We use 4 × V100 GPUs as the default training device but we will see that even using only one RTX 3090 GPU for training, our method still achieves comparable performance. The training takes 4 hours on MOT17 or MOT20 and 11 hours on DanceTrack.

4.4.2 Benchmark Results

For benchmarking, we only report the performance of online tracking algorithms as offline post-processing [93, 486] gives unfair advantages and blurs the discussion about visual representation discriminativeness. We first benchmark on MOT17 and MOT20 motdtin Table 4.1. On MOT17, CSC-Tracker achieves the highest HOTA and

Table 4.2: Benchmarking results on DanceTrack test set.

Tracker	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
CenterTrack [500]	41.8	78.1	22.6	86.8	35.7
FairMOT [485]	39.7	66.7	23.8	82.2	40.8
QDTrack [269]	45.7	72.1	29.2	83.0	44.8
TraDes [415]	43.3	74.5	25.4	86.2	41.2
ByteTrack [483]	47.3	71.6	31.4	89.5	52.5
OC-SORT [46]	55.7	81.7	38.3	92.0	54.6
Deep OC-SORT [238]	61.3	82.2	45.8	92.3	61.5
DST-Tracker [47]	51.9	72.3	34.6	84.9	51.0
SUSHI [53]	63.3	80.1	50.1	88.7	63.4
TransTrk[353]	45.5	75.9	27.5	88.4	45.2
MOTR [467]	54.2	73.5	40.2	79.7	51.5
GTR [502]	48.0	72.5	31.9	84.7	50.3
CSC-Tracker (Ours)	**55.5**	**77.3**	**43.1**	**89.5**	**54.0**

AssA score among transformer-based methods. MOT20 is a more challenging dataset with crowded pedestrian flows. Though CSC-Tracker shows better performance than MeMOT [41] on MOT17, its performance is inferior on MOT20. This is probably related to the long-time heavy and frequent occlusion on MOT20. To solve this problem, the long temporal buffer of historical object appearance in MeMOT shows effectiveness. However, MeMOT requires 8×A100 GPUs for training to support such a long buffering (22 frames v.s. 8 frames by CSC-Tracker) and uses COCO [211] dataset as the additional pretraining data, which makes it not an apple-to-apple comparison.

We also benchmark on DanceTrack-test in Table 4.2. CSC-Tracker achieves state-of-the-art performance among transformer-based methods. Also, CSC-Tracker shows advanced time efficiency. For example, training on MOT17 takes MOTR [467] 2.5 days on 8×V100 GPUs while only 4 hours on 4×V100 GPUs for our proposed method. The inference speed is 6.3FPS for MOTR while 21.3FPS for our method on the same machine (V100 GPU). Compared to GTR [502], CSC-Tracker achieves a more significant outperforming on DanceTrack than on MOT17. As other variables and design choices are strictly controlled, it suggests our proposed visual hierarchy representation is more powerful than the naive bounding box features when the occlusion is heavier.

Given the aforementioned results, we have demonstrated CSC-Tracker to be the state-of-the-art among transformer-based methods with a lightweight design. More importantly, we show that the proposed hierarchical representation is more effective and efficient in discriminatively distinguishing objects. CSC-Tracker builds a new

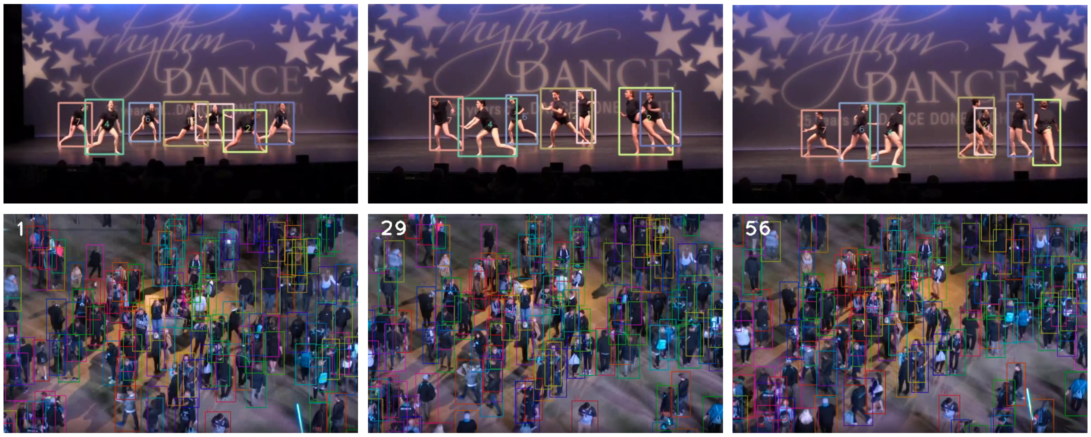


Figure 4.3: Upper line: Results from DanceTrack-test set where targets have occlusion, crossover and similar appearance. Bottom line: Results on a MOT20-test video where the pedestrians are in the crowd and heavily occluded.

baseline for future research in this line of methods. The commonly adopted techniques of query propagation and iteration [249, 353, 467], deformable attention [41, 353] and long-time feature buffering [41] are all compatible to be integrated with CSC-Tracker. Compared to the overall state-of-the-art methods, such as OC-SORT [46] and SUSHI [53], CSC-Tracker still shows inferior performance. But their performance is reported with a more advanced detector, i.e. YOLOX [110]. This makes a fair comparison hard to present. But still, there is a performance gap between the SOTAs and the transformer-based methods. For inference speed, given detections on MOT17, OC-SORT runs at 300FPS and SUSHI runs at 21FPS while CSC-Tracker runs at 93FPS.

4.4.3 Ablation Study

We now ablate the contribution of key variables in the design and implementation to the performance of CSC-Tracker. Many previous works in the multi-object tracking community follow the practice of CenterTrack [500] on MOT17 [256] to use the latter half of training video sequences as the validation set. However, this makes the ablation study on the validation set not fair because the data distribution of the training set and validation set is so close that the performance gap reflected on the validation set might degrade or even disappear on the test set. Therefore, we turn to DanceTrack [354] for the ablation study as an independent validation set is provided.

Table 4.3: Ablation of video clip length for training.

T	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
6	51.0	70.7	33.4	81.4	51.4
8	51.9	71.4	34.0	81.9	**52.2**
10	52.4	71.7	34.5	81.8	51.4
12	**52.6**	**71.9**	**34.7**	**82.0**	51.7

Table 4.4: Ablation of video clip length for Inference.

T	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
8	50.2	70.7	32.9	81.1	51.2
16	51.6	71.2	33.6	81.5	51.7
24	**51.9**	**71.4**	**34.0**	81.9	**52.2**
32	51.7	71.2	33.9	**82.0**	51.9

For the following tables, we highlight our default implementation choice in yellow, which corresponds to the entries previously reported on benchmarks to compare with other methods.

Table 4.5: The ablation study about the contribution from *semantic*, *compositional*, and *contextual* features.

Semantic	Compo.	Context.	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
✓			47.8	69.1	30.1	80.8	49.1
✓	✓		49.6	69.3	31.3	81.2	50.4
✓		✓	50.5	70.6	32.6	81.5	51.2
✓	✓	✓	51.9	71.4	34.0	81.9	52.2

Table 4.6: Different implementation choices to fit multiple training device configurations.

Training Device	Train_len	Image Size	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
1x RTX 3090-24GB	6	1280 × 1280	50.9	71.0	33.3	81.3	51.2
1x V100-32GB	8	1560 × 1560	51.2	71.7	33.7	82.0	52.0
4x V100-32GB	8	1280 × 1280	51.9	71.4	34.0	81.9	52.2

Video Length. Table 4.3 and 4.4 show the influence of video clip length in the training and inference stages respectively. The result suggests that training the association model with longer video clips can continuously improve performance. Limited by the GPU memory, we cannot increase the video clip length to longer than 12 frames here. On the contrary, during the inference stage, the sliding window size does not have a significant impact on the performance. Increasing the window size beyond a plateau will even hurt the performance.

Three levels in CSC-hierarchy. We study the contribution of each level of the

Table 4.7: Ablation of detector models.

Detector	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
CenterNet	51.9	71.4	34.0	81.9	52.2
YOLOv4 [8]	52.6	73.8	34.5	84.0	53.4
YOLOX [110]	**53.5**	**74.7**	**35.1**	**85.1**	**54.7**

Table 4.8: Ablation about feature fusion strategies.

Method	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
Bbox only	47.8	69.1	30.1	80.8	49.1
Multi-Region CNN[115]	47.4	69.5	29.5	80.8	48.6
CSC-Attention	**51.9**	**71.4**	**34.0**	**81.9**	**52.2**

CSC hierarchy in Table 4.5. Here, only the semantic information is necessary for the evaluation with bounding box-based ground truth annotations and we can manipulate the other two levels in the CSC-hierarchy by not adding the corresponding feature in the generation of the CSC-Tokens. Here we note that adding the compositional and contextual features only brings subtle computation overhead as the required self-attention and cross-attention operation are highly in parallel. Compared to only using the *semantic* feature, CSC-Tracker achieves a significant performance improvement indicated by higher HOTA and AssA scores. Also, integrating the features of the union area shows better effectiveness than solely integrating the features of body parts. This is probably because the cross attention between object body and union areas can provide critical information to compare object targets with their neighboring objects, preventing potential mismatch. On the other hand, integrating the body part features can't explicitly avoid the mismatch with other instances. Fusing the features from all the levels turns out the best choice.

Input size. We try different parameter configurations in Table 4.6 for the input clip length and image size. With only a single RTX 3090 GPU for training and inference, its performance is still comparable to the default setting with 4 × V100 GPUs. This makes the notorious computation barrier of transformer-based methods not that terrible anymore.

Detector. The highest priority for experiments is to validate the effectiveness of our proposed representations instead of racing on the leaderboard. For a fair comparison with the closest baseline GTR [502], we follow it to choose CenterNet [498] as the default detector. But CSC-Tracker is a tracking-by-detection method, flexible to integrate with different detectors. We compare CenterNet with the other detectors, i.e., YOLOv4 [8] and YOLOX [110] (used by ByteTrack, OC-SORT, SUSHI, etc.) in Table 4.7. Advanced detectors can boost tracking performance.

Fusion strategy of hierarchical features. As a main contribution of this paper, we propose CSC-Attention module to fuse the features from the CSC-hierarchy. In a naive fashion, the multi-region CNN applies a *split-and-concatenate* strategy to fuse the features from different bins inside a bounding box. We conduct a comparison with the multi-region CNN [115] in Table 4.8. Though multi-region CNN achieves improvement over the naive bounding box representation for object detection, this advantage is not observed anymore for multi-object tracking. Its performance gap with the features fused by CSC-Attention is even more significant than solely using the bounding box. This experiment suggests the effectiveness of the proposed three-level hierarchy and fusing them with the proposed CSC-Attention module.

4.4.4 Robustness to Detection Noise

With the enforcement of the part region (compositional) features, we expect CSC-Tracker to show better robustness to the noise in detections. The intuition is that even if the bounding box is not accurate, as long as a distinct part is recognized, the model should be able to track an object consistently. To validate it, we add noise to the detection positions and observe its influence on the tracking performance. We apply random shifting and random resizing to add noise. For random shifting, we have a 25% chance to shift the bounding box to the four directions independently, the shift stride is a random value in the range of $[0, \min(0.2d, 20)]$, where d is the bounding box width or height. We resize the bounding box width or height independently with a ratio of α_w and α_h, both of which are random values in the range of [0.9, 1.1]. The results on Dancetrack-val are shown in Table 4.9. Compared to the motion-based baseline OC-SORT and the full-box-only baseline GTR, CSC-Tracker shows better robustness to the noise of detections as expected.

4.4.5 Time Efficiency

Time efficiency is a bottleneck of query-based methods, especially for those using graph network [74], long-history buffers [41] or temporal aggregation [467]. Collecting

Table 4.9: Effect of detection noise (* indicates adding noise).

Method	HOTA↑	AssA↑	IDF1↑
OC-SORT [46]	52.1	35.3	51.6
OC-SORT*	49.5 (↓ 2.6)	31.3 (↓ 4.0)	48.5 (↓ 3.1)
GTR [502]	47.2	28.2	47.0
GTR*	45.0 (↓ 2.2)	26.7 (↓ 1.5)	45.6 (↓ 1.4)
CSC-Tracker	51.9	34.0	52.2
CSC-Tracker*	50.8 (↓ 1.1)	33.2 (↓ 0.8)	51.5 (↓ 0.7)

Table 4.10: Time efficiency (MOT17-test).

Method	HOTA	training time	inference speed
Transtrack [353]	54.1	18 hrs	10FPS
Trackformer [249]	-	-	7.4FPS
MOTR [467]	57.2	63 hrs	6.5FPS
TransCenter [430]	54.5	-	11FPS
GTR [502]	59.1	4 hrs	22.4FPS
CSC-Tracker	60.8	4 hrs	21.3FPS

the methods that report the time efficiency or have open-sourced implementation, we report the required training time and inference speed in Table 4.10 by default settings on MOT17. The speed is tested on Nvidia V100 GPU and the training time is evaluated on 4xV100 GPUs. CSC-Tracker achieves the best accuracy with one of the best time efficiency for both training time and the inference speed.

4.5 Conclusion

In this paper, we propose to construct discriminative visual representations by a *compositional-semantic-contextual* visual hierarchy combining different visual cues to distinguish a target. To leverage them comprehensively, we propose a CSC-Attention to gather and fuse the visual features. These are the two main contributions of this paper. We have demonstrated that they are connected to show power. The designs are integrated into CSC-Tracker for multi-object tracking. The results on multiple datasets demonstrate its efficiency and effectiveness. We hope the study of this paper can provide new knowledge in the visual representation of objects and an advanced baseline model to solve multi-object tracking problems. The method is also more robust to the detection noises and computation-economic.

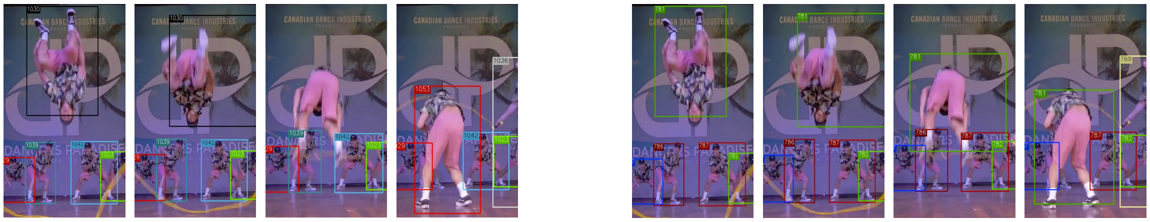
Chapter 5

Parametric Linear Filtering for Multi-Human Tracking in Crowds

5.1 Introduction

We aim to develop a motion model-based multi-object tracking (MOT) method that is robust to occlusion and non-linear motion. Most existing motion model-based algorithms assume that the tracking targets have a constant velocity within a time interval, which is called the linear motion assumption. This assumption breaks in many practical scenarios, but it still works because when the time interval is small enough, the object's motion can be reasonably approximated as linear. In this work, we are motivated by the fact that most of the errors from motion model-based tracking methods occur when occlusion and non-linear motion happen together. To mitigate the adverse effects caused, we first rethink current motion models and recognize some limitations. Then, we propose addressing them for more robust tracking performance, especially in occlusion.

As the main branch of motion model-based tracking, filtering-based methods assume a transition function to predict the state of objects on future time steps, which are called state "estimations". Besides estimations, they leverage an observation model, such as an object detector, to derive the state measurements of target objects, also called "observations". Observations usually serve as auxiliary information to help update the posteriori parameters of the filter. The trajectories are still extended



(a) SORT

(b) The proposed OC-SORT

Figure 5.1: Samples from the results on DanceTrack [354]. SORT and OC-SORT use the same detection results. On the third frame, SORT encounters an ID switch for the backflip target while ours not.

by the state estimations. Among this line of work, the most widely used one is SORT [27], which uses a Kalman filter (KF) to estimate object states and a linear motion function as the transition function between time steps. However, SORT shows insufficient tracking robustness when the object motion is non-linear, and no observations are available when updating the filter posteriori parameters.

In this work, we recognize **three limitations** of SORT. First, although the high frame rate is the key to approximating the object motion as linear, it also amplifies the model's sensitivity to the noise of state estimations. Specifically, between consecutive frames of a high frame-rate video, we demonstrate that the noise of displacement of the object can be of the same magnitude as the actual object displacement, leading to the estimated object velocity by KF suffering from a significant variance. Also, the noise in the velocity estimate will accumulate into the position estimate by the transition process. Second, the noise of state estimations by KF is accumulated along the time when there is no observation available in the update stage of KF. We show that the error accumulates very fast with respect to the time of the target object's being untracked. The noise's influence on the velocity direction often makes the track lost again even after re-association. Last, given the development of modern detectors, the object state by detections usually has lower variance than the state estimations propagated along time steps by a fixed transition function in filters. However, SORT is designed to prolong the object trajectories by state estimations instead of observations.

To relieve the negative effect of these limitations, we propose two main innovations in this work. First, we design a module to use object state observations to reduce the accumulated error during the track's being lost in a backcheck fashion. To be precise, besides the traditional stages of *predict* and *update*, we add a stage of *re-update* to correct the accumulated error. The *re-update* is triggered when a track is re-activated by associating to an observation after a period of being untracked. The *re-update* uses virtual observations on the historical time steps to prevent error accumulation. The virtual observations come from a trajectory generated using the last-seen observation before untracked and the latest observation re-activating this track as anchors. We name it *Observation-centric Re-Update (ORU)*.

Besides ORU, the assumption of linear motion provides the consistency of the object motion direction. But this cue is hard to be used in SORT's association because of the heavy noise in direction estimation. But we propose an observation-centric manner to incorporate the direction consistency of tracks in the cost matrix for the association. We name it *Observation-Centric Momentum (OCM)*. We also provide analytical justification for the noise of velocity direction estimation in practice.

The proposed method, named as **O**bservation-**C**entric **SORT** or **OC-SORT** in short, remains simple, online, real-time and significantly improves robustness over occlusion and non-linear motion. Our contributions are summarized as the following:

1. We recognize, analytically and empirically, three limitations of SORT, i.e.,sensitivity to the noise of state estimations, error accumulation over time, and being estimation-centric;
2. We propose OC-SORT for tracking under occlusion and non-linear motion by fixing SORT's limitations. It achieves state-of-the-art performance on multiple datasets in an online and real-time fashion.

5.2 Related Works

Motion Models.. Many modern MOT algorithms [27, 69, 409, 483, 500] use motion models. Typically, these motion models use Bayesian estimation [195] to predict the next state by maximizing a posterior estimation. As one of the most classic motion models, Kalman filter (KF) [172] is a recursive Bayes filter that follows a typical predict-update cycle. The true state is assumed to be an unobserved Markov process, and the measurements are observations from a hidden Markov model [298]. Given that the linear motion assumption limits KF, follow-up works like Extended KF [342] and Unscented KF [169] were proposed to handle non-linear motion with first-order and third-order Taylor approximation. However, they still rely on approximating the Gaussian prior assumed by KF and require motion pattern assumption. On the other hand, particle filters [128] solve the non-linear motion by sampling-based posterior estimation but require exponential order of computation. Therefore, these variants of Kalman filter and particle filters are rarely adopted in the visual multi-object tracking and the mostly adopted motion model is still based on Kalman filter [27].

Multi-object Tracking.. As a classic computer vision task, visual multi-object tracking is traditionally approached from probabilistic perspectives, e.g.,joint probabilistic association [16]. And modern video object tracking is usually built upon modern object detectors [308, 313, 498]. SORT [27] adopts the Kalman filter for motion-based multi-object tracking given observations from deep detectors. DeepSORT [409] further introduces deep visual features [140, 340] into object association under the framework of SORT. Re-identification-based object association[269, 409, 485] has also become popular since then but falls short when scenes are crowded and objects are represented coarsely (e.g.,enclosed by bounding boxes), or object appearance is not distinguishable. More recently, transformers [380] have been introduced to MOT [47, 249, 352, 465] to learn deep representations from both visual information and object trajectories. However, their performance still has a significant gap between state-of-the-art tracking-by-detection methods in terms of both accuracy and time efficiency.

5.3 Rethink the Limitations of SORT

In this section, we review Kalman filter and SORT [27]. We recognize some of their limitations, which are significant with **occlusion** and **non-linear object motion**. We are motivated to improve tracking robustness by fixing them.

5.3.1 Preliminaries

Kalman filter (KF) [172] is a linear estimator for dynamical systems discretized in the time domain. KF only requires the state estimations on the previous time step and the current measurement to estimate the target state on the next time step. The filter maintains two variables, the posteriori state estimate $\mathbf{x}$, and the posteriori estimate covariance matrix $\mathbf{P}$. In the task of object tracking, we describe the KF process with the state transition model $\mathbf{F}$, the observation model $\mathbf{H}$, the process noise $\mathbf{Q}$, and the observation noise $\mathbf{R}$. At each step t, given observations $\mathbf{z}_t$, KF works in an alternation of *predict* and *update* stages:

$$\begin{aligned} &\textit{predict} \begin{cases} \hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_t \hat{\mathbf{x}}_{t-1|t-1} \\ \mathbf{P}_{t|t-1} = \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^\top + \mathbf{Q}_t \end{cases}, \\ &\textit{update} \begin{cases} \mathbf{K}_t = \mathbf{P}_{t|t-1} \mathbf{H}_t^\top (\mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^\top + \mathbf{R}_t)^{-1} \\ \hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t (\mathbf{z}_t - \mathbf{H}_t \hat{\mathbf{x}}_{t|t-1}) \\ \mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_{t|t-1} \end{cases}. \end{aligned} \tag{5.1}$$

The stage of *predict* is to derive the state estimations on the next time step t. Given a measurement of target states on the next step t, the stage of *update* aims to update the posteriori parameters in KF. Because the measurement comes from the observation model $\mathbf{H}$, it is also called “observation” in many scenarios.

SORT [27] is a multi-object tracker built upon KF. The KF's state $\mathbf{x}$ in SORT is defined as $\mathbf{x} = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^\top$, where $(u,\ v)$ is the 2D coordinates of the object center in the image. s is the bounding box scale (area) and r is the bounding box aspect ratio. The aspect ratio r is assumed to be constant. The other three variables, $\dot{u}$, $\dot{v}$ and $\dot{s}$ are the corresponding time derivatives. The observation is a bounding box $\mathbf{z} = [u, v, w, h, c]^\top$ with object center position (u, v), object width w, and height h and the detection confidence c respectively. SORT assumes linear motion as the transition model $\mathbf{F}$ which leads to the state estimation as

$$u_{t+1} = u_t + \dot{u}_t \Delta t, \quad v_{t+1} = v_t + \dot{v}_t \Delta t. \tag{5.2}$$

To leverage KF (Eq 5.1) in SORT for visual MOT, the stage of *predict* corresponds to estimating the object position on the next video frame. And the observations used for the update stage usually come from a detection model. The update stage is to update Kalman filter parameters and does not directly edit the tracking outcomes.

When the time difference between two steps is constant during the transition,

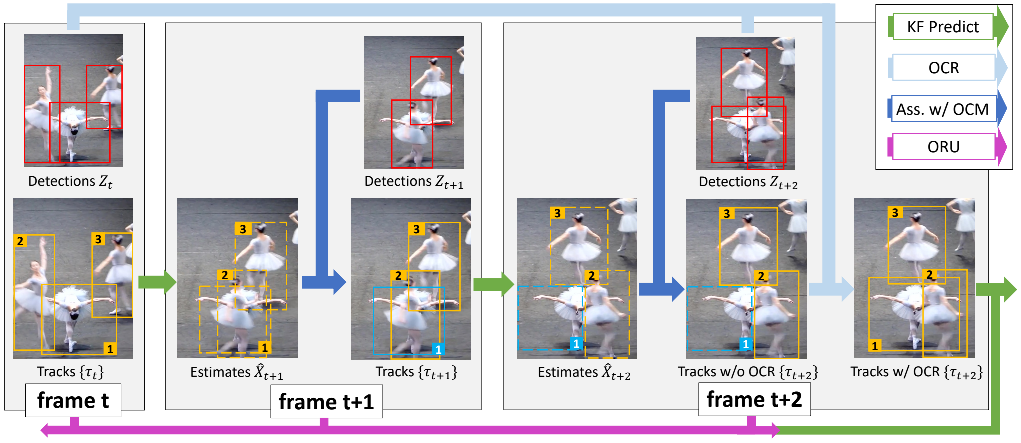


Figure 5.2: The pipeline of our proposed OC-SORT. The red boxes are detections, orange boxes are active tracks, blue boxes are untracked tracks, and dashed boxes are the estimates from KF. During association, OCM is used to add the velocity consistency cost. The target #1 is lost on the frame t+1 because of occlusion. But on the next frame, it is recovered by referring to its observation of the frame t by OCR. It being re-tracked triggers ORU from t to t+2 for the parameters of its KF.

e.g.,, the video frame rate is constant, we can set $\Delta t = 1$. When the video frame rate is high, SORT works well even when the object motion is non-linear globally, (e.g.,dancing, fencing, wrestling) because the motion of the target object can be well approximated as linear within short time intervals. However, in practice, observations are often absent on some time steps, e.g.,the target object is occluded in multi-object tracking. In such cases, we cannot update the KF parameters by the update operation as in Eq. 5.1 anymore. SORT uses the priori estimations directly as posterior. We call this **"dummy update"**, namely

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1}, \mathbf{P}_{t|t} = \mathbf{P}_{t|t-1}. \tag{5.3}$$

The philosophy behind such a design is to trust estimations when no observations are available to supervise them. We thus call the tracking algorithms following this scheme "estimation-centric". However, we will see that this estimation-centric mechanism can cause trouble when non-linear motion and occlusion happen together.

5.3.2 Limitations of SORT

In this section, we identify three main limitations of SORT which are connected. This analysis lays the foundation of our proposed method.

Sensitive to State Noise

Now we show that SORT is sensitive to the noise from KF's state estimations. To begin with, we assume that the estimated object center position follows $u \sim \mathcal{N}(\mu_u, \sigma_u^2)$ and $v \sim \mathcal{N}(\mu_v, \sigma_v^2)$, where (μ_u, μ_v) is the underlying true position. Then, if we assume that the state noises are independent on different steps, by Eq.5.2, the object speed between two time steps, $t \longrightarrow t + \Delta t$, is

$$\dot{u} = \frac{u_{t+\Delta t} - u_t}{\Delta t}, \qquad \dot{v} = \frac{v_{t+\Delta t} - v_t}{\Delta t}, \tag{5.4}$$

making the noise of estimated speed $\delta_{\dot{u}} \sim \mathcal{N}(0, \frac{2\sigma_u^2}{(\Delta t)^2})$, $\delta_{\dot{v}} \sim \mathcal{N}(0, \frac{2\sigma_v^2}{(\Delta t)^2})$. Therefore, a small Δt will amplify the noise. This suggests that SORT will suffer from the heavy noise of velocity estimation on high-frame-rate videos. The analysis above is simplified from the reality. In pratice, velocity won't be determined by the state on future time steps. For a more strict analysis, please refer to 5.7.9.

Moreover, for most multi-object tracking scenarios, the target object displacement is only a few pixels between consecutive frames. For instance, the average displacement is 1.93 pixels and 0.65 pixels along the image width and height for the MOT17 [256] training dataset. In such a case, even if the estimated position has a shift of only a single pixel, it causes a significant variation in the estimated speed. In general, the variance of the speed estimation can be of the same magnitude as the speed itself or even greater. This will not make a massive impact as the shift is only of few pixels from the ground truth on the next time step and the observations, whose variance is independent of the time, will be able to fix the noise when updating the posteriori parameters. However, we find that such a high sensitivity to state noise introduces significant problems in practice after being amplified by the error accumulation across multiple time steps when no observation is available for KF *update*.

Temporal Error Magnification

For analysis above in Eq. 5.4, we assume the noise of the object state is i.i.d on different time steps (this is a simplified version, a more detailed analysis is provided in 5.7.9). This is reasonable for object detections but not for the estimations from KF. This is because KF's estimations always rely on its estimations on previous time steps. The effect is usually minor because KF can use observation in *update* to prevent the posteriori state estimation and covariance, i.e.,$\hat{\mathbf{x}}_{t|t}$ and $\mathbf{P}_{t|t}$, deviating from the true value too far away. However, when no observations are provided to KF, it cannot use observation to update its parameters. Then it has to follow Eq. 5.3 to prolong the estimated trajectory to the next time step. Consider a track is occluded on the time steps between t and $t+T$ and the noise of speed estimate follows $\delta_{\dot{u}_t} \sim \mathcal{N}(0, 2\sigma_u^2)$, $\delta_{\dot{v}_t} \sim \mathcal{N}(0, 2\sigma_v^2)$ for SORT. On the step $t+T$, state estimation would be

$$u_{t+T} = u_t + T\dot{u}_t, \qquad v_{t+T} = v_t + T\dot{v}_t, \tag{5.5}$$

whose noise follows $\delta_{u_{t+T}} \sim \mathcal{N}(0, 2T^2\sigma_u^2)$ and $\delta_{v_{t+T}} \sim \mathcal{N}(0, 2T^2\sigma_v^2)$. So without the observations, the estimation from the linear motion assumption of KF results in a fast error accumulation with respect to time. Given σ_v and σ_u is of the same magnitude as object displacement between consecutive frames, the noise of final object position (u_{t+T}, v_{t+T}) is of the same magnitude as the object size. For instance, the size of pedestrians close to the camera on MOT17 is around 50×300 pixels. So even assuming the variance of position estimation is only 1 pixel, 10-frame occlusion can accumulate a shift in final position estimation as large as the object size. Such error magnification leads to a major accumulation of errors when the scenes are crowded.

Estimation-Centric

The aforementioned limitations come from a fundamental property of SORT that it follows KF to be estimation-centric. It allows *update* without the existence of observations and purely trusts the estimations. A key difference between state estimations and observations is that we can assume that the observations by an object detector in each frame are affected by i.i.d. noise $\delta_{\mathbf{z}} \sim \mathcal{N}(0, {\sigma'}^2)$ while the noise in

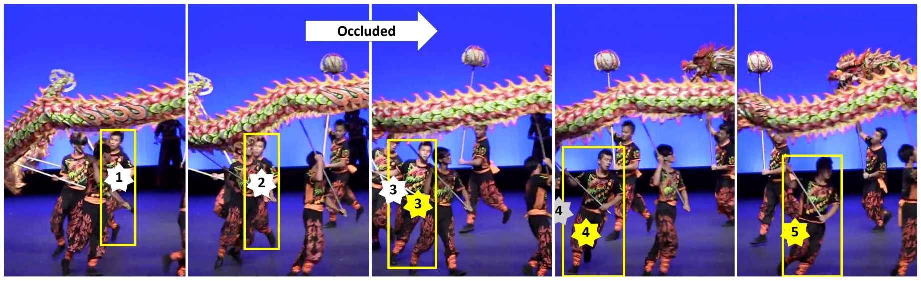


Figure 5.3: Example of how *Observation-centric Re-Update (ORU)* reduces the error accumulation when a track is broken. The target is occluded between the second and the third time step and the tracker finds it back at the third step. Yellow boxes are the state observations by the detector. White stars are the estimated centers without ORU. Yellow stars are the estimated centers fixed by ORU. The gray star on the fourth step is the estimated center without ORU and fails to match observations.

state estimations can be accumulated along the hidden Markov process. Moreover, modern object detectors use powerful object visual features [313, 340]. It makes that, even on a single frame, it is usually safe to assume $\sigma' < \sigma_u$ and $\sigma' < \sigma_v$ because the object localization is more accurate by detection than from the state estimations through linear motion assumption. Combined with the previously mentioned two limitations, being estimation-centric makes SORT suffer from heavy noise when there is occlusion and the object motion is not perfectly linear.

5.4 Observation-Centric SORT

In this section, we introduce the proposed *Observation-Centric SORT (OC-SORT)*. To address the limitations of SORT discussed above, we use the momentum of the object moving into the association stage and develop a pipeline with less noise and more robustness over occlusion and non-linear motion. The key is to design the tracker as **observation-centric** instead of **estimation-centric**. If a track is recovered from being untracked, we use an *Observation-centric Re-Update (ORU)* strategy to counter the accumulated error during the untracked period. OC-SORT also adds an *Observation-Centric Momentum (OCM)* term in the association cost. Please refer to Algorithm 1 for the pseudo-code of OC-SORT. The pipeline is shown in Fig. 5.2.

5.4.1 Observation-centric Re-Update (ORU)

In practice, even if an object can be associated again by SORT after a period of being untracked, it is probably lost again because its KF parameters have already deviated far away from the correct due to the temporal error magnification. To alleviate this problem, we propose *Observation-centric Re-Update (ORU)* to reduce the accumulated error. Once a track is associated with an observation again after a period of being untracked ("re-activation"), we backcheck the period of its being lost and re-update the parameters of KF. The re-update is based on "observations" from a virtual trajectory. The virtual trajectory is generated referring to the observations on the steps starting and ending the untracked period. For example, by denoting the last-seen observation before being untracked as $\mathbf{z}_{t_1}$ and the observation triggering the re-association as $\mathbf{z}_{t_2}$, the virtual trajectory is denoted as

$$\tilde{\mathbf{z}}_t = Traj_{\text{virtual}}(\mathbf{z}_{t_1}, \mathbf{z}_{t_2}, t), t_1 < t < t_2. \tag{5.6}$$

Then, along the trajectory of $\tilde{\mathbf{z}}_t(t_1 < t < t_2)$, we run the loop of *predict* and *re-update*. The *re-update* operation is

$$\textit{re-update} \begin{cases} \mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}_t^\top(\mathbf{H}_t\mathbf{P}_{t|t-1}\mathbf{H}_t^\top + \mathbf{R}_t)^{-1} \\ \hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\tilde{\mathbf{z}}_t - \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1}) \\ \mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\mathbf{P}_{t|t-1} \end{cases} \tag{5.7}$$

As the observations on the virtual trajectory match the motion pattern anchored by the last-seen and the latest association real observations, the update will not suffer from the error accumulated through the dummy update anymore. We call the proposed process *Observation-centric Re-Update*. It serves as an independent stage outside the *predict-update* loop and is triggered only a track is re-activated from a period of having no observations.

5.4.2 Observation-Centric Momentum (OCM)

In a reasonably short time interval, we can approximate the motion as linear. And the linear motion assumption also asks for consistent motion direction. But the

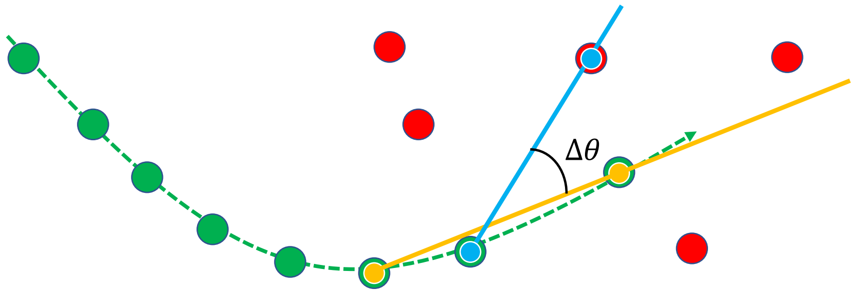


Figure 5.4: Calculation of motion direction difference in OCM. The green line indicates an existing track and the dots are the observations on it. The red dots are the new observations to be associated. The blue link and the yellow link form the directions of θ^{track} and $\theta^{\text{intention}}$ respectively. The included angle is the difference of direction $\Delta\theta$.

noise prevents us from leveraging the consistency of direction. To be precise, to determine the motion direction, we need the object state on two steps with a time difference Δt. If Δt is small, the velocity noise would be significant because of the estimation's sensitivity to state noise. If Δt is big, the noise of direction estimation can also be significant because of the temporal error magnification and the failure of linear motion assumption. As state observations have no problem of temporal error magnification that state estimations suffer from, we propose to use observations instead of estimations to reduce the noise of motion direction calculation and introduce the term of its consistency to help the association.

With the new term, given N existing tracks and M detections on the new-coming time step, the association cost matrix is formulated as

$$C(\hat{\mathbf{X}}, \mathbf{Z}) = C_{\text{IoU}}(\hat{\mathbf{X}}, \mathbf{Z}) + \lambda C_v(\mathcal{Z}, \mathbf{Z}), \tag{5.8}$$

where $\hat{\mathbf{X}} \in \mathbb{R}^{N\times 7}$ is the set of object state estimations and $\mathbf{Z} \in \mathbb{R}^{M\times 5}$ is the set of observations on the new time step. λ is a weighting factor. $\mathcal{Z}$ contains the trajectory of observations of all existing tracks. $C_{\text{IoU}}(\cdot,\cdot)$ calculates the negative pairwise IoU (Intersection over Union) and $C_v(\cdot,\cdot)$ calculates the consistency between the directions of i) linking two observations on an existing track (θ^{track}) and ii) linking a track's historical observation and a new observation ($\theta^{\text{intention}}$). C_v contains all pairs of $\Delta\theta = |\theta^{\text{track}} - \theta^{\text{intention}}|$. In our implementation, we calculate the motion direction in radians, namely $\theta = \arctan(\frac{v_1 - v_2}{u_1 - u_2})$ where (u_1, v_1) and (u_2, v_2) are the observations on two different time steps. The calculation of this is also illustrated in Figure 5.4.

Following the assumptions of noise distribution mentioned before, we can derive a closed-form probability density function of the distribution of the noise in the direction estimation. The derivation is explained in detail in 5.7.1. By analyzing the property of this distribution, we reach a conclusion that, under the linear-motion model, the scale of the noise of direction estimation is negatively correlated to the time difference between the two observation points, i.e.,Δt. This suggests increasing Δt to achieve a low-noisy estimation of θ. However, the assumption of linear motion typically holds only when Δt is small enough. Therefore, the choice of Δt requires a trade-off.

Besides ORU and OCM, we also find it empirically helpful to check a track's last presence to recover it from being lost. We thus apply a heuristic *Observation-Centric Recovery (OCR)* technique. OCR will start a second attempt of associating between the last observation of unmatched tracks to the unmatched observations after the usual association stage. It can handle the case of an object stopping or being occluded for a short time interval.

5.5 Experiments

5.5.1 Experimental Setup

Datasets. We evaluate our method on multiple multi-object tracking datasets including MOT17 [256], MOT20 [87], KITTI [112], DanceTrack [354] and CroHD [357]. MOT17 [256] and MOT20 [87] are for pedestrian tracking, where targets mostly move linearly, while scenes in MOT20 are more crowded. KITTI [112] is for pedestrian and car tracking with a relatively low frame rate of 10FPS. CroHD is a dataset for head tracking in the crowd and the results on it are included in 5.7. DanceTrack [354] is a recently proposed dataset for human tracking. For the data in DanceTrack, object localization is easy, but the object motion is highly non-linear. Furthermore, the objects have a close appearance, severe occlusion, and frequent crossovers. Considering our goal is to improve tracking robustness under occlusion and non-linear object motion, we would emphasize the comparison on DanceTrack.

Implementations. For a fair comparison, we directly apply the object detections

Table 5.1: Results on MOT17-test with the private detections. ByteTrack and OC-SORT share detections.

Tracker	HOTA↑	MOTA↑	IDF1↑	FP(10^4)↓	FN(10^4)↓	IDs↓	Frag↓	AssA↑	AssR↑
FairMOT [485]	59.3	73.7	72.3	2.75	11.7	3,303	8,073	58.0	63.6
TransCt [430]	54.5	73.2	62.2	2.31	12.4	4,614	9,519	49.7	54.2
TransTrk [352]	54.1	75.2	63.5	5.02	8.64	3,603	4,872	47.9	57.1
GRTU [394]	62.0	74.9	75.0	3.20	10.8	**1,812**	**1,824**	62.1	65.8
QDTrack [269]	53.9	68.7	66.3	2.66	14.7	3,378	8,091	52.7	57.2
MOTR [465]	57.2	71.9	68.4	2.11	13.6	2,115	3,897	55.8	59.2
PermaTr [375]	55.5	73.8	68.9	2.90	11.5	3,699	6,132	53.1	59.8
TransMOT [74]	61.7	76.7	75.1	3.62	9.32	2,346	7,719	59.9	66.5
GTR [503]	59.1	75.3	71.5	2.68	11.0	2,859	-	61.6	-
DST-Tracker [47]	60.1	75.2	72.3	2.42	11.0	2,729	-	62.1	-
MeMOT [40]	56.9	72.5	69.0	2.72	11.5	2,724	-	55.2	-
UniCorn [435]	61.7	77.2	75.5	5.01	**7.33**	5,379	-	-	-
ByteTrack [483]	63.1	**80.3**	77.3	2.55	8.37	2,196	2,277	62.0	**68.2**
OC-SORT	**63.2**	78.0	**77.5**	**1.51**	10.8	1,950	2,040	**63.2**	67.5

Table 5.2: Results on MOT20-test with private detections. ByteTrack and OC-SORT share detections.

Tracker	HOTA↑	MOTA↑	IDF1↑	FP(10^4)↓	FN(10^4)↓	IDs↓	Frag↓	AssA↑	AssR↑
FairMOT [485]	54.6	61.8	67.3	10.3	8.89	5,243	7,874	54.7	60.7
TransCt [430]	43.5	58.5	49.6	6.42	14.6	4,695	9,581	37.0	45.1
Semi-TCL [205]	55.3	65.2	70.1	6.12	11.5	4,139	8,508	56.3	60.9
CSTrack [208]	54.0	66.6	68.6	2.54	14.4	3,196	7,632	54.0	57.6
GSDT [397]	53.6	67.1	67.5	3.19	13.5	3,131	9,875	52.7	58.5
TransMOT [74]	61.9	77.5	75.2	3.42	**8.08**	1,615	2,421	60.1	66.3
MeMOT [40]	54.1	63.7	66.1	4.79	13.8	1,938	-	55.0	-
ByteTrack [483]	61.3	**77.8**	75.2	2.62	8.76	1,223	1,460	59.6	66.2
OC-SORT	**62.1**	75.5	**75.9**	**1.80**	10.8	**913**	**1,198**	**62.0**	**67.5**

Table 5.3: Results on DanceTrack test set. Methods in the blue block share the same detections.

Tracker	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
CenterTrack [500]	41.8	78.1	22.6	86.8	35.7
FairMOT [485]	39.7	66.7	23.8	82.2	40.8
QDTrack [269]	45.7	72.1	29.2	83.0	44.8
TransTrk[352]	45.5	75.9	27.5	88.4	45.2
TraDes [415]	43.3	74.5	25.4	86.2	41.2
MOTR [465]	54.2	73.5	40.2	79.7	51.5
GTR [503]	48.0	72.5	31.9	84.7	50.3
DST-Tracker [47]	51.9	72.3	34.6	84.9	51.0
SORT [27]	47.9	72.0	31.2	91.8	50.8
DeepSORT [409]	45.6	71.0	29.7	87.8	47.9
ByteTrack [483]	47.3	71.6	31.4	89.5	52.5
OC-SORT	54.6	**80.4**	40.2	89.6	54.6
OC-SORT + Linear Interp	**55.1**	**80.4**	**40.4**	**92.2**	**54.9**

Table 5.4: Results on KITTI-test. Our method uses the same detections as PermaTr [375]

	Car					Pedestrian				
Tracker	HOTA↑	MOTA↑	AssA↑	IDs↓	Frag↓	HOTA↑	MOTA↑	AssA↑	IDs↓	Frag↓
IMMDP [418]	68.66	82.75	69.76	211	181	-	-	-	-	-
SMAT [118]	71.88	83.64	72.13	198	294	-	-	-	-	-
TrackMPNN [306]	72.30	87.33	70.63	481	237	39.40	52.10	35.45	626	669
MPNTrack [35]	-	-	-	-	-	45.26	46.23	47.28	397	1,078
CenterTr [500]	73.02	88.83	71.18	254	227	40.35	53.84	36.93	425	618
LGM [388]	73.14	87.60	72.31	448	**164**	-	-	-	-	-
TuSimple [69]	71.55	86.31	71.11	292	218	45.88	57.61	47.62	246	651
PermaTr [375]	**77.42**	**90.85**	**77.66**	275	271	47.43	65.05	43.66	483	703
OC-SORT	74.64	87.81	74.52	257	318	52.95	62.00	57.81	**181**	**598**
OC-SORT + HP	76.54	90.28	76.39	250	280	**54.69**	**65.14**	**59.08**	184	609

from existing baselines. For MOT17, MOT20, and DanceTrack, we use the publicly available YOLOX [110] detector weights by ByteTrack [483]. For KITTI [112], we use the detections from PermaTrack [375] publicly available in the official release[1]. For ORU, we generate the virtual trajectory during occlusion with the constant-velocity assumption. Therefore, Eq. 5.6 is adopted as $\tilde{\mathbf{z}}_t = \mathbf{z}_{t_1} + \frac{t-t_1}{t_2-t_1}(\mathbf{z}_{t_2} - \mathbf{z}_{t_1}), t_1 < t < t_2$. For OCM, the velocity direction is calculated using the observations three time steps apart, i.e.,$\Delta t = 3$. The direction difference is measured by the absolute difference of angles in radians. We set $\lambda = 0.2$ in Eq. 5.8. Following the common practice of SORT, we set the detection confidence threshold at 0.4 for MOT20 and 0.6 for other datasets. The IoU threshold during association is 0.3.

Metrics. We adopt HOTA [226] as the main metric as it maintains a better balance between the accuracy of object detection and association [226]. We also emphasize AssA to evaluate the association performance. IDF1 is also used for association performance evaluation. Other metrics we report, such as MOTA, are highly related to detection performance. It is fair to use these metrics only when all methods use the same detections for tracking, which is referred to as "public tracking" as reported in 5.7.5.

[1] https://github.com/TRI-ML/permatrack/

5.5.2 Benchmark Results

Here we report the benchmark results on multiple datasets. We put all methods that use the shared detection results in a block at the bottom of each table.

Table 5.5: Ablation on MOT17-val and DanceTrack-val.

			MOT17-val			DanceTrack-val		
ORU	OCM	OCR	HOTA↑	AssA↑	IDF1↑	HOTA↑	AssA↑	IDF1↑
			64.9	66.8	76.9	47.8	31.0	48.3
✓			66.3	68.0	77.2	48.5	32.2	49.8
✓	✓		66.4	**69.0**	**77.8**	**52.1**	35.0	50.6
✓	✓	✓	**66.5**	68.9	77.7	**52.1**	**35.3**	**51.6**

Table 5.6: Ablation on the trajectory hypothesis in ORU.

	MOT17-val			DanceTrack-val		
	HOTA↑	AssA↑	IDF1↑	HOTA↑	AssA↑	IDF1↑
Const. Speed	**66.5**	**68.9**	**77.7**	**52.1**	**35.3**	**51.6**
GPR	63.1	65.2	75.7	49.5	33.7	49.6
Linear Regression	64.3	66.5	76.0	49.3	33.4	49.2
Const. Acceleration	66.2	67.9	77.4	51.3	34.8	50.9

Table 5.7: Influence from the value of Δt in OCM.

	MOT17-val			DanceTrack-val		
	HOTA↑	AssA↑	IDF1↑	HOTA↑	AssA↑	IDF1↑
$\Delta t = 1$	66.1	67.5	76.9	51.3	34.3	51.3
$\Delta t = 2$	66.3	68.0	77.3	**52.2**	**35.4**	51.4
$\Delta t = 3$	**66.5**	**68.9**	**77.7**	52.1	35.3	51.6
$\Delta t = 6$	66.0	67.5	76.9	52.1	**35.4**	**51.8**

MOT17 and MOT20. We report OC-SORT's performance on MOT17 and MOT20 in Table 5.1 and Table 5.2 using private detections. To make a fair comparison, we use the same detection as ByteTrack [483]. OC-SORT achieves performance comparable to other state-of-the-art methods. Our gains are especially significant in MOT20 under severe pedestrian occlusion, setting a state-of-the-art HOTA of 62.1. As our method is designed to be simple for better generalization, we do not use adaptive detection thresholds as in ByteTrack. Also, ByteTrack uses more detections of low-confidence to achieve higher MOTA scores but we keep the detection confidence threshold the same as on other datasets, which is the common practice in the community. We inherit the linear interpolation on the two datasets as baseline methods for a fair comparison. To more clearly discard the variance from the detector, we also perform public tracking on MOT17 and MOT20, which is reported in Table 5.12 and Table 5.13 I 5.7.5. OC-SORT still outperforms the existing state-of-the-art in public tracking settings.

DanceTrack. To evaluate OC-SORT under challenging non-linear object motion, we report results on the DanceTrack in Table 5.3. OC-SORT sets a new state-of-the-art, outperforming the baselines by a great margin under non-linear object motions. We compare the tracking results of SORT and OC-SORT under extreme non-linear situations in Fig.5.1 and more samples are available in Fig. 5.8 in 5.7.7. We also visualize the output trajectories by OC-SORT and SORT on randomly selected DanceTrack video clips in Fig. 5.9 in 5.7.7. For multi-object tracking in occlusion and non-linear motion, the results on DanceTrack are strong evidence of the effectiveness of OC-SORT.

KITTI. In Table 5.4 we report the results on the KITTI dataset. For a fair comparison, we adopt the detector weights by PermaTr [375] and report its performance in the table as well. Then, we run OC-SORT given the shared detections. As initializing SORT's track requires continuous tracking across several frames ("minimum hits"), we observe that the results not recorded during the track initialization make a significant difference. To address this problem, we perform offline head padding (HP) post-processing by writing these entries back after finishing the online tracking stage. The results of the car category on KITTI show an essential shortcoming of the default implementation version of OC-SORT that it chooses the IoU matching for the association. When the object velocity is high or the frame rate is low, the IoU of object bounding boxes between consecutive frames can be very low or even zero. This issue does not come from the intrinsic design of OC-SORT and is widely observed when using IoU as the association cue. Adding other cues [314, 492, 500] and appearance similarity [239, 409] have been demonstrated [409] efficient to solve this. In contrast to the relatively inferior car tracking performance, OC-SORT improves pedestrian tracking performance to a new state-of-the-art. Using the same detections, OC-SORT achieves a large performance gap over PermaTr with 10x faster speed.

The results on multiple benchmarks have demonstrated the effectiveness and efficiency of OC-SORT. We note that we use a shared parameter stack across datasets. Carefully tuning the parameters can probably further boost the performance. For example, the adaptive detection threshold is proven useful in previous work [483]. Besides the association accuracy, we also care about the inference speed. Given off-the-shelf detections, OC-SORT runs at 793 FPS on an Intel i9-9980XE CPU @ 3.00GHz. Therefore, OC-SORT can still run in an online and real-time fashion.

5.5.3 Ablation Study

Component Ablation. We ablate the contribution of proposed modules on the validation sets of MOT17 and DanceTrack in Table 5.5. The splitting of the MOT17 validation set follows a popular convention [500]. The results demonstrate the efficiency of the proposed modules in OC-SORT. The results show that the performance gain from ORU is significant on both datasets but OCM only shows good help on DanceTrack dataset where object motion is more complicated and the occlusion is heavy. It suggests the effectiveness of our proposed method to improve tracking robustness in occlusion and non-linear motion.

Virtual Trajectory in ORU. For simplicity, we follow the naive hypothesis of constant speed to generate a virtual trajectory in ORU. There are other alternatives like constant acceleration, regression-based fitting such as Linear Regression (LR) or Gaussian Process Regression (GPR), and Near Constant Acceleration Model (NCAM) [162]. The results of comparing these choices are shown in Table 5.6. For GPR, we use the RBF kernel [57] $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x}-\mathbf{x}'||^2}{50}\right)$. We provide more studies on the kernel configuration in 5.7.2. The results show that local hypotheses such as Constant Speed/Acceleration perform much better than global hypotheses such as LR and GPR. This is probably because, as virtual trajectory generation happens in an online fashion, it is hard to get a reliable fit using only limited data points on historical time steps.

Δt in OCM. There is a trade-off when choosing the time difference Δt in OCM (Section 5.4). A large Δt decreases the noise of velocity estimation. but is also likely to discourage approximating object motion as linear. Therefore, we study the influence of varying Δt in Table 5.7. Our results agree with our analysis that increasing Δt from $\Delta t = 1$ can boost the association performance. But increasing Δt

higher than the bottleneck instead hurts the performance because of the difficulty of maintaining the approximation of linear motion.

5.6 Conclusion

We analyze the popular motion-based tracker SORT and recognize its intrinsic limitations from using Kalman filter. These limitations significantly hurt tracking accuracy when the tracker fails to gain observations for supervision - likely caused by unreliable detectors, occlusion, or fast and non-linear target object motion. To address these issues, we propose *Observation-Centric SORT (OC-SORT)*. OC-SORT is more robust to occlusion and non-linear object motion while keeping simple, online, and real-time. In our experiments on diverse datasets, OC-SORT significantly outperforms the state-of-the-art. The gain is especially significant for multi-object tracking under occlusion and non-linear object motion.

5.7 More Analysis

5.7.1 Velocity Direction Variance in OCM

In this section, we work on the setting of linear motion with noisy states. We provide proof that the trajectory direction estimation has a smaller variance if the two states we use for the estimation have a larger time difference. We assume the motion model is $\mathbf{x}_t = f(t) + \epsilon$ where ϵ is gaussian noise and the ground-truth center position of the target is (μ_{u_t}, μ_{v_t}) at time step t. Then the true motion direction between the two time steps is

$$\theta = \arctan(\frac{\mu_{v_{t_1}} - \mu_{v_{t_2}}}{\mu_{u_{t_1}} - \mu_{u_{t_2}}}). \tag{5.9}$$

And we have $|\mu_{v_{t_1}} - \mu_{v_{t_2}}| \propto |t_1 - t_2|$, $|\mu_{u_{t_1}} - \mu_{u_{t_2}}| \propto |t_1 - t_2|$. As the detection results do not suffer from the error accumulation due to propagating along Markov process as Kalman filter does, we can assume the states from observation suffers some i.i.d. noise, i.e., $u_t \sim \mathcal{N}(\mu_{u_t}, \sigma_u^2)$ and $v_t \sim \mathcal{N}(\mu_{v_t}, \sigma_v^2)$. We now analyze the noise of the estimated $\tilde{\theta} = \frac{v_{t_1} - v_{t_2}}{u_{t_1} - u_{t_2}}$ by two observations on the trajectory. Because the function of $\arctan(\cdot)$ is monotone over the whole real field, we can study $\tan\tilde{\theta}$ instead which simplifies the analysis. We denote $w = u_{t_1} - u_{t_2}$, $y = v_{t_1} - v_{t_2}$, and $z = \frac{y}{w}$, first we can see that y and w jointly form a Gaussian distribution:

$$\begin{bmatrix} y \\ w \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_y \\ \mu_w \end{bmatrix}, \begin{bmatrix} \sigma_y^2 & \rho\sigma_y\sigma_w \\ \rho\sigma_y\sigma_w & \sigma_w^2 \end{bmatrix}\right), \tag{5.10}$$

where $\mu_y = \mu_{v_{t_1}} - \mu_{v_{t2}}$, $\mu_w = \mu_{u_{t_1}} - \mu_{u_{t_2}}$, $\sigma_w = \sqrt{2}\sigma_u$ and $\sigma_y = \sqrt{2}\sigma_v$, and ρ is the correlation coefficient between y and w. We can derive a closed-form solution of the probability density function [147] of z as

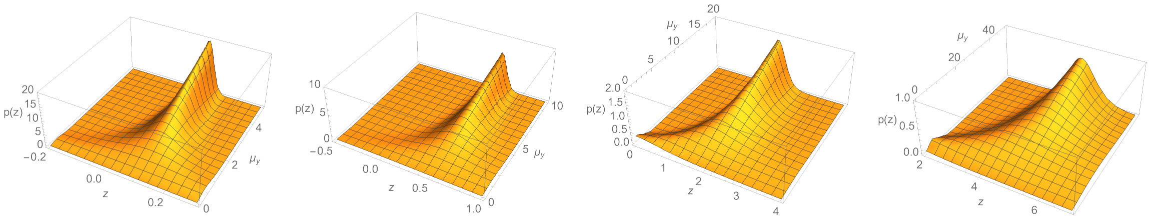
$$\begin{aligned} p(z) =& \frac{g(z)e^{\frac{g(z)^2 - \alpha r(z)^2}{2\beta^2 r(z)^2}}}{\sqrt{2\pi}\sigma_w\sigma_y r(z)^3}\left[\Phi\left(\frac{g(z)}{\beta r(z)}\right) - \Phi\left(-\frac{g(z)}{\beta r(z)}\right)\right] \\ &+ \frac{\beta e^{-2\alpha/\beta}}{\pi\sigma_w\sigma_y r(z)^2} \end{aligned} \tag{5.11}$$

where

$$\begin{aligned} r(z) &= \sqrt{\frac{z^2}{\sigma_y^2} - \frac{2\rho z}{\sigma_y\sigma_w} + \frac{1}{\sigma_w^2}}, \\ g(z) &= \frac{\mu_y z}{\sigma_y^2} - \frac{\rho(\mu_y + \mu_w z)}{\sigma_y\sigma_w} + \frac{\mu_w}{\sigma_w^2}, \\ \alpha &= \frac{\mu_w^2 + \mu_y^2}{\sigma_y^2} - \frac{2\rho\mu_y\mu_w}{\sigma_w\sigma_y}, \qquad \beta = \sqrt{1-\rho^2}, \end{aligned} \tag{5.12}$$

and Φ is the cumulative distribution function of the standard normal. Without loss of generality, we can assume $\mu_w > 0$ and $\mu_y > 0$ because negative ground-truth displacements enjoy the same property. This solution has a good property that larger μ_w or μ_y makes the probability density at the true value, i.e. $\mu_z = \frac{\mu_y}{\mu_w}$, higher, and the tails decay more rapidly. So the estimation of $\arctan\theta$, also θ, has smaller noise when μ_w or μ_y is larger. Under the assumption of linear motion, we thus should select two observations with a large temporal difference to estimate the direction.

It is reasonable to assume the noise of detection along the u-axis and v-axis are independent so $\rho = 0$. And when representing the center position in pixel, it is also moderate to assume $\sigma_w = \sigma_y = 1$ (also for the ease of presentation). Then, with different true value of $\mu_z = \frac{\mu_y}{\mu_w}$, the visualizations of $p(z)$ over z and μ_y are shown in Figure 5.5. The visualization demonstrates our analysis above. Moreover, it shows



(a) $\mu_z = 0.1$ (b) $\mu_z = 0.5$ (c) $\mu_z = 2$ (d) $\mu_z = 5$

Figure 5.5: The probability density of $z = \tan\theta$ under different true value of z, i.e. $\mu_z = \frac{\mu_y}{\mu_w}$. We set μ_y and z as two variables. It shows that under different settings of true velocity direction when μ_y is smaller, the probability of estimated value with a significant shift from the true value is higher. As μ_y is proportional to the time difference of the two selected observations under linear motion assumption, it relates to the case that the two steps for velocity direction estimation has a shorter time difference.

that when the value of μ_y or μ_w is small, the cluster peak of the distribution at μ_z is not significant anymore, as the noise σ_y and σ_w can be dominant. Considering the visualization shows that happens when μ_y is close to σ_y, this can happen when we estimate the speed by observations from two consecutive frames because the variance of observation can be close to the absolute displacement of object motion. This makes another support to our analysis in the main paper about the sensitivity to state estimation noise.

Table 5.8: Ablation study about the interpolation post-processing.

	MOT17-val				DanceTrack-val			
	HOTA↑	AssA↑	MOTA↑	IDF1↑	HOTA↑	AssA↑	MOTA↑	IDF1↑
w/o interpolation	66.5	68.9	74.9	77.7	52.1	35.3	87.3	51.6
Linear Interpolation	**68.0**	**69.9**	**77.9**	**79.3**	**52.8**	**35.6**	**89.8**	**52.1**
GPR Interpolation	65.2	67.0	72.9	75.9	51.6	35.0	86.1	51.2

5.7.2 Interpolation by Gaussian Progress Regression

Interpolation as post-processing. Although we focus on developing an online tracking algorithm, we are also interested in whether post-process can further optimize the tracking results in diverse conditions. Despite the failure of GPR in online tracking in Table 5.6, we continue to study if GPR is better suited for interpolation in Table 5.8.

Table 5.9: Ablation study about using Gaussian Process Regression for object trajectory interpolation. LI indicates Linear Interpolation, which is used to interpolate the trajectory before smoothing the trajectory by GPR. MT indicates Median Trick for kernel choice in regression. L_τ is the length of trajectory.

	MOT17-val				DanceTrack-val			
Interpolation Method	HOTA	AssA	MOTA	IDF1	HOTA	AssA	MOTA	IDF1
w/o interpolation	66.5	68.9	74.9	77.7	52.1	35.3	87.3	51.6
Linear Interpolation	**69.6**	**69.9**	**77.9**	**79.3**	52.8	**35.6**	89.8	**52.1**
GPR Interp, $l = 1$	66.2	67.6	74.3	76.6	51.8	35.0	86.6	50.8
GPR Interp, $l = 5$	66.3	67.0	72.9	75.9	51.8	35.1	86.5	51.1
GPR Interp, $l = L_\tau$	66.1	67.0	73.1	77.8	51.6	35.1	86.4	50.7
GPR Interp, $l = 1000/L_\tau$	65.9	67.0	73.0	77.8	51.8	35.0	86.9	51.0
GPR Interp, $l = \mathrm{MT}(\tau)$	65.9	67.0	73.1	77.8	51.7	35.1	86.7	50.9
LI + GPR Smoothing, $l = 1$	69.5	69.6	77.8	**79.3**	52.8	**35.6**	**89.9**	**52.1**
LI + GPR Smoothing, $l = 5$	69.5	69.7	77.8	**79.3**	52.9	34.9	89.7	**52.1**
LI + GPR Smoothing, $l = L_\tau$	69.6	69.5	77.8	79.2	52.9	**35.6**	**89.9**	**52.1**
LI + GPR Smoothing, $l = 1000/L_\tau$	69.5	**69.9**	77.8	**79.3**	**53.0**	**35.6**	**89.9**	**52.1**
LI + GPR Smoothing, $l = \mathrm{MT}(\tau)$	69.5	69.6	77.8	**79.3**	52.8	**35.6**	89.8	**52.1**

We compare GPR with the widely-used linear interpolation. The maximum gap for interpolation is set as 20 frames and we use the same kernel for GPR as mentioned above. The results suggest that the GPR's non-linear interpolation is simply not efficient. We think this is due to limited data points which results in an inaccurate fit of the object trajectory. Further, the variance in regressor predictions introduces extra noise. Although GPR interpolation decreases the performance on MOT17-val significantly, its negative influence on DanceTrack is relatively minor where the object motion is more non-linear. We believe how to fit object trajectory with non-linear hypothesis still requires more study.

From the analysis in the main paper, the failure of SORT can mainly result from occlusion (lack of observations) or the non-linear motion of objects (the break of the linear-motion assumption). So the question arises naturally whether we can extend SORT free of the linear-motion assumption or at least more robust when it breaks.

One way is to extend from KF to non-linear filters, such as EKF [172, 342] and UKF [169]. However, for real-world online tracking, they can be hard to be adopted as they need knowledge about the motion pattern or still rely on the techniques fragile to non-linear patterns, such as linearization [170]. Another choice is to gain the knowledge beyond linearity by regressing previous trajectory, such as combing Gaussian Process (GP) [183, 309, 406]: given a observation $\mathbf{z}_\star$ and a kernel function

$k(\cdot,\cdot)$, GP defines gaussian functions with mean $\mu_{\mathbf{z}_\star}$ and variance $\Sigma_{\mathbf{z}_\star}$ as

$$\begin{aligned}\mu_{\mathbf{z}_\star} &= \mathbf{k}_\star^\top[\mathbf{K}+\sigma^2\mathbf{I}]^{-1}\mathbf{y},\\ \Sigma_{\mathbf{z}_\star} &= k(\mathbf{z}_\star,\mathbf{z}_\star) - \mathbf{k}_\star^\top[\mathbf{K}+\sigma^2\mathbf{I}]^{-1}\mathbf{k}_\star,\end{aligned} \tag{5.13}$$

where $\mathbf{k}_\star$ is the kernel matrix between the input and training data and $\mathbf{K}$ is the kernel matrix over training data, $\mathbf{y}$ is the output of data. Until now, we have shown the primary study of using Gaussian Process Regression (GPR) in the online generation of the virtual trajectory in ORU and offline interpolation. But neither of them successfully boosts the tracking performance. Now, We continue to investigate in detail the chance of combining GPR and SORT for multi-object tracking for interpolation as some designs are worth more study.

5.7.3 Choice of Kernel Function in Gaussian Process

The kernel function is a key variable of GPR. There is not a generally efficient guideline to choose the kernel for Gaussian Process Regression though some basic observations are available [94]. When there is no additional knowledge about the time sequential data to fit, the RBF kernel is one of the most common choices:

$$k(\mathbf{x},\mathbf{x}') = \sigma^2 \exp\left(-\frac{||\mathbf{x}-\mathbf{x}'||^2}{2l^2}\right), \tag{5.14}$$

where l is the lengthscale of the data to be fit. It determines the length of the "wiggles" of the target function. σ^2 is the output variance that determines the average distance of the function away from its mean. This is usually just a scale factor [94]. GPR is considered sensitive to l in some situations. So we conduct an ablation study over it in the offline interpolation to see if we can use GPR to outperform the linear interpolation widely used in multi-object tracking.

5.7.4 GPR for Offline Interpolation

We have presented the use of GPR in online virtual trajectory fitting and offline interpolation where we use $l^2 = 25$ and $\sigma = 1$ for the kernel in Eq. 5.14. Further, we make a more thorough study of the setting of GPR. We follow the settings of

Table 5.10: Results on CroHD Head Tracking dataset [357]. Our method uses the detections from HeadHunter [357] or FairMOT [485] to generate new tracks.

Tracker	HOTA↑	MOTA↑	IDF1↑	FP(10^4)↓	FN(10^4)↓	IDs↓	Frag↓
HeadHunter [357]	36.8	57.8	53.9	**5.18**	30.0	4,394	15,146
HeadHunter dets + OC-SORT	39.0	60.0	56.8	**5.18**	28.1	**4,122**	10,483
FairMOT [485]	43.0	60.8	62.8	11.8	19.9	12,781	41,399
FairMOT dets + OC-SORT	**44.1**	**67.9**	**62.9**	10.2	**16.4**	4,243	**10,122**

Table 5.11: Results on DanceTrack test set. "Ours (MOT17)" uses the YOLOX detector trained on MOT17-training set.

Tracker	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
SORT	47.9	72.0	31.2	91.8	50.8
OC-SORT	55.1	80.3	38.0	89.4	54.2
OC-SORT (MOT17)	48.6	71.0	33.3	84.2	51.5

experiments in the main paper that only trajectories longer than 30 frames are put into interpolation. And the interpolation is only applied to the gap shorter than 20 frames. We conduct the experiments on the validation sets of MOT17 and DanceTrack.

For the value of l, we try fixed values, i.e. $l = 1$ and $l = 5$ ($2l^2 = 50$), value adaptive to trajectory length, i.e. $l = L_\tau$ and $l = 1000/L_\tau$, and the value output by Median Trick (MT) [109]. The training data is a series of quaternary $[u, v, w, h]$, normalized to zero-mean before being fed into training. The results are shown in Table 5.9. Linear interpolation is simple but builds a strong baseline as it can stably improve the tracking performance concerning multiple metrics. Directly using GPR to interpolate the missing points hurts the performance and the results of GPR are not sensitive to the setting of l.

There are two reasons preventing GPR from accurately interpolating missing segments. First, the trajectory is usually limited to at most hundreds of steps, providing very limited data points for GPR training to converge. On the other hand, the missing intermediate data points make the data series discontinuous, causing a huge challenge. We can fix the second issue by interpolating the trajectory with Linear Interpolation (LI) first and then smoothing the interpolated steps by GPR. This outperforms LI on DanceTrack but still regrades the performance by LI on MOT17. This is likely promoted by the non-linear motion on DanceTrack. By fixing

Table 5.12: Results on MOT17 test set with the public detections. LI indicates Linear Interpolation.

Tracker	HOTA$\uparrow$	MOTA$\uparrow$	IDF1$\uparrow$	FP(10^4)$\downarrow$	FN(10^4)$\downarrow$	IDs$\downarrow$	Frag$\downarrow$	AssA$\uparrow$	AssR$\uparrow$
CenterTrack [500]	-	61.5	59.6	1.41	20.1	2,583	-	-	-
QDTrack [269]	-	64.6	65.1	1.41	18.3	2,652	-	-	-
Lif_T [151]	51.3	60.5	65.6	1.50	20.7	1,189	3,476	54.7	59.0
TransCt [430]	51.4	68.8	61.4	2.29	**14.9**	4,102	8,468	47.7	52.8
TrackFormer [249]	-	**62.5**	60.7	3.28	17.5	2,540	-	-	-
OC-SORT	52.4	58.2	65.1	**0.44**	23.0	**784**	2,006	**57.6**	63.5
OC-SORT + LI	**52.9**	59.4	65.7	0.66	22.2	801	**1,030**	57.5	**63.9**

Table 5.13: Results on MOT20 test set with the public detections. LI indicates Linear Interpolation.

Tracker	HOTA$\uparrow$	MOTA$\uparrow$	IDF1$\uparrow$	FP(10^4)$\downarrow$	FN(10^4)$\downarrow$	IDs$\downarrow$	Frag$\downarrow$	AssA$\uparrow$	AssR$\uparrow$
MPNTrack [35]	46.8	57.6	59.1	17.0	20.1	1,210	1,420	47.3	52.7
TransCt [430]	43.5	61.0	49.8	4.92	**14.8**	4,493	8,950	36.1	44.5
ApLift [152]	46.6	58.9	56.5	1.77	19.3	2,241	2,112	45.2	48.1
TMOH [346]	48.9	60.1	61.2	3.80	16.6	2,342	4,320	48.4	52.9
LPC_MOT [79]	49.0	56.3	62.5	1.17	21.3	1,562	1,865	52.4	54.7
OC-SORT	54.3	59.9	67.0	**0.44**	20.2	554	2,345	59.5	65.1
OC-SORT + LI	**55.2**	**61.7**	**67.9**	0.57	19.2	**508**	**805**	**59.8**	**65.9**

the missing data issue of GPR, GPR can have a more accurate trajectory fitting over LI for the non-linear trajectory cases. But considering the outperforming from GPR is still minor compared with the Linear Interpolation-only version and GPR requires much heavier computation overhead, we do not recommend using such a practice in most multi-object tracking tasks. More careful and deeper study is still required on this problem.

5.7.5 Results on More Benchmarks

Results on HeadTrack [357].. When considering tracking in the crowd, focusing on only a part of the object can be beneficial [48] as it usually suffers less from occlusion than the full body. This line of study is conducted over hand tracking [252, 334], human pose [421] and head tracking [19, 281, 357] for a while. Moreover, with the knowledge of more fine-grained part trajectory, it can be useful in downstream tasks, such as action recognition [98, 106] and forecasting [43, 56, 182, 189]. As we are interested in the multi-object tracking in the crowd, we also evaluate the proposed OC-SORT on a recently proposed human head tracking dataset CroHD [357]. To

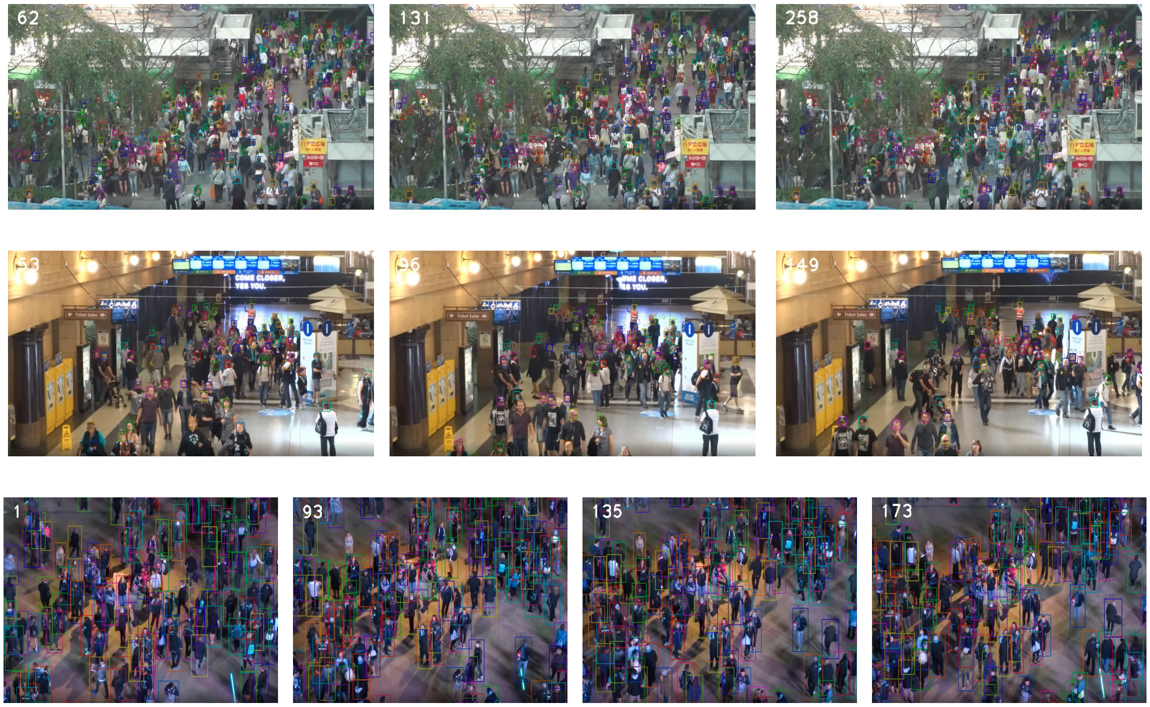


Figure 5.6: The visualization of the output of OC-SORT on randomly selected samples from the test set of HeadTrack [357] (the first two rows) and MOT20 [87] (the bottom row). These two datasets are both challenging because of the crowded scenes where pedestrians have heavy occlusion with each other. OC-SORT achieves superior performance on both datasets.

make a fair comparison on only the association performance, we adopt OC-SORT by directing using the detections from existing tracking algorithms. The results are shown in Table 5.10. The detections of FairMOT [485] and HeadHunter [357] are extracted from their tracking results downloaded from the official leaderboard [2]. We use the same parameters for OC-SORT as on the other datasets. The results suggest a significant tracking performance improvement compared with the previous methods [357, 485] for human body part tracking. But the tracking performance is still relatively low (HOTA=$\sim$ 40). It is highly related to the difficulty of having accurate detections of tiny objects. Some samples from the test set of HeadTrack are shown in the first two rows of Figure 5.6.

Public Tracking on MOT17 and MOT20.. Although we use the same object detectors as some selected baselines, there is still variances in detections when compared with other methods. Therefore, we also report with the public detections on MOT17/MOT20 in Table 5.12 and Table 5.13. OC-SORT still outperforms the existing state-of-the-arts in the public tracking setting. And the outperforming of OC-SORT is more significant on MOT20 which has more severe occlusion scenes. Some samples from the test set of MOT20 are shown in the last row in Figure 5.6.

5.7.6 Pseudo-code of OC-SORT

See the pseudo-code of OC-SORT in Algorithm. 1.

5.7.7 More Results on DanceTrack

To gain more intuition about the improvement of OC-SORT over SORT, we provide more comparisons. In Figure 5.8, we show more samples where SORT suffers from ID switch or Fragmentation caused by non-linear motion or occlusion but OC-SORT survives. Furthermore, in Figure 5.9, we show more samples of trajectory visualizations from SORT and OC-SORT on DanceTrack-val set.

DanceTrack [354] is proposed to encourage better association algorithms instead of carefully tuning detectors. We train YOLOX [110] detector on MOT17 training set only to provide detections on DanceTrack. We find the tracking performance of

[2] https://motchallenge.net/results/Head_Tracking_21/

OC-SORT is already higher than the baselines (Table 5.11). We believe the potential to improve multi-object tracking by better association strategy is still promising and DanceTrack is a good platform for the evaluation.

5.7.8 Integrate Appearance into OC-SORT

OC-SORT is pure motion-based but flexible to integrate with other association cues, such as object appearance. We make an attempt of adding appearance information into OC-SORT and achieve significant performance improvements, validated by experiments on MOT17, MOT20, and DanceTrack. Please refer to Deep OC-SORT [239] for details.

5.7.9 More Discussion of State Noise Sensitivity

In Section 5.3.2, we show that the noise of state estimate will be amplified to the noise of velocity estimate. This is because the velocity estimate is correlated to the state estimate. But the analysis is in a simplified model in which velocity itself does not gain noise from the transition directly and the noise of state estimate is i.i.d on different steps. However, in the general case, such a simplification does not hold. We now provide a more general analysis of the state noise sensitivity of SORT.

For the process in Eq 5.1, we follow the most commonly adapted implementation of Kalman filter [3] and SORT [4] for video multi-object tracking. Instead of writing the mean state estimate, we consider the noisy prediction of state estimate now, which is formulated as

$$\mathbf{x}_{t|t-1} = \mathbf{F}_t \mathbf{x}_{t|t-1} + \mathbf{w}_t, \tag{5.15}$$

where $\mathbf{w}_t$ is the process noise, drawn from a zero mean multivariate normal distribution, $\mathcal{N}$, with covariance, $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}_t)$. As $\mathbf{x}_t$ is a seven-tuple, i.e.,$\mathbf{x}_t = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^\top$, the process noise applies to not just the state estimate but also the velocity estimates. Therefore, for a general form of analysis of temporal error magnification in Eq 5.5, we would get a different result because not just the position terms but also the velocity terms gain noise from the transition process. And the noise of velocity terms will

[3] https://github.com/rlabbe/filterpy
[4] https://github.com/abewley/sort

amplify the noise of position estimate by the transition at the next step. We note the process noise as in practice:

$$\mathbf{Q}_t = \begin{bmatrix} \sigma_u^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_v^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_s^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_r^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{\dot{u}}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{\dot{v}}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{\dot{s}}^2 \end{bmatrix}, \tag{5.16}$$

and the linear transition model as

$$\mathbf{F}_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.17}$$

We assume the time step when a track gets untracked is t_1 and don't consider the noise from previous steps. For simplicity, we assume the motion in the x-direction and y-direction do not correlate. We take the motion on the x-direction as an example without loss of generality:

$$\delta_{u_{t_0}} \sim \mathcal{N}(0, \sigma_u^2), \quad \delta_{\dot{u}_{t_0}} \sim \mathcal{N}(0, {\sigma_{\dot{u}}}^2). \tag{5.18}$$

On the next step, with no correction from the observation, the error would be accumulated ($\Delta t = 1$),

$$\delta_{u_{t_0+1}} \sim \mathcal{N}(0, 2\sigma_u^2 + {\sigma_{\dot{u}}}^2), \quad \delta_{\dot{u}_{t_0+1}} \sim \mathcal{N}(0, 2{\sigma_{\dot{u}}}^2). \tag{5.19}$$

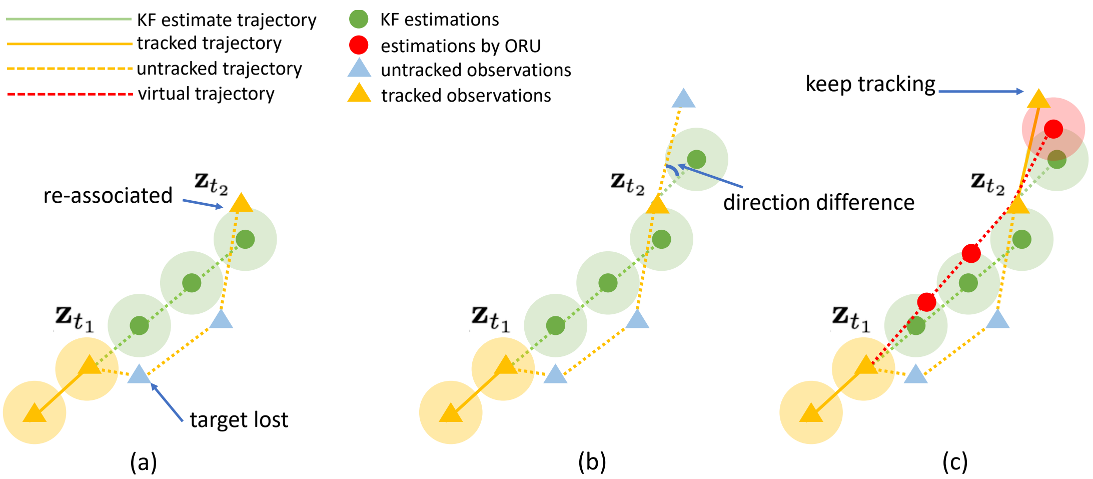


Figure 5.7: Illustration of how ORU changes the behaviors of SORT after an untracked track is re-associated to an observation. The circle area with shadow indicates the range that an estimate can be associated with observations close enough to it. **(a).** The track is re-associates with an observation $\mathbf{z}_{t_2}$ at the step t_2 after being untracked since the time step t_1. **(b).** Without ORU, on the next step of re-association, even though the KF state is updated by $\mathbf{z}_{t_2}$, there is still a direction difference between the true object trajectory and the KF estimates. Therefore, the track is unmatched with detections again (in blue). **(c).** With ORU, we get a more significant change in the state, especially the motion direction by updating velocity. Now, the state estimate (in red) is closer to the state observation and they can be associated again.

Therefore, the accumulation is even faster than we analyze in Section 5.3.2 as

$$\delta_{u_{t_0+T}} \sim \mathcal{N}(0, (T+1)\sigma_u^2 + \frac{1}{2}T(T+1)\sigma_{\dot{u}}^2). \tag{5.20}$$

In the practice of SORT, we have to suppress the noise from velocity terms because it is too sensitive. We achieve it by setting a proper value for the process noise $\mathbf{Q}_t$. For example, the most commonly adopted value [5] of $\mathbf{Q}_t$ in SORT is

$$\mathbf{Q}_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0001 \end{bmatrix}. \tag{5.21}$$

In such a parameter setting, we have the ratio between the noise from position terms and velocity terms as

$$\beta = \frac{(T+1)\sigma_u^2}{0.5T(T+1)\sigma_{\dot{u}}^2} = \frac{200}{T}. \tag{5.22}$$

In practice, a track is typically deleted if it keeps untracked for T_{del} time steps. Usually we set $T_{\text{del}} < 10$, so we have $\beta > 20$. Therefore, we usually consider the noise from velocity terms as secondary. Such a convention allows us to use the simplified model in Section 5.3.2 for noise analysis. But it also brings a side-effect that SORT can't allow the velocity direction of a track to change quickly in a short time interval. We will see later (Section 5.7.10) that it makes trouble to SORT when non-linear motion and occlusion come together and motivates the design of ORU in OC-SORT.

5.7.10 Intuition behind ORU

ORU is designed to fix the error accumulated during occlusion when an untracked track is re-associated with an observation. But in general, the bias in the state

[5] https://github.com/abewley/sort/blob/master/sort.py#L111

estimate $\hat{\mathbf{x}}$ after being untracked for T time steps can be fixed by the update stage once it gets re-associated with an observation. To be precise, the Optimal Kalman gain, i.e.,$\mathbf{K}_t$, can use the re-associated observation to update the KF posteriori parameters. In general, such an expectation of KF's behavior is reasonable. But because we usually set the corresponding covariance for velocity terms very small (Eq 5.21), it is difficult for SORT to steer to the correct velocity direction at the step of re-association.

Motivated by such observations, we design ORU. In the simplified model shown in Figure 5.7, the circle area with the shadow around each estimate footage is the eligible range to associate with observations inside. ORU is designed for the case that a track is re-associated after being untracked. Therefore, the typical situation is as shown in the figure that the true trajectory first goes away from the linear trajectory of KF estimates and then goes closer to it so that there can be a re-association. After the re-association, there would be a cross of the two trajectories.

In SORT, after re-associating with an observation, the direction of the velocity of the previously untracked track still has a significant difference from the true value. This is shown in Figure 5.7(b). This makes the estimate on the future steps lost again (the blue triangle). The reason is the convention of $\mathbf{Q}$ discussed in Appendix 5.7.9. Therefore, even though the canonical KF can support fixing the accumulated error during being untracked theoretically, it is very rare in practice. In ORU, we follow the virtual trajectory where we have multiple virtual observations. In this way, even if the value of $\mathbf{Q}[4:,4:]$ is small, we can still have a better-calibrated velocity direction after the time step t_2. We would like to note that the intuition behind ORU is from our observations in practice and based on the common convention of using Kalman filter for multi-object tracking. It does not make fundamental changes to upgrade the power of the canonical Kalman filter.

Here we provide a more formal mathematical expression to compare the behaviors of SORT and OC-SORT. Assume that the track was lost at the time step t_1 and re-associated at t_2. We assume the mean state estimate is

$$\hat{\mathbf{x}}_{t_1|t_1} = [u_1, v_1, s_1, r_1, \dot{u}_1, \dot{v}_1, \dot{s}_1]^\top, \tag{5.23}$$

and the covariance at t_1 is

$$\mathbf{P}_{t_1|t_1} = \begin{bmatrix} \sigma^2_{u_1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{v_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{s_1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{r_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_{\dot{u}_1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2_{\dot{v}_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{\dot{s}_1} \end{bmatrix}. \tag{5.24}$$

Then, because the covariance expands from the input of process noise at each step of *predict*, at t_2, we have the priori estimates ($t_\Delta = t_2 - t_1$) of state

$$\hat{\mathbf{x}}_{t_2|t_2-1} = [u_2, v_2, s_2, r_2, \dot{u}_2, \dot{v}_2, \dot{s}_2]^\top, \tag{5.25}$$

with

$$\begin{aligned} u_2 &= u_1 + \dot{u}_1 t_\Delta, \\ v_2 &= v_1 + \dot{v}_1 t_\Delta, \\ s_2 &= s_1 + \dot{s}_1 t_\Delta, \\ r_2 &= r_1, \\ \dot{u}_2 &= \dot{u}_1, \\ \dot{v}_2 &= \dot{v}_1, \\ \dot{s}_2 &= \dot{s}_1. \end{aligned} \tag{5.26}$$

And the priori covariance

$$\mathbf{P}_{t_2|t_2-1} = \begin{bmatrix} \sigma^2_{u_2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{v_2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{s_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{r_2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_{\dot{u}_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2_{\dot{v}_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{\dot{s}_2} \end{bmatrix}, \tag{5.27}$$

with

$$\begin{aligned}
\sigma_{u_2}^2 &= \sigma_{u_1}^2 + t_\Delta(\sigma_u^2 + \sigma_{\dot{u}_1}^2) + \frac{t_\Delta(t_\Delta - 1)}{2}\sigma_{\dot{u}}^2, \\
\sigma_{v_2}^2 &= \sigma_{v_1}^2 + t_\Delta(\sigma_v^2 + \sigma_{\dot{v}_1}^2) + \frac{t_\Delta(t_\Delta - 1)}{2}\sigma_{\dot{v}}^2, \\
\sigma_{s_2}^2 &= \sigma_{s_1}^2 + t_\Delta(\sigma_s^2 + \sigma_{\dot{s}_1}^2) + \frac{t_\Delta(t_\Delta - 1)}{2}\sigma_{\dot{s}}^2, \\
\sigma_{r_2}^2 &= \sigma_{r_1}^2 + t_\Delta\sigma_r^2, \\
\sigma_{\dot{u}_2}^2 &= \sigma_{\dot{u}_1}^2 + t_\Delta\sigma_{\dot{u}}^2, \\
\sigma_{\dot{v}_2}^2 &= \sigma_{\dot{v}_1}^2 + t_\Delta\sigma_{\dot{v}}^2, \\
\sigma_{\dot{s}_2}^2 &= \sigma_{\dot{s}_1}^2 + t_\Delta\sigma_{\dot{s}}^2.
\end{aligned} \tag{5.28}$$

Now, SORT will keep going forward as normal. Therefore, with the re-associated observation $\mathbf{z}_{t_2}$, we have

$$SORT \begin{cases} \hat{\mathbf{x}}_{t_2|t_2} = \hat{\mathbf{x}}_{t_2|t_2-1} + \mathbf{K}_{t_2}(\mathbf{z}_{t_2} - \mathbf{H}\hat{\mathbf{x}}_{t_2|t_2-1}), \\ \mathbf{P}_{t_2|t_2} = (\mathbf{I} - \mathbf{K}_{t_2}\mathbf{H}_{)}\mathbf{P}_{t_2|t_2-1} \end{cases} \tag{5.29}$$

where the observation model is

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \tag{5.30}$$

and the Kalman gain is

$$\mathbf{K}_{t_2} = \mathbf{P}_{t_2|t_2-1}\mathbf{H}^\top(\mathbf{H}\mathbf{P}_{t_2|t_2-1}\mathbf{H}^\top + \mathbf{R}_{t_2})^{-1}. \tag{5.31}$$

On the other hand, OC-SORT will replay Kalman filter *predict* on a generated virtual trajectory to gain the posteriori estimates on t_2 (ORU). With the default linear motion analysis, we have the virtual trajectory as

$$\tilde{\mathbf{z}}_t = \mathbf{z}_{t_1} + \frac{t - t_1}{t_2 - t_1}(\mathbf{z}_{t_2} - \mathbf{z}_{t_1}), t_1 < t < t_2. \tag{5.32}$$

Now, to derive the posteriori estimate, we will run the loop between *predict* and *re-update* from t_1 to t_2.

$$\textit{OC-SORT} \begin{cases} \hat{\mathbf{x}}_{t|t} = \mathbf{F}\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{K}_t(\tilde{\mathbf{z}}_t - \mathbf{H}\mathbf{F}\hat{\mathbf{x}}_{t-1|t-1}) \\ \mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H})(\mathbf{F}\mathbf{P}_{t-1|t-1}\mathbf{F}^\top + \mathbf{Q}_t) \end{cases} \tag{5.33}$$

where the Kalman gain is

$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}_t^\top(\mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^\top + \mathbf{R}_t)^{-1}, \tag{5.34}$$

and we can always rewrite it with

$$\mathbf{P}_{t|t-1} = \mathbf{F}\mathbf{P}_{t-1|t-1}\mathbf{F}^\top + \mathbf{Q}_t. \tag{5.35}$$

In the common practice of Kalman filter, we assume a constant set of Gaussian noise for the process noise $\mathbf{Q}_t$. This assumption typically can't hold in practice. This makes the conflict that when there are consistent observations over time, we require a small process noise for multi-object tracking in high-frame-rate videos. However, when there is a period of observation missing, the direction difference between the true direction and the direction maintained by the linear motion assumption grows. This causes the failure of SORT to consistently track previously lost targets even after re-association.

We show the different outcomes of SORT and OC-SORT upon re-associating lost targets in Eq 5.29 and Eq 5.33. Analyzing their difference more deeply will require more assumptions of the underlying true object trajectory and the observations. Therefore, instead of theoretical proof, we demonstrate the gain of performance from OC-SORT over SORT empirically as shown in the experiments.

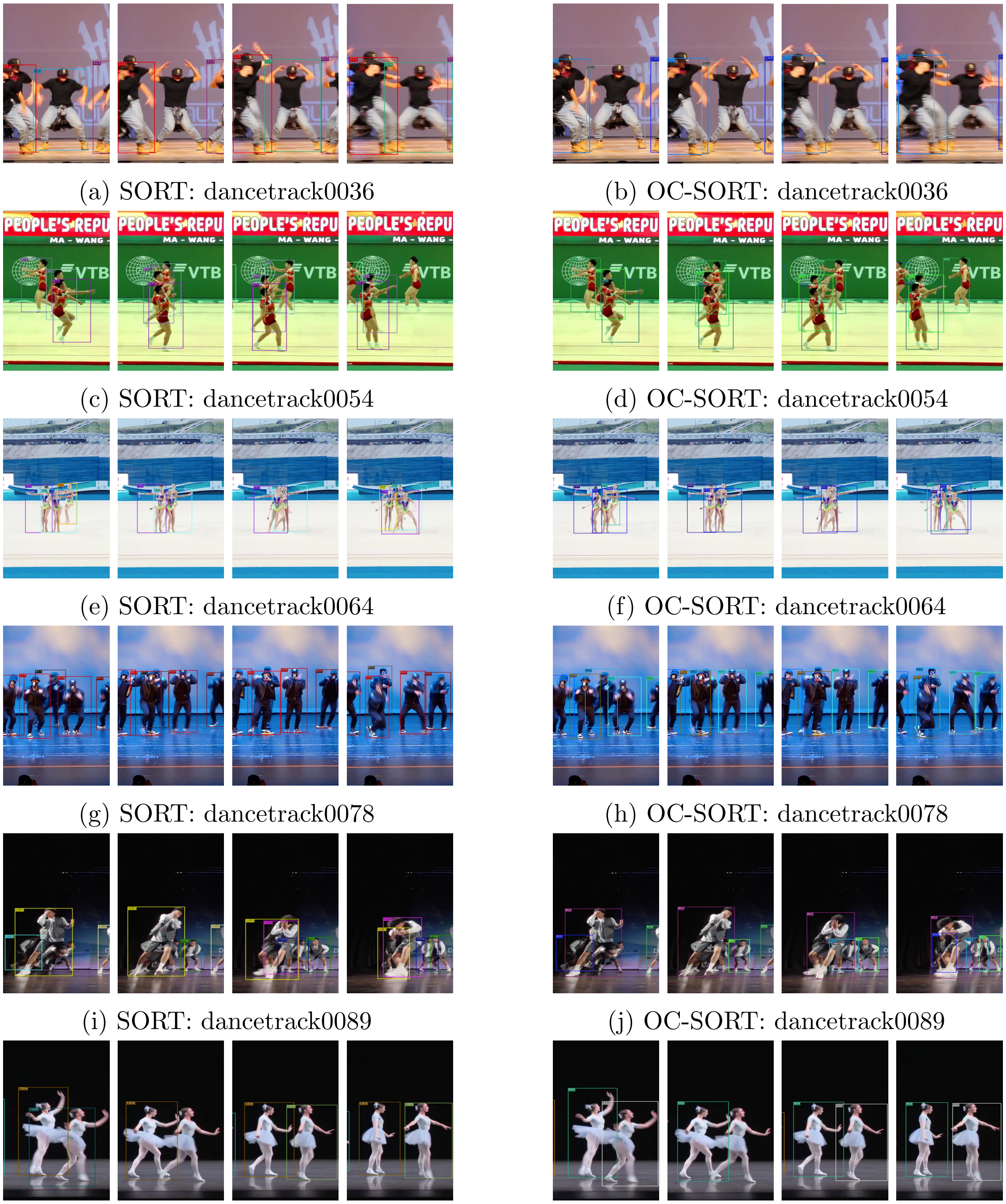
```
Input: Detections $\mathcal{Z} = \{\mathbf{z}_k^i | 1 \leq k \leq T, 1 \leq i \leq N_k\}$; Kalman Filter KF; threshold to remove untracked tracks $t_{\text{expire}}$
Output: The set of tracks $\mathcal{T} = \{\tau_i\}$
Initialization: $\mathcal{T} \leftarrow \emptyset$ and KF;
for timestep $t \leftarrow 1 : T$ do
    /* Step 1: match track prediction with observations */
    $\mathbf{Z}_t \leftarrow [\mathbf{z}_t^1, ..., \mathbf{z}_t^{N_t}]^\top$ /* Obervations */
    $\hat{\mathbf{X}}_t \leftarrow [\hat{\mathbf{x}}_t^1, ..., \hat{\mathbf{x}}_t^{|\mathcal{T}|}]^\top$ from $\mathcal{T}$ /* Estimations by KF.predict */
    $\mathcal{Z} \leftarrow$ Historical observations on the existing tracks
    $C_t \leftarrow C_{\text{IoU}}(\hat{\mathbf{X}}_t, \mathbf{Z}_t) + \lambda C_v(\mathcal{Z}, \mathbf{Z}_t)$ /* Cost Matrix with OCM term */
    Linear assignment by Hungarians with cost $C_t$
    $\mathcal{T}_t^{\text{matched}} \leftarrow$ tracks matched to an observation
    $\mathcal{T}_t^{\text{remain}} \leftarrow$ tracks not matched to any observation
    $\mathbf{Z}_t^{\text{remain}} \leftarrow$ observations not matched to any track

    /* Step 2: perform OCR to find lost tracks back */
    $\mathbf{Z}^{\mathcal{T}_t^{\text{remain}}} \leftarrow$ last matched observations of tracks in $\mathcal{T}_t^{\text{remain}}$
    $C_t^{\text{remain}} \leftarrow C_{\text{IoU}}(\mathbf{Z}^{\mathcal{T}_t^{\text{remain}}}, \mathbf{Z}_t^{\text{remain}})$
    Linear assignment by Hungarians with cost $C_t^{\text{remain}}$
    $\mathcal{T}_t^{\text{recovery}} \leftarrow$ tracks from $\mathcal{T}_t^{\text{remain}}$ and matched to observations in $\mathbf{Z}^{\mathcal{T}_t^{\text{remain}}}$
    $\mathbf{Z}_t^{\text{unmatched}} \leftarrow$ observations from $\mathbf{Z}^{\mathcal{T}_t^{\text{remain}}}$ that are still unmatched to tracks
    $\mathcal{T}_t^{\text{unmatched}} \leftarrow$ tracks from $\mathcal{T}_t^{\text{remain}}$ that are still unmatched to observations
    $\mathcal{T}_t^{\text{matched}} \leftarrow \{\mathcal{T}_t^{\text{matched}}, \mathcal{T}_t^{\text{recovery}}\}$

    /* Step 3: update status of matched tracks */
    for $\tau$ in $\mathcal{T}_t^{matched}$ do
        if $\tau.tracked = False$ then
            /* Perform ORU for track from untracked to tracked */
            $\mathbf{z}_{t'}^\tau, t' \leftarrow$ The last observation matched to $\tau$ and the time step
            Rollback KF parameters to $t'$
            /* Generate virtual observation trajectory */
            $\hat{\mathbf{Z}}_t^\tau \leftarrow [\hat{\mathbf{z}}_{t'+1}^\tau, ..., \hat{\mathbf{z}}_{t-1}^\tau]$
            Online smooth KF parameters along $\hat{\mathbf{Z}}_t^\tau$
        end
        $\tau.tracked = True$
        $\tau.untracked = 0$
        Append the new matched associated observation $\mathbf{z}_t^\tau$ to $\tau$'s observation history
        Update KF parameters for $\tau$ by $\mathbf{z}_t^\tau$
    end

    /* Step 4: initialize new tracks and remove expired tracks */
    $\mathcal{T}_t^{new} \leftarrow$ new tracks generated from $\mathbf{Z}_t^{\text{unmatched}}$
    for $\tau$ in $\mathcal{T}_t^{unmatched}$ do
        $\tau.tracked = False$
        $\tau.untracked = \tau.untracked + 1$
    end
    $\mathcal{T}_t^{\text{reserved}} \leftarrow \{\tau | \ \tau \in \mathcal{T}_t^{\text{unmatched}}$ and $\tau.untacked < t_{\text{expire}}\}$ /* remove expired unmatched tracks */
    $\mathcal{T} \leftarrow \{\mathcal{T}_t^{\text{new}}, \mathcal{T}_t^{\text{matched}}, \mathcal{T}_t^{\text{reserved}}\}$ /* Conclude */
end
$\mathcal{T} \leftarrow$ Postprocess($\mathcal{T}$) /* [Optional] offline post-processing */
Return: $\mathcal{T}$
```

algorithm 1: Pseudo-code of OCSORT.



(a) SORT: dancetrack0036

(b) OC-SORT: dancetrack0036

(c) SORT: dancetrack0054

(d) OC-SORT: dancetrack0054

(e) SORT: dancetrack0064

(f) OC-SORT: dancetrack0064

(g) SORT: dancetrack0078

(h) OC-SORT: dancetrack0078

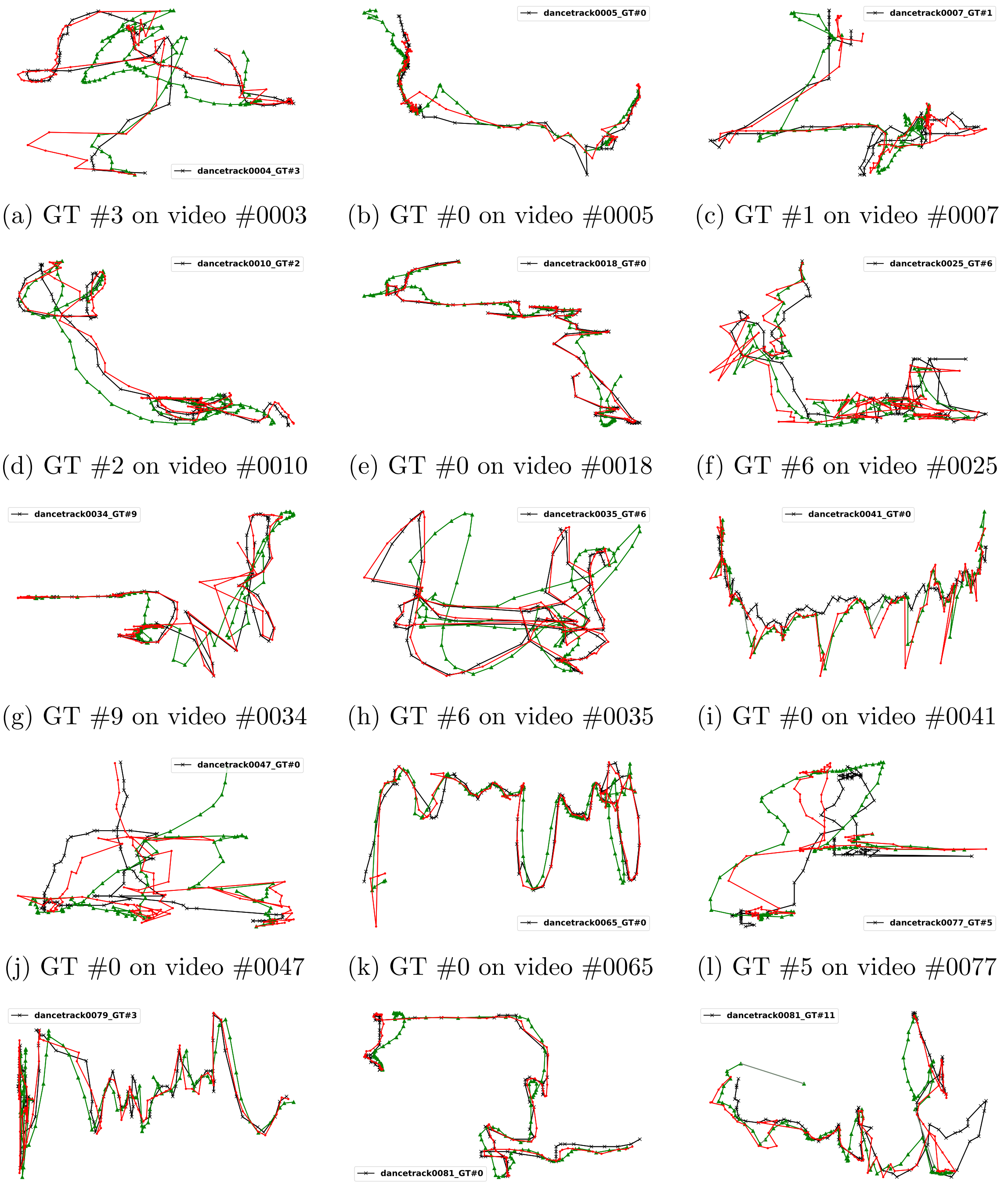
(i) SORT: dancetrack0089

(j) OC-SORT: dancetrack0089

(k) SORT: dancetrack0100

(l) OC-SORT: dancetrack0100

Figure 5.8: More samples where SORT suffers from the fragmentation and ID switch of tracks from occlusion or non-linear motion but OC-SORT survives. To be precise, the issue happens on the objects by SORT at: (a) #322 → #324; (c) ID switch between #672 and #673, later #673 being lost; (e) #760 → #761; (g) #871 → #872; (i) #1063 → #1090, then ID switch with #1081; (l) #1295 → #1304. We select samples from diverse scenes, including street dance, classic dance and gymnastics. Best viewed in color and zoomed in.



(a) GT #3 on video #0003 (b) GT #0 on video #0005 (c) GT #1 on video #0007

(d) GT #2 on video #0010 (e) GT #0 on video #0018 (f) GT #6 on video #0025

(g) GT #9 on video #0034 (h) GT #6 on video #0035 (i) GT #0 on video #0041

(j) GT #0 on video #0047 (k) GT #0 on video #0065 (l) GT #5 on video #0077

(m) GT #3 on video #0079 (n) GT #0 on video #0081 (o) GT #11 on video #0081

Figure 5.9: Randomly selected object trajectories on the videos from the DanceTrack-val set. The **black cross** indicates the ground truth trajectory. The **red dots** indicate the trajectory output by OC-SORT and associated to the selected GT trajectory. The **green triangles** indicate the trajectory output by SORT and associated to the selected GT trajectory. SORT and OC-SORT use the same hyperparameters and detections. Trajectories are sampled at the first 100 frames of each video sequence.

Chapter 6

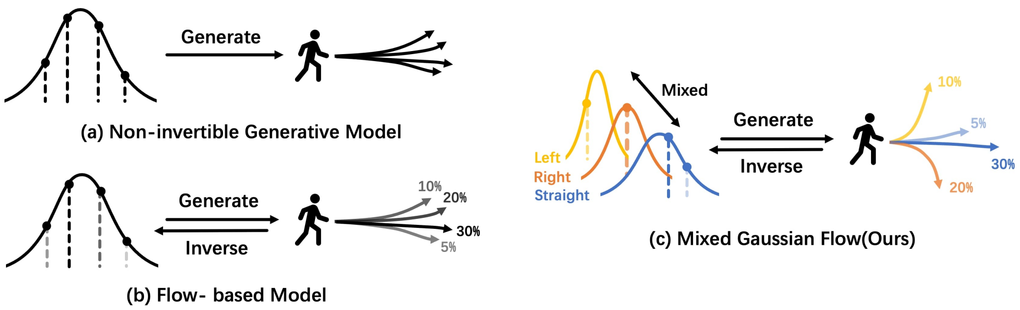
Mixed Gaussian Prior for Human Trajectory Generation

6.1 Introduction

In this work, we aim to improve the diversity for probabilistic trajectory prediction. In trajectory prediction, diversity describes the fact that agents (pedestrians) can move in different directions, speeds, and interact with other agents. Because the motion intentions of agents can not be determined by their historical positions, there is typically no global-optimal strategy to predict a single outcome of future trajectories. Therefore, recent works have focused on probabilistic methods to generate multiple likely outcomes. However, existing solutions are argued to lack good diversity and they often fail to generate the under-represented future trajectory patterns in the training data.

Different motion patterns are usually imbalanced in a dataset. For example, agents are more likely to move straight than turn around in most datasets. Thus, many motion patterns are highly under-represented though discoverable. Therefore, intuitively, an ideal distribution to represent the possible future trajectories should be asymmetric, multi-modal, and expressive to represent long-tailed patterns.

However, most existing generative models solve the problem of trajectory prediction by modeling it as a single-modal and symmetric distribution, i.e., standard Gaussian. This is because the standard Gaussian is tractable and there is a belief that it



(a) Non-invertible Generative Model

(b) Flow- based Model

(c) Mixed Gaussian Flow(Ours)

Figure 6.1: Non-invertible generative models (a), e.g., CVAE, GAN, and diffusions, lack the invertibility for probability density estimation. Flow-based methods (b) are invertible while, sampling from the symmetric standard Gaussian, undermines the diversity and controllability of generation. Our proposed Mixed Gaussian flow (c) maps from a mixed Gaussian prior instead. Summarizing distributions from data and controllable edits, it achieves better **diversity** and **controllability** for trajectory prediction.

can be transformed into any desired distribution of the same or a lower dimension. However, deriving a desired complex distribution from a simple and symmetric prior distribution is challenging, especially with limited and imbalanced data. Moreover, when we derive the target distribution by transforming from the tractable original distribution as Normalizing Flows, GANs, and VAEs do, a dilemma arises: an over-smoothing transformation model can neglect under-represented samples while an over-decorated transformation model will overfit. Especially for normalizing flow, some studies[33, 187] discussed the difficulty of training normalizing flow in practice to represent a complex target distribution.

To solve this dilemma, we propose a prior distribution with more expressiveness and data-driven statistics. It is asymmetric, multi-modal, and adaptive to the training data in the form of a mixed set of Gaussians. Compared to the standard Gaussian, the mixture of Gaussians can better summarize the under-represented samples scattered far away in the representation space. This relieves the sparsity issue of rare cases and thus enhances the diversity of the generated outcomes. Besides diversity, as the mixed Gaussian prior is parametric and transparent during construction, we could control the generation by manipulating this prior, such as adjusting the weights of different sub-Gaussians or manipulating the mean value of them. All these manipulations change generation results statistically without requiring fine-tuning or other re-training. Upon the prior distribution, we choose to construct the generative model by normalizing flow with the unique advantage of being invertible. We thus could estimate the likelihood of each single generated outcome. By combining the designs, we propose a normalizing flow-based trajectory prediction model named Mixed Gaussian Flow (MGF). It enjoys better diversity, controllability, interruptibility, and invertibility for trajectory prediction. During our study, we find that though several evaluation tools have been proposed for measuring diversity[260, 274, 332, 459], they employ varying calculation method and have not gained widespread adoption within the research community. The most popular evaluation metrics (APD/FDE scores) focus on how similar a generated trajectory is to the single ground truth. It is calculated in a "best-of-M" fashion where only one candidate in a batch of M predictions is taken into the measurement. This protocol encourages the methods to generate outcomes approaching the mean (most likelihood) of the learned distribution and provides no sense of the diversity of generation outcomes. Therefore, building upon previous research in the field of human motion prediction[459], we formulate a metric set of Average Pairwise Displacement (APD) and Final Pairwise Displacement (FPD), which measure the diversity of a batch of M generated samples. This helps us to have a concrete study about generation diversity and avoid bias from the "best-of-M" evaluation protocol. With the proposed metrics, we demonstrate that the proposed architecture design improves the diversity of generated trajectories. Still, we estimate the "best-of-M" candidate's alignment with the ground truth under widely adopted APD/FDE metrics. Surprisingly, MGF also achieves state-of-the-art performance.

To conclude, In this work, we focus on enhancing the diversity of trajectory prediction. We propose Mixed Gaussian Flow (MGF) by reforming the prior distribution of normalizing flows as a novel design of mixed Gaussians. It achieves state-of-the-art performance with respect to both the "best-of-M" alignment metrics and diversity metrics. We demonstrate that the proposed MGF model is capable of diverse and controllable trajectory predictions.

6.2 Related Works

Generative Models for Trajectory Prediction. Trajectory prediction aims to predict the positions in a future horizon given historical position observations of multiple participants (agents). Early studies solve the problem by deterministic trajectory prediction [182] where Social forces [144], RNNs [6, 262, 381], and the Gaussian Process [389] are proposed to model the agent motion intentions. Recent works mostly seek multi-modal and probabilistic solutions for trajectory prediction instead, which is a more challenging but faithful setting. Though some of them leverage reinforcement learning [44, 197], the mainstream uses generative models to solve the problem as generating likely future trajectories. Auto-encoder [178] and its variants, such as CVAE [193, 461], are widely adopted. GANs make another line of work [126]. More recently, the diffusion [149] model is also used in this area [246]. However, they are typically not capable of estimating outcome probability as the generation process is not invertible. Normalizing flow [184] is preferred in many cases for being invertible.

Normalizing Flow for Trajectory Prediction. In this work, we would like the predicted trajectories diverse and controllable. We prefer the generation process invertible to allow tractable generation likelihood. We thus choose normalizing flow [184] generative models. Normalizing flow [271] constructs complex distributions by transforming a probability density through invertible mappings from tractable distribution. Normalizing flow has been studied for trajectory prediction in some previous works [120, 315, 316]. In the commonly adopted evaluation protocol of "best-of-M" trajectory candidates, normalizing flow-based methods are once considered not capable of achieving state-of-the-art performance. However, we will show in this paper that with proper design of architecture, normalizing flow can be state-of-the-art. And much more importantly, its invertibility allows more controllable and explainable trajectory prediction.

Gaussian Mixture models as prior. Though the standard Gaussian is chosen by mainstream generative models as the original sampling distribution, some previous works explored how Gaussian mixture models (GMM) can be an alternative to help with generation or classification tasks. [90] uses a GMM prior in VAE models to enhance the clustering and classification performance. [20] adopts GMM to enhance the conditional generation of GAN networks. FlowGMM [160] uses GMM as the prior for flow-based models to deal with the classification task in a semi-supervised way. A recent work PluGen [410] proposes to mix two Gaussians to model the conditional presence of two binary labels to control generation tasks. Existing methods mostly use GMM to describe the presence of multiple auxiliary labels and they typically require additional annotations to construct the GMM. In this work, we use GMM as the distribution prior for normalizing flows without requiring any label annotations. It is designed to enhance the diversity of the generation and relieve the difficulty of learning transforming the tractable prior distribution to the desired complex and multi-modal target distribution for future trajectory generation.

6.3 Method

Our proposed method is based on the normalizing flow paradigm for invertible trajectory generation while we construct a mixed Gaussian prior as the original distribution instead of the naive standard Gaussian to allow more diverse and controllable outcomes. In this section, we first provide the formal problem formulation in section 6.3.1. Then we introduce normalizing flow in section 6.3.2 and the proposed Mixed Gaussian Flow (MGF) model in section 6.3.3. We detail the training/inference implementations in section 6.3.4. At last, we introduce the proposed metrics set to measure the diversity of generated trajectories in section 6.3.5. The overall illustration of MGF is shown in Figure 6.2.

6.3.1 Problem Formulation

We focus on 2D agent trajectory forecasting and represent the agent positions by 2D coordinates. Given a set of multiple agents, i.e., pedestrians in our case, we denote the 2D position of an agent a at time t as $\mathbf{x}_t^a$ and a trajectory from t_i to $t_j (t_i < t_j)$ as $\mathbf{x}_{t_i:t_j}^a$. Given a fixed scene with map $\mathbf{M}$ and a period $\mathbf{T}$: $t_0, t_1, t_2, ..., t_c, ..., t_T$, there are N agents that have appeared during the period $\mathbf{T}$, denoted as $A_{t_0:t_T} = \{a_0, a_1, ..., a_{N-1}\}$. Without loss of generality, given a current time step $t_c \in (t_0, t_T)$, the task of trajectory prediction aims to obtain a set of likely trajectories $\mathbf{x}_{t_c:t_T}^a$ with the past trajectories of all observed agents $\mathbf{X}_{t_0:t_c}^{A_{t_0:t_c}} = \{\mathbf{x}_{t_0:t_c}^a, a \in A_{t_0:t_c}\}$ as input, where a is an arbitrary agent that has shown up during $t : t_0 \longrightarrow t_c$. For each agent $a \in A_{t_0:t_c}$ we seek to sample plausible and likely trajectories of it over the remaining time steps $t_c \longrightarrow t_T$

by a generative model Φ, i.e.,

$$\hat{\mathbf{x}}^{a}_{t_c:t_T} = \Phi(\mathbf{X}^{A_{t_0:t_c}}_{t_0:t_c}), \tag{6.1}$$

at the same time, when there are other variables such as the observations of the maps are provided, we can use them as additional input information. By denoting the observations until t as $\mathbf{O}_{t_0:t_c}$ we have

$$\hat{\mathbf{x}}^{a}_{t_c:t_T} = \Phi(\mathbf{x}^{A_{t_0:t_c}}_{t_0:t_c}; \mathbf{O}_{t_0:t_c}). \tag{6.2}$$

If the generation process is probabilistic instead of deterministic, the outcome of the solution is a set of trajectories instead of a single one. The formulation thus turns to

$$\{^{(i)}\hat{\mathbf{x}}^{a}_{t_c:t_T}\} = \Phi(\mathbf{x}^{A_{t_0:t_c}}_{t_0:t_c}; \mathbf{O}_{t_0:t_c}), \tag{6.3}$$

where i is the index of one candidate in the predicted batch.

For some generative models relying on transforming from a sample point in a known distribution $\mathcal{D}_0$ to the target distribution, e.g., GANs and normalizing flows, the set is generated by mapping from different sample points, i.e., $p \in \mathcal{D}_0$. Therefore, the full formulation becomes

$$\{^{(i)}\hat{\mathbf{x}}^{a}_{t_c:t_T}\} = \Phi(\mathbf{x}^{A_{t_0:t_c}}_{t_0:t_c}; \mathbf{O}_{t_0:t_c}, \mathbf{P}), \tag{6.4}$$

where $\mathbf{P} = \{p_0, ..., p_K\}$ is a set of sampled points from $\mathcal{D}_0$.

Implicitly, the model Φ is required to construct a transformation (Jacobians) between the two distributions. Usually, $\mathcal{D}_0$ is chosen as a symmetric and tractable distribution, such as a standard Gaussian. However, the distribution of the target distribution can be shaped by many data-biased asymmetries thus posing a challenge to learning the transformation effectively and inclusively. This often causes failure of generating under-represented trajectory patterns for trajectory forecasting and hurts the diversity of the outcomes. This observation motivates us to propose a probabilistic generative model for more diverse outcomes by representing the original distribution with more expressiveness.

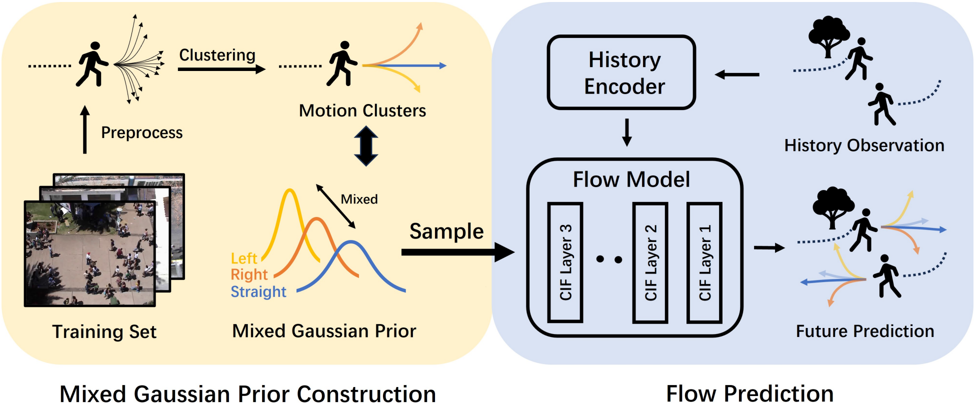


Figure 6.2: The illustration of our proposed Mixed Gaussian Flow (MGF). During training, we construct a mixed Gaussian prior by statistics from the training set. During sampling, the initial noise samples are from the constructed mixed Gaussian prior. MGF keeps a tractable prior distribution and an invertible inference process while the novel mixed Gaussian prior provides more diversity and controllability to the generation outcomes.

6.3.2 Normaling Flow

Normalizing flow [184] is a genre of generative model that constructs complex distributions by transforming a simple distribution through a series of invertible mappings. We choose normalizing flow over other probabilistic generative models as it can provide per-sample likelihood estimates thanks to being invertible. This property is critical to more comprehensively understand the distribution of different future trajectory patterns, especially when typically only sampling dozens of outcomes and considering the existence of long-tailed trajectory patterns. We denote a normalizing flow as a bijective mapping f which transforms a simple distribution $p(\mathbf{z})$ to a complex distribution $p(\mathbf{x})$. The transformation is often conditioned on context information $\mathbf{c}$. With the change-of-variables formula, we can derive the transformation connecting two smooth distributions as follows:

$$\begin{aligned} \mathbf{x} &= f(\mathbf{z};\mathbf{c}), \\ p(\mathbf{x}) &= p(\mathbf{z}) \cdot |\det(\nabla_{\mathbf{x}} f^{-1}(\mathbf{x};\mathbf{c}))|, \\ -\log(p(\mathbf{x})) &= -\log(p(\mathbf{z})) - \log(|\det(\nabla_{\mathbf{x}} f^{-1}(\mathbf{x};\mathbf{c}))|). \end{aligned} \tag{6.5}$$

Given the formulations, with a known distribution $\mathbf{z} \sim \mathcal{D}_0$, we can calculate the density of $p(\mathbf{x})$ following the transformations and vice versa. However, the equations require the Jacobian determinant of the function f to obtain the distribution density $p(\mathbf{x})$. The calculation of it in the high-dimensional space is not trivial. Recent works propose to use deep neural networks to approximate the Jacobians. To maintain the inevitability of the normalizing flows, some carefully designed layers are inserted into the deep models and the coupling layers [91] are one of the most widely adopted ones.

More recently, FlowChain [237] is proposed to enhance the standard normalizing flow models by using a series of Conditional Continuously-indexed Flows (CIFs) [77] to estimate the density of outcomes. CIFs are obtained by replacing the single bijection f in normalizing flows with an indexed family $F(\cdot;u)_{u\in U}$, where $U \subseteq R$ is the index set and each $F(\cdot;u) : \mathbf{z} \longrightarrow \mathbf{x}$ is a bijection. Then, the transformation is changed to

$$\mathbf{z} \sim p(\mathbf{z}), \quad U \sim p_{U|\mathbf{z}}(\cdot|\mathbf{z}), \quad \mathbf{x} := F(\mathbf{z};U). \tag{6.6}$$

Please refer to [77] for more details about CIFs and their connection with variational inference. In this work, we follow the idea of using a stack of CIFs from [237] to achieve fast inference and the updates of trajectory density estimates.

Normalizing flow based model samples from a standard Gaussian, $\mathbf{z} \sim \mathcal{N}(0,1)$, usually results in overfitting to the most-likelihood for trajectory prediction. This is because each data sample from the training sample is considered extracted as the mode of a standard Gaussian. Only the mode value (the ground truth) is directly supervised and the underlying target distribution is assumed to be perfectly symmetric, which is not aligned with the usual real-world facts. Related discussion can be found in many previous literatures[180, 324]. This typically results in degraded expressiveness of the model to fail to capture under-represented motion patterns from the data and thus hurts the outcome diversity.

6.3.3 Mixed Gaussian Flow (MGF)

We propose Mixed Gaussian Flow (MGF) to enhance the diversity and controllability in trajectory prediction. MGF consists of two stages as summarized in Figure 6.2. First, we construct the mixed Gaussian prior by fitting the parametric model of a combination of K Gaussians, $\{\mathcal{N}(\mu_k, \sigma_k^2)\}, (1 \leq k \leq K)$. The parametric model is obtained with the data samples from training sets. Then, during inference, we sample points from the mixture of Gaussian and map them into a trajectory latent in the target distribution by a stack of CIF layers with the historical trajectories of all involved agents as the condition. We will introduce the two stages in detail below.

MGF maps from a mixture of Gaussians instead of a single Gaussian to the target distribution. To maintain the inevitability of the model, the mixed Gaussian prior can not be arbitrary. We obtain the parametric construction of the mixed Gaussian by fitting it with training data. In this fashion, we can derive multiple Gaussians to represent different motion patterns in the dataset, such as going straight or turning left and right. In a simplified perspective, we regard the mixture as combining multiple clusters, each of which represents a certain sub-distribution. By sampling from the mixture of Gaussians instead of a standard Gaussian, our constructed model has more powerful expressiveness than the standard normalizing flow model. This results in more diverse trajectory predictions. Also, by manipulating the mixed Gaussian prior, we can achieve controllable trajectory prediction.

Mixed Gaussian Prior Construction. For the data pre-processing, we transfer motion directions into relative directions with respect to a zero-degree direction. All position footage is represented in meters. Given the trajectory between $t_0 \longrightarrow t_c$ to predict the trajectory between $t_c \longrightarrow t_T$, we would put the position pivot at t_c, i.e., $\mathbf{x}_{t_c}$, as the origin and convert the position on all other time steps to be the offset from $\mathbf{x}_{t_c}$. Then, we cluster the preprocessed future trajectories into K clusters, which is a hyper-parameter. We note the mean of the clusters as $\boldsymbol{\mu} = \{\mu_i\}_{i=1,\dots,K}$.

These cluster centers reveal the mean value of K representative patterns of pedestrians' motion, e.g. go straight, turn left. They will be the means of the Gaussians. The variances of the Gaussian, i.e., σ_k^2, can be pre-determined or learned. The final mixture of Gaussians is denoted as

$$\mathcal{D}^{\Sigma} = \sum_{k=1}^{K} \beta_k \mathcal{N}(\mu_k, \sigma_k^2), \tag{6.7}$$

where β_k are the weights assigned to each cluster following the k-means clustering of the training data. By default, we perform clustering by K-means with $K = 8$.

Flow Prediction. Once the mixed Gaussian prior is built, we can do trajectory prediction by mapping samples from the distribution to future trajectories conditioned on historical information(e.g. social interaction features extracted by a Trajectron++[328] encoder). Here, we ignore the intermediate transformation by CIFs as eq. (6.6) shows while following the original formulations of normalizing flows as eq. (6.5) for simplicity. We distribute the samples from different Gaussians by their weights. Given the i-th sample from $\mathcal{N}(\mu_k, \sigma_k^2)$, we can transform it to the i-th predicted trajectories

$$\mathbf{z}_i \sim \mathcal{D}^{\Sigma}, \quad {}^{(i)}\hat{\mathbf{x}}^a_{t_c:t_T} = \Phi(\mathbf{x}^{A_{t_0:t_c}}_{t_0:t_c}; \mathbf{O}_{t_0:t_c}, \mathbf{z}_i). \tag{6.8}$$

For a sample $\frac{\mathbf{z}_i}{\beta_k} \sim \mathcal{N}(\mu_k, \sigma_k^2)$, we have the probability estimate

$$p(\mathbf{z}_i) = \beta_k \frac{1}{\sigma_k\sqrt{2\pi}} e^{-\frac{(z_i-\mu_k)^2}{2\sigma_k^2}}, \tag{6.9}$$

and the transformation is converted to

$$p({}^{(i)}\hat{\mathbf{x}}^a_{t_c:t_T}) = \exp(-\frac{(z_i-\mu_k)^2}{2\sigma_k^2} + \log\frac{\beta_k}{\sigma_k\sqrt{2\pi}}) \cdot |\det(\nabla_{f(\mathbf{z_i};\mathbf{O}_{t_0:t_c})}\mathbf{z_i})|, \tag{6.10}$$

which can be also invested back for the density estimate by the normalizing flow law

$$\hat{p}(\mathbf{z}_i) = \beta_k \frac{1}{\sigma_k\sqrt{2\pi}} \exp(-\frac{[f^{-1}({}^{(i)}\hat{x}^a_{t_c:t_T}; O_{t_0:t_c}) - \mu_k]^2}{2\sigma_k^2}). \tag{6.11}$$

6.3.4 Training and Inference

The training loss of MGF comes from two directions: the forward process to get mixed flow loss and the inverse process to get minimum ℓ_2 loss.

Forward process. Given a ground truth trajectory sample $\mathbf{x}^a_{t_c:t_T}$, we need to assign it to a cluster in the mixed Gaussian prior by measuring its distance to the centroids

$$\hat{k} = \arg\min_i (\mathbf{x}^a_{t_c:t_T} - \mu_i)^2, \quad \mathcal{D}^{\hat{k}} := \beta_{\hat{k}}\mathcal{N}(\mu_{\hat{k}}, \sigma_{\hat{k}^2}), \tag{6.12}$$

with a tractable probability density function $p_{\hat{k}}(\cdot)$. Through the inverse process f^{-1} of flow model, we transform $\mathbf{x}^a_{t_c:t_T}$ into its corresponding latent representation, here

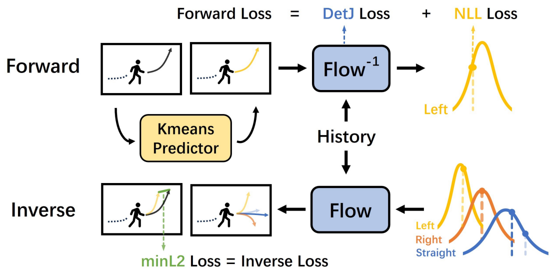


Figure 6.3: During training, the model is trained at both the forward and the inverse process of the normalizing flow.

denoted as

$$\hat{\mathbf{z}} = f^{-1}(\mathbf{x}^a_{t_c:t_T}; \mathbf{O}_{t_0:t_c}). \tag{6.13}$$

Then we can compute the forward mixed flow loss:

$$L_{forward} = -\log(p(\mathbf{x}^a_{t_c:t_T})) = -\log(p_{\hat{k}}(\hat{\mathbf{z}})) - \log(|\det(\nabla_{\mathbf{x}^a_{t_c:t_T}} \hat{\mathbf{z}})|). \tag{6.14}$$

Instead of computing negative-log-likelihood(NLL) loss of $\hat{\mathbf{z}}$ in the mixed distribution $\sum_{k=1}^{K} \beta_k \mathcal{N}(\mu_k, \sigma_k^2)$, we compute NLL loss in the sub-Gaussian with the nearest centroid $\beta_{\hat{k}} \mathcal{N}(\mu_{\hat{k}}, \sigma_{\hat{k}^2})$ because each centroid is independent to others in the mixed distribution and we encourage the model to learn specified motion patterns to avoid overwhelming by the major data patterns. Calculating NLL loss over the mixed distribution may fuse other centroids and damage the diversity of model outputs. By our design, the mixed Gaussian prior can maintain more capacity for expressing complicated multi-modal distribution than the traditional single Gaussian prior, which typically constrains the target distribution to be single-modal and symmetric.

Inverse Process. This process repeats the flow prediction process to get generated trajectories. To predict M candidates, we sample $\mathbf{z}_i \sim \sum_{k=1}^{K} \beta_k \mathcal{N}(\mu_k, \sigma_k^2), i = 1, 2, ..., M$ and transform them into M trajectories

$$\{{}^{(i)}\hat{\mathbf{x}}^a_{t_c:t_T}\} = \{f(\mathbf{z}_i; \mathbf{O}_{t_0:t_c})\}, i = 1, 2, ..., M. \tag{6.15}$$

We compute the minimum ℓ_2 loss between M predictions and ground truth trajectory as [126] does:

$$L_{inverse} = \min_{i=1}^{M} \frac{\left({}^{(i)}\hat{\mathbf{x}}^a_{t_c:t_T} - \mathbf{x}^a_{t_c:t_T}\right)^2}{t_T - t_c}. \tag{6.16}$$

We sample $\mathbf{z}_i$ from sub Gaussians by their weight. This is approximately equal to sampling from the original mixed Gaussians but makes the reparameterization trick doable.

Although approximated differential backpropagation techniques, such as the Gumbel-Softmax trick, can be employed to make the sampling process of mixed Gaussians differentiable, computing the Negative Log-Likelihood (NLL) loss between a sample point and the mixed Gaussian distribution remains challenging because

$$-\log(p_{\mathcal{D}^\Sigma}(\hat{\mathbf{z}})) = -\log(\sum_{k=1}^{K} \frac{\beta_k}{\sigma_k} \cdot e^{-\frac{(\hat{z}-\mu_k)^2}{2\sigma_k^2}}) + C, \tag{6.17}$$

contains exponential operations on matrices, which can be simplified through logarithmic operations in single Gaussian condition. Computing this term requires iterative optimization methods, such as the Expectation-Maximization algorithm[85] for approximation[331, 433], which makes the computing process much more complex. Therefore, in practice, sampling from individual Gaussian components is preferred for computing efficiency. Furthermore, applying the Gumble-softmax to learn a mixture of Gaussians in generative models has been reported difficult in practice in some cases[294] due to gradient vanishing problem.

The forward and inverse losses encourage the model to predict a well-aligned sample in a sub-space from the prior without hurting the flexibility and expressiveness of other sub-spaces. We combine the forward and inverse losses by a ratio γ to be a Symmetric Cross-Entropy loss [315], which was proved beneficial for better balancing the "diversity" and "precision" of predicted trajectories:

$$L = L_{forward} + \gamma \cdot L_{inverse}. \tag{6.18}$$

6.3.5 Diversity Metrics

The widely adopted average/final displacement error (ADE/FDE) scores measure the alignment (precision) between the ground truth future trajectory and one predicted trajectory. Under the common "best-of-M" evaluation protocol, ADE/FDE scores encourage nothing but finding a single "aligned" trajectory with the ground truth. ADE encourages the position on all time steps to be aligned with the single ground truth and FDE chooses the trajectory with the closest endpoint while all other trajectories are neglected in score calculating. Such an evaluation protocol overwhelmingly encourages the methods to fit the most likelihood from a certain distribution and all generated candidates race to be the most similar one as the distribution mean. Under the single-mode and symmetric assumption, this usually tends to fit into a Gaussian with a smaller variance. However, this tendency hurts the diversity of predicted trajectory hypotheses.

To provide a tool for quantitative trajectory diversity evaluation, we formulate a set of metrics. Following the idea of average displacement error (ADE) and final displacement error (FDE), we measure the diversity of trajectories by their pairwise displacement along the whole generated trajectories and the final step. Follow Dlow[459], we name that average pairwise displacement (APD) and final pairwise displacement (FPD). We note that the diversity metrics are measured in the complete set of generated trajectory candidates instead of between a single candidate and the ground truth. The formulation of APD and FPD are as below

$$\mathrm{APD} = \frac{\sum_{i=1}^{M} \sum_{j=1}^{M} \sqrt{\sum_{t=t_c}^{t_T} ({}^{(i)}\hat{\mathbf{x}}_t^a -{}^{(j)} \hat{\mathbf{x}}_t^a)^2}}{M^2 \cdot (t_T - t_c)}, \quad \mathrm{FPD} = \frac{\sum_{i=1}^{M} \sum_{j=1}^{M} \sqrt{({}^{(i)}\hat{\mathbf{x}}_{t_T}^a -{}^{(j)} \hat{\mathbf{x}}_{t_T}^a)^2}}{M^2}, \tag{6.19}$$

where APD measures the average displacement along the whole predicted trajectories and FPD measures the displacement of trajectory endpoints. We would mainly follow the widely adopted ADE/FDE for benchmarking purposes while using APD/FPD as a secondary metric set to better understand the diversity of the generated future trajectories.

6.4 Experiments

In this section, we provide experiments to demonstrate the effectiveness of our method. We first introduce experiment setup in section 6.4.1 and benchmark with related works to evaluate the trajectory prediction alignment and diversity in section 6.4.2. Then,

Table 6.1: **Results on *ETH/UCY* dataset with Best-of-20 metrics.** Scores are in meters, lower is better. **bold** and underlined scores denote the best and the second-best scores.

Method	ETH		HOTEL		UNIV		ZARA1		ZARA2		Mean	
	ADE	**FDE**	**ADE**	**FDE**	**ADE**	**FDE**	**ADE**	**FDE**	**ADE**	**FDE**	**ADE**	**FDE**
Social-GAN [126]	0.87	1.62	0.67	1.37	0.76	1.52	0.35	0.68	0.42	0.84	0.61	1.21
STGAT [157]	0.65	1.12	0.35	0.66	0.52	1.10	0.34	0.69	0.29	0.60	0.43	0.83
Social-STGCNN [259]	0.64	1.11	0.49	0.85	0.44	0.79	0.34	0.53	0.30	0.48	0.44	0.75
Trajectron++ [328]	0.61	1.03	0.20	0.28	0.30	0.55	0.24	0.41	0.18	0.32	0.31	0.52
MID [119]	0.55	0.88	0.20	0.35	0.30	0.55	0.29	0.51	0.20	0.38	0.31	0.53
PECNet [245]	0.54	0.87	0.18	0.24	0.35	0.60	0.22	0.39	0.17	0.30	0.29	0.48
GroupNet [422]	0.46	0.73	0.15	0.25	0.26	0.49	0.21	0.39	0.17	0.33	0.25	0.44
AgentFormer [461]	0.45	0.75	0.14	0.22	0.25	0.45	0.18	0.30	0.14	0.24	0.23	0.39
EqMotion [424]	0.40	0.61	**0.12**	**0.18**	0.23	0.43	0.18	0.32	**0.13**	**0.23**	0.21	0.35
FlowChain [237]	0.55	0.99	0.20	0.35	0.29	0.54	0.22	0.40	0.20	0.34	0.29	0.52
MGF(Ours)	**0.39**	**0.59**	0.13	0.20	**0.21**	**0.39**	**0.17**	**0.29**	0.14	0.24	**0.21**	**0.34**

Table 6.2: **Evaluation results on *SDD* (in pixels)**.

Method	**ADE**	**FDE**
Social-GAN [126]	27.25	41.44
STGAT [157]	14.85	28.17
Social-STGCNN [259]	20.76	33.18
Trajectron++ [328]	19.30	32.70
MID [119]	10.31	17.37
PECNet [245]	9.97	15.89
GroupNet [422]	9.31	16.11
EqMotion [424]	8.80	14.35
MemoNet [423]	8.56	12.66
FlowChain [237]	9.93	17.17
MGF (Ours)	**7.74**	**12.07**

we showcase the diversity and controllability of MGF in section 6.4.3 and section 6.4.4. Finally, we ablate key implementation components in section 6.4.5.

6.4.1 Setup

Datasets. We evaluate on two major benchmarks, i.e., ETH/UCY [196, 280] and SDD [318]. ETH/UCY consists of five subsets. We follow the widely used Social-GAN [126] benchmark. SDD dataset consists of 20 scenes captured in bird's eye view. We follow the TrajNet [327] benchmark. We note that in the community of trajectory prediction, previous works have inconsistent evaluation protocol details and thus have made unfair comparisons.

Metrics. We use the widely used average displacement error (ADE) and final

Table 6.3: **Results on *ETH/UCY* dataset with diversity metrics.** Scores are in meters, higher means more diverse prediction. **bold** and underlined scores denote the best and the second-best scores.

Method	ETH		HOTEL		UNIV		ZARA1		ZARA2		Mean	
	APD	**FPD**	**APD**	**FPD**	**APD**	**FPD**	**APD**	**FPD**	**APD**	**FPD**	**APD**	**FPD**
Social-GAN [126]	0.680	1.331	0.566	1.259	0.657	1.502	0.617	1.360	0.515	1.119	0.607	1.314
Social-STGCNN [259]	0.404	0.633	0.591	0.923	0.333	0.497	0.490	0.762	0.417	0.657	0.447	0.694
Trajectron++ [328]	0.704	1.532	0.568	1.240	0.648	1.404	0.697	1.528	0.532	1.161	0.630	1.373
AgentFormer [461]	**1.998**	**4.560**	0.995	2.333	1.049	**2.445**	0.774	1.772	0.849	1.982	1.133	**2.618**
MemoNet [423]	1.232	2.870	0.950	2.030	0.847	1.822	0.844	1.919	0.880	2.120	0.951	2.152
FlowChain [237]	0.814	1.481	0.484	0.833	0.636	1.094	0.505	0.890	0.492	0.859	0.586	1.031
MGF(Ours)	1.624	3.555	**1.138**	**2.387**	**1.115**	2.163	**1.029**	**2.119**	**1.065**	**2.182**	**1.194**	2.481

displacement error (FDE) to measure the alignment of the predicted trajectories and the ground truth. ADE is the average L2 distance between the ground truth and the predicted trajectory. FDE is the L2 distance between the ground truth endpoints and predictions. Most previous works choose the "Best-of-M" evaluation protocol and we follow it to choose $M = 20$ as default.

Here, we note that, under different assumptions of distribution spreading and variance, the evaluation is ideally done with different values of M. However, most existing methods only provide results with $M = 20$ and many of them do not open-source the code of the models so we can not rebenchmark with other value choices of M. Besides the metrics for trajectory alignment, we also use the proposed metrics set APD and FPD to measure the diversity of the predicted trajectory candidates.

Implementation Details. We enhance our model using a similar technique as "intension clustering" [423] and we name it "prediction clustering". The key difference is that we directly cluster the entire trajectory instead of the endpoints. To make a fair comparison, we followed the data processing from FlowChain [237] and Trajectron++ [328]. We also follow FlowChain's implementations of CIFs that each layer consists of a RealNVP [92] with a 3-layer MLP and 128 hidden units. We use a Trajectron++ [328] encoder to encode historical trajectories. All models were trained on a single NVIDIA V100 GPU for 100 epochs(approximately 4 to 8 hours).

6.4.2 Benchmark Results

We benchmark MGF with a line of recent related works on ETH/UCY dataset in table 6.1. The results of Trajectron++ and MID are updated according to a reported implementation issue [1]. MGF achieves on-par state-of-the-art performance with Eqmotion [424]. Specifically, Our method achieves the best ADE and FDE in 3 out of 5 subsets and the best ADE and FDE score by averaging all splits. Here we note that we build MGF as a normalizing flow-based method as its invertibility is key property we desire, though normalizing flow is usually considered inferior regarding the alignment evaluation. Therefore, such a good performance on the alignment is surprising to us. To compare with other normalizing flow-based methods, our method significantly improves the performance compared to FlowChain, achieving **27.6%** improvement by ADE and **34.6%** improvement by FDE.

On the SDD dataset, where the motion pattern is considered more diverse than UCY/ETH, the benchmark results are shown in table 6.2. Our method outperforms all baselines measured by ADE/FDE for trajectory alignment. Specifically, Our method reduces ADE from 8.56 to 7.74 compared to the current state-of-the-art method MemoNet, achieving **9.6%** improvement. Our method also significantly improves the performance of FlowChain for **22.1%** by ADE and **29.7%** by FDE. According to the benchmarking on the two popular datasets, we demonstrate the state-of-the-art alignment (precision) of our proposed method. Here we note again that the alignment with the deterministic ground truth is not the highest priority when we design our method, we will discuss the main advantages of MGF, diversity, and controllability, in the next paragraphs.

6.4.3 Diverse Generation

By leveraging the mixed Gaussian prior, our model can generate trajectories from the corresponding clusters, resulting in a more diverse set of trajectories than sampling from a Gaussian. This is intuitively due to less difficulty in learning the Jacobians for distribution transformation. We present examples in fig. 6.4. Given a past trajectory, there is a single ground truth future trajectory possibility from the dataset. We select four samples with different ground truth intentions, i.e., going straight, U-turn, left-turn, and right-turn. By sampling noise from the clustered distributions, we could generate future trajectories with diverse intentions. From the visualizations, we could notice, of course, that we generate outcomes that are very similar to the

[1] https://github.com/StanfordASL/Trajectron-plus-plus/issues/53

Table 6.4: Ablation study on the ETH/UCY and SDD dataset. All components are demonstrated beneficial to the prediction alignment.

Inv. Loss	Mixed Gaus.	Learnable Var.	Pred. Clustering	ETH/UCY		SDD	
				ADE	**FDE**	**ADE**	**FDE**
				0.30	0.55	10.00	16.59
✓				0.28 (↓0.02)	0.53 (↓0.02)	9.65 (↓0.35)	16.39 (↓0.20)
✓	✓			0.27 (↓0.01)	0.44 (↓0.09)	9.26 (↓0.39)	15.48 (↓0.91)
✓	✓	✓		0.25 (↓0.02)	0.40 (↓0.04)	9.20 (↓0.06)	14.78 (↓0.70)
✓	✓	✓	✓	**0.21** (↓0.04)	**0.34** (↓0.06)	**7.74** (↓1.46)	**12.17** (↓2.61)

ground truth with close intentions while we also generate outcomes that have very diverse intentions. The well-aligned single trajectory accounts for the high ADE and FDE score our method achieves. And the impressive diversity demonstrates the effectiveness of our design, especially considering they are well controlled by the clusters where they are sampled from.

Quantitatively, we evaluate the generation diversity according to our proposed metrics on ETH/UCY dataset since most existing methods did not either make experiments on SDD or open-source training code/checkpoint on SDD. The results are presented in table 6.3. We can observe that MGF achieves the best or second-best APD and FPD score on all splits among sota methods. Besides, our method significantly improves the performance compared to FlowChain, achieving **103.7%** improvement by APD and **140.6%** improvement by FPD. The only method that can achieve close diversity with our method is Agentformer [461], which designs sampling from a set of conditional VAE to improve the diversity. However, compared to MGF, Agentformer is more computation-intensive and shows significantly lower alignment according to ADE/FDE scores in table 6.1. Also, Agentformer is not fully invertible, which is considered a key property we desire for trajectory forecasting. The superior quantitative performance according to the alignment (precision) and diversity metrics suggests the effectiveness of our method by balancing these two adversarial features.

6.4.4 Controllable Generation

The generated sample from MGF is highly correlated with the original sample drawn from the mixed Gaussian prior. If the prior distribution is a standard Gaussian as in the canonical normalizing flow method, we can have almost no control over the

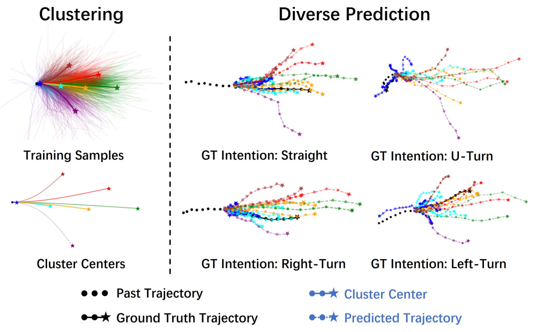


Figure 6.4: MGF predictions on ETH dataset. The color of trajectories corresponds to the cluster in the mixed Gaussian prior, from which the sample belongs to.

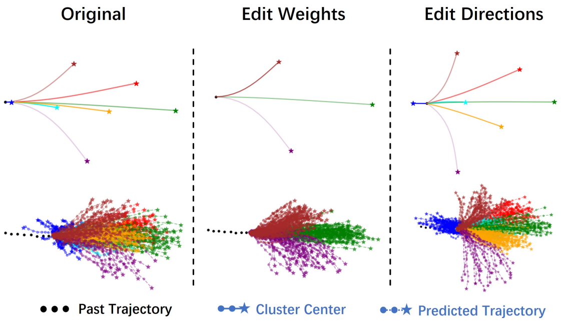


Figure 6.5: Controllable generation on ETH dataset. By editing cluster centers, we can control the predictions.

generated sample. The only controllability is to sample near the mode to generate a sample similar to the learned most-likelihood outcome or far from the mode to make them more different. However, as we discussed, after sufficient training and supervision by the forward loss, the variance of the latent Gaussian distribution of the outcome is usually very small, which further hurts the controllability. However, as we chose a transparent mixed Gaussian prior for the sampling, we can control the generation flexibility. First, by adjusting sub-Gaussians in the mixture prior, we can manipulate the generation process statistically. fig. 6.5 shows that by editing cluster compositions, we can control the predictions of MGF with good interpretability. By editing the weights of sub-Gaussians, we can control the ratio of splatting into directions. By editing the directions of the cluster means, we can control the intentions of samples statistically. Besides cluster centers, we can also edit the variance of Gaussian to control the density of generated trajectories or combine a set of operations to get expected predictions. We provide more discussions and examples in the appendix in the supplement.

6.4.5 Ablation Study

We ablate some key components of our implementation for both ADE/FDE and APD/FPD metrics, see table 6.5 and table 6.6. (1)**Prediction clustering** is a

Table 6.5: Ablation study of ADE/FDE on the ETH/UCY and SDD dataset.

Pred. Clustering	Mixed Gaus.	Learnable Var.	Inv. Loss	ETH/UCY		SDD	
				ADE	**FDE**	**ADE**	**FDE**
-	-	-	-	0.33	0.61	11.90	21.33
-	✓	-	-	0.29 (↓0.04)	0.48 (↓0.13)	11.38 (↓0.52)	19.28 (↓2.05)
✓	-	-	-	0.29	0.54	10.63	18.80
✓	✓	-	-	0.27 (↓0.02)	0.48 (↓0.06)	9.19 (↓1.44)	15.86 (↓2.94)
✓	✓	✓	-	0.23 (↓0.04)	0.39 (↓0.09)	8.71 (↓0.48)	14.86 (↓1.00)
✓	✓	✓	✓	**0.21** (↓0.02)	**0.34** (↓0.05)	**7.74** (↓0.97)	**12.07** (↓2.79)

Table 6.6: Ablation study of APD/FPD on the ETH/UCY and SDD dataset.

Pred. Clustering	Mixed Gaus.	Learnable Var.	Inv. Loss	ETH/UCY		SDD	
				APD	**FPD**	**APD**	**FPD**
-	-	-	-	0.39	0.76	14.82	27.22
-	✓	-	-	0.78(↑0.39)	1.70(↑0.94)	23.18(↑8.36)	44.90(↑17.68)
✓	-	-	-	0.41	0.80	15.52	28.50
✓	✓	-	-	1.09(↑0.68)	2.33(↑1.53)	32.42(↑16.9)	65.43(↑36.93)
✓	✓	✓	-	0.96(↓0.13)	2.12(↓0.21)	30.10(↓2.32)	60.20(↓5.23)
✓	✓	✓	✓	1.19(↑0.77)	2.48(↑0.36)	31.56(↑1.46)	64.52(↑4.32)

common post-processing method, which improves the ADE/FDE as expected. However, it hurts the diversity for nomalizing flow model with single Gaussian prior. This is reasonable as the single Gaussian prior tends to generate trajectories densely close to the most likelihood and prediction clustering can't cluster them into well-separated clusters for different motion intentions. (2)**Mixed Gaussian prior** help the model generates more diverse outputs and achieves higher APD/FPD scores and this improvement can be further enhanced by prediction clustering. It also increases ADE/FDE scores a lot, we believe this is because mixed Gaussian prior relieves the difficulty of learning the Jacobians for distribution transformation. Thus more under-explored patterns, which may be selected as the "best-of-M" samples in rare but plausible scenarios, have the chance to be expressed. (3)**Learnable variance** improve ADE/FDE while bring down APD/FPD a bit. We find that the learnable variance usually converges to a smaller value than the fixed situation. This is encouraged by the supervision from the ground truth (most likelihood) to a desired steeper Gaussian, thus hurting the diversity. However, its substantial improvement in ADE/FDE indicates that it remains a valuable component of the model architecture. (4)**Inverse loss** provides a straightforward supervision of the trajectory in the coordinate space, which is also proved beneficial for ADE/FDE and APD/FPD scores.

6.5 Conclusion

We focus on improving the diversity while keeping the estimated probability tractable for trajectory forecasting in this work. We noticed the poor expressiveness of Gaussian distribution as the original sampling distribution for normalizing flow-based methods to generate complicated and clustered outcome patterns. We thus propose to construct a mixed Gaussian prior to help learn Jacobians for distribution transformation with less difficulty and higher flexibility. Based on this main innovation, we propose Mixed Gaussian Flow (MGF) model for the diverse and controllable trajectory generation. The cooperating strategy of constructing the prior distribution and training the model is also designed. According to the evaluation of popular benchmarks, we demonstrate that MGF achieves state-of-the-art prediction alignment and diversity. It also has other good properties such as controllability and being invertible for probability estimates.

Part II

Human Motion from Human-Scene Interaction

Chapter 7

Physics-Based Human Motion Imitation

7.1 Introduction

Physics-based motion imitation has captured the imagination of vision and graphics communities due to its potential for creating realistic human motion, enabling plausible environmental interactions, and advancing virtual avatar technologies of the future. However, controlling high-degree-of-freedom (DOF) humanoids in simulation presents significant challenges, as they can fall, trip, or deviate from their reference motions, and struggle to recover. For example, controlling simulated humanoids using poses estimated from noisy video observations can often lead humanoids to fall to the ground[229, 231, 456, 457]. These limitations prevent the widespread adoption of physics-based methods, as current control policies cannot handle noisy observations such as video or language.

In order to apply physically simulated humanoids for avatars, the first major challenge is learning a motion imitator (controller) that can faithfully reproduce human-like motion with a high success rate. While reinforcement learning (RL)-based imitation policies have shown promising results, successfully imitating motion from a large dataset, such as AMASS (ten thousand clips, 40 hours of motion), with a *single* policy has yet to be achieved. Attempts to use larger or a mixture of expert policies have been met with some success [395, 411], although they have not yet scaled to

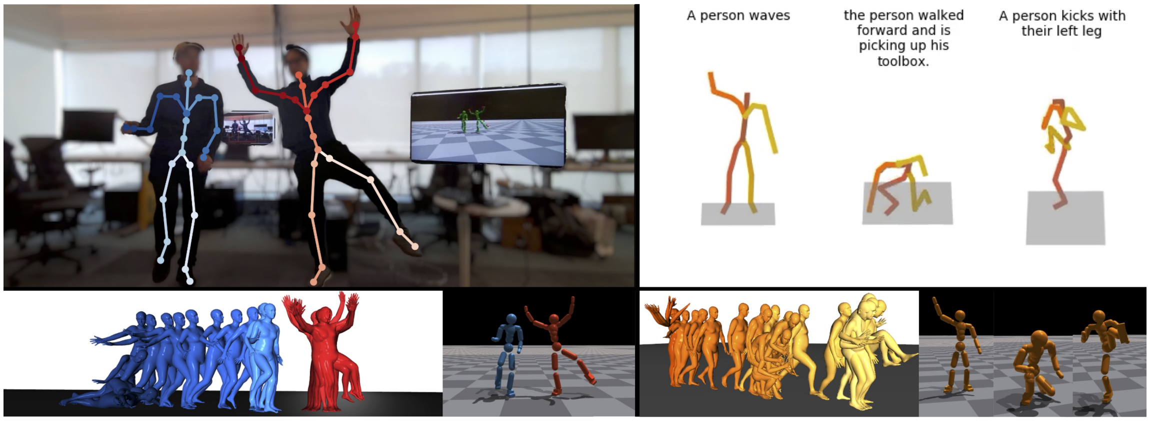

A person waves
the person walked forward and is picking up his toolbox.
A person kicks with their left leg

Figure 7.1: We propose a motion imitator that can naturally recover from falls and walk to far-away reference motion, perpetually controlling simulated avatars without requiring reset. Left: real-time avatars from video, where the blue humanoid recovers from a fall. Right: Imitating 3 *disjoint* clips of motion generated from language, where our controller fills in the blank. The color gradient indicates the passage of time.

the largest dataset. Therefore, researchers have resorted to using external forces to help stabilize the humanoid. Residual force control (RFC) [458] has helped to create motion imitators that can mimic up to 97% of the AMASS dataset [229], and has seen successful applications in human pose estimation from video[117, 230, 460] and language-based motion generation [463]. However, the external force compromises physical realism by acting as a "hand of God" that puppets the humanoid, leading to artifacts such as flying and floating. One might argue that, with RFC, the realism of simulation is compromised, as the model can freely apply a non-physical force on the humanoid.

Another important aspect of controlling simulated humanoids is how to handle noisy input and failure cases. In this work, we consider human poses estimated from video or language input. Especially with respect to video input, artifacts such as floating [463], foot sliding [509], and physically impossible poses are prevalent in popular pose estimation methods due to occlusion, challenging view point and lighting, fast motions *etc.*. To handle these cases, most physics-based methods resort to resetting the humanoid when a failure condition is triggered [229, 231, 457]. However, resetting successfully requires a high-quality reference pose, which is often difficult to obtain due to the noisy nature of the pose estimates, leading to a vicious cycle of falling and resetting to unreliable poses. Thus, it is important to have a controller that can gracefully handle unexpected falls and noisy input, naturally recover from fail-state, and resume imitation.

In this work, our aim is to create a humanoid controller specifically designed to control real-time virtual avatars, where video observations of a human user are used to control the avatar. We design the Perpetual Humanoid Controller (PHC), a *single* policy that achieves a high success rate on motion imitation **and** can recover from fail-state naturally. We propose a progressive multiplicative control policy (PMCP) to learn from motion sequences in the entire AMASS dataset without suffering catastrophic forgetting. By treating harder and harder motion sequences as a different "task" and gradually allocating new network capacity to learn, PMCP retains its ability to imitate easier motion clips when learning harder ones. PMCP also allows the controller to learn fail-state recovery tasks *without compromising* its motion imitation capabilities. Additionally, we adopt Adversarial Motion Prior (AMP)[286] throughout our pipeline and ensure natural and human-like behavior during fail-state recovery. Furthermore, while most motion imitation methods require both estimates of link position and rotation as input, we show that we can design controllers that require only the link positions. This input can be generated more easily by vision-based 3D keypoint estimators or 3D pose estimates from VR controllers.

To summarize, our contributions are as follows: (1) we propose a Perpetual Humanoid Controller that can successfully imitate 98.9% of the AMASS dataset without applying any external forces; (2) we propose the progressive multiplicative control policy to learn from a large motion dataset without catastrophic forgetting and unlock additional capabilities such as fail-state recovery; (3) our controller is task-agnostic and is compatible with off-the-shelf video-based pose estimators as a drop-in solution. We demonstrate the capabilities of our controller by evaluating on both Motion Capture (MoCap) and estimated motion from videos. We also show a live (30 fps) demo of driving perpetually simulated avatars using a webcam video as input.

7.2 Related Works

Physics-based Motion Imitation. Governed by the laws of physics, simulated characters [22, 66, 105, 117, 132, 254, 283, 284, 285, 286, 288, 395, 407, 458] have the distinct advantage of creating natural human motion, human-to-human interaction [217, 412], and human-object interactions [254, 288]. Since most modern physics simulators are not differentiable, training these simulated agents requires RL, which is time-consuming & costly. As a result, most of the work focuses on small-scale use cases such as interactive control based on user input [22, 286, 288, 395], playing sports [217, 254, 412], or other modular tasks (reaching goals [413], dribbling [286], moving around [283], *etc.*). On the other hand, imitating large-scale motion datasets is a challenging yet fundamental task, as an agent that can imitate reference motion can be easily paired with a motion generator to achieve different tasks. From learning to imitate a single clip [284] to datasets [66, 385, 395, 411], motion imitators have demonstrated their impressive ability to imitate reference motion, but are often limited to imitating high-quality MoCap data. Among them, ScaDiver [411] uses a mixture of expert policy to scale up to the CMU MoCap dataset and achieves a success rate of around 80% measured by time to failure. Unicon[395] shows qualitative results in imitation and transfer, but does not quantify the imitator's ability to imitate clips from datasets. MoCapAct[385] first learns single-clip experts on the CMU MoCap dataset, and distills them into a single that achieves around 80% of the experts' performance. The effort closest to ours is UHC [229], which successfully imitates 97% of the AMASS dataset. However, UHC uses residual force control [457], which applies a non-physical force at the root of the humanoid to help balance. Although effective in preventing the humanoid from falling, RFC reduces physical realism and creates artifacts such as floating and swinging, especially when motion sequences become challenging [229, 230]. Compared to UHC, our controller does not utilize any external force.

Fail-state Recovery for Simulated Characters. As simulated characters can easily fall when losing balance, many approaches [66, 288, 337, 367, 457] have been proposed to help recovery. PhysCap [337] uses a floating-base humanoid that does not require balancing. This compromises physical realism, as the humanoid is no longer properly simulated. Egopose [457] designs a fail-safe mechanism to reset the humanoid to the kinematic pose when it is about to fall, leading to potential teleport behavior in which the humanoid keeps resetting to unreliable kinematic poses. NeruoMoCon [154] utilizes sampling-based control and reruns the sampling process if the humanoid falls. Although effective, this approach does not guarantee success and prohibits real-time use cases. Another natural approach is to use an additional recovery policy [66] when the humanoid has deviated from the reference motion. However, since such a recovery policy no longer has access to the reference motion, it produces unnatural behavior, such as high-frequency jitters. To combat this, ASE [288] demonstrates the ability to rise naturally from the ground for a sword-swinging policy. While impressive, in motion imitation the policy not only needs to get up from the ground, but also goes back to tracking the reference motion. In this work, we propose a comprehensive solution to the fail-state recovery problem in motion imitation: our PHC can rise from fallen state and naturally walks back to the reference motion and resume imitation.

Progressive Reinforcement Learning. When learning from data containing diverse patterns, catastrophic forgetting [102, 248] is observed when attempting to perform multi-task or transfer learning by fine-tuning. Various approaches [84, 163, 181] have been proposed to combat this phenomenon, such as regularizing the weights of the network [181], learning multiple experts [163], or increasing the capacity using a mixture of experts [335, 411, 504] or multiplicative control [285]. A paradigm has been studied in transfer learning and domain adaption as progressive learning [42, 59] or curriculum learning [21]. Recently, progressive reinforcement learning [26] has been proposed to distill skills from multiple expert policies. It aims to find a policy that best matches the action distribution of experts instead of finding an optimal mix of experts. Progressive Neural Networks (PNN) [326] proposes to avoid catastrophic forgetting by freezing the weights of the previously learned subnetworks and initializing additional subnetworks to learn new tasks. The experiences from previous subnetworks are forwarded through lateral connections. PNN requires manually choosing which subnetwork to use based on the task, preventing it from being used in motion imitation since reference motion does not have the concept of task labels.

7.3 Method

We define the reference pose as $\hat{\boldsymbol{q}}_t := (\hat{\boldsymbol{\theta}}_t, \hat{\boldsymbol{p}}_t)$, consisting of 3D joint rotation $\hat{\boldsymbol{\theta}}_t \in \mathbb{R}^{J\times 6}$ and position $\hat{\boldsymbol{p}}_t \in \mathbb{R}^{J\times 3}$ of all J links on the humanoid (we use the 6 DoF rotation representation [505]). From reference poses $\hat{\boldsymbol{q}}_{1:T}$, one can compute the reference velocities $\hat{\dot{\boldsymbol{q}}}_{1:T}$ through finite difference, where $\hat{\dot{\boldsymbol{q}}}_t := (\hat{\boldsymbol{\omega}}_t, \hat{\boldsymbol{v}}_t)$ consist of angular $\hat{\boldsymbol{\omega}}_t \in \mathbb{R}^{J\times 3}$ and linear velocities $\hat{\boldsymbol{v}}_t \in \mathbb{R}^{J\times 3}$. We differentiate rotation-based and keypoint-based motion imitation by input: rotation-based imitation relies on reference poses $\hat{\boldsymbol{q}}_{1:T}$ (both rotation and keypoints), while keypoint-based imitation only requires 3D keypoints $\hat{\boldsymbol{p}}_{1:T}$. As a notation convention, we use $\tilde{\cdot}$ to represent kinematic quantities (without physics simulation) from pose estimator/keypoint detectors, $\hat{\cdot}$ to denote ground truth quantities from Motion Capture (MoCap), and normal symbols without accents for values from the physics simulation. We use “imitate”, “track”, and “mimic” reference motion interchangeably. In Sec.7.3.1, we first set up the preliminary of our main framework. Sec.7.3.2 describes our progressive multiplicative control policy to learn to imitate a large dataset of human motion and recover from fail-states. Finally, in Sec.7.3.3, we briefly describe how we connect our task-agnostic controller to off-the-shelf video pose estimators and generators for real-time use cases.

7.3.1 Goal Conditioned Motion Imitation with Adversarial Motion Prior

Our controller follows the general framework of goal-conditioned RL (Fig.7.3), where a goal-conditioned policy π_{PHC} is tasked to imitate reference motion $\hat{\boldsymbol{q}}_{\boldsymbol{1:t}}$ or keypoints $\hat{\boldsymbol{p}}_{1:T}$. Similar to prior work [229, 284], we formulate the task as a Markov Decision Process (MDP) defined by the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \boldsymbol{\mathcal{R}}, \gamma \rangle$ of states, actions, transition dynamics, reward function, and discount factor. The physics simulation determines state $\boldsymbol{s}_t \in \mathcal{S}$ and transition dynamics $\mathcal{T}$ while our policy π_{PHC} computes per-step action $\boldsymbol{a}_t \in \mathcal{A}$. Based on the simulation state $\boldsymbol{s}_t$ and reference motion $\hat{\boldsymbol{q}}_t$, the reward function $\boldsymbol{\mathcal{R}}$ computes a reward $r_t = \boldsymbol{\mathcal{R}}(\boldsymbol{s}_t, \hat{\boldsymbol{q}}_t)$ as the learning signal for our policy. The policy's goal is to maximize the discounted reward $\mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} r_t\right]$, and we use the proximal policy gradient (PPO) [330] to learn π_{PHC}.

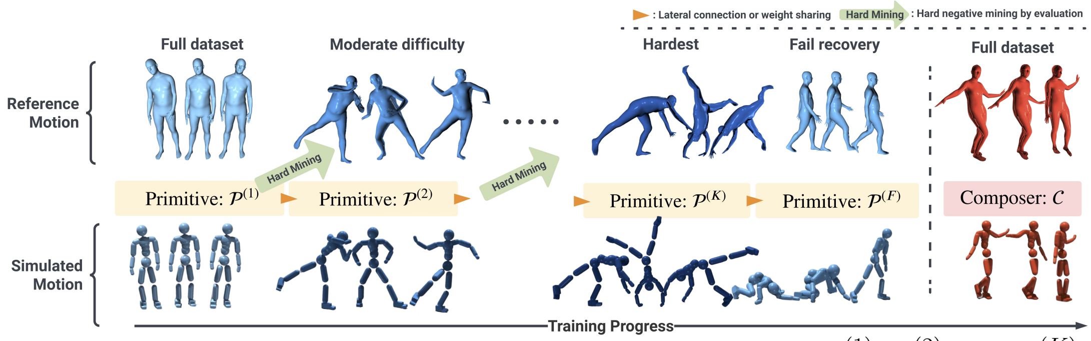

: Lateral connection or weight sharing
Hard Mining
: Hard negative mining by evaluation
Full dataset
Moderate difficulty
Hardest
Fail recovery
Full dataset
Reference Motion
Hard Mining
Hard Mining
Primitive: $\mathcal{P}^{(1)}$
Primitive: $\mathcal{P}^{(2)}$
Primitive: $\mathcal{P}^{(K)}$
Primitive: $\mathcal{P}^{(F)}$
Composer: $\mathcal{C}$
Simulated Motion
Training Progress

Figure 7.2: Our progressive training procedure to train primitives $\mathcal{P}^{(1)}, \mathcal{P}^{(2)}, \cdots, \mathcal{P}^{(K)}$ by gradually learning harder and harder sequences. Fail recovery $\mathcal{P}^{(F)}$ is trained in the end on simple locomotion data; a composer is then trained to combine these frozen primitives.

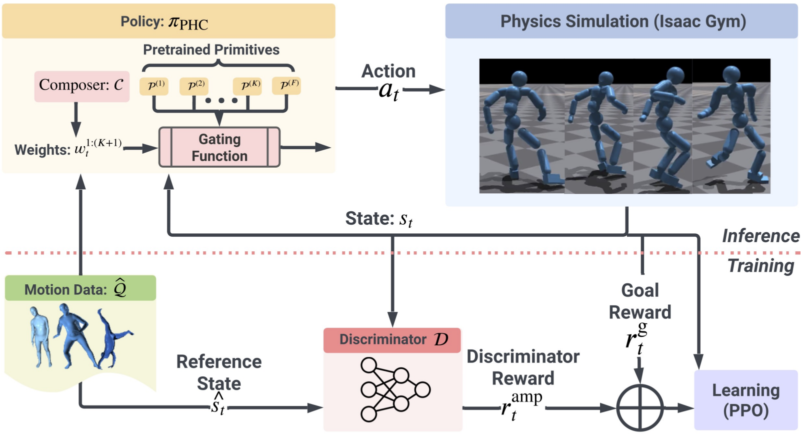


Figure 7.3: Goal-conditioned RL framework with Adversarial Motion Prior. Each primitive $\mathcal{P}^{(k)}$ and composer $\mathcal{C}$ is trained using the same procedure, and here we visualize the final product π_{PHC}.

State. The simulation state $\boldsymbol{s}_t := (\boldsymbol{s}_t^{\mathrm{p}}, \boldsymbol{s}_t^{\mathrm{g}})$ consists of humanoid proprioception $\boldsymbol{s}_t^{\mathrm{p}}$ and the goal state $\boldsymbol{s}_t^{\mathrm{g}}$. Proprioception $\boldsymbol{s}_t^{\mathrm{p}} := (\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t, \boldsymbol{\beta})$ contains the 3D body pose $\boldsymbol{q}_t$, velocity $\dot{\boldsymbol{q}}_t$, and (optionally) body shapes $\boldsymbol{\beta}$. When trained with different body shapes, $\boldsymbol{\beta}$ contains information about the length of the limb of each body link [231]. For rotation-based motion imitation, the goal state $\boldsymbol{s}_t^{\mathrm{g}}$ is defined as the difference between the next time step reference quantitives and their simulated counterpart:

$$\boldsymbol{s}_t^{\text{g-rot}} := (\hat{\boldsymbol{\theta}}_{t+1} \ominus \boldsymbol{\theta}_t, \hat{\boldsymbol{p}}_{t+1} - \boldsymbol{p}_t, \hat{\boldsymbol{v}}_{t+1} - \boldsymbol{v}_t, \hat{\boldsymbol{\omega}}_{t+1} - \boldsymbol{\omega}_t, \hat{\boldsymbol{\theta}}_{t+1}, \hat{\boldsymbol{p}}_{t+1})$$

where $\ominus$ calculates the rotation difference. For keypoint-only imtiation, the goal state becomes

$$\boldsymbol{s}_t^{\text{g-kp}} := (\hat{\boldsymbol{p}}_{t+1} - \boldsymbol{p}_t, \hat{\boldsymbol{v}}_{t+1} - \boldsymbol{v}_t, \hat{\boldsymbol{p}}_{t+1}).$$

All of the above quantities in $\boldsymbol{s}_t^{\mathrm{g}}$ and $\boldsymbol{s}_t^{\mathrm{p}}$ are normalized with respect to the humanoid's current facing direction and root position [229, 413].

Reward. Unlike prior motion tracking policies that only use a motion imitation reward, we use the recently proposed Adversarial Motion Prior [286] and include a discriminator reward term throughout our framework. Including the discriminator term helps our controller produce stable and natural motion and is especially crucial in learning natural fail-state recovery behaviors. Specifically, our reward is defined as the sum of a task reward r_t^{g}, a style reward r_t^{amp}, and an additional energy penalty r_t^{energy} [284]:

$$r_t = 0.5 r_t^{\mathrm{g}} + 0.5 r_t^{\mathrm{amp}} + r_t^{\mathrm{energy}}. \tag{7.1}$$

For the discriminator, we use the same observations, loss formulation, and gradient penalty as AMP [286]. The energy penalty is expressed as $-0.0005 \cdot \sum_{j \in \text{ joints }} |\boldsymbol{\mu}_{\boldsymbol{j}} \boldsymbol{\omega}_{\boldsymbol{j}}|^2$ where $\boldsymbol{\mu}_{\boldsymbol{j}}$ and $\boldsymbol{\omega}_{\boldsymbol{j}}$ correspond to the joint torque and the joint angular velocity, respectively. The energy penalty [104] regulates the policy and prevents high-frequency jitter of the foot that can manifest in a policy trained without external force (see Sec.7.4.1). The task reward is defined based on the current training objective, which can be chosen by switching the reward function for motion imitation $\boldsymbol{\mathcal{R}}^{\text{imitation}}$ and fail-state recovery $\boldsymbol{\mathcal{R}}^{\text{recover}}$. For motion tracking, we use:

$$\begin{aligned} r_t^{\text{g-imitation}} &= \boldsymbol{\mathcal{R}}^{\text{imitation}}(\boldsymbol{s}_t, \hat{\boldsymbol{q}}_t) = w_{\text{jp}} e^{-100\|\hat{\boldsymbol{p}}_t - \boldsymbol{p}_t\|} \\ &+ w_{\text{jr}} e^{-10\|\hat{\boldsymbol{q}}_t \ominus \boldsymbol{q}_t\|} + w_{\text{jv}} e^{-0.1\|\hat{\boldsymbol{v}}_t - \boldsymbol{v}_t\|} + w_{\text{j}\omega} e^{-0.1\|\hat{\boldsymbol{\omega}}_t - \boldsymbol{\omega}_t\|} \end{aligned} \tag{7.2}$$

where we measure the difference between the translation, rotation, linear velocity, and angular velocity of the rigid body for all links in the humanoid. For fail-state recovery, we define the reward $r_t^{\text{g-recover}}$ in Eq.7.3.

Action. We use a proportional derivative (PD) controller at each DoF of the humanoid and the action $\boldsymbol{a}_t$ specifies the PD target. With the target joint set as $\boldsymbol{q}_t^d = \boldsymbol{a}_t$, the torque applied at each joint is $\boldsymbol{\tau}^i = \boldsymbol{k}^p \circ (\boldsymbol{a}_t - \boldsymbol{q}_t) - \boldsymbol{k}^d \circ \dot{\boldsymbol{q}}_t$. Notice that this is different from the residual action representation [229, 275, 458] used in prior motion imitation methods, where the action is added to the reference pose: $\boldsymbol{q}_t^d = \hat{\boldsymbol{q}}_t + \boldsymbol{a}_t$ to speed up training. As our PHC needs to remain robust to noisy and ill-posed reference motion, we remove such a dependency on reference motion in our action space. We do not use any external forces [458] or meta-PD control[460].

Control Policy and Discriminator. Our control policy $\pi_{\text{PHC}}(\boldsymbol{a}_t|\boldsymbol{s}_t) = \mathcal{N}(\mu(\boldsymbol{s}_t), \sigma)$ represents a Gaussian distribution with fixed diagonal covariance. The AMP discriminator $\boldsymbol{\mathcal{D}}(\boldsymbol{s}_{t-10:t}^{\text{p}})$ computes a real and fake value based on the current prioproception of the humanoid. All of our networks (discriminator, primitive, value function, and discriminator) are two-layer multilayer perceptrons (MLP) with dimensions [1024, 512].

Humanoid. Our humanoid controller can support any human kinematic structure, and we use the SMPL [219] kinematic structure following prior arts [229, 230, 460]. The SMPL body contains 24 rigid bodies, of which 23 are actuated, resulting in an action space of $\boldsymbol{a}_t \in \mathbb{R}^{23 \times 3}$. The body proportion can vary based on a body shape parameter $\beta \in \mathbb{R}^{10}$.

Initialization and Relaxed Early Termination. We use reference state initialization (RSI) [284] during training and randomly select a starting point for a motion clip for imitation. For early termination, we follow UHC [229] and terminate the episode when the joints are more than 0.5 meters globally on average from the reference motion. Unlike UHC, we remove the ankle and toe joints from the termination condition. As observed by RFC [458], there exists a dynamics mismatch between simulated humanoids and real humans, especially since the real human foot is multisegment [272]. Thus, it is not possible for the simulated humanoid to have the exact same foot movement as MoCap, and blindly following the reference foot movement may lead to the humanoid losing balance. Thus, we propose Relaxed Early Termination (RET), which allows the humanoid's ankle and toes to slightly deviate from the MoCap motion to remain balanced. Notice that the humanoid still receives imitation and discriminator rewards for these body parts, which prevents these joints from moving in a nonhuman manner. We show that though this is a small detail, it is conducive to achieving a good motion imitation success rate.

Hard Negative Mining. When learning from a large motion dataset, it is essential to train on harder sequences in the later stages of training to gather more informative experiences. We use a similar hard negative mining procedure as in UHC [229] and define hard sequences by whether or not our controller can successfully imitate this sequence. From a motion dataset $\hat{\boldsymbol{Q}}$, we find hard sequences $\hat{\boldsymbol{Q}}_{\text{hard}} \subseteq \hat{\boldsymbol{Q}}$ by evaluating our model over the entire dataset and choosing sequences that our policy fails to imitate.

7.3.2 Progressive Multiplicative Control Policy

As training continues, we notice that the performance of the model plateaus as it forgets older sequences when learning new ones. Hard negative mining alleviates the problem to a certain extent, yet suffers from the same issue. Introducing new tasks, such as fail-state recovery, may further degrade imitation performance due to catastrophic forgetting. These effects are more concretely categorized in the Appendix (App. C). Thus, we propose a progressive multiplicative control policy (PMCP), which allocates new subnetworks (primitives $\boldsymbol{\mathcal{P}}$) to learn harder sequences.

Progressive Neural Networks (PNN). A PNN [326] starts with a single primitive network $\boldsymbol{\mathcal{P}}^{(1)}$ trained on the full dataset $\hat{\boldsymbol{Q}}$. Once $\boldsymbol{\mathcal{P}}^{(1)}$ is trained to convergence on the entire motion dataset $\hat{\boldsymbol{Q}}$ using the imitation task, we create a subset of hard motions by evaluating $\boldsymbol{\mathcal{P}}^{(1)}$ on $\hat{\boldsymbol{Q}}$. We define convergence as the success rate on $\hat{\boldsymbol{Q}}^{(k)}_{\text{hard}}$ no longer increases. The sequences that $\boldsymbol{\mathcal{P}}^{(1)}$ fails on is formed as $\hat{\boldsymbol{Q}}^{(1)}_{\text{hard}}$. We then freeze the parameters of $\boldsymbol{\mathcal{P}}^{(1)}$ and create a new primitive $\boldsymbol{\mathcal{P}}^{(2)}$ (randomly initialized) along with lateral connections that connect each layer of $\boldsymbol{\mathcal{P}}^{(1)}$ to $\boldsymbol{\mathcal{P}}^{(2)}$. For more information about PNN, please refer to our supplementary material. During training, we construct each $\hat{\boldsymbol{Q}}^{(k)}_{\text{hard}}$ by selecting the failed sequences from the previous step $\hat{\boldsymbol{Q}}^{(k-1)}_{\text{hard}}$, resulting in a smaller and smaller hard subset: $\hat{\boldsymbol{Q}}^{(k)}_{\text{hard}} \subseteq \hat{\boldsymbol{Q}}^{(k-1)}_{\text{hard}}$. In this way, we ensure that each newly initiated primitive $\boldsymbol{\mathcal{P}}^{(k)}$ is responsible for learning a new and harder subset of motion sequences, as can be seen in Fig.7.2. Notice that this is different from hard-negative mining in UHC [229], as we initialize a new primitive $\boldsymbol{\mathcal{P}}^{(k+1)}$ to train. Since the original PNN is proposed to solve completely new tasks (such as different Atari games), a lateral connection mechanism is proposed to allow later tasks to choose between reuse, modify, or discard prior experiences. However, mimicking human motion is highly correlated, where fitting to harder sequences $\hat{\boldsymbol{Q}}^{(k)}_{\text{hard}}$ can effectively draw experiences from previous motor control experiences. Thus,

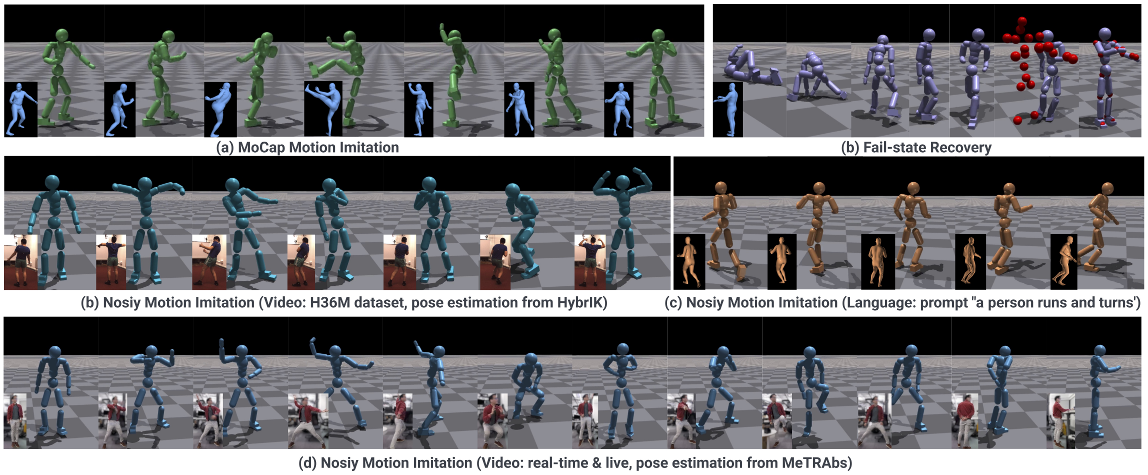

(a) MoCap Motion Imitation
(b) Fail-state Recovery
(b) Nosiy Motion Imitation (Video: H36M dataset, pose estimation from HybrIK)
(c) Nosiy Motion Imitation (Language: prompt "a person runs and turns')
(d) Nosiy Motion Imitation (Video: real-time & live, pose estimation from MeTRAbs)

Figure 7.4: (a) Imitating high-quality MoCap – spin and kick. (b) Recover from fallen state and go back to reference motion (indicated by red dots). (b) Imitating noisy motion estimated from video. (c) Imitating motion generated from language. (d) Using poses estimated from a webcam stream for a real-time simulated avatar.

we also consider a variant of PNN where there are **no lateral** connections, but the new primitives are initialized from the weights of the prior layer. This weight sharing scheme is similar to fine-tuning on the harder motion sequences using a new primitive $\boldsymbol{\mathcal{P}}^{(k+1)}$ and preserve $\boldsymbol{\mathcal{P}}^{(k)}$'s ability to imitate learned sequences.

Fail-state Recovery. In addition to learning harder sequences, we also learn new tasks, such as recovering from fail-state. We define three types of fail-state: 1) fallen on the ground; 2) faraway from the reference motion ($> 0.5m$); 3) their combination: fallen and faraway. In these situations, the humanoid should get up from the ground, approach the reference motion in a natural way, and resume motion imitation. For this new task, we initialize a primitive $\boldsymbol{\mathcal{P}}^{(F)}$ at the end of the primitive stack. $\boldsymbol{\mathcal{P}}^{(F)}$ shares the same input and output space as $\boldsymbol{\mathcal{P}}^{(1)} \cdots \boldsymbol{\mathcal{P}}^{(k)}$, but since the reference motion does not provide useful information about fail-state recovery (the humanoid should not attempt to imitate the reference motion when lying on the ground), we modify the state space during fail-state recovery to remove all information about the reference motion except the root. For the reference joint rotation $\hat{\boldsymbol{\theta}}_t = [\hat{\boldsymbol{\theta}}^0_t, \hat{\boldsymbol{\theta}}^1_t, \cdots \hat{\boldsymbol{\theta}}^J_t]$ where $\hat{\boldsymbol{\theta}}^i_t$ corresponds to the i^{th} joint, we construct $\hat{\boldsymbol{\theta}}'_t = [\hat{\boldsymbol{\theta}}^0_t, \boldsymbol{\theta}^1_t, \cdots \boldsymbol{\theta}^j_t]$ where all joint rotations except the root are replaced with simulated values (without $\hat{\cdot}$). This amounts to setting the non-root joint goals to be identity when computing the goal states: $\boldsymbol{s}^{\text{g-Fail}}_t := (\hat{\boldsymbol{\theta}}'_t \ominus \boldsymbol{\theta}_t, \hat{\boldsymbol{p}}'_t - \boldsymbol{p}_t, \hat{\boldsymbol{v}}'_t - \boldsymbol{v}_t, \hat{\boldsymbol{\omega}}'_t - \boldsymbol{\omega}_t, \hat{\boldsymbol{\theta}}'_t, \hat{\boldsymbol{p}}'_t)$. $\boldsymbol{s}^{\text{g-Fail}}_t$ thus collapse from an imitation objective to a point-goal [413] objective where the only information provided is the relative position and orientation of the target root. When the reference root is too far ($> 5m$), we normalize $\hat{\boldsymbol{p}}'_t - \boldsymbol{p}_t$ as $\frac{5\times(\hat{\boldsymbol{p}}'_t - \boldsymbol{p}_t)}{\|\hat{\boldsymbol{p}}'_t - \boldsymbol{p}_t\|_2}$ to clamp the goal position. Once the humanoid is close enough (e.g.,$< 0.5m$), the goal will switch back to full-motion imitation:

$$\boldsymbol{s}_t^{\text{g}} = \begin{cases} \boldsymbol{s}_t^{\text{g}} & \|\hat{\boldsymbol{p}}_t^0 - \boldsymbol{p}_t^0\|_2 \leq 0.5 \\ \boldsymbol{s}_t^{\text{g-Fail}} & \text{otherwise.} \end{cases} \tag{7.3}$$

To create fallen states, we follow ASE [288] and randomly drop the humanoid on the ground at the beginning of the episode. The faraway state can be created by initializing the humanoid $2 \sim 5$ meters from the reference motion. The reward for fail-state recovery consists of the AMP reward r_t^{amp}, point-goal reward $r_t^{\text{g-point}}$, and energy penalty r_t^{energy}, calculated by the reward function $\boldsymbol{\mathcal{R}}^{\text{recover}}$:

$$r_t^{\text{g-recover}} = \boldsymbol{\mathcal{R}}^{\text{recover}}(\boldsymbol{s}_t, \hat{\boldsymbol{q}}_t) = 0.5 r_t^{\text{g-point}} + 0.5 r_t^{\text{amp}} + 0.1 r_t^{\text{energy}}, \tag{7.4}$$

The point-goal reward is formulated as $r_t^{\text{g-point}} = (d_{t-1} - d_t)$ where d_t is the distance between the root reference and simulated root at the time step t [413]. For training $\boldsymbol{\mathcal{P}}^{(F)}$, we use a handpicked subset of the AMASS dataset named $\boldsymbol{Q}^{\text{loco}}$ where it contains mainly walking and running sequences. Learning using only $\boldsymbol{Q}^{\text{loco}}$ coaxes the discriminator $\boldsymbol{\mathcal{D}}$ and the AMP reward r_t^{amp} to bias toward simple locomotion such as walking and running. We do not initialize a new value function and discriminator while training the primitives and continuously fine-tune the existing ones.

Multiplicative Control. Once each primitive has been learned, we obtain $\{\boldsymbol{\mathcal{P}}^{(1)} \cdots \boldsymbol{\mathcal{P}}^{(K)}, \boldsymbol{\mathcal{P}}^{(F)}\}$, with each primitive capable of imitating a subset of the dataset $\hat{\boldsymbol{Q}}$. In Progressive Networks [326], task switching is performed manually. In motion imitation, however, the boundary between hard and easy sequences is blurred. Thus, we utilize Multiplicative Control Policy (MCP) [285] and train an additional composer $\boldsymbol{\mathcal{C}}$ to dynamically combine the learned primitives. Essentially, we use the pretrained primitives as a informed search space for the composer $\boldsymbol{\mathcal{C}}$, and $\boldsymbol{\mathcal{C}}$ only needs to select which primitives to activate for imitation. Specifically, our composer $\boldsymbol{\mathcal{C}}(\boldsymbol{w}_t^{1:K+1}|\boldsymbol{s}_t)$ consumes the same input as the primitives and outputs a weight vector $\boldsymbol{w}_t^{1:K+1} \in \mathbb{R}^{k+1}$ to activate the primitives. Combining our composer and primitives, we have the PHC's output distribution:

$$\pi_{\text{PHC}}(\boldsymbol{a}_t \mid \boldsymbol{s}_t) = \frac{1}{\boldsymbol{\mathcal{C}}(\boldsymbol{s}_t)} \prod_i^k \boldsymbol{\mathcal{P}}^{(i)}(\boldsymbol{a}_t^{(i)} \mid \boldsymbol{s}_t)^{\boldsymbol{\mathcal{C}}(\boldsymbol{s}_t)}, \quad \boldsymbol{\mathcal{C}}(\boldsymbol{s}_t) \geq 0. \tag{7.5}$$

Table 7.1: Quantitative results on imitating MoCap motion sequences (* indicates removing sequences containing human-object interaction). AMASS-Train*, AMASS-Test*, and H36M-Motion* contain 11313, 140, and 140 high-quality MoCap sequences, respectively.

		AMASS-Train*					AMASS-Test*					H36M-Motion*				
Method	RFC	Succ ↑	$E_{\text{g-mpjpe}}$ ↓	E_{mpjpe} ↓	E_{acc} ↓	E_{vel} ↓	Succ ↑	$E_{\text{g-mpjpe}}$ ↓	E_{mpjpe} ↓	E_{acc} ↓	E_{vel} ↓	Succ ↑	$E_{\text{g-mpjpe}}$ ↓	E_{mpjpe} ↓	E_{acc} ↓	E_{vel} ↓
UHC	✓	97.0 %	36.4	25.1	4.4	5.9	96.4 %	50.0	31.2	9.7	12.1	87.0%	59.7	35.4	4.9	7.4
UHC	✗	84.5 %	62.7	39.6	10.9	10.9	62.6%	58.2	98.1	22.8	21.9	23.6%	133.14	67.4	14.9	17.2
Ours	✗	**98.9 %**	**37.5**	**26.9**	**3.3**	**4.9**	96.4%	**47.4**	**30.9**	6.8	**9.1**	92.9%	50.3	**33.3**	3.7	**5.5**
Ours-kp	✗	98.7%	40.7	32.3	3.5	5.5	**97.1%**	53.1	39.5	**7.5**	10.4	**95.7%**	**49.5**	39.2	**3.7**	5.8

As each $\boldsymbol{\mathcal{P}}^{(k)}$ is an independent Gaussian, the action distribution:

$$\mathcal{N}\left(\frac{1}{\sum_l^k \frac{\boldsymbol{\mathcal{C}}_i(\boldsymbol{s}_t)}{\sigma_l^j(\boldsymbol{s}_t)}} \sum_i^k \frac{\boldsymbol{\mathcal{C}}_i(\boldsymbol{s}_t)}{\sigma_i^j(\boldsymbol{s}_t)} \mu_i^j(\boldsymbol{s}_t), \sigma^j(\boldsymbol{s}_t) = \left(\sum_i^k \frac{\boldsymbol{\mathcal{C}}_i(\boldsymbol{s}_t)}{\sigma_i^j(\boldsymbol{s}_t)}\right)^{-1}\right), \tag{7.6}$$

where $\mu_i^j(\boldsymbol{s}_t)$ corresponds to the $\boldsymbol{\mathcal{P}}^{(\text{i})}$'s j^{th} action dimension. Unlike a Mixture of Expert policies that only activates one at a time (top-1 MOE), MCP combines the actors' distribution and activates all actors at the same (similar to top-inf MOE). Unlike MCP, we progressively train our primitives and make the composer and actor share the same input space. Since primitives are independently trained for different harder sequences, we observe that the composite policy sees a significant boost in performance. During composer training, we interleave fail-state recovery training. The training process is described in Alg.2 and Fig.7.2.

7.3.3 Connecting with Motion Estimators

Our PHC is task-agnostic as it only requires the next time-step reference pose $\tilde{\boldsymbol{q}}_t$ or the keypoint $\tilde{\boldsymbol{p}}_t$ for motion tracking. Thus, we can use any off-the-shelf video-based human pose estimator or generator compatible with the SMPL kinematic structure. For driving simulated avatars from videos, we employ HybrIK [199] and MeTRAbs [358, 359], both of which estimate in the metric space with the important distinction that HybrIK outputs joint rotation $\tilde{\boldsymbol{\theta}}_t$ while MeTRAbs only outputs 3D keypoints $\tilde{\boldsymbol{p}}_t$. For language-based motion generation, we use the Motion Diffusion Model (MDM) [371]. MDM generates disjoint motion sequences based on prompts, and we use our controller's recovery ability to achieve in-betweening.

Table 7.2: Motion imitation on noisy motion. We use HybrIK[199] to estimate the joint rotations $\tilde{\boldsymbol{\theta}}_t$ and uses MeTRAbs [358] for global 3D keypoints $\tilde{\boldsymbol{p}}_t$. HybrIK + MeTRAbs (root): using joint rotations $\tilde{\boldsymbol{\theta}}_t$ from HybrIK and root position $\tilde{\boldsymbol{p}}_t^0$ from MeTRAbs. MeTRAbs (all keypoints): using all keypoints $\tilde{\boldsymbol{p}}_t$ from MeTRAbs, only applicable to our keypoint-based controller.

			H36M-Test-Video*		
Method	RFC	Pose Estimate	Succ ↑	$E_{\text{g-mpjpe}}$ ↓	E_{mpjpe} ↓
UHC	✓	HybrIK + MeTRAbs (root)	58.1%	75.5	49.3
UHC	✗	HybrIK + MeTRAbs (root)	18.1%	126.1	67.1
Ours	✗	HybrIK + MeTRAbs (root)	88.7%	**55.4**	**34.7**
Ours-kp	✗	HybrIK + MeTRAbs (root)	90.0%	55.8	41.0
Ours-kp	✗	MeTRAbs (all keypoints)	**91.9%**	55.7	41.1

7.4 Experiments

We evaluate and ablate our humanoid controller's ability to imitate high-quality MoCap sequences and noisy motion sequences estimated from videos in Sec.7.4.1. In Sec.7.4.2, we test our controller's ability to recovery from fail-state. As motion is best in videos, we provide extensive qualitative results in the supplementary materials. All experiments are run three times and averaged.

Baselines. We compare with the SOTA motion imitator UHC [229] and use the official implementation. We compare against UHC both *with and without* residual force control.

Implementation Details. We uses four primitives (including fail-state recovery) for all our evaluations. PHC can be trained on a single NVIDIA A100 GPU; it takes around a week to train all primitives and the composer. Once trained, the composite policy runs at > 30 FPS. Physics simulation is carried out in NVIDIA's Isaac Gym [242]. The control policy is run at 30 Hz, while simulation runs at 60 Hz. For evaluation, we do not consider body shape variation and use the mean SMPL body shape.

Datasets. PHC is trained on the training split of the AMASS [241] dataset. We follow UHC [229] and remove sequences that are noisy or involve interactions of human objects, resulting in 11313 high-quality training sequences and 140 test sequences. To evaluate our policy's ability to handle unseen MoCap sequences and noisy pose estimate from pose estimation methods, we use the popular H36M dataset [158]. From H36M, we derive two subsets *H36M-Motion** and *H36M-Test-Video**. H36M-Motion* contains 140 high-quality MoCap sequences from the entire H36M dataset. H36M-Test-Video* contains 160 sequences of noisy poses estimated from videos in the H36M test split (since SOTA pose estimation methods are trained on H36M's training split). * indicates the removal of sequences containing human-chair interaction.

Metrics. We use a series of pose-based and physics-based metrics to evaluate our motion imitation performance. We report the success rate (Succ) as in UHC [229], deeming imitation unsuccessful when, at *any point* during imitation, the body joints are on average $> 0.5m$ from the reference motion. Succ measures whether the humanoid can track the reference motion without losing balance or significantly lags behind. We also report the root-relative mean per-joint position error (MPJPE) E_{mpjpe} and the global MPJPE $E_{\text{g-mpjpe}}$ (in mm), measuring our imitator's ability to imitate the reference motion both locally (root-relative) and globally. To show physical realism, we also compare acceleration E_{acc} (mm/frame^2) and velocity E_{vel} (mm/frame) difference between simulated and MoCap motion. All the baseline and our methods are physically simulated, so we do not report any foot sliding or penetration.

7.4.1 Motion Imitation

Motion Imitation on High-quality MoCap. Table7.1 reports our motion imitation result on the AMASS train, test, and H36M-Motion* dataset. Comparing with the baseline **with RFC**, our method outperforms it on almost all metrics across training and test datasets. On the training dataset, PHC has a better success rate while achieving better or similar MPJPE, showcasing its ability to better imitate sequences from the training split. On testing, PHC shows a high success rate on unseen MoCap sequences from both the AMASS and H36M data. Unseen motion poses additional challenges, as can be seen in the larger per-joint error. UHC trained without residual force performs poorly on the test set, showing that it lacks the ability to imitate unseen reference motion. Noticeably, it also has a much larger acceleration error because it uses high-frequency jitter to stay balanced. Compared to UHC, our controller has a low acceleration error even when facing unseen motion sequences, benefiting from the energy penalty and motion prior. Surprisingly, our keypoint-based controller is on par and sometimes outperforms the rotation-based one. This validates that the keypoint-based motion imitator can be a simple and strong alternative to the rotation-based ones.

Motion Imitation on Noisy Input from Video. We use off-the-shelf pose estimators HybrIK [199] and MeTRAbs [358] to extract joint rotation (HybrIK) and keypoints (MeTRAbs) using images from the H36M test set. As a post-processing step, we apply a Gaussian filter to the extracted pose and keypoints. Both HyBrIK and MeTRAbs are per-frame models that do not use any temporal information. Due to depth ambiguity, monocular global pose estimation is highly noisy [358] and suffers from severe depth-wise jitter, posing significant challenge to motion imitators. We find that MeTRAbs outputs better global root estimation $\tilde{\boldsymbol{p}}_t^0$, so we use its $\tilde{\boldsymbol{p}}_t^0$ combined with HybrIK's estimated joint rotation $\tilde{\boldsymbol{\theta}}_t$ (HybrIK + Metrabs (root)). In Table7.2, we report our controller and baseline's performance on imitating these noisy sequences. Similar to results on MoCap Imitation, PHC outperforms the baselines by a large margin and achieves a high success rate ($\sim 90\%$). This validates our hypothesis that PHC is robust to noisy motion and can be used to drive simulated avatars directly from videos. Similarly, we see that keypoint-based controller (ours-kp) outperforms rotation-based, which can be explained by 1) estimating 3D keypoint directly from images is an easier task than estimating joint rotations, so keypoints from MeTRABs are of higher quality than joint rotations from HybrIK; 2) our keypoint-based controller is more robust to noisy input as it has the freedom to use any joint configuration to try to match the keypoints.

Ablations. Table7.3 shows our controller trained with various components disabled. We perform ablation on the noisy input from H36M-Test-Image* to better showcase the controller's ability to imitate noisy data. First, we study the performance of our controller before training to recover from fail-state. Comparing row 1 (R1) and R2, we can see that relaxed early termination (RET) allows our policy to better use the ankle and toes for balance. R2 vs R3 shows that using MCP directly without our progressive training process boosts the network performance due to its enlarged network capacity. However, using the PMCP pipeline significantly boosts robustness and imitation performance (R3 vs. R4). Comparing R4 and R5 shows that PMCP is effective in adding fail-state recovery capability **without** compromising motion imitation. Finally, R5 vs. R6 shows that our keypoint-based imitator can be on-par with rotation-based ones, offering a simpler formulation where only keypoints is needed. For additional ablation on MOE vs. MCP, number of primitives, please refer to the supplement.

Real-time Simulated Avatars. We demonstrate our controller's ability to imitate pose estimates streamed in real-time from videos. Fig.7.4 shows a qualitative result on a live demonstration of using poses estimated from an office environment. To achieve this, we use our keypoint-based controller and MeTRAbs-estimated keypoints in a streaming fashion. The actor performs a series of motions, such as posing and jumping, and our controller can remain stable. Fig.7.4 also shows our controller's ability to imitate reference motion generated directly from a motion language model MDM [371]. We provide extensive qualitative results in our supplementary materials for our real-time use cases.

Table 7.3: Ablation on components of our pipeline, performed using noisy pose estimate from HybrIK + Metrabs (root) on the H36M-Test-Video* data. RET: relaxed early termination. MCP: multiplicative control policy. PNN: progressive neural networks.

H36M-Test-Video*							
RET	MCP	PNN	Rotation	Fail-Recover	Succ ↑	$E_{\text{g-mpjpe}}$ ↓	E_{mpjpe} ↓
✗	✗	✗	✓	✗	51.2%	56.2	34.4
✓	✗	✗	✓	✗	59.4%	60.2	37.2
✓	✓	✗	✓	✗	66.2%	59.0	38.3
✓	✓	✓	✓	✗	86.9%	**53.1**	**33.7**
✓	✓	✓	✓	✓	88.7%	55.4	34.7
✓	✓	✓	✗	✓	**90.0%**	55.8	41.0

Table 7.4: We measure whether our controller can recover from the fail-states by generating these scenarios (dropping the humanoid on the ground & far from the reference motion) and measuring the time it takes to resume tracking.

	Fallen-State		Far-State		Fallen + Far-State	
Method	Succ-5s ↑	Succ-10s ↑	Succ-5s ↑	Succ-10s ↑	Succ-5s ↑	Succ-10s ↑
Ours	95.0%	98.8%	83.7%	99.5%	93.4%	98.8%
Ours-kp	92.5%	94.6%	95.1%	96.0%	79.4%	93.2%

7.4.2 Fail-state Recovery

To evaluate our controller's ability to recover from fail-state, we measure whether our controller can successfully reach the reference motion within a certain time frame. We consider three scenarios: 1) fallen on the ground, 2) far away from reference motion, and 3) fallen and far from reference. We use a single clip of standing-still reference motion during this evaluation. We generate fallen-states by dropping the humanoid on the ground and applying random joint torques for 150 time steps. We create the far-state by initializing the humanoid 3 meters from the reference motion. Experiments are run randomly 1000 trials. From Tab.7.4 we can see that both of our keypoint-based and rotation-based controllers can recover from fall state with high success rate (> 90%) even in the challenging scenario when the humanoid is both fallen and far away from the reference motion. For a more visual analysis of fail-state recovery, see our supplementary videos.

7.5 Discussions

Limitations. While our purposed PHC can imitate human motion from MoCap and noisy input faithfully, it does not achieve a 100% success rate on the training set. Upon inspection, we find that highly dynamic motions such as high jumping and back flipping are still challenging. Although we can train single-clip controller to **overfit** on these sequences (see the supplement), our full controller often fails to learn these sequences. We hypothesize that learning such highly dynamic clips (together with simpler motion) requires more planning and intent (e.g.,running up to a high jump), which is not conveyed in the single-frame pose target $\hat{\boldsymbol{q}}_{t+1}$ for our controller. The training time is also long due to our progressive training procedure. Furthermore, to achieve better downstream tasks, the current disjoint process (where the video pose estimator is unaware of the physics simulation) may be insufficient; tighter integration with pose estimation [230, 460] and language-based motion generation [463] is needed.

Conclusion and Future Work. We introduce Perpetual Humanoid Controller, a general purpose physics-based motion imitator that achieves high quality motion imitation while being able to recover from fail-states. Our controller is robust to noisy estimated motion from video and can be used to perpetually simulate a real-time avatar without requiring reset. Future directions include 1) improving imitation capability and learning to imitate 100% of the motion sequences of the training set; 2) incorporating terrain and scene awareness to enable human-object interaction; 3) tighter integration with downstream tasks such as pose estimation and motion generation, *etc.*.

Acknowledgements. We thank Zihui Lin for her help in making the plots in this paper. Zhengyi Luo is supported by the Meta AI Mentorship (AIM) program.

```
Function TrainPPO($\pi$, $\hat{\boldsymbol{Q}}^{(k)}$, $\boldsymbol{\mathcal{D}}$, $\boldsymbol{\mathcal{V}}$, $\boldsymbol{\mathcal{R}}$):
    while not converged do
        $\boldsymbol{M} \leftarrow \emptyset$ initialize sampling memory
        while $\boldsymbol{M}$ not full do
            $\hat{\boldsymbol{q}}_{1:T} \leftarrow$ sample motion from $\hat{\boldsymbol{Q}}$
            for $t \leftarrow 1...T$ do
                $\boldsymbol{s}_t \leftarrow (\boldsymbol{s}_t^{\mathrm{p}}, \boldsymbol{s}_t^{\mathrm{g}})$ ;
                $\boldsymbol{a}_t \sim \pi(\boldsymbol{a}_t|\boldsymbol{s}_t)$
                $\boldsymbol{s}_{t+1} \leftarrow \mathcal{T}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$            // simulation
                $\boldsymbol{r}_t \leftarrow \boldsymbol{\mathcal{R}}(\boldsymbol{s}_t, \hat{\boldsymbol{q}}_{t+1})$
                store $(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{r}_t, \boldsymbol{s}_{t+1})$ into memory $\boldsymbol{M}$
        $\boldsymbol{\mathcal{P}}^{(k)}, \boldsymbol{\mathcal{V}} \leftarrow$ PPO update using experiences collected in $\boldsymbol{M}$
        $\boldsymbol{\mathcal{D}} \leftarrow$ Discriminator update using experiences collected in $\boldsymbol{M}$
    return $\pi$

Input: Ground truth motion dataset $\hat{\boldsymbol{Q}}$;
$\boldsymbol{\mathcal{D}}$, $\boldsymbol{\mathcal{V}}$, $\hat{\boldsymbol{Q}}^{(1)}_{\mathrm{hard}} \leftarrow \hat{\boldsymbol{Q}}$    // Initialize discriminator, value function, and dataset
for $k \leftarrow 1...K$ do
    Initialize $\boldsymbol{\mathcal{P}}^{(k)}$                    // Lateral connection/weight sharing
    $\boldsymbol{\mathcal{P}}^{(k)} \leftarrow$ TrainPPO($\boldsymbol{\mathcal{P}}^{(k)}$, $\hat{\boldsymbol{Q}}^{(k)}_{hard}$, $\boldsymbol{\mathcal{D}}$, $\boldsymbol{\mathcal{V}}$, $\boldsymbol{\mathcal{R}}^{imitation}$)
    $\hat{\boldsymbol{Q}}^{(k+1)}_{\mathrm{hard}} \leftarrow$ eval( $\boldsymbol{\mathcal{P}}^{(k)}$, $\hat{\boldsymbol{Q}}^{(k)}$ )
    $\boldsymbol{\mathcal{P}}^{(k)} \leftarrow$ freeze $\boldsymbol{\mathcal{P}}^{(k)}$
$\boldsymbol{\mathcal{P}}^{(F)} \leftarrow$ TrainPPO($\boldsymbol{\mathcal{P}}^{(F)}$, $\boldsymbol{Q}^{loco}$, $\boldsymbol{\mathcal{D}}$, $\boldsymbol{\mathcal{V}}$, $\boldsymbol{\mathcal{R}}^{recover}$)          // Fail-state Recovery
$\pi_{\mathrm{PHC}} \leftarrow \{\boldsymbol{\mathcal{P}}^{(1)} \cdots \boldsymbol{\mathcal{P}}^{(K)}, \boldsymbol{\mathcal{P}}^{(F)}, \boldsymbol{\mathcal{C}}\}$
$\pi_{\mathrm{PHC}} \leftarrow$ TrainPPO($\pi_{PHC}$, $\hat{\boldsymbol{Q}}$, $\boldsymbol{\mathcal{D}}$, $\boldsymbol{\mathcal{V}}$, $\{\boldsymbol{\mathcal{R}}^{imitation}, \boldsymbol{\mathcal{R}}^{recover}\}$)       // Train Composer
```

algorithm 3: Learn Progressive Multiplicative Control Policy

Chapter 8

Language-Guided Human Motion Generation in Simulated Scenes

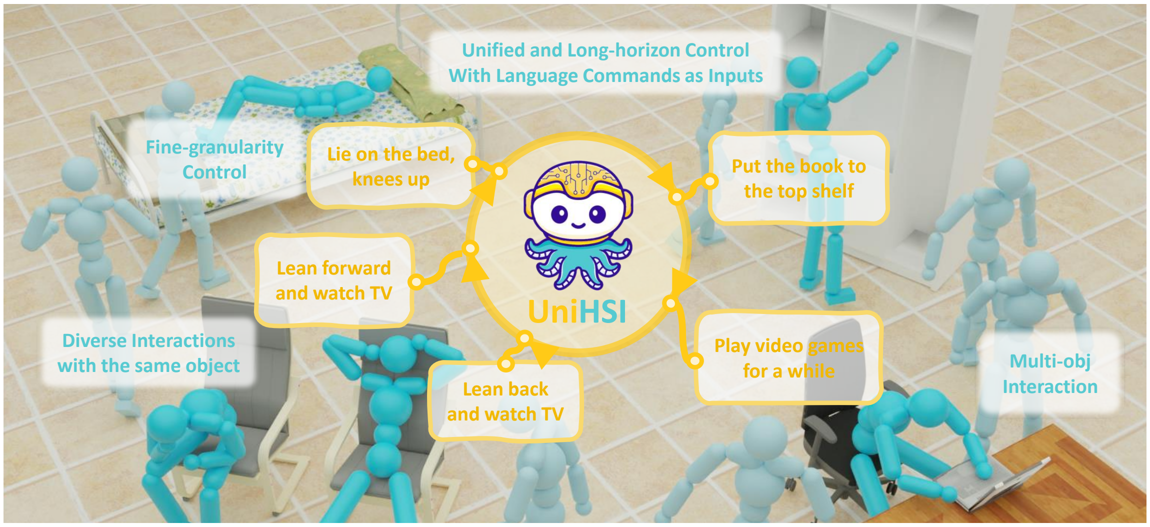


Figure 8.1: UniHSI facilitates unified and long-horizon control in response to natural language commands, offering notable features such as diverse interactions with a singular object, multi-object interactions and fine-granularity control.

8.1 Introduction

Human-Scene Interaction (HSI) constitutes a crucial element in various applications, including embodied AI and virtual reality. Despite the great efforts in this domain to promote motion quality [134, 135, 150, 348, 349, 390, 489] and physical plausibility [134, 135, 150, 348, 349, 390, 489], two key factors, versatile interaction control and the development of a user-friendly interface, are yet to be explored before HSI can be put into practical usage.

This paper aims to provide an HSI system that supports versatile interaction control through language commands, one of the most uniform and accessible interfaces for users. Such a system requires: 1) Aligning language commands with precise interaction execution, 2) Unifying diverse interactions within a single model to ensure scalability. To achieve this, the initial effort involves the uniform definition of different interactions. We propose that interaction itself contains a strong prior in the form of human-object contact regions. For example, in the case of "lie down on the bed", it can be interpreted as "first the pelvis contacts the mattress of the bed, then the head contacts the pillow". To this end, we formulate interaction as ordered sequences of human joint-object part contact pairs, which we refer to as *Chain of Contacts (CoC)*. Unlike previous contact-driven methods, which are limited to supporting specific interactions through manual design, our interaction definition is generalizable to versatile interactions and capable of modeling multi-round transitions. The recent advancements in Large Language Models have made it possible to translate language commands into CoC. The structured formulation then can be uniformly processed for the downstream controller to execute.

Following the above formulation, we propose **UniHSI**, the first **Uni**fied physical **HSI** framework with language commands as inputs. UniHSI consists of a high-level **LLM Planner** to translate language inputs into the task plans in the form of CoC and a low-level **Unified Controller** for executing these plans. Combining language commands and background information such as body joint names and object part layout, we harness prompt engineering techniques to instruct LLMs to plan interaction step by step. We design the TaskParser to support the unified execution. It serves as the core of the Unified Controller. Following CoC, the TaskParser collects information including joint poses and object point clouds from the physical environment, then formulates them into uniform task observations and task objectives.

As illustrated in Fig. 8.1, the Unified Controller models whole-body joints and arbitrary parts of objects in the scenarios to enable fine-granularity control and multi-object interaction. With different language commands, we can generate diverse interactions with the same object. Unlike previous methods that only model a limited horizon of interactions, like "sitting down", we design the TaskParser to evaluate the completion of the current steps and sequentially fetch the next step, resulting in multi-round and long-horizon transition control. The Unified control leverages the adversarial motion prior framework [287] that uses a motion discriminator for realistic motion synthesis and a physical simulation [243] to ensure physical plausibility.

Another impressive feature of our framework is the training is interaction annotation-free. Previous methods typically require datasets that capture both target objects and the corresponding motion sequences, which demand numerous laboring. In contrast, we leverage the interaction knowledge of LLMs to generate interaction plans. It significantly reduces the annotation requirements and makes versatile interaction training feasible. To this end, we create a novel dataset named **ScenePlan**. It encompasses thousands of interaction plans based on scenarios constructed from PartNet [258] and ScanNet [78] datasets. We conduct comprehensive experiments on ScenePlan. The results illustrate the effectiveness of the model in versatile interaction control and good generalizability on real scanned scenarios.

8.2 Related Works

Kinematics-based Human-Scene Interaction.. How to synthesize realistic human behavior is a long-standing topic. Most existing methods focus on promoting the quality and diversity of humanoid movements [18, 131, 278, 372, 436, 474, 481] but do not consider scene influence. Recently, there has been a growing interest in synthesizing motion with human-scene interactions, driven by its applications in various applications like embodied AI and virtual reality. Many previous methods [134, 135, 150, 348, 349, 390, 401, 479, 489] use data-driven kinematic models to generate static or dynamic interactions. These methods are typically inferior in physical plausibility and prone to synthesizing motions with artifacts, such as penetration, floating, and sliding. The need for additional post-processing to mitigate these artifacts hinders the real-time applicability of these frameworks.

Physics-based Human-Scene Interaction.. Recent advances in physics-based methods (e.g., [136, 171, 267, 287, 289] hold promise for ensuring physical realism through physics-aware simulators. However, they have limitations: 1) They typically require separate policy networks for each task, limiting their ability to learn versatile interactions within a unified controller. 2) These methods often focus on basic action-based control, neglecting finer-grained interaction details. 3) They heavily rely on annotated motion sequences for human-scene interactions, which can be challenging to obtain. In contrast, our UniHSI redesigns human-scene interactions into a uniform representation, driven by world knowledge from our high-level LLM Planner. This allows us to train a unified controller with versatile

Table 8.1: Comparative Analysis of Key Features between UniHSI and Preceding Methods.

Methods	Unified Interact.	Language Input	Long horizon Transit.	Interact. Annot. free	Control Joints	Multi-obj Interact.
NSM [348]			✓		3 (pelvis, hands)	✓
SAMP [134]					1 (pelvis)	
COUCH [479]					3 (pelvis, hands)	✓
HUMANISE [401]	✓	✓			-	
ScenDiffuser [156]	✓	✓			-	
PADL [171]		✓	✓	✓	-	
InterPhys [136]					4 (pelvis, head, hands)	
Ours	✓	✓	✓	✓	15 (whole-body)	✓

interaction skills without the need for annotated motion sequences. Key feature comparisons are in Tab. 8.1.

Languages in Human Motion Control.. Incorporating language understanding into human motion control has become a recent research focus. Existing methods primarily focus on scene-agnostic motion synthesis [63, 165, 370, 372, 472, 474, 481] [13]. Generating human-scene interactions using language commands poses additional challenges because the output movements must align with the commands and be coherent with the environment. Zhao et al. [489] generates static interaction gestures through rule-based mapping of language commands to specific tasks. Juravsky et al. [171] utilized BERT [89] to infer language commands, but their method requires pre-defined tasks and different low-level policies for task execution. Wang et al. [401] unified various tasks in a CVAE [442] network with a language interface, but their performance was limited due to challenges in grounding target objects and contact areas for the characters. Recently, there have been some explorations on LLM-based agent control. Brohan et al. [38] uses fine-tuned VLM (Vision Language Model) to directly output actions for low-level robots. Rocamonde et al. [319] employs CLIP-generated cos-similarity as RL training rewards. In contrast, UniHSI utilizes large language models to transfer language commands into the formation of *Chain of Contacts* and design a robust unified controller to execute versatile interaction based on the structured formation.

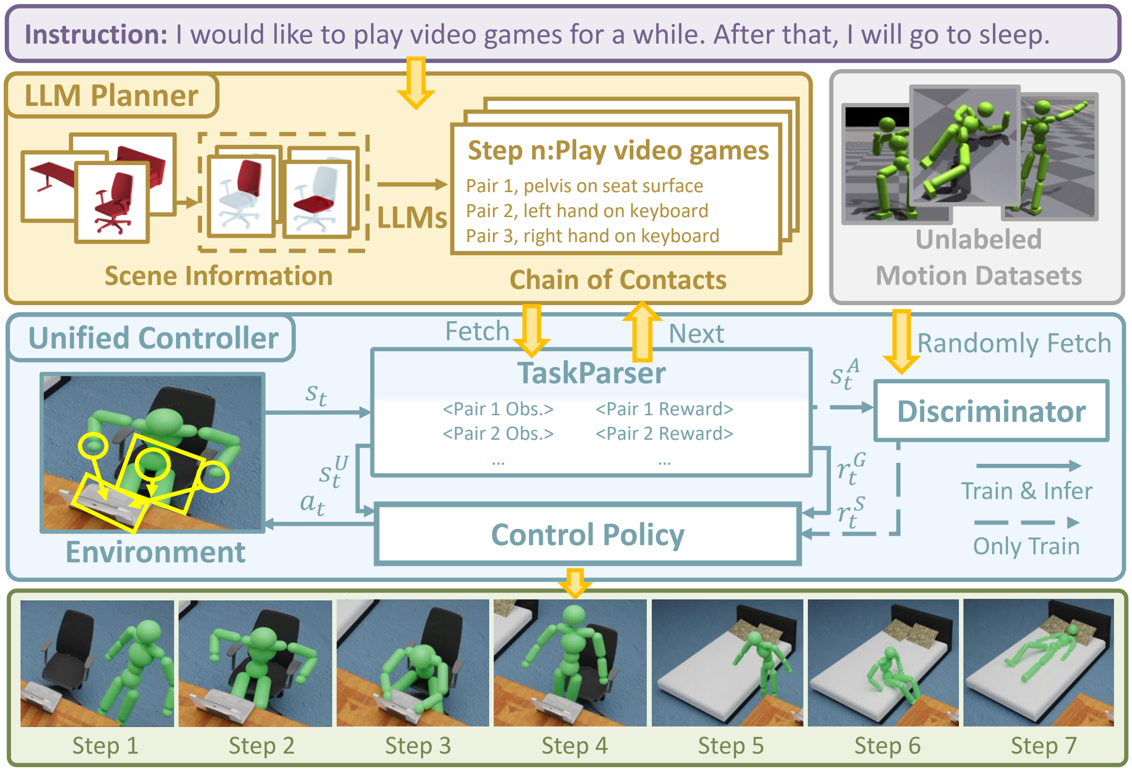


Figure 8.2: **Comprehensive Overview of UniHSI.** The entire pipeline comprises two principal components: the LLM Planner and the Unified Controller. The LLM Planner processes language inputs and background scenario information to generate multi-step plans in the form of CoC. Subsequently, the Unified Controller executes CoC step by step, producing interaction movements.

8.3 Methodology

As shown in Fig. 8.2, UniHSI supports versatile human-scene interaction control following language commands. In the following subsections, we first illustrate how we design the unified interaction formulation as CoC(Sec. 8.3.1). Then we show how we translate language commands into the unified formulation by the LLM Planner (Sec. 8.3.2). Finally, we elaborate on the construction of the Unified Controller (Sec. 8.3.3).

8.3.1 Chain of Contacts

The initial effort of UniHSI lies in the unified formulation of interaction. Inspired by Hassan et al. [135], which infers contact regions of humans and objects based on the interaction gestures of humans, we propose a high correlation between contact regions and interaction types. Further, interactions are not limited to a single gesture but involve sequential transitions. To this end, we can universally define interaction as CoC $\mathcal{C}$, with the formulation as

$$\mathcal{C} = \{\mathcal{S}_1, \mathcal{S}_2, ...\}, \tag{8.1}$$

where $\mathcal{S}_i$ is the i^{th} contact step. Each step $\mathcal{S}$ includes several contact pairs. For each contact pair, we control whether a joint contacts the corresponding object part and the direction of the contact. We construct each contact pair with five elements: an object o, an object part p, a humanoid joint j, the contact type c of j and p, and the relative direction d from j to p. The contact type includes "contact", "not contact", and "not care". The relative direction includes "up", "down", "front", "back", "left", and "right". For example, one contact unit $\{o, p, j, c, d\}$ could be {chair, seat surface, pelvis, contact, up}. In this way, we can formulate each $\mathcal{S}$ as

$$\mathcal{S} = \{\{o_1, p_1, j_1, c_1, d_1\}, \{o_2, p_2, j_2, c_2, d_2\}, ...\}. \tag{8.2}$$

CoC is the output of the LLM Planner and the input of the Unified Controller.

8.3.2 Large Language Model Planner

We leverage LLMs as our planners to infer language commands $\mathcal{L}$ into manageable plans $\mathcal{C}$. As shown in Fig. 8.3, the inputs of the LLM Planner include language commands $\mathcal{L}$, background scenario information $\mathcal{B}$, humanoid joint information $\mathcal{J}$ together with pre-set instructions, rules and examples. Specifically, $\mathcal{B}$ includes several objects $\mathcal{O}$ and their optional spatial layouts. Each object consists of several parts $\mathcal{P}$, i.e.,, a chair could consist of arms, the back, and the seat. The humanoid joint information is pre-defined for all scenarios. We use prompt engineering to combine these elements together and instruct LLMs to output task plans. By modifying instructions in the prompts, we can generate specified numbers of plans for diverse ways of interactions. We can also let LLMs automatically generate plausible plans given the scenes. In this way, we build our interaction datasets to train and evaluate the Unified Controller.

8.3.3 Unified Controller

The Unified Controller takes multi-step plans $\mathcal{C}$ and background scenarios in the form of meshes and point clouds as input and outputs realistic movements coherent to the environments.

Preliminary.. We build the controller upon AMP [287]. AMP is a goal-conditioned reinforcement learning framework incorporated with an adversarial discriminator to model the motion prior. Its objective is defined by a reward function $R(\cdot)$ as

$$R(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}, \mathcal{G}) = w^G R^G(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}, \mathcal{G}) + w^S R^S(\boldsymbol{s}_t, \boldsymbol{s}_{t+1}). \tag{8.3}$$

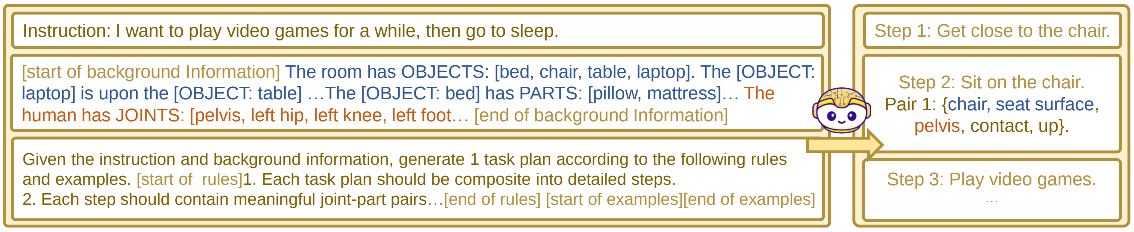
The task reward R^G defines the high-level goal $\mathcal{G}$ an agent should achieve. The style reward R^S encourages the agent to imitate low-level behaviors from motion datasets. w^G and w^S are empirical weights of R^G and R^S, respectively. $\boldsymbol{s}_t$, $\boldsymbol{a}_t$, $\boldsymbol{s}_{t+1}$ are the state at time t, the action at time t, the state at time $t+1$, respectively. The style reward R^S is modeled using an adversarial discriminator D, which is trained according to the objective:

$$\begin{aligned} \arg\min_{D} \; & -\mathbb{E}_{d^{\mathcal{M}}(\boldsymbol{s}_t, \boldsymbol{s}_{t+1})}\left[\log\left(D(\boldsymbol{s}_t^A, \boldsymbol{s}_{t+1}^A)\right)\right] - \mathbb{E}_{d^{\pi}(\boldsymbol{s}, \boldsymbol{s}_{t+1})}\left[\log\left(1 - D(\boldsymbol{s}^A, \boldsymbol{s}_{t+1}^A)\right)\right] \\ & + w^{\mathrm{gp}} \; \mathbb{E}_{d^{\mathcal{M}}(\boldsymbol{s}, \boldsymbol{s}_{t+1})}\left[\left|\left|\nabla_{\phi} D(\phi)\Big|_{\phi=(\boldsymbol{s}^A, \boldsymbol{s}_{t+1}^A)}\right|\right|^2\right], \end{aligned} \tag{8.4}$$

where $d^{\mathcal{M}}(\boldsymbol{s}, \boldsymbol{s}_{t+1})$ and $d^{\pi}(\boldsymbol{s}, \boldsymbol{s}_{t+1})$ denote the likelihood of a state transition from $\boldsymbol{s}_t$ to $\boldsymbol{s}_{t+1}$ in the dataset $\mathcal{M}$ and the policy π respectively. w^{gp} is an empirical coefficient to regularize gradient penalty. $\boldsymbol{s}^A = \Phi(\boldsymbol{s})$ is the observation for discriminator. The style reward $r^S = R^S(\cdot)$ for the policy is then formulated as:

$$R^S(\boldsymbol{s}_t, \boldsymbol{s}_{t+1}) = -\log(1 - D(\boldsymbol{s}_t^A, \boldsymbol{s}_{t+1}^A)). \tag{8.5}$$

We adopt the key design of motion discriminator for realistic motion modeling. In our implementation, we feed 10 adjacent frames together into the discriminator to assess the style. Our main contribution to the controller parts lies in unifying different tasks. As shown in the left part of Fig. 8.4 (a), AMP [287], as well as most of the previous methods [171, 490], design specified task observations, task objectives, and hyperparameters to train task-specified control policy. In contrast, we unify different tasks into Chains of Contacts and devise a TaskParser to process the uniform representation.



Instruction: I want to play video games for a while, then go to sleep.

[start of background Information] The room has OBJECTS: [bed, chair, table, laptop]. The [OBJECT: laptop] is upon the [OBJECT: table] …The [OBJECT: bed] has PARTS: [pillow, mattress]… The human has JOINTS: [pelvis, left hip, left knee, left foot… [end of background Information]

Given the instruction and background information, generate 1 task plan according to the following rules and examples. [start of rules]1. Each task plan should be composite into detailed steps.
2. Each step should contain meaningful joint-part pairs…[end of rules] [start of examples][end of examples]

Step 1: Get close to the chair.

Step 2: Sit on the chair.
Pair 1: {chair, seat surface, pelvis, contact, up}.

Step 3: Play video games.
…

Figure 8.3: The Procedure for Translating Language Commands into Chains of Contacts.

TaskParser.. As the core of the Unified Controller, the TaskParser is responsible for formulating CoC into uniform task observations and task objectives. It also sequentially fetches steps for multi-round interaction execution.

Given one specific contacting pair $\{o, p, j, c, d\}$, for task observation, the TaskParser collects the corresponding position $\boldsymbol{v}^j \in \mathbb{R}^3$ of the joint j, and point clouds $\boldsymbol{v}^p \in \mathbb{R}^{m\times 3}$ of the object part p from the simulation environment, where m is the point number of point clouds. It selects the nearest point $\boldsymbol{v}^{np} \in \boldsymbol{v}^p$ from $\boldsymbol{v}^p$ to $\boldsymbol{v}^j$ as the target point for contact. We formulate task observation of the single pair as $\{\boldsymbol{v}^{np} - \boldsymbol{v}^j, c, d\}$. For the task observation in the network, we map c and d into digital numbers, but we still use the same notation for simplicity. Combining these contact pairs together, we get the uniform task observations $s^U = \{\{\boldsymbol{v}_1^{np} - \boldsymbol{v}_1^j, c_1, d_1\}, \{\boldsymbol{v}_2^{np} - \boldsymbol{v}_2^j, c_2, d_2\}, ..., \{\boldsymbol{v}_n^{np} - \boldsymbol{v}_n^j, c_n, d_n\}\}$.

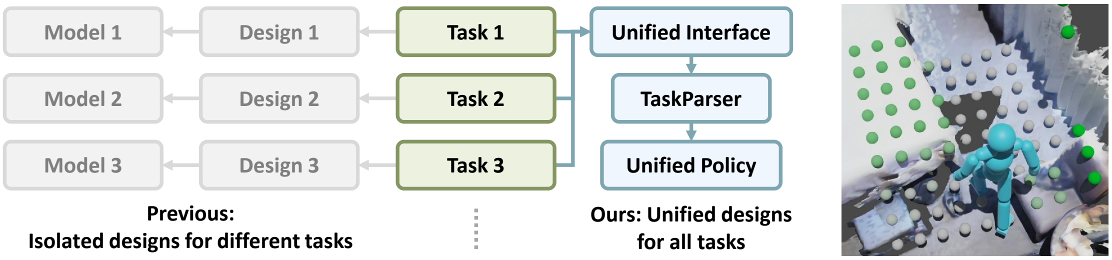
The task reward $r^G = R^G(\cdot)$ is the summarization of all contact pair rewards:

$$R^G = \sum_k w_k R_k, \;\; k = 1, 2, ..., n. \tag{8.6}$$

We model each contact reward R_k according to the contact type c_k. When $c_k =$ contact, the contact reward encourages the joint j to be close to the part p, satisfying the specified direction d. When $c_k =$ notcontact, we hope the joint j is not close to the part p. If $c_k =$ not care, we directly set the reward to max. Following the idea, the k^{th} contact reward R_k is defined as

$$R_k = \begin{cases} w_{\text{dis}}\exp(-w_{dk}||\boldsymbol{d}_k||) + w_{\text{dir}}\max(\overline{\boldsymbol{d}}_k\hat{\boldsymbol{d}}_k, 0), & c_k = \text{contact} \\ 1 - \exp(-w_{dk}||\boldsymbol{d}_k||), & c_k = \text{not contact} \\ 1, & c_k = \text{not care} \end{cases} \tag{8.7}$$

where $\boldsymbol{d}_k = \boldsymbol{v}^{np} - \boldsymbol{v}^j$ indicates the k^{th} distance vector, $\overline{\boldsymbol{d}}_k$ is the normalized unit vector of



(a) Framework Comparison

(b) Ego-centric Heightmap

Figure 8.4: **Design Visualization.** (a) Our framework ensures a unified design across tasks using the unified interface and the TaskParser. (b) The ego-centric height map in a ScanNet scene is depicted by green dots, with darker shades indicating greater height.

$\boldsymbol{d}_k$, $\hat{\boldsymbol{d}}_k$ is the unit direction vector specified by direction d_k, and c_k is the k^{th} contact type. w_{dis}, w_{dir}, w_{dk} are corresponding weights. We set the scale interval of R_k as $[0, 1]$ and use *exp* to ensure it.

Similar to the formulation of contact reward, the TaskParser considers a step to be completed if All $k = 1, 2, ..., n$ satisfy: if $c_k = \text{contact} : ||\boldsymbol{d}_k|| < 0.1$ and $\overline{\boldsymbol{d}}_k \hat{\boldsymbol{d}}_k > 0.8$, if $c_k = \text{not contact} : ||\boldsymbol{d}_k|| > 0.1$, if $c_k = \text{not care}, True$.

Adaptive Contact Weights.. The formulation of 8.6 includes lots of weights to balance different contact parts of the rewards. Empirically setting them requires much laboring and is not generalizable to versatile tasks. To this end, we adaptively set these weights based on the current optimization process. The basic idea is to give parts of rewards that are hard to optimize high rewards while lowering the weights of easier parts. Given R_1, R_2, ..., R_n, we heuristically set their weights to

$$w_k = \frac{1 - R_k}{n - \sum_{k=1,2,...,n} R_k + e}, \tag{8.8}$$

Ego-centric Heightmap.. The humanoid must be scene-aware to avoid collision when navigating or interacting in a scene. We adopt similar approaches in Starke et al. [348], Wang et al. [390], Won et al. [414] that sample surrounding information as the humanoid's observation. We build1 a square ego-centric heightmap that samples the height of surrounding objects (Fig. 8.4 (b)). It is important to extend our methods into real scanned scenarios such as ScanNet [78] in which various objects are densely distributed and easily collide.

Table 8.2: Performance Evaluation on the ScenePlan Dataset.

Source	Success Rate (%) ↑			Contact Error ↓			Success Steps		
	Simple	Mid	Hard	Simple	Mid	Hard	Simple	Mid	Hard
PartNet [258]	85.5	67.9	40.5	0.035	0.037	0.040	2.1	4.1	4.8
wo Adaptive Weights	21.2	5.3	0.1	0.181	0.312	0.487	0.7	1.2	0.0
wo Heightmap	61.6	45.7	0.0	0.068	0.076	-	1.8	3.4	0.0
ScanNet [78]	73.2	43.1	22.3	0.061	0.072	0.062	2.2	3.5	4.8

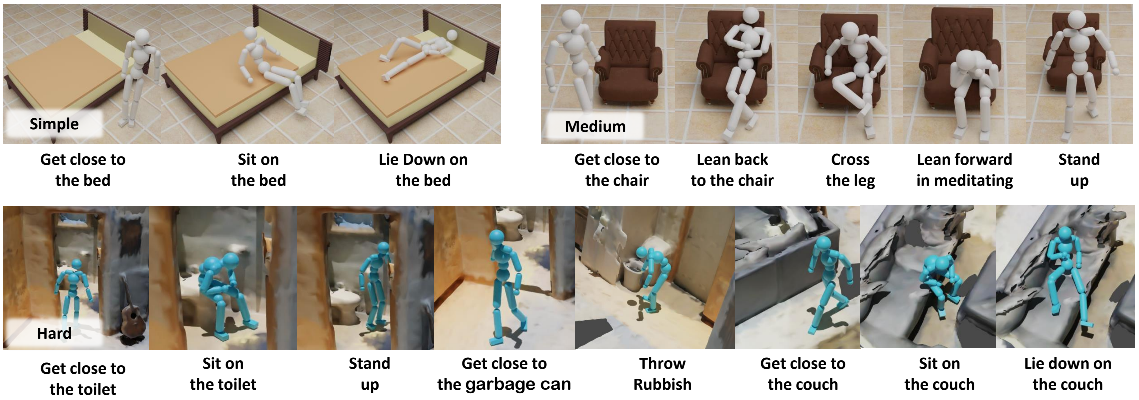


Figure 8.5: Visual Examples Illustrating Tasks of Varying Difficulty Levels.

8.4 Experiments

Existing methods and datasets related to human-scene interactions mainly focus on short and limited tasks [134, 136, 287, 401]. To the best of our knowledge, we are the first method that supports arbitrary horizon interactions with language commands as input. To this end, we construct a novel dataset for training and evaluation. We also conduct various ablations with vanilla baselines and key components of our framework.

8.4.1 Datasets and Metrics

To facilitate the training and evaluation of UniHSI, we construct a novel ScenePlan dataset comprising various indoor scenarios and interaction plans. The indoor scenarios are collected and constructed from object datasets and scanned scene datasets. We leverage our LLM Planner to generate interaction plans based on these scenarios. The training of our model also requires motion datasets to train the motion discriminator, which constrains our agents to interact in natural ways. We follow the practice of Hassan et al. [136] to evaluate the performance of our method.

ScenePlan.. We gather scenarios for ScenePlan from PartNet [258] and ScanNet [78] datasets. PartNet offers indoor objects with fine-grained part annotations, ideal for LLM Planners. We select diverse objects from PartNet and compose them into scenarios. For ScanNet, which contains real indoor room scenes, we collect scenes and annotate key object parts based on fragmented area annotations. We then employ the LLM Planner to generate various interaction plans from these scenarios. Our training set includes 40 objects from PartNet, with 5-20 plausible interaction steps generated for each object. During training, we randomly choose 1-4 objects from this set for each scenario and select their steps as interaction plans. The evaluation set consists of 40 PartNet objects and 10 ScanNet scenarios. We construct objects from PartNet into scenarios either manually or randomly. We generated 1,040 interaction plans for PartNet scenarios and 100 interaction plans for ScanNet scenarios. These plans encompass diverse interactions, including different types, horizons, and multiple objects.

Motion Datasets.. We use the SAMP dataset [134] and CIRCLE [10] as our motion dataset. SAMP includes 100 minutes of MoCap clips, covering common walking, sitting, and lying down behaviors. CIRCLE contains diverse right and left-hand reaching data. We use all clips in SAMP and pick 20 representative clips in CIRCLE for training.

Metrics.. We follow Hassan et al. [136] that uses *Success Rate* and *Contact Error* (*Precision* in Hassan et al. [136]) as the main metrics to measure the quality of interactions quantitatively. Success Rate records the percentage of trials that humanoids successfully complete every step of the whole plan. In our experiments, we consider a trial of n steps to be successfully completed if humanoids finish it in $n \times 10$ seconds. We also record the average error of all contact pairs:

$$\text{ContactError} = \sum_{i,c_i \neq 0} er_i / \sum_{i,c_i \neq 0} 1, \qquad er_i = \begin{cases} ||\boldsymbol{d}_k||, & c_i = \text{contact} \\ \min(0.3 - ||\boldsymbol{d}_k||, 0). & c_i = \text{not contact} \end{cases} \tag{8.9}$$

We further record *Success Steps*, which denotes the average success step in task execution.

8.4.2 Performance on ScenePlan

We initially conducted experiments on our ScenePlan dataset. To measure performance in detail, we categorize task plans into three levels: simple, medium, and hard. We classify plans within 3 steps as simple tasks, those with more than 3 steps but with a single object as medium-level tasks, and those with multiple objects as hard tasks. Simple task plans

Table 8.3: Ablation Study on Baseline Models and Vanilla Implementations.

Methods	Success Rate (%) ↑			Contact Error ↓		
	Sit	Lie Down	Reach	Sit	Lie Down	Reach
NSM - Sit [348]	75.0	-	-	0.19	-	-
SAMP - Sit [134]	75.0	-	-	0.06	-	-
SAMP - Lie Down[134]	-	50.0	-	-	0.05	-
InterPhys - Sit [136]	93.7	-	-	0.09	-	-
InterPhys - Lie Down[136]	-	80.0	-	-	0.30	-
AMP [287]-Sit	77.3	-	-	0.090	-	-
AMP-Lie Down	-	21.3	-	-	0.112	-
AMP-Reach	-	-	**98.1**	-	-	0.016
AMP-Vanilla Combination (VC)	62.5	20.1	90.3	0.093	0.108	0.032
UniHSI	**94.3**	**81.5**	97.5	**0.032**	**0.061**	**0.016**

Table 8.4: UniHSI with different LLMs.

LLM Type	ESR (%) ↑	PC (%) ↑
Human	73.2	-
w. GPT-3.5	35.6	49.1
w. GPT-4	57.3	71.9

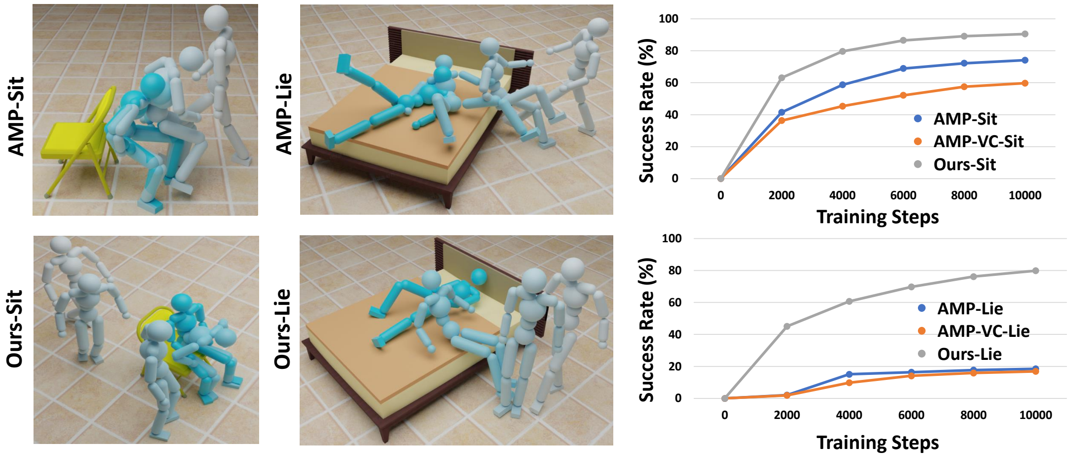
typically involve straightforward interactions. Medium-level plans encompass more diverse interactions with multiple rounds of transitions. Hard task plans introduce multiple objects, requiring agents to navigate between these objects and interact with one or more objects simultaneously. Examples of tasks are illustrated in Fig. 8.5.

As shown in Table 8.2, UniHSI performs well in simple task plans, exhibiting a high Success Rate and low Error. However, as task plans become more diverse and complex, the performance of our model experiences a noticeable decline. Nevertheless, the Success Steps metric continues to increase, indicating that our model still performs well in parts of the plans. It's important to note that the scenarios in the ScenePlan test set are unseen during training, and scenes from ScanNet exhibit a modality gap with the training set. The overall performance on the test set demonstrates the versatile capability, robustness, and generalization ability of UniHSI.

8.4.3 Ablation Studies

Key Components Ablation

Choice of LLMs for UniHSI.. We evaluated different Language Model (LM) choices for the LLM Planner using 100 sets of language commands. We compared task plan Execution Success Rate (ESR) and Planning Correctness (PC) among humans, GPT-



(a) Visual comparisons on task performance

(b) Comparisons on Success Rate v.s. Training Steps

Figure 8.6: **Visual Ablations.** (a) Our model exhibits superior natural and accurate performance compared to baselines in tasks such as “Sit” and “Lie Down”. (b) Our model demonstrates more efficient and effective training procedures.

3.5[265], and GPT-4[266] across 10 tests per plan. PC is evaluated by humans, with choices of ”correct” and ”not correct”. GPT-4 outperformed GPT-3.5, but both LLMs still lag behind human performance. Failures typically involved incomplete planning and out-of-distribution interactions, like GPT-3.5 occasionally skipping transitions or generating out-of-distribution actions like opening a laptop. While using more rules in prompts and GPT-4 can mitigate these issues, errors can still occur.

Adaptive Weights.. Table 8.2 demonstrates that removing Adaptive Weights from our controller leads to a substantial performance decline across all task levels. Adaptive Weights are crucial for optimizing various contact pairs effectively. They automatically adjust weights, reducing them for unused or easily learned pairs and increasing them for more challenging pairs. This becomes especially vital as tasks become more complex.

Ego-centric Heightmap.. Removing the Ego-centric Heightmap results in performance degradation, especially for difficult tasks. This heightmap is essential for agent navigation within scenes, enabling perception of surroundings and preventing collisions with objects. This is particularly critical for challenging tasks involving complex scenarios and numerous objects. Additionally, the Ego-centric Heightmap is key to our model’s ability to generalize to real scanned scenes.

Design Comparison with Previous Methods

Baseline Settings.. We compared our approach to previous methods using simple interaction tasks like “Sit,” “Lie Down,” and “Reach.” Direct comparisons are challenging due to differences in training data and code unavailability for a closely related method [134, 136, 348]. Thus we list the results from their papers and implement a simple version of InterPhys [136]. We integrated key design elements from Hassan et al. [136] into our baseline model [287] to ensure fairness. Task observations and objectives were manually formulated for various tasks, following Hassan et al. [136], with task objectives expressed as:

$$R^G = \begin{cases} 0.7R^{\text{near}} + 0.3R^{\text{far}}, & \text{if distance} > 0.5\text{m} \\ 0.7R^{\text{near}} + 0.3, & \text{otherwise} \end{cases} \tag{8.10}$$

In this equation, R^{far} encourages character movement toward the object, and R^{near} encourages specific task performance when the character is close, necessitating task-specific designs.

We also created a vanilla baseline by consolidating multiple tasks within a single model. We combined task observations from various tasks and included task choices within these observations. We randomly selected tasks and trained them with their respective rewards during training. This experiment involved a total of 70 objects (30 for sitting, 30 for lying down, and 10 for reaching) with 4096 trials per task and random variations in orientation and object placement during evaluation.

Quantitative Comparison.. In Table 8.3, UniHSI consistently outperforms or matches baseline implementations across various metrics. The performance advantage is most pronounced in complex tasks, especially the challenging “Lie Down” task. This improvement stems from our approach of breaking tasks into multi-step plans, reducing task complexity. Additionally, our model benefits from shared motion transitions among tasks, enhancing its adaptability. Figure 8.6 (b) shows that our methods achieve higher success rates and converge faster than baseline implementations. Importantly, the vanilla combination of AMP [287] results in a noticeable performance drop in all tasks while our methods remain effective. This difference is because the vanilla combination introduces interference and inefficiencies in training, whereas our approach unifies tasks into consistent representations and objectives, enhancing multi-task learning.

Qualitative Comparison.. In Figure 8.6 (a), we qualitatively visualize the performance of baseline methods and our model. Our model performs more naturally and accurately than the baselines in tasks like “Sit” and “Lie Down”. This is primarily attributed to the differences in task objectives. Baseline objectives (Eq. 8.10) model the combination of sub-tasks, such as walking close and sitting down, as simultaneous processes. Consequently, agents tend to perform these different goals simultaneously. For example, they may attempt to sit down even if they are not in the correct position or throw themselves like a projectile onto the bed, disregarding the natural task progression. On the other hand, our methods decompose tasks into natural movements through language planners, resulting in more realistic interactions.

8.5 Conclusion

UniHSI is a unified Human-Scene Interaction (HSI) system adept at diverse interactions and language commands. Defined as Chains of Contacts (CoC), interactions involve sequences of human joint-object part contact pairs. UniHSI integrates a Large Language Planner for command translation into CoC and a Unified Controller for uniform execution. Comprehensive experiments showcase UniHSI's effectiveness and generalizability, representing a significant advancement in versatile and user-friendly HSI systems.

Part III

Human Motion from Human-Object Interaction

Chapter 9

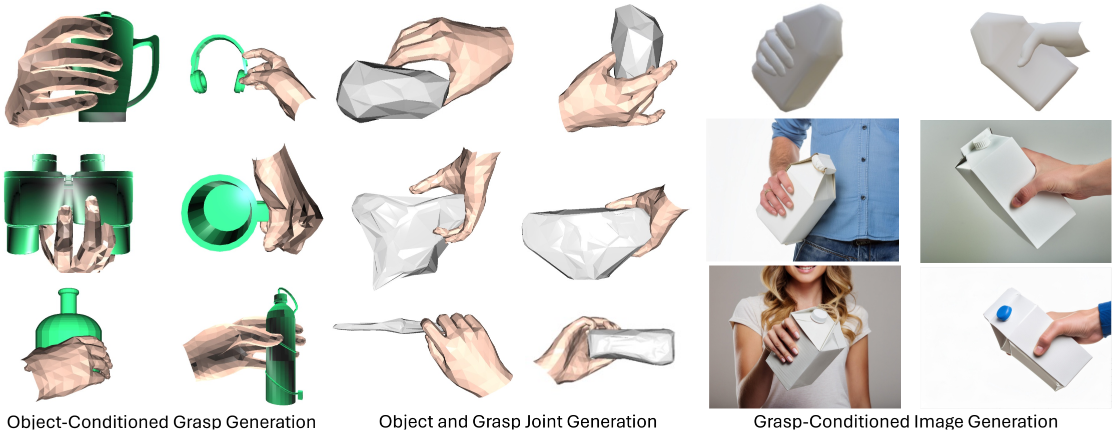
Static Hand-Object Grasp Generation

9.1 Introduction

In this paper, we study generating hand-object grasps. Instead of fitting to a small set of objects presented in 3D hand-object interaction datasets, we wish to generalize to diverse object geometries. We build a joint diffusion model capable of generating the hand poses either conditional to or jointly with the object shapes. The proposed method uses objects in a pure geometric perspective instead of requiring language descriptions or category labels. With access to limited hand-object interaction datasets, it learns an inclusive object shape embedding by leveraging large-scale object shape datasets and generates the paired hand and object in a grasp by denoising a joint latent representation.

Existing works [97, 224, 360] for hand-object grasp generation typically relies on datasets with *full-stack* 3D annotations [97, 360, 441]. However, with the difficulty of capturing 3D object models and hand gestures, this area of research faces notorious limitations of annotated data,e.g., only dozens of object shapes are available in a dataset [97, 360]. Moreover, existing datasets are designed for different hand parametric models [1, 164, 257, 322, 360, 393, 441], which prevents combining different data resources. The limited scale of object annotations causes the method to overfit to biased object shapes and thus bad generalizability of generating grasps on unseen objects.

On the other hand, human beings can transfer a grasping pattern to different objects and make plausible grasping choices for an unseen object based on the object's geometry.


Object-Conditioned Grasp Generation

Object and Grasp Joint Generation

Grasp-Conditioned Image Generation

Figure 9.1: Applications of our proposed method Joint Hand-Object Diffusion (JHOD). Left: generating the hand grasp on an unseen object. Middle: jointly sampling a combination of the hand and the object for a plausible grasp. Right: the generated grasp can be used as guidance for generating photo-realistic grasping images using existing image generation tools [3].

We call such ability *universal grasp generation*. We argue that the key to generating the hand-object grasp universally is an inclusive object shape embedding, which is not possible if we only train the object embedding with limited object shapes. Though *full-stack* 3D annotation is limited, we have large-scale 3D object datasets available. Therefore, in this work, we aim to study if we can combine the large-scale 3D object datasets to help hand-object grasp generation by jointly modeling hand and object representations in a latent space. Specifically, our model can accept grasping data with *partial supervision* in training, such as hand pose and object shape whichever is available, which significantly increases the datasets it can use for learning. Our model can generate either the full 3D grasping scenes with both the posed hand and object or a posed hand conditioned on a given object geometry. We believe such a flexible hand-object interaction generation ability without domain-specific auxiliary information makes one step forward for human-like universal grasp generation.

Following such an intention, we propose our Joint Hand-Object Diffusion (JHOD) model. It follows the latent diffusion model [320] (LDM) to encode and ensemble different modalities into a latent space and then generate plausible samples by the probabilistic denoising process [149]. To learn from more diverse objects, we construct the model by leveraging the object shape encoding and decoding networks pre-trained on the large-scale generic 3D object shape dataset [55]. As we aim to derive a universal grasp generation solution, we remove the category-specific information when constructing the latent code for objects and encode the object shapes in a purely geometric fashion.

As the hand articulation and positions are always coupled to the object shape in hand-object grasping, we design the latent diffusion to denoise the latent code from different modalities as a whole. We decouple each modality representation to allow the model training by fusing data from different resources with heterogeneous annotations. Therefore, our method has independent encoder and decoder networks for different modalities and can optimize the generation of each modality solely. Such a design solves the limitation of object diversity in hand-object interaction datasets as we could leverage the data from generic object shape datasets to tune the object generation part. The model is trained to generate the hand part regarding the corresponding object shape embedding so an inclusive object generation improves the corresponding hand grasp generation on diverse object shapes. To achieve disentangled modality representations to boost training, we propose to use asynchronous denoising schedulers for different modalities during noise diffusion and denoising in training.

To conclude, we develop a joint hand-object grasp generation model to sample from either joint distribution for both hand and object or the object-conditioned distribution for hand generation. It has the advantage of incorporating training resources with different annotations. By the qualitative and quantitative evaluations, our proposed method shows a good performance for grasp generation in both unconditional and object-conditioned settings. Thanks to the posterior generation learned from rich data resources, it shows significantly better performance in generating grasp over out-of-domain object shapes, which is critical for universal grasp generation. Considering that no published works have supported the generation of both hand and object to form a grasp in an end-to-end fashion, we believe our work is pioneering for the universal grasp generation task.

With the flexibility to generate both modalities in hand grasp, our model can facilitate many downstream applications. The grasps are good references to generate photo-realistic images with good geometry alignment and significantly fewer artifacts. We first generate grasps in 3D and use them as conditional signals for image generation [3] can help produce high-quality hand images with challenging gestures and view angles. We demonstrate using grasp from our model as the condition improves grasp alignment on images and relieves artifacts. Some examples of object-conditioned grasp generation, joint generation, and image generation with grasp rendering are presented in Figure 9.1. The main contribution of this work is to propose a generative model capable of generating hand grasp in either unconditional or object-conditioned fashions and the corresponding strategy to enhance the performance with limited available 3D hand-object grasp annotations.

9.2 Related Works

Hand Grasp Generation. Modern generative models [149] are recently introduced into hand grasp reconstruction [445] and generation. Hand grasp reconstruction asks for hand grasp rendering aligned with images while hand grasp generation asks for hand grasp visually or physically plausible. GrabNet [360] is a commonly used method for hand grasp generation but it can hardly be generalized to diverse object shapes. HOIDiffusion [473] uses the 3D hand grasp from GrabNet and an image diffusion model to generate hand-object interaction images. AffordanceDiffusion [446] also studies the generation of hand-object interaction for image synthesis. Instead of generating hand-object interaction images or videos, we focus on improving hand-object interaction in 3D space. Many recent works are based on diffusion models [73, 224, 448] or physics-based simulators [234, 429, 470]. Compared to existing works which typically require a given object [360, 446, 473, 495] shape as input, we desire generating a hand grasp over a given object or generating both the object and hand to form a plausible grasp.

Multi-Modal Generation. With the rise of diffusion models, multi-modal generation has been extended in many areas, such as image+text generation [15] and audio+pixel generation [325]. Our model can jointly generate the hand and object in a grasp, making another type of multi-modal generation. A concurrent work UGG [224] studies to generate both hand and object to form a grasp but it uses the ShadowHand parametric model to leverage the large-scale synthetic ShadowHand grasping datasets [393] and the final results are optimization-based instead of directly from the generative model. On the other hand, another concurrent work G-HOP [448] builds a multi-modal hand-object grasp prior by encoding object shape and hand poses into a unified latent space but a language prompt is required to provide necessary conditional information. In this work, we focus on a dual-modal latent diffusion model to generate the modalities of a 3D object and hand at the same time without requiring optimization or auxiliary conditions. The hand grasp is purely geometric-based. It allows more flexibility for downstream tasks, such as generating photo-realistic images by using the rendering of generated 3D hand-object interaction as a condition.

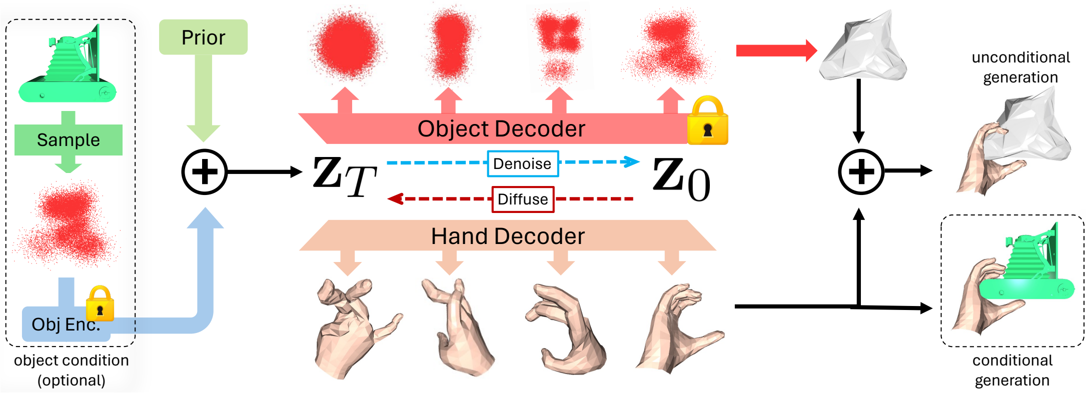


Figure 9.2: The illustration of our proposed Joint Hand-Object Diffusion (JHOD). We present the main modules of the model while removing the secondary modules for simplicity. The optional object condition can turn the generation into object-conditioned.

9.3 Method

In Section 9.3.1, we first provide the formal problem formulation. We then review the related preliminary knowledge for methodology and data representation in Section 9.3.2. Finally, we introduced our proposed method in Section 9.3.3.

9.3.1 Problem Formulation

An instance of hand-object grasp consists of a posed hand, denoted as **H**, and a posed object, denoted as **O**. To generate a hand-object grasp unconditionally, we model the joint distribution of hand and object:

$$\{\mathbf{O}, \mathbf{H}\} \sim \mathcal{D}_\Phi. \tag{9.1}$$

Also, in the community of visual perception, animation, and robotics, people are interested in generating hand grasp conditioned on certain objects, which can be formulated as

$$\{\mathbf{H}\} \sim \mathcal{D}_\Phi|_{\mathbf{O}_g}, \tag{9.2}$$

where $\mathbf{O}_g$ is a provided object shape. In the previous works, the unconditional generation has been under-explored. The methods for conditional hand grasp generation are usually required with no flexibility for object shape generation. They suffer from bad robustness to different object shapes. This is because the existing datasets with 3D annotations of both modalities are notoriously limited in covered object shapes due to the expansiveness of scanning the object shape and capturing the corresponding hand poses. These methods' limitation is underestimated because the object shapes used in training and testing in existing datasets share similar scales and geometries. In this work, we aim to solve both unconditional and object-conditioned grasp generation by a single model. We also wish to enhance the robustness of object shapes by deriving a more inclusive object embedding which also improves the generalizability of the hand part generation.

9.3.2 Preliminaries

Latent Diffusion

Our method follows the latent diffusion models (LDM) [320, 343] to generate samples in two modality spaces, i.e., articulable hands, and rigid objects. LDM diffuses and denoises in a latent space instead of on the raw data representations.

To diffuse a data sample, given a clean latent code z_0 from a data sample $y \sim \mathcal{D}_y$, i.e., $z_0 = \mathcal{E}(y) \sim \mathcal{D}_z$, we add noise following a Markov noise process:

$$q(z_t|z_{t-1}) = \mathcal{N}(\sqrt{\alpha_t} z_{t-1}, (1-\alpha_t)I), \quad t \in [1, T], \tag{9.3}$$

where $\alpha_t \in (0, 1)$ are constant hyper-parameters determined by the time step t and a noise scheduler. Though the typical diffusion model [149] trains a denoiser to predict the noise added per time step during the diffusion stage, there is another line of diffusion models [304, 372] that learns to recover the clean data sample directly. The loss turns to

$$L_{\text{LDM}} := \mathbb{E}_{z_0 \sim \mathcal{D}_z, t \sim [1,T]} \Big[||z_0 - G_\Phi(z_t, t)||_2^2 \Big], \tag{9.4}$$

requiring a generative model G_Φ to predict a clean latent code conditioned to a noisy latent code on a corresponding time step. We follow this paradigm of training diffusion models in this work. Besides the diffusion model itself, we also need the conversion between the raw data representation and the latent code. After recovering a clean code $\hat{z}_0$, we can convert it to the raw data representation by a decoder network $\mathcal{X}$ which is trained to be the inverse of $\mathcal{E}$, i.e., ideally $\mathcal{X}(z_0) \sim \mathcal{D}_y$. Therefore, there are three modules with learnable weights: an encoder network $\mathcal{E}$, a decoder network $\mathcal{X}$, and the denoiser network G_Φ.

Modality Representations

Here we introduce the raw representations of the modalities involved in interrupting a hand-object interaction.

Hand. We follow the MANO [322] parametric model to represent hands. Compared to another parametric model ShadowHand [1], MANO is designed for animating non-rigid real human hands instead of robotic hands thus a better fit for animation and photo-realistic generalization tasks. However, its higher complexity causes difficulty in grasping data synthesis. Therefore, there is no MANO-based hand grasping dataset as large-scale as the synthetic ShadowHand-based datasets [393] yet. This prevents research on the joint generation of MANO hands and objects in a straightforward supervised learning fashion. In this work, we directly use MANO parameters as joint rotation angles, i.e. $\theta \in \mathbb{R}^{48}$, and the translation of hands $\mathbf{t}_h \in \mathbb{R}^3$. We use constant mean shape parameters $\beta_0 = \mathbf{0}^{10}$ during linear blend skinning. The parameters of hands are concatenated and encoded by a hand encoder $\mathcal{E}_H$ to be the hand latent code $\mathbf{y}_H \in \mathbb{R}^{51}$.

Object. For object representation, we use the point cloud latent code as $\mathbf{y}_O = \mathbf{h} \in \mathbb{R}^{N \times 4}$, where N is the number of points. As the grasp is modeled in the object-centric frame, no translation or orientation is required in the object pose and shape representation. This convention is aligned with the definition of data in many hand-object grasp datasets, such as OakInk [441].

Object Generation

The limited amount and high category-specific bias of the objects in hand-object interaction (HOI) datasets prevent learning generating grasps robust and generalizable to diverse object shapes. When training on these datasets only, the learned object embedding is usually biased and overfit. By leveraging the large-scale object-only 3D datasets, we can derive a more robust object encoding for the grasp generation than learning from the HOI datasets solely. For the object encoding part, we borrow the point-cloud-based object shape generation method LION [377]. Existing HOI generation methods can only learn object shapes from HOI datasets, including only dozens of objects, while LION learns from a much larger basis, i.e., more than 50,000 objects in ShapeNet [55]. LION generates objects in two stages. In the first step, it derives a global latent code $\mathbf{z}_0^G \in \mathbb{R}^{D_G}$ from a posterior distribution $q_\phi(\mathbf{z}_0^G|\mathbf{P})$ where $\mathbf{P} \in \mathbb{R}^{u \times 3}$ is the object point cloud. Then it samples a point cloud latent $\mathbf{h}_0$ from a posterior $q_{\mathcal{E}_O}(\mathbf{h}_0|\mathbf{P}, \mathbf{z}_0^G)$. LION generates the object shape by a reverse process from sampled latent code

$$p_{\mathcal{E}_O,\psi,\gamma}(\mathbf{P},\mathbf{h}_0,\mathbf{z}_0^G) = p_{\mathcal{X}_O}(\mathbf{P}|\mathbf{h}_0,\mathbf{z}_0^G)p_\psi(\mathbf{h}_0|\mathbf{z}_0^G)p_\gamma(\mathbf{z}_0^G), \tag{9.5}$$

where $\mathcal{X}_O$ is the decoder network, ψ is the generator of point cloud latent code and γ is the distribution of global latent code. In this work, we remove the global latent code from LION to consider the object shape from a purely geometric perspective. We encode and decode the object between point clouds and latent code from the point cloud latent $\mathbf{h}_0$ only. We convert the object generation part to

$$p_{\mathcal{E}_O,\psi'}(\mathbf{P},\mathbf{h}_0) = p_{\mathcal{X}_O}(\mathbf{P}|\mathbf{h}_0)p_{\psi'}(\mathbf{h}_0), \tag{9.6}$$

where ψ' is fine-tuned from ψ by replacing the global prior γ with a standard Gaussian. Thanks to the pre-training from large-scale object datasets [55], we could use the weights of $\mathcal{E}_O$ and $\mathcal{X}_O$ from LION directly. By freezing the encoder and decoder, we fine-tune the object prior distribution with the object shapes from HOI datasets.

9.3.3 Joint Hand-Object Diffusion (JHOD)

We now introduce our proposed Joint Hand-Object Diffusion (JHOD). There are two main goals. First, we would like to use the training data with heterogeneous annotations so that we can go beyond the limited object shapes in HOI datasets. Second, we want a single model capable of object-conditioned hand grasp generation and unconditional hand-object joint generation. We present the overall illustration and key designs of our proposed method in Figure 9.2.

Latent Codes for Different Modalities

For each modality, we have an MLP-based encoding network to convert the raw representation into latent codes:

$$\mathbf{z}_0^H = \mathcal{E}_H(\mathbf{y}_H), \quad \mathbf{z}_0^O = \mathcal{E}_O(\mathbf{y}_O), \tag{9.7}$$

with the dimensions $\mathbf{z}_0^H \in \mathbb{R}^{1\times 512}$ amd $\mathbf{z}_0^O \in \mathbb{R}^{4\times 512}$. They are concatenated to be the final latent code for the whole hand-object interaction configuration

$$\mathbf{z}_0 = \mathbf{z}_0^H \oplus \mathbf{z}_0^O \in \mathbb{R}^{5\times 512}. \tag{9.8}$$

This design allows us to jointly process different modalities without unnecessary entanglement. So we can train the model with partial modality supervision without corrupting the parameters for other modalities.

Asynchronous Denoising Schedulers

We propose to generate either both hand and object or only the hand given an object shape. To realize this, we need a certain degree of entanglement between the two modalities as we need them to cooperate to form a valid grasp. However, we still desire certain disentanglement so that we could supervise the training with partially annotated data, e.g., 3D object shape data. In the usual fashion of organizing multi-modal latent codes in latent diffusion models, the latent codes of different modalities always have the same scale of noise corruption as they share the noise scheduler. In contrast, we desire that the two modalities can be denoised with respect to an arbitrary level of noise in the other modality. Therefore, we propose to use asynchronous denoising schedulers for this purpose during training. Instead of using a single noise scheduler, we have two schedulers t_H and t_O to control the noise patterns in the hand latent code and the object latent code respectively. Following Equation (9.3) we derive the corrupted latent codes as

$$\mathbf{z}_{t_H}^{H} = \Pi_{t=1}^{t_H} q(\mathbf{z}_t^H | \mathbf{z}_{t-1}^H), \quad \mathbf{z}_{t_O}^{O} = \Pi_{t=1}^{t_O} q(\mathbf{z}_t^O | \mathbf{z}_{t-1}^O), \tag{9.9}$$

with the same time step range $t_O, t_H \in [1, T]$. This design allows the diffusion to deal with the noise in each modality separately. During training, it allows using object-only data to supervise the object part only. During sampling, it allows the model to generate only the hand given an object shape as the condition. Similar to Bao et al. [15], we will learn the joint distribution for hand-object pairs and the marginal distribution for the hand at the same time in an end-to-end fashion by manipulating the time schedulers.

Training and Sampling

Training. When training on HOI data samples, with the asynchronous noise schedulers, we can derive the assembly of two corrupted latent codes as

$$\mathbf{z}_{t_H, t_O} = \mathbf{z}_{t_H}^{H} \oplus \mathbf{z}_{t_O}^{O}. \tag{9.10}$$

Due to the element-wise property of MSE loss, we can calculate and back-propagate the gradient to a certain part of the latent codes independently. We leverage this property to train the unconditional and conditional generation at the same time. To learn the joint distribution, we apply the unconditional generation loss

$$L_{\text{uncond.}} = \mathbb{E}_{z_0 \sim \mathcal{D}_{HOI}, \{t_H, t_O\} \sim [1,T]} \Big[||z_0 - G_\Phi(z_{t_H, t_O}, t_H, t_O)||_2^2 \Big], \tag{9.11}$$

where $\mathcal{D}_{HOI}$ is the distribution of the HOI datasets [360, 441]. We also train the model to learn the object-conditioned grasp generation. Given an object latent code $\mathbf{z}_0^O$, we set $t_O = 0$ indicating that the object part is noise-free and serves as a condition to derive the marginal distribution. The object-conditioned generation loss is thus

$$L_{\text{cond.}} = \mathbb{E}_{z_0 \sim \mathcal{D}_{HOI}, t_H \sim [1,T]} \Big[||z_0 - G_\Phi(z_{t_H}, t_H)||_2^2 \Big], \tag{9.12}$$

with $z_{t_H} = z_{t_H}^H \oplus \mathbf{z}_0^O$. The conditional and unconditional generation losses require uncorrupted and corrupted object latent codes respectively. In practice, we combine these two training objectives in a 1:1 ratio for a single draw of training data batch. This training strategy is similar to the classifier-free guidance [148] for diffusion model training. However, the optional condition (object) exists in the input and output instead of in an independent condition vector.

Finally, we could leverage the large-scale 3D object shape datasets to train the object distribution only. We select object samples from both HOI datasets $\mathcal{D}_{HOI}$ and object-only datasets $\mathcal{D}_O$. The loss turns to

$$L_{\text{obj.}} = \mathbb{E}_{z_0^O \sim \mathcal{D}_O \cup \mathcal{D}_{HOI}, t_O \sim [1,T]} \Big[||z_0 - G_\Phi(z_{t_O}, t_O)||_2^2 \Big], \tag{9.13}$$

The latent codes only take the object part as the variable:

$$\mathbf{z}_0 = \epsilon \oplus \mathbf{z}_0^O, \quad \mathbf{z}_{t_O} = \epsilon \oplus \mathbf{z}_{t_O}^O, \quad s.t. \quad \epsilon \sim \mathcal{N}(0, I). \tag{9.14}$$

Obviously, the gradient only influences the object-related parts while keeping the hand part as-is. During training, the object encoder and decoder networks are pre-trained on the ShapeNet [55] and frozen. All other encoder, decoder, and denoiser networks are trained end to end.

Similar to the proximity sensor features [363] and the Grasping Filed [175], we use a distance field to help capture the relation between the hand and the object. Since we eliminate the requirement of object templates to accommodate more general datasets, we can not define the field using object keypoints [97] or templated vertices. Instead, we define the distance field as the vectors between each hand joint and the 10 closest points in the object points. Therefore, we have the distance field as $\mathbf{y}_f \in \mathbb{R}^{16\times 30}$. When the data sample is drawn from HOI datasets, we would also supervise the distance field calculated from the recovered hand gesture and object shape:

$$L_{\text{distance}} = ||\mathbf{y}_f - \hat{\mathbf{y}}_f||_2^2, \tag{9.15}$$

Besides supervision of the raw MANO parameters for hand, we also supervise the joint position by the loss

$$L_{\text{hand\_xyz}} = ||\text{LBS}(\mathbf{y}_H) - \text{LBS}(\hat{\mathbf{y}}_H)||_2^2, \tag{9.16}$$

where LBS$(\cdot)$ is the linear blend skinning process by MANO parametric model to derive the joint kinematic positions. With all the losses introduced, we could train the diffusion model using data with heterogeneous annotations and for unconditional and conditional generation at the same time. The overall loss is

$$L = \mathbb{1}_{z_0 \sim D_{HOI}}(L_{\text{uncond.}} + L_{\text{cond.}} + L_{\text{distance}} + L_{\text{hand\_xyz}}) + L_{\text{obj.}}. \tag{9.17}$$

Sampling. We follow the canonical progressive denoising process for diffusion models in the sampling stage. To sample for joint (unconditional) HOI generation, we synchronize the schedulers $t_H = t_O = t$. Similar to Tevet et al. [372], at each step t, we predict a clean sample by $\hat{z}_0 = G_\Phi(z_t, t_H, t_O)$ from the corrupted code z_t and then noise it back to z_{t-1}. During sampling for object-conditioned grasp generation, we keep $t_O = 0$ and $z_{t_H} = z_{t_H}^H \oplus z_0^O$. Then the denoiser predicts $\hat{z}_0 = G_\Phi(z_{t_H}, t_H)$ and then noise it back to z_{t-1}. In either the unconditional or the conditional generation, we repeat the process above along $t = T \longrightarrow 1$ until the final $\hat{z}_0$ is achieved after T iterations.

9.4 Experiments

9.4.1 Setups

Datasets. We combine the data from multiple resources to train the model. GRAB [360] contains human full-body poses together with 3D objects. We extract the hands from the full-body annotations. For OakInk [441], we use the official training split for training. We also use the contact-adapted synthetic grasp from the OakInk-Shape dataset for training. Besides the hand-object interaction data, we also leverage the rich resources of 3D object

Table 9.1: Quantitative evaluation of grasp generation. All models are trained using GRAB and OakInk-shape training set. We evaluate the methods on the objects from OakInk-test set, objects generated by JHOD, and objects from the ARCTIC dataset.

Objects	Methods	FID ↓	Pene. Dep. ↓	Intsec. Vol. ↓	Sim. Disp. Mean ↓	User Scor.↑
OakInk-test	GrabNet	26.96	0.70	11.93	3.14	3.80
	GrabNet-Refine	25.26	0.63	4.44	2.78	4.00
	JHOD	20.73	0.32	3.87	2.52	4.87
ARCTIC	GrabNet-Refine	37.23	1.22	14.20	12.17	1.33
	JHOD	23.92	0.42	4.55	2.91	4.13
Self-generated	GrabNet-Refine	-	1.17	13.82	10.03	1.33
	JHOD (uncond.)	-	0.51	6.71	7.13	3.53

Table 9.2: The ablation study about the impact of training data on the generation quality over unseen objects from ARCTIC [97]. Adding more training data consistently boosts the generation quality on unseen objects.

OakInk-Shape	GRAB	Object-only Data	FID ↓	Pene. Dep. ↓	Intsec. Vol. ↓	Sim. Disp. Mean ↓	User Scor.↑
✓			27.29	0.51	4.89	3.22	3.87
✓	✓		25.31	0.47	4.61	3.08	4.00
✓	✓	✓	23.92	0.42	4.55	2.91	4.13

data to help train the object part in our model. The object data is also used to fine-tune the LION prior distribution. Fine-tuning prevents improper object shapes from the original distribution (pre-trained from ShapeNet [55]) such as sofas, chairs, and bookcases. We combine the objects from GRAB, OakInk (both the objects from OakInk-Image with grasp annotation and the objects from OakInk-shape without grasps), Affordpose [446] and DexGraspNet [393] as the object data resource. We also leverage the DeepSDF [273] as trained in Ink-base [441] to provide synthetic object data and include them in OakInk-Shape.

Data Pre-Processing. We follow the convention in OakInk-Shape to transform the hands into the object-centric frame. For data from GRAB [360], we extract the MANO parameters from its original SMPL-X [277] annotations. Then, we filter out the frames where the right-hand mesh has no contact with the object in training. For the object representation, we randomly sample 2048 points from mesh vertices to represent each object every time we draw the object in both training and inference to avoid overfitting a fixed set of point clouds. For the objects in the OakInk-Shape dataset, we sample the point clouds from the object mesh surfaces uniformly instead of sampling from the vertices because the annotated mesh vertices are too sparse on smooth surfaces.

Table 9.3: The ablation study about the impact of training data on the generation quality for joint generation of hand and object. Adding more data, either hand-obejct grasp data or object-only data, improves the generation quality.

OakInk-Shape	GRAB	Object-only Data ↓	Pene. Dep. ↓	Intsec. Vol. ↓	Sim. Disp. Mean ↓	User Scor.↑
✓			0.92	8.12	8.23	3.33
✓	✓		0.51	6.71	7.13	3.53
✓	✓	✓	0.43	5.55	6.98	3.73

Table 9.4: Ablation about asynchronous denoising schedulers. We evaluate on the generated object shapes by JHOD.

Asyn.	Pene. Dep. ↓	Intsec. Vol. ↓	Sim. Disp. Mean ↓
	0.57	6.52	7.88
✓	0.43	5.55	6.98

Implementation. The DeepSDF in Ink [441] is trained per category to derive better details and fidelity when interpolating object shapes. We use the DeepSDF to provide synthetic object shapes. For the encoder network to transform modality features to latent codes, we always use 2-layer MLP networks with a hidden dimension of 1024 and an output dimension of 512. For the denoiser, we follow MDM [372] to use a transformer encoder-only backbone-based diffusion network. After decoding the object latent code by the LION decoder, we derive the point cloud set. For visualization purposes, we conduct surface reconstruction from the generated object point cloud. Though advanced surface reconstruction techniques such as some learned solver [282] can provide more details, this part is not our focus and we wish to keep consistency for out-of-domain objects generated. Therefore, we choose the classic Alpha Shape algorithm [95]. Without category labels or other constraints, it is challenging to generate object shapes with highly carved details and it is not our focus in this paper.

Baseline Methods There is no commonly adopted benchmark in the area of grasp generation and some similar methods can follow different evaluation protocols. For example, the provided evaluation protocol of G-HOP [448] is for grasp reconstruction instead of generation and text description or object label is necessary for its generation mode which is not required in our proposed method. On the other hand, another line of works, such as UGG [224] and DexDiffuer [405] focus on grasp generation but within the domain of robotic dexterous hand thus there is no trivial way to compare with their in the same setting. In this work, we focus on hand grasp generation with only object shape as condition or without any condition. We select the widely adopted method GrabNet [360] as the baseline method to compare with in this section. We also use its refined version GrabNet-Refine in some certain comparisons.

Evaluation and Metrics. Unlike the reconstruction tasks where the ground truth is available, it is difficult to do the quantitative evaluation for generation models. We hold the objects from the OakInk-Shape test set and ARCTIC [97] dataset for quantitative evaluations. Referring to the evaluation of image generation quality, we first measure the quality of generated grasp by FID (Frechet Inception Distance) [146] between the images rendered from the ground truth grasps and the generated grasps. We can not directly calculate the FID score on generated 3D parameters as there is no commonly used encoder for this purpose and implementing it ourselves can cause many ambiguities. We also measure the model performance by directly checking the 3D asset quality. We follow previous studies [439, 441] to use three metrics: (1) the penetration depth (Pene. Depth) between hand and object mesh, (2) the solid intersection volume (Intsec. Vol.) between them and (3) the mean displacement in simulation following [137]. Finally, we perform a user study to measure the plausibility and the truthfulness of the generated grasps by scoring (1-5). We invite 15 participants to rate the quality of batches of the mixture of 10 rendering of ground truth grasps and 10 generated grasps. The score 1 indicates "totally fake and implausible" and 5 indicates "plausible enough to be real". Each participant evaluates by averaging the scores over 10 randomly selected batches.

9.4.2 Object-Conditioned Grasp Generation

Our method can generate visually plausible grasp configurations even though the model has not seen these objects during training. We generate samples on the objects from GRAB and OakInk test sets as shown in Figure 9.3 and Figure 9.4 . We perform the quantitative evaluation on the objects unseen during training with the previously introduced metrics. We follow the previous practice [441] to use the widely adopted GrabNet [360] and its refined version GrabNet-refine [441] as the baseline models. We train the models on the GRAB and OakInk-shape train splits. The results are shown in Table 9.1.

Compared to the baseline methods, our proposed method achieves better generation quality per four metrics: FID, Mesh Penetration Depth, Intersection Volume, and Mean Simulation Displacement. The performance advantages are demonstrated on objects from the OakInk test set, generated by our method or ARCTIC dataset.

OakInk training set and training set contain similar objects though not the same. Such a

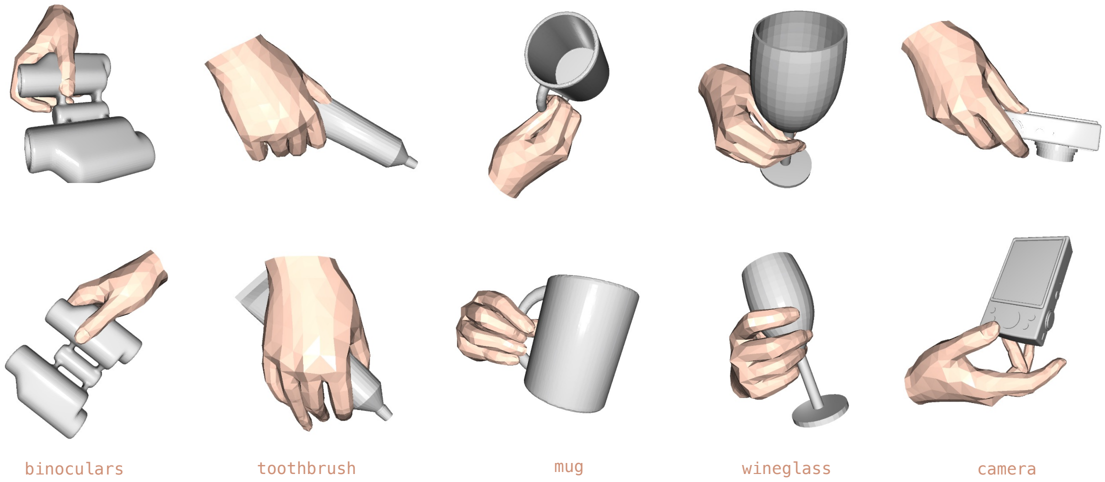


Figure 9.3: Generated samples on objects from GRAB [360] dataset.

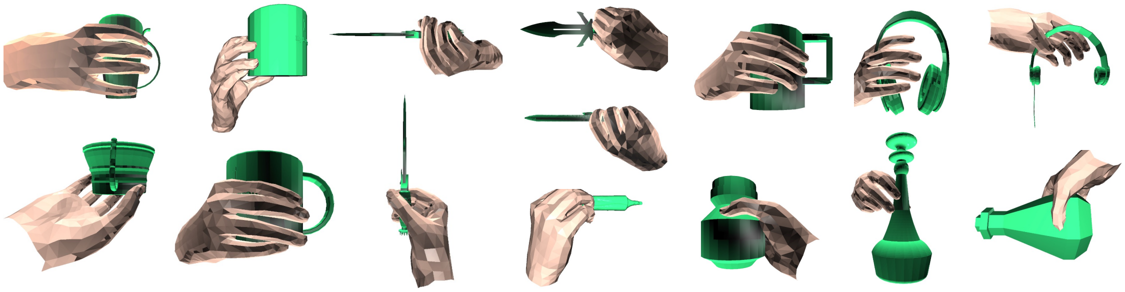


Figure 9.4: Generated samples on objects from OakInk [441] dataset.

biased similarity between the training and test sets exists in many HOI datasets and conceals the limitation of generating grasps on unseen objects. So we also test on the ARCTIC dataset, which is more confidently out-of-distribution from training. GrabNet-Refine fails to generate grasps with decent quality while the advantage of our method becomes more significant. GrabNet-Refine's generation quality is inferior to our method with a much larger margin by all metrics. The experiment reveals that some existing methods, such as GrabNet, face difficulty in generalizing to object shapes that are significantly different from training data. On the other hand, our method learns more generalizable and diverse object prior information and more universal grasp generation ability.

9.4.3 Unconditional Grasp Generation

Our proposed method learns the joint distribution of the latent representation of hand and object. It leverages large-scale object datasets to learn a sufficiently generalizable object shape embedding. To measure the quality of unconditional hand-object grasp, the results are shown in the last row in Table 9.1. The metric scores indicate that the unconditional hand grasp generation quality is also good and close to the generation quality on ARCTIC objects. Compared to the object-conditioned generated grasps from JHOD, the quality of unconditional generation is inferior, which is in fact as expected. Here, we also provide the evaluation results by GrabNet-Refine on the same set of objects generated by our method for reference. And GrabNet-Refine fails to generalize to this set of unseen objects again just as its failure on ARCTIC objects. Such failure of generalizability is previously under-explored because the training/test splits of a HOI dataset usually contain objects from certain categories, with similar scale, geometry, and affordance. The method trained on the training split can learn significantly biased in-domain knowledge about the objects in the test splits. To generate hand grasps on more out-of-domain object shapes would be a challenge to all existing methods.

By designing the method capable of training with object-only data and disentangling the modality representations in the latent diffusion, our method not only has the advantage of generating the object and the hand grasp simultaneously but also achieves much better robustness to generate grasps on out-of-domain objects.

9.4.4 Ablation Study

We now provide ablation studies to show the contributions of different resources to our method's final performance.

Ablation of grasp generation on unseen objects. To provide transparent experimental conclusions, we ablate the training data on the model generation quality. We use the objects from ARCTIC [97] to measure the conditional generation quality on unseen objects. The quantitative evaluation results are shown in Table 9.2. Compared to using only OakInk-Shape data for training, adding GRAB data and object shapes from Object-only datasets both improves the generalization qualities. The improvement from extra GRAB data is easy to explain as training with more HOI data improves the generalizability of the model while it is interesting that adding the object-only datasets also boosts the performance. We believe this is because the extra object data improves the robustness and generalizability of the object encoding. It extends the latent representation expressiveness that the diffusion model learns to generate.

Ablation of unconditional generation quality. The ability to generate both hand and object to form a grasp is another main contribution. We are interested in whether it can benefit from the additional training data as well. We present the ablation study in Table 9.3. Similar to the conditional generation ablation study results, the unconditional generation quality also benefits from additional training data consistently. The metrics of Pene. Dep., Intsec. Vol. and Sim. Disp. Mean indicate that the generated hand-object pair improves plausibility along with adding more training data. The increasing user scoring result suggests that the generated object and hand also become more and more visually realistic along with adding more training data.

Ablation of synchronous denoising schedulers. As one of the main implementation innovations of JHOD, we conduct an ablation about Asynchronous Denoising Schedulers in Table 9.4. For the fairness of the comparison, we keep the curriculum of the training the same by mixing data samples from different resources. Intuitively, we designed this module to decouple the noise level in hand latent and object latent so that the model could better use the data with heterogeneous annotations and learn to denoise a certain modality. Without Asynchronous Denoising Schedulers, the noise diffusion and denoising time step of the object part is the same as the hand latent code. It makes the training biased to the HOI dataset where both modalities are available for supervision. Adding Asynchronous Denoising Schedulers, the denoising is learned with more independence for the object part and the hand part. The model can better learn object shape coding to generate hand grasps. Therefore, when evaluating on the out-of-domain objects generated by JHOD, the generation quality is improved as expected.

9.5 Conclusion

In this work, we propose a joint diffusion model to generate hand-object grasps. By encoding the modalities of objects and hands into a unified latent space, we could generate grasps either jointly or conditionally. We reveal the limitations of existing methods by using limited HOI data for training and explore how to learn grasp generation generalizable to more diverse objects. Our model benefits from data with partial annotation thus relieving the limitation of full-stack annotated data. The generated grasps in both unconditional and conditional settings are of good quality. We believe that the proposed method provides a good exploration of learning generalizable and robust hand-object grasp generation with limited full-stack annotated data.

9.6 More Results

9.6.1 Qualitative Results

We showcase the qualitative visualization results of our proposed method here. We select the object shapes from GRAB and OakInk-Shape that are not used in training as the object condition. The generated hand grasps are presented in Figure 9.5 and Figure 9.6. On the same object or objects with similar shapes, we can observe different hand grasp patterns, which are visually plausible. Given that these object shapes are unseen during training and no text/category description is provided as an extra condition, we believe such grasping generation is made from the model's ability to generate grasping from a pure geometric perspective of the object shape. Moreover, by sampling noise from the latent space, we could generate the object shape and a corresponding hand grasp unconditionally. The results are presented in Figure 9.7. We could notice the good plausibility of the generated hand-object grasp. The hand pose and articulation are well adapted to the object shapes generated. Compared to the object shapes from real-world scanning, the objects generated by our method lack the same level of well carving. This is also related to the surface reconstruction algorithm, i.e., Alpha Shape [95], which we leveraged to reconstruct mesh from point clouds. Applying a more advanced surface reconstruction method can improve the details and fidelity of the object shapes but that is out of the scope of this work. Moreover, as we study the object shapes from a pure geometric perspective, we can't use the surface reconstruction arts that require category prior.

9.6.2 Grasp Diversity

Given a single object shape, we could generate a set of different grasps onto it. Though diversity is expected to be constrained by the hand grasp data available in training, there are some other implicit but more fundamental cues to help generate diverse hand grasp. This underlying but fundamental constraint can be leveraged to enhance the generation robustness over unseen objects if the feature of the object shape is sufficiently generalizable. For example, the model can learn to avoid penetration between the object volume and the hand shape and certain fingers should be close to the surface of objects to form a physically valid grasp. We generate grasps with unseen objects from the OakInk-shape test set as the condition in Figure 9.8. We observe some grasp patterns not provided in the training set, for example, holding a dagger between two fingers. As the implicit concept of forming a valid grasp is always conditioned to object shapes, our strategy of exposing the model

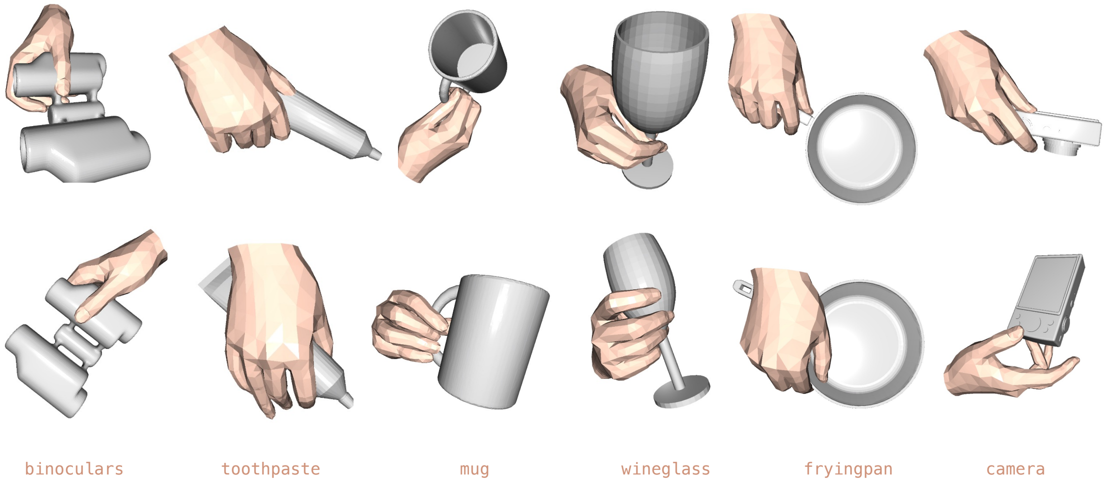


Figure 9.5: Samples of object-conditioned grasp generation on GRAB.

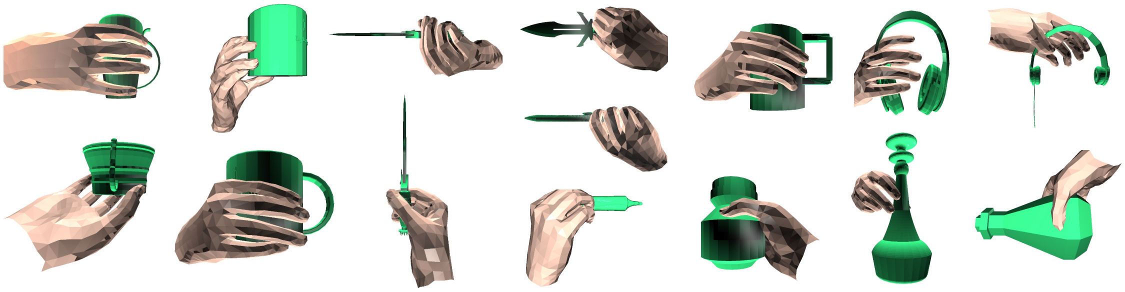


Figure 9.6: Samples of object-conditioned grasp generation on OakInk-Shape.

training to more diverse object shapes can help to learn grasp patterns on unseen and even out-of-domain objects. We believe this is a key reason for making our method outstanding when generating grasps on out-of-domain objects.

9.6.3 Dataset Statistics

Here we provide more details about the statistics of the datasets involved in our training and evaluation in Table 9.5. The HOI datasets with deformable and articulable hand models, i.e., MANO [322], face severe limitations of object resources. Combining the training set of GRAB [360] and OakInk-Image [441] datasets still make just ∼100 objects. On the other hand, AffordPose [164] and DexGraspNet [393] contain more object shapes but the different choices of hand models make them hard to integrate with MANO-parameterized datasets. Fortunately, we have the large-scale 3D object shape dataset ShapeNet [55] with more than

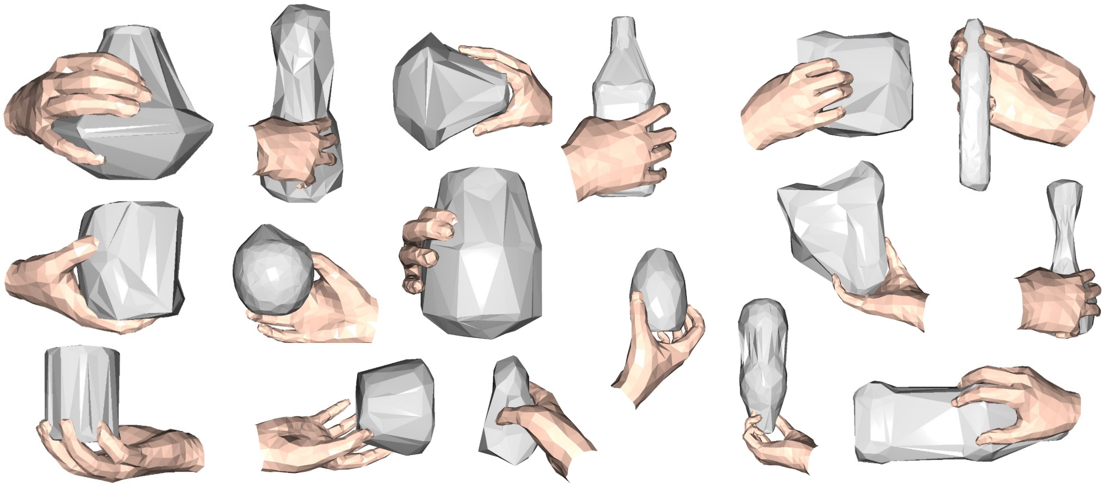


Figure 9.7: Samples of the unconditional generation of hand grasp together with objects.

Table 9.5: Statistics of the datasets. ShapeNet contains the largest number of object assets. However, it is a generic object shape dataset, thus many included objects are not proper for hand grasp. OakInk-Shape contains rich grasp and object data but can not provide annotation for object transformation. We use the grasp data from GRAB and the object data from AffordPose as the supplement during training.

Datasets	#obj	#grasp	real/syn.	hand model
ShapeNet	51,300	-	real	-
GRAB [360]	51	1.3k	real	MANO [322]
OakInk-Image [441]	100	49k	real	MANO [322]
OakInk-Shape [441]	1,700	-	real + synthetic	MANO [322]
AffordPose [164]	641	26k	synthetic	GraspIt [257]
DexGraspNet [393]	5,355	1.32M	synthetic	ShadowHand [1]

50,000 objects. We combine the objects from these datasets in the training for the object part. They help to construct a more universal prior distribution for object generation and the grasp posterior distribution.

9.6.4 Grasp-conditioned Image Generation

We demonstrate using JHOD as a creativity tool to generate 3D grasps and the corresponding images. In Figure 9.9, we first use JHOD to sample interaction pairs of 3D hands and objects. With the rendering of the 3D hand-object grasp as the input, we applied Adobe Firefly [3], an existing tool to generate the images with the depth and edge reference of the

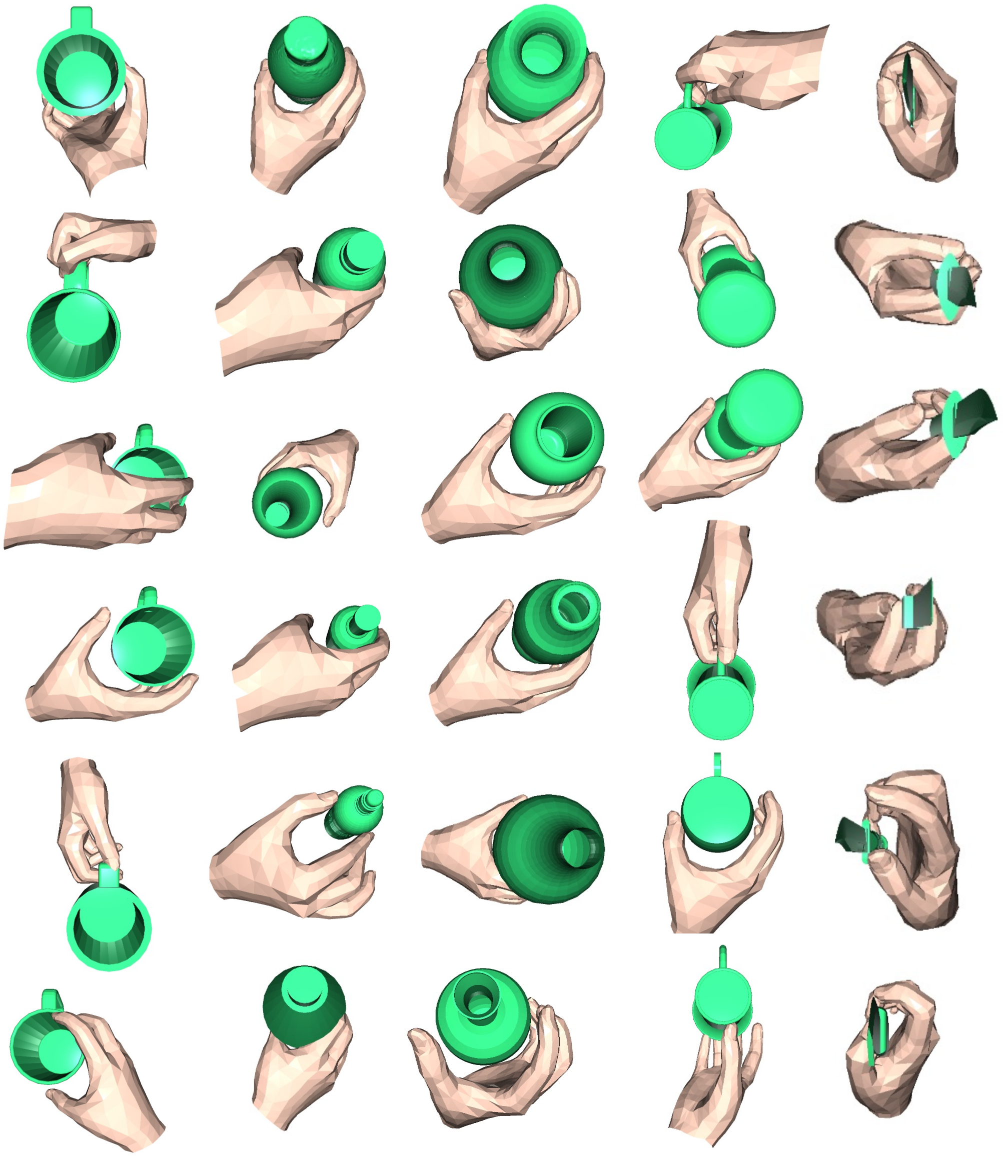


Figure 9.8: Given a single object, our method is capable of generating different grasps on it.

3D pairs. The hand pose and the geometry provide guidance for Firefly to complement the details given simple text prompts. In Figure 9.10, compared to grasping images authorized purely by text prompts, images guided by JHOD's output have more consistent gestures, better-aligned object geometry and boundaries, and fewer hand artifacts. Therefore, with the rise of text-to-image generation, our hand-object grasp generation model can serve as a proxy to enhance the consistency among multiple instances and the visual plausibilities. We also provide a closer look at their detailed in Figure 9.11 to compare the results with and without the generated grasp as a proxy. At each row, the images are generated from the same text prompt as shown at the bottom. There are two main benefits of generating images conditioned to the grasp. It first allows a set of images with aligned hand and object geometry and boundaries which can be useful for image and video editing. On the other hand, even though the Adobe Firefly generator has shown a significant advantage over the public Stable Diffusion in eliminating artifacts, we still observed many finger artifacts when generating the images without a grasp condition. Fortunately, with the generated grasp, including the object shape and the hand, as the condition, the image generator can produce significantly fewer artifacts, especially artifact fingers, which have been a notorious issue in image generation recently. We provided some zoomed-in examples at the bottom. Moreover, compared to previous works using hand shapes to guide image generation [264, 446], we could generate both the object and the hand. Therefore, we could use the whole scene of hand-object interaction as the condition and thus enable more controllable details in the generated image.

9.6.5 Failure cases

To provide a transparent evaluation of our proposed method, we still notice some failure cases on the unseen objects from OakInk and GRAB datasets as shown in Figure 9.12. There are in general three patterns of failures: (1) the hand and the object have no contact, making the grasp physically implausible; (2) the hand skin is twisted or the pose is not able to grasp the object in human common sense; (3) the mesh of object and hand has penetration and intersection. There can be some reasons that make these happen. First of all, we do not explicitly supervise the contact between the object surface and hand as this is a rare annotation on many data sources, and calculating it can be computationally expensive. On the other hand, some implausible grasps have pretty good visual quality as shown in the middle two columns in the figure. However, according to our life experience, we know that the grasp is not physically plausible to manipulate the object. This has gone

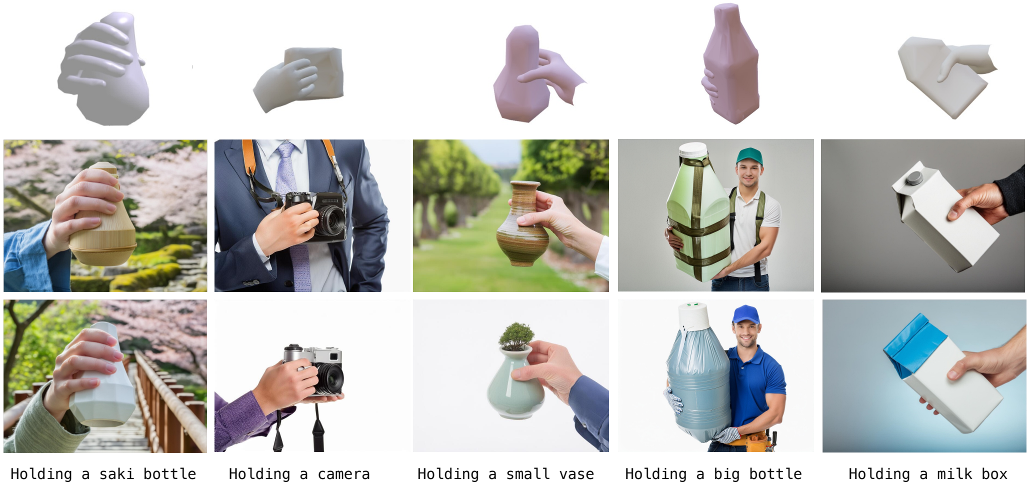


Figure 9.9: Samples of images generated by Adobe Firefly with synthesized hand grasp by our proposed method as the condition.

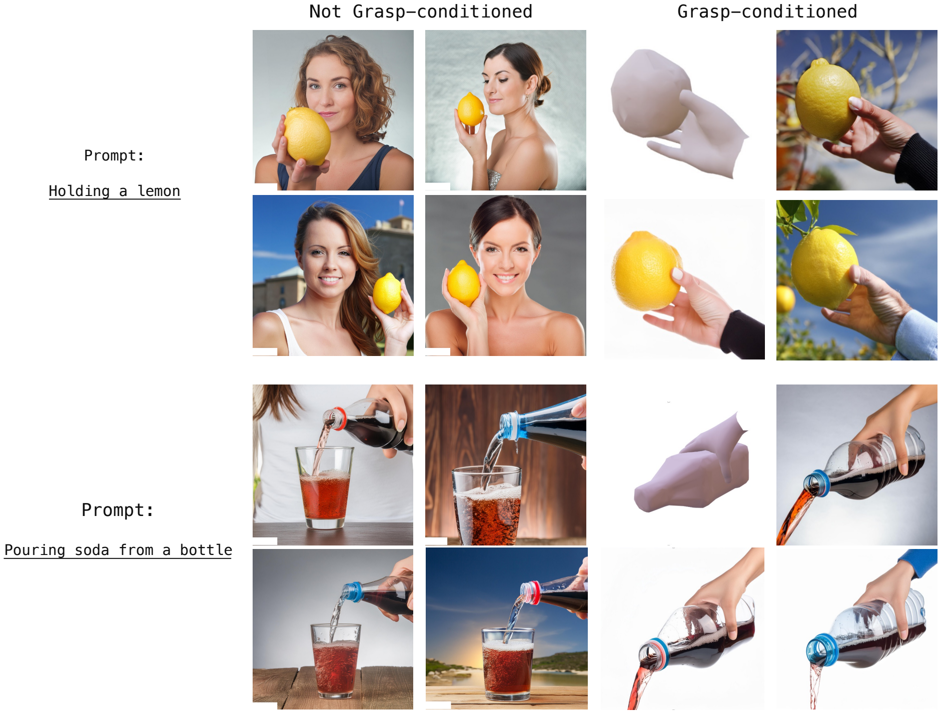


Figure 9.10: Using the same text-prompted image generation tool, we synthesize photo-realistic images with and without the grasp as the geometry condition. The grasp-conditioned image generation can have broad applications in image and video editing.

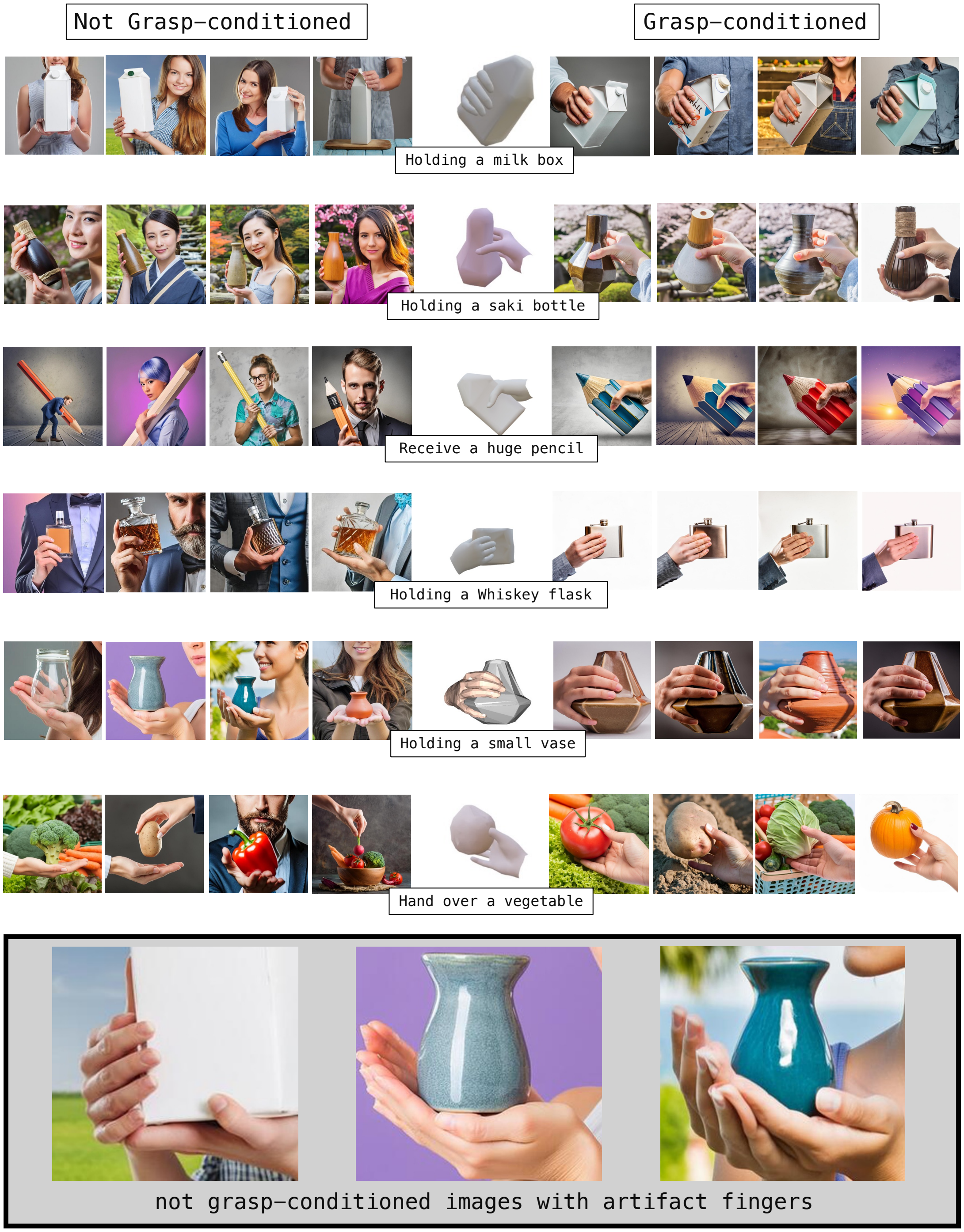


Figure 9.11: More examples of generating images of grasps with and without the condition from our generated grasps by the same image generation tool. Without a plausible grasp as the condition, there are more frequent unrealistic artifacts in the generated images, especially the number and pose of fingers. We provide some zoomed-in bad examples at the bottom.

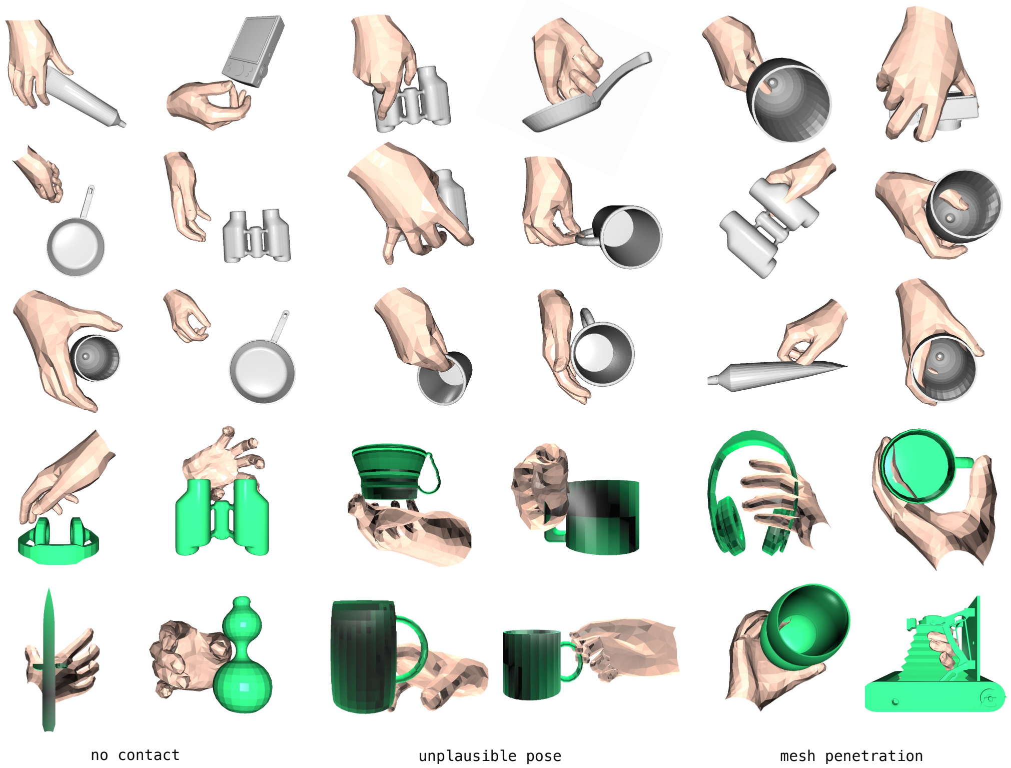


Figure 9.12: Bad samples of conditional grasp generation given objects from GRAB (gray) and OakInk-Shape (green).

beyond our scope in this work as we do not have any physics-aware supervision such as the physical demonstration in a physics simulator. Finally, we use the object point cloud to represent object shapes to allow grasp generation on universal objects without a template, this causes the potential penetration between object mesh and hand mesh as the object mesh is ambiguous and unknown to our method. Despite the failure cases, we note that they make only a very small portion of the generated samples (less than 10% by a rough estimation). We show them here for transparency and to help discussion about future works in this area.

Chapter 10

Whole-Body Motion Generation in Physics with Object Interaction

10.1 Introduction

Given an object mesh, we aim to control a simulated humanoid equipped with two dexterous hands to pick up the object and follow plausible trajectories, as shown in Fig.5.1. This capability could be broadly applied to creating human-object interactions for animation and AV/VR, with potential extensions to humanoid robotics [142]. However, controlling a simulated humanoid with dexterous hands for precise object manipulation poses significant challenges. The bipedal humanoid must maintain balance to enable detailed movements of the arms and fingers. Moreover, interacting with objects requires forming stable grasps that accommodate diverse object shapes. Combining these demands with the inherent difficulties of controlling a humanoid with a high degree of freedom (e.g.,153 DoF) significantly complicates the learning process.

These challenges have led previous methods of simulated grasping to employ a disembodied hand [71, 81, 303, 432] to grasp and transport. While this approach can generate physically plausible grasps, employing a floating hand compromises physical realism: the hands' root position and orientation are controlled by invisible forces, allowing it to remain nearly perfectly stable during grasping. Moreover, studying the hand in isolation does not accurately reflect its typical use, which is when it is attached to a mobile and flexible body. A naive approach to supporting hands is to use existing full-body motion imitators [233] to provide body control and train additional hand controllers for grasping. However, the presence of a body introduces instability, limits hand movement, and requires synchronizing the entire body to facilitate finger motion. State-of-the-art (SOTA) full-body imitators also have an average 30mm tracking error for the hands, which can cause the humanoid to miss objects. Due to the above challenges, previous work that studies full-body object manipulations often limits its scope to only one sequence of object interaction [396] and encounters difficulties in trajectory following [36], even when trained with highly specialized motion priors.

Another challenge of grasping is the diversity of the object shapes and trajectories. Each object may require a unique type of grasping, and scaling to thousands of different objects often requires training procedures such as generalist-specialist training [432] or curriculum [386, 487]. There is also infinite variability in potential object trajectories, and each trajectory may necessitate precise full-body coordination. Thus, prior work typically focuses on simple trajectories, such as vertical lifting [71, 432], or on learning a single, fixed, and pre-recorded trajectory per policy [81]. The flexibility with which humans manipulate objects to follow various trajectories while holding them remains unobtainable for current humanoids, even in simulations.

In this work, we introduce a full-body and dexterous humanoid controller capable of picking up and following diverse object trajectories using Reinforcement Learning (RL). Our proposed method, Omnigrasp, presents a scalable approach that generalizes to unseen object shapes and trajectories. Here, “Omni” refers to following any trajectory in all directions within a reasonable range and grasping diverse objects. Our key insight lies in using a pretrained universal dexterous motion representation as the action space. Directly training a policy on the joint actuation space using RL results in unnatural motion and leads to a severe exploration problem. Exploration noise in the torso can lead to a large deviation in the location of the arm and wrist as the noise propagates through the kinematic chain. This can lead to the humanoid quickly knocking the object away, which hinders training progress. Prior work has explored using a separate body and hand latent space trained using adversarial learning [36]. However, as the adversarial latent space can only cover small-scale and curated datasets, these methods do not achieve a high grasping success rate. The separation of hands and body motion prior also adds complexity to the system. We propose using a unified *universal and dexterous* humanoid motion latent space [232]. Learned from a large-scale human motion database [241], our motion representation provides a compact and efficient action space for RL exploration. We enhance the dexterity of this latent space by incorporating articulated hand motions into the existing body-only human motion dataset.

Equipped with a universal motion representation, our humanoid controller does not require any specialized interaction graph [396, 488] to learn human-object interactions. Our input to the policy consists only of object and trajectory-following information and is devoid of any grasp or reference body motion. For training, we use randomly generated trajectories and do not require paired full-body human-object motion data. We also identify the importance of pre-grasps [81] (the hand pose right before grasping) and utilize it in our reward design. The resulting policy can be directly applied to transport new objects without additional processing and achieve a SOTA success rate on following object trajectories captured by Motion Capture (MoCap).

To summarize, our contributions are: (1) we design a dexterous and universal humanoid motion representation that significantly increases sample efficiency and enables learning to grasp with simple yet effective state and reward designs; (2) we show that leveraging this motion representation, one can learn grasping policies with synthetic grasp poses and trajectories, without using any paired full-body and object motion data. (3) we demonstrate the feasibility of training a humanoid controller that can achieve a high success rate in grasping objects, following complex trajectories, scaling up to diverse training objects, and generalizing to unseen objects.

10.2 Related Works

Simulated Humanoid Control. Simulated humanoids can be used to create animations [136, 214, 284, 285, 286, 288, 411, 463, 488], estimate full-body pose from sensors [117, 154, 194, 229, 235, 407, 456, 457, 460], and transfer to real humanoid robots [104, 142, 143, 299, 300]. Since there are no ground truth data for joint actuation and physics simulators are often non-differentiable, model-based control [153], trajectory optimization [214, 419], and deep RL [65, 284] are used instead of supervised learning. Due to its flexibility and scalability, deep RL has been popular among efforts in simulated humanoids, where a policy/controller is trained via trial and error. Most of the previous work on humanoids does not consider articulated fingers, except for a few [14, 36, 214, 254]. A dexterous humanoid controller is essential for humanoids to perform meaningful tasks in simulation and in the real world.

Dexterous Manipulation. Dexterous manipulation is an essential topic in robotics [37, 38, 61, 62, 71, 72, 99, 215, 302, 386, 432, 464, 470, 471] and animation [5, 36, 198, 487]. This task usually involves pick-and-place [37, 38], lifting [386, 432, 470], articulating objects [471], and following predefined object trajectories [36, 39, 81]. Most of these efforts use a disembodied hand for grasping and employ non-physical virtual forces to control the hand.

Among them, D-Grasp [71] leverages the MANO [321] hand model for physically plausible grasp synthesis and 6DoF target reaching. UniDexGrasp [432] and its followup [386] use the Shadow Hand [2]. PGDM [81] trains a grasping policy for individual object trajectories and identifies pre-grasp initialization (initializing the hand in a pose right before grasping) as a crucial factor for successful grasping. For the works that consider both hands and body, PMP [14] and PhysHOI [396] train one policy for each task or object. Braun et al.[36] studies a similar setting to ours but relies on MoCap human-object interaction data and only uses one hand. Compared to prior work, Omnigrasp trains one policy to transport diverse objects, supports bimanual motion, and achieves a high success rate in lifting and object trajectory following.

Kinematic Grasp Synthesis. Synthesizing hand grasp can be widely applied in robotics and animation. A line of work [34, 49, 49, 96, 108, 216, 244, 263, 420, 444] focuses on reconstructing and predicting grasp from images or videos, while others [264, 446] study hand grasp generation to help image generation. Among them, Manipnet and CAMS [469] predict finger poses given a hand object trajectory. TOCH [494] and GeneOH [218] denoise dynamic hand pose predictions for object interactions. More research in this area focuses on generating static or sequential hand poses with a given object as the condition [166, 361, 447]. For synthesizing body and hand poses jointly, there are limited MoCap data available [360] due to difficulties in capturing synchronized full-body and object trajectories. Some generative methods [114, 203, 362, 363, 368, 416, 449] can create paired human-object interactions, but they require initialization from the ground truth [114, 362, 416], or only predict static full-body grasps [368]. In this work, we use GrabNet [361] trained on object shapes from OakInk [440] to generate hand poses as reward guidance for our policy training.

Humanoid Motion Representation. Due to the high DoF of a humanoid and the sample inefficiency of RL training, the search space within which the policy operates during trial and error is crucial. A more structured action space such as motion primitives [129, 133, 253, 307] or motion latent space [288, 369] can significantly increase sample efficiency since the policy can sample coherent motion instead of relying on random "jittering" noise. This is especially important for humanoids with dexterous hands, where the torso motion can drastically affect the hand movement and lead to the humanoid knocking the object away. Thus, prior work in this space utilizes part-based motion priors [14, 36] trained on specialized datasets. While effective in the single task setting where the humanoid only needs to perform actions close to the ones in the specialized datasets, these motion priors can hardly scale to more free-formed motion, such as following randomly generated object trajectories. We extend the recently proposed universal humanoid motion representation, PULSE [232], to the dexterous humanoid setting and demonstrate that a 48-dimensional, full-body-and-hand motion latent space can be used to pick up and follow randomly generated trajectories.

10.3 Preliminaries

We define the human pose as $\boldsymbol{q}_t := (\boldsymbol{\theta}_t, \boldsymbol{p}_t)$, consisting of 3D joint rotation $\boldsymbol{\theta}_t \in \mathbb{R}^{J \times 6}$ and position $\boldsymbol{p}_t \in \mathbb{R}^{J \times 3}$ of all J links on the humanoid (hands and body), using the 6 degree-of-freedom (DOF) rotation representation [505]. To define velocities $\dot{\boldsymbol{q}}_{1:T}$, we have $\dot{\boldsymbol{q}}_t := (\boldsymbol{\omega}_t, \boldsymbol{v}_t)$ as angular $\boldsymbol{\omega}_t \in \mathbb{R}^{J \times 3}$ and linear velocities $\boldsymbol{v}_t \in \mathbb{R}^{J \times 3}$. For objects, we define their 3D trajectories $\boldsymbol{q}_t^{\text{obj}}$ using object position $\boldsymbol{p}_t^{\text{obj}}$, orientation $\boldsymbol{\theta}_t^{\text{obj}}$, linear velocity $\boldsymbol{v}_t^{\text{obj}}$, and angular velocity $\boldsymbol{\omega}_t^{\text{obj}}$. As a notation convention, we use $\widehat{\cdot}$ to denote the kinematic quantities from Motion Capture (MoCap) or trajectory generator and normal symbols without accents for values from the physics simulation. $\hat{\boldsymbol{O}}$ refers to a dataset of diverse object meshes.

Goal-conditioned Reinforcement Learning for Humanoid Control. We define the object grasping and transporting task using the general framework of goal-conditioned RL. Namely, a goal-conditioned policy π is trained to control a simulated humanoid to grasp an object and follow object trajectories $\hat{\boldsymbol{q}}_{1:T}^{\text{obj}}$ using dexterous hands. The learning task is formulated as a Markov Decision Process (MDP) defined by the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \boldsymbol{\mathcal{R}}, \gamma \rangle$ of states, actions, transition dynamics, reward function, and discount factor. The simulation determines the state $\boldsymbol{s}_t \in \mathcal{S}$ and transition dynamics $\mathcal{T}$, where a policy computes the action $\boldsymbol{a}_t$. The state $\boldsymbol{s}_t$ contains the proprioception $\boldsymbol{s}_t^{\text{p}}$ and the goal state $\boldsymbol{s}_t^{\text{g}}$. Proprioception is defined as $\boldsymbol{s}_t^{\text{p}} := (\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t, \boldsymbol{c}_t)$, which contains the 3D body pose $\boldsymbol{q}_t$, velocity $\dot{\boldsymbol{q}}_t$, and contact forces $\boldsymbol{c}_t$ on the hand. The goal state $\boldsymbol{s}_t^{\text{g}}$ is defined based on the states of the objects. When computing the states $\boldsymbol{s}_t^{\text{g}}$ and $\boldsymbol{s}_t^{\text{p}}$, all values are normalized with respect to the humanoid heading (yaw). Based on proprioception $\boldsymbol{s}_t^{\text{p}}$ and the goal state $\boldsymbol{s}_t^{\text{g}}$, we define a reward $r_t = \boldsymbol{\mathcal{R}}(\boldsymbol{s}_t^{\text{p}}, \boldsymbol{s}_t^{\text{g}})$ for training the policy. We use proximal policy optimization (PPO) [330] to maximize discounted reward $\mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} r_t\right]$. Our humanoid follows the kinematic structure of SMPL-X [276] using the mean shape. It has 52 joints, of which 51 are actuated. 21 joints are body joints, and the remaining 30 joints are for two hands. All joints have 3 DoF, resulting in an actuation space of $\boldsymbol{a}_t \in \mathbb{R}^{51 \times 3}$. Each degree of freedom is actuated by a proportional derivative (PD) controller, and the action $\boldsymbol{a}_t$ specifies the PD target.

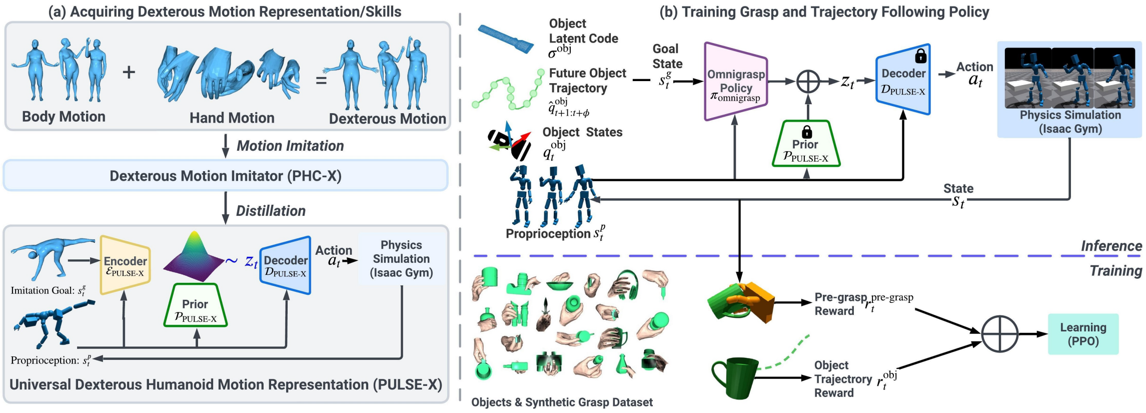

(a) Acquiring Dexterous Motion Representation/Skills
Body Motion
+
Hand Motion
=
Dexterous Motion
Motion Imitation
Dexterous Motion Imitator (PHC-X)
Distillation
Encoder $\mathcal{E}_{\text{PULSE-X}}$
Imitation Goal: s_t^g
$\sim z_t$
Prior $\mathcal{P}_{\text{PULSE-X}}$
Decoder $\mathcal{D}_{\text{PULSE-X}}$
Action a_t
Physics Simulation (Isaac Gym)
Proprioception: s_t^p
Universal Dexterous Humanoid Motion Representation (PULSE-X)
(b) Training Grasp and Trajectory Following Policy
Object Latent Code σ^{obj}
Future Object Trajectory $\hat{q}^{\text{obj}}_{t+1:t+\phi}$
Goal State s_t^g
Omnigrasp Policy $\pi_{\text{omnigrasp}}$
z_t
Decoder $\mathcal{D}_{\text{PULSE-X}}$
Action a_t
Physics Simulation (Isaac Gym)
Object States q_t^{obj}
Prior $\mathcal{P}_{\text{PULSE-X}}$
State s_t
Proprioception s_t^p
Inference
Training
Objects & Synthetic Grasp Dataset
Pre-grasp Reward $r_t^{\text{pre-grasp}}$
Object Trajectroy Reward r_t^{obj}
Learning (PPO)

Figure 10.1: *Omnigrasp* is trained in two stages. (a) A universal and dexterous humanoid motion representation is trained via distillation. (b) Pre-grasp guided grasping training using a pretrained motion representation.

10.4 Omnigrasp: Grasping Diverse Objects and Follow Object Trajectories

To tackle the challenging problem of picking up objects and following diverse trajectories, we first acquire a universal dexterous humanoid motion representation in Sec.10.4.1. Using this motion representation, we design a hierarchical RL framework (Sec. 10.4.2) for grasping objects using simple[1] state and reward designs guided by pre-grasps. Our architecture is visualized in Figure 10.1.

10.4.1 PULSE-X: Physics-based Universal Dexterous Humanoid Motion Representation

We introduce PULSE-X that extends PULSE [232] to the dexterous humanoid by adding articulated fingers. We first train a humanoid motion imitator [233] that can scale to a large-scale human motion dataset with finger motion. Then, we distill the motion imitator into a motion representation using a variational information bottleneck (similar to a VAE [179]).

Data Augmentation. Since full-body motion datasets that contain finger motion are rare (e.g.,, 91% of the AMASS sequences do not have finger motion), we first augment existing

[1] Here, the "simple reward" refers to not needing paired full-body-and-hand MoCap data when computing the reward, which increases complexity.

sequences with articulated finger motion and construct a dexterous full-body motion dataset. Similarly to the process in BEDLAM [31], we randomly pair full-body motion from AMASS [241] with hand motion sampled from GRAB [360] and Re:InterHand [261] to create a dexterous AMASS dataset. Intuitively, training on this dataset increases the dexterity of the imitator and the subsequent motion representation.

PULSE-X: Humanoid Motion Imitation with Articulated Fingers. Inspired by PHC [233], we design PULSE-X $\pi_{\text{PHC-X}}$ for humanoid motion imitation with articulated fingers. For the finger joints, *we treat them similarly as the rest of the body (e.g.,toe or wrist)* and find this formulation sufficient to acquire the dexterity needed for grasping. Formally, the goal state for training $\pi_{\text{PHC-X}}$ with RL is $\boldsymbol{s}_t^{\text{g-mimic}} := (\hat{\boldsymbol{\theta}}_{t+1} \ominus \boldsymbol{\theta}_t, \hat{\boldsymbol{p}}_{t+1} - \boldsymbol{p}_t, \hat{\boldsymbol{v}}_{t+1} - \boldsymbol{v}_t, \hat{\boldsymbol{\omega}}_{t+1} - \boldsymbol{\omega}_t, \hat{\boldsymbol{\theta}}_{t+1}, \hat{\boldsymbol{p}}_{t+1})$, which contains the difference between proprioception and one frame reference pose $\hat{\boldsymbol{q}}_{t+1}$.

Learning Motion Representation via Online Distillation. In PULSE [232], an encoder $\boldsymbol{\mathcal{E}}_{\text{PULSE-X}}$, decoder $\boldsymbol{\mathcal{D}}_{\text{PULSE-X}}$, and prior $\boldsymbol{\mathcal{P}}_{\text{PULSE-X}}$ are learned to compress motor skills into a latent representation. For downstream tasks, the frozen decoder and prior will translate the latent code to joint actuation. Formally, the encoder $\boldsymbol{\mathcal{E}}_{\text{PULSE-X}}(\boldsymbol{z}_t | \boldsymbol{s}_t^{\text{p}}, \boldsymbol{s}_t^{\text{g-mimic}})$ computes the latent code distribution based on current input states. The decoder $\boldsymbol{\mathcal{D}}_{\text{PULSE-X}}(\boldsymbol{a}_t | \boldsymbol{s}_t^{\text{p}}, \boldsymbol{z}_t)$ produces action (joint actuation) based on the latent code $\boldsymbol{z}_t$. The prior $\boldsymbol{\mathcal{P}}_{\text{PULSE-X}}(\boldsymbol{z}_t | \boldsymbol{s}_t^{\text{p}})$ defines a Gaussian distribution based on proprioception and replaces the unit Gaussian distribution used in VAEs [179]. The prior increases the expressiveness of the latent space and guides downstream task learning by forming a residual action space (see Sec.10.4.2). We model the encoder and prior distribution as diagonal Gaussian:

$$\boldsymbol{\mathcal{E}}_{\text{PULSE-X}}(\boldsymbol{z}_t | \boldsymbol{s}_t^{\text{p}}, \boldsymbol{s}_t^{\text{g-mimic}}) = \mathcal{N}(\boldsymbol{z}_t | \boldsymbol{\mu}_t^e, \boldsymbol{\sigma}_t^e), \boldsymbol{\mathcal{P}}_{\text{PULSE-X}}(\boldsymbol{z}_t | \boldsymbol{s}_t^{\text{p}}) = \mathcal{N}(\boldsymbol{z}_t | \boldsymbol{\mu}_t^p, \boldsymbol{\sigma}_t^p). \tag{10.1}$$

To train the models, we use online distillation similar to DAgger [323] by rolling out the encoder-decoder in simulation and querying $\pi_{\text{PHC-X}}$ for action labels $\boldsymbol{a}_t^{\text{PHC-X}}$. For more information and evaluation of PULSE-X and PULSE-X, please refer to the Appendix 10.7.1.

10.4.2 Pre-grasp Guided Object Manipulation

Using hierarchical RL and PULSE-X's trained decoder $\boldsymbol{\mathcal{D}}_{\text{PULSE-X}}$ and prior $\boldsymbol{\mathcal{P}}_{\text{PULSE-X}}$, the action space for our object manipulation policy becomes the latent motion representation $\boldsymbol{z}_t$. Since the action space serves as a strong human-like motion prior, we can use simple state and reward design and do not require any paired object and human motion to learn grasping policies. We use only hand pose before grasping (pregraps), either from a generative method or MoCap, to train our policy.

State. To provide the task policy $\pi_{\text{Omnigrasp}}$ with information about the object and the desired object trajectory, we define the goal state as

$$\boldsymbol{s}_t^{\text{g}} := (\hat{\boldsymbol{p}}_{t+1:t+\phi}^{\text{obj}} - \boldsymbol{p}_t^{\text{obj}}, \hat{\boldsymbol{\theta}}_{t+1:t+\phi}^{\text{obj}} \ominus \boldsymbol{\theta}_t^{\text{obj}}, \hat{\boldsymbol{v}}_{t+1:t+\phi}^{\text{obj}} - \boldsymbol{v}_t^{\text{obj}}, \hat{\boldsymbol{\omega}}_{t+1:t+\phi}^{\text{obj}} - \boldsymbol{\omega}_t^{\text{obj}}, \boldsymbol{p}_t^{\text{obj}}, \boldsymbol{\theta}_t^{\text{obj}}, \boldsymbol{\sigma}^{\text{obj}}, \boldsymbol{p}_t^{\text{obj}} - \boldsymbol{p}_t^{\text{hand}}), \tag{10.2}$$

which contains the reference object pose and the difference between the reference object trajectory for the next ϕ frames and the current object state. $\boldsymbol{\sigma}^{\text{obj}} \in \mathcal{R}^{512}$ is the object shape latent code computed using the canonical object pose and Basis Point Set (BPS) [295]. $\boldsymbol{p}_t^{\text{obj}} - \boldsymbol{p}_t^{\text{hand}}$ is the difference between the current object position and each hand joint position. All values are normalized with respect to the humanoid heading. Notice that the state $\boldsymbol{s}_t^{\text{g}}$ does not contain body pose, grasp, or phase variables [36], which makes our method applicable to unseen objects and reference trajectories at test time.

Action. Similar to downstream task policies in PULSE, we form the action space of $\pi_{\text{Omnigrasp}}$ as the residual action with respect to prior's mean $\boldsymbol{\mu}_t^p$ and compute the PD target $\boldsymbol{a}_t$:

$$\boldsymbol{a}_t = \boldsymbol{\mathcal{D}}_{\text{PULSE-X}}(\pi_{\text{PHC}}(\boldsymbol{z}_t^{\text{omnigrasp}} | \boldsymbol{s}_t^{\text{p}}, \boldsymbol{s}_t^{\text{g}}) + \boldsymbol{\mu}_t^p), \tag{10.3}$$

where $\boldsymbol{\mu}_t^p$ is computed by the prior $\boldsymbol{\mathcal{P}}_{\text{PULSE-X}}(\boldsymbol{z}_t | \boldsymbol{s}_t^{\text{p}})$. The policy $\pi_{\text{Omnigrasp}}$ computes $\boldsymbol{z}_t^{\text{omnigrasp}} \in \mathcal{R}^{48}$ instead of the target $\boldsymbol{a}_t \in \mathcal{R}^{51\times 3}$ directly, and leverages the latent motion representation of PULSE-X to produce human-like motion.

Reward. While our policy does not take any grasp guidance or reference body trajectory *as input*, we utilize pre-grasp guidance in the *reward*. We refer to pre-grasp $\hat{\boldsymbol{q}}^{\text{pre-grasp}} := (\hat{\boldsymbol{p}}^{\text{pre-grasp}}, \hat{\boldsymbol{\theta}}^{\text{pre-grasp}})$ as a single frame of hand pose consisting of hand translation $\hat{\boldsymbol{p}}^{\text{pre-grasp}}$ and rotation $\hat{\boldsymbol{\theta}}^{\text{pre-grasp}}$. PGDM [81] shows that initializing a floating hand to pre-grasps can help the policy better reach objects and initiate manipulation. As we do not initialize the humanoid with the pre-grasp pose as in PGDM, we design a stepwise pre-grasp reward:

$$\boldsymbol{r}_t^{\text{omnigrasp}} = \begin{cases} r_t^{\text{approach}}, & \|\hat{\boldsymbol{p}}^{\text{pre-grasp}} - \boldsymbol{p}_t^{\text{hand}}\|_2 > 0.2 \text{ and } t < \lambda \\ r_t^{\text{pre-grasp}}, & \|\hat{\boldsymbol{p}}^{\text{pre-grasp}} - \boldsymbol{p}_t^{\text{hand}}\|_2 \leq 0.2 \text{ and } t < \lambda \\ r_t^{\text{obj}}, & t \geq \lambda, \end{cases} \tag{10.4}$$

based on time and the distance between the object and hands. Here, $\lambda = 1.5s$ indicates the frame in which grasping should occur, and $\boldsymbol{p}_t^{\text{hand}}$ indicates the hand position. When the object is far away from the hands ($\|\hat{\boldsymbol{p}}^{\text{pre-grasp}} - \boldsymbol{p}_t^{\text{hand}}\|_2 > 0.2$), we use an approach reward r_t^{approach} similar to a point-goal [233, 413] reward $r_t^{\text{approach}} = \|\hat{\boldsymbol{p}}^{\text{pre-grasp}} - \boldsymbol{p}_t^{\text{hand}}\|_2 - \|\hat{\boldsymbol{p}}^{\text{pre-grasp}} - \boldsymbol{p}_{t-1}^{\text{hand}}\|_2,$, where the policy is encouraged to get close to the pre-grasp. After the hands are close enough ($\leq$ 0.2m), we use a more precise hand imitation reward: $r_t^{\text{pre-grasp}} = w_{\text{hp}} e^{-100\|\hat{\boldsymbol{p}}^{\text{pre-grasp}} - \boldsymbol{p}_t^{\text{hand}}\|_2 \times \mathbb{1}\{\|\hat{\boldsymbol{p}}^{\text{pre-grasp}} - \hat{\boldsymbol{p}}_t^{\text{obj}}\|_2 \leq 0.2\}} + w_{\text{hr}} e^{-100\|\hat{\boldsymbol{\theta}}^{\text{pre-grasp}} - \boldsymbol{\theta}_t^{\text{hand}}\|_2}$, to encourage the hands to be close to pre-grasps. For grasps that involve only one hand, we use an indicator variable $\mathbb{1}\{\|\hat{\boldsymbol{p}}^{\text{pre-grasp}} - \hat{\boldsymbol{p}}_t^{\text{obj}}\|_2 \leq 0.2\}$ to filter out hands that are too far away from the object. After timestep λ, we use only the object trajectory following reward:

$$r_t^{\text{obj}} = (w_{\text{op}} e^{-100\|\hat{\boldsymbol{p}}_t^{\text{obj}} - \boldsymbol{p}_t^{\text{obj}}\|_2} + w_{\text{or}} e^{-100\|\hat{\boldsymbol{\theta}}_t^{\text{obj}} - \boldsymbol{\theta}_t^{\text{obj}}\|_2} + w_{\text{ov}} e^{-5\|\hat{\boldsymbol{v}}_t^{\text{obj}} - \boldsymbol{v}_t^{\text{obj}}\|_2} + w_{\text{oav}} e^{-5\|\hat{\boldsymbol{\omega}}_t^{\text{obj}} - \boldsymbol{\omega}_t^{\text{obj}}\|_2}) \cdot \mathbb{1}\{\text{C}\} + \mathbb{1}\{\text{C}\} \cdot w_{\text{c}}, \tag{10.5}$$

r_t^{obj} computes the difference between the current and reference object pose, which is filtered by an indicator variable $\mathbb{1}\{\text{C}\}$ that is set to true if the object is in contact with the humanoid hands. The reward $\mathbb{1}\{\text{C}\} \cdot w_{\text{c}}$ encourages the humanoid's hand to have contact with the object. Hyperparameters can be found in Appendix 10.7.2.

Object 3D Trajectory Generator. As there is a limited number of ground-truth object trajectories [81], either collected from MoCap or animators, we design a 3D object trajectory generator that can create trajectories with varying speed and direction. Using the trajectory generator, our policy can be trained without any ground-truth object trajectories. This strategy provides better coverage of potential object trajectories, and the resulting policy achieves higher success in following unseen trajectories (see Table 10.1). Specifically, we extend the 2D trajectory generator used in PACER [312, 392] to 3D, and create our trajectory generator $\mathcal{T}^{\text{3D}}(\boldsymbol{q}_0^{\text{obj}}) = \hat{\boldsymbol{q}}_{1:T}^{\text{obj}}$. Given initial object pose $\boldsymbol{q}_0^{\text{obj}}$, $\mathcal{T}^{\text{3D}}$ can generate a sequence of plausible reference object motion $\hat{\boldsymbol{q}}_{1:T}^{\text{obj}}$. We limit the z-direction trajectory to between 0.03m and 1.8m and leave the xy direction unbounded. For more information and sampled trajectories, please refer to Appendix 10.7.2.

Training. Our training process is depicted in Algo 4. One of the main sources of performance improvement for motion imitation is hard-negative mining [229, 233], where the policy is evaluated regularly to find the failure sequences to train on. Thus, instead of using object curriculum [386, 432, 487], we use a simple hard-negative mining process to pick hard objects $\hat{\boldsymbol{O}}_{\text{hard}}$ to train on. Specifically, let s_j be the number of failed lifts for object j over all previous runs. The probability of choosing object j among all objects is $P(j) = \frac{s_j}{\sum_i^J s_i}$.

Object and Humanoid Initial State Randomization. Since objects can have diverse initial positions and orientations with respect to the humanoid, it is crucial to have the policy

```
Function TrainOmnigrasp($\mathcal{D}_{PULSE\text{-}X}, \mathcal{P}_{PULSE\text{-}X}, \pi_{PHC}, \hat{O}, \mathcal{T}^{3D}$):
    Input: Pretrained PULSE-X's decoder $\mathcal{D}_{\text{PULSE-X}}$ and prior $\mathcal{P}_{\text{PULSE-X}}$, Object mesh dataset $\hat{O}$, 3D trajectory Generator $\mathcal{T}^{\text{3D}}$
    while not converged do
        $M \leftarrow \emptyset$ initialize sampling memory
        while $M$ not full do
            $q_0^{\text{obj}}, \hat{p}^{\text{pre-grasp}}, s_t^{\text{p}} \sim$ randomly sample initial object state, pre-grasp, and humanoid state
            $\hat{q}_{1:T}^{\text{obj}} \sim$ sample reference object trajectory using $\mathcal{T}^{\text{3D}}$
            for $t \leftarrow 1 ... T$ do
                $z_t^{\text{omnigrasp}} \sim \pi_{\text{PHC}}(z_t^{\text{omnigrasp}} | s_t^{\text{p}}, s_t^{\text{g}})$        // use pretrained latent space as action space
                $\mu_t^p, \sigma_t^p \leftarrow \mathcal{P}_{\text{PULSE-X}}(z_t | s_t^{\text{p}})$        // compute prior latent code
                $a_t \leftarrow \mathcal{D}_{\text{PULSE-X}}(a_t | s_t^{\text{p}}, z_t^{\text{omnigrasp}} + \mu_t^p)$        // decode action using pretrained decoder
                $s_{t+1} \leftarrow \mathcal{T}(s_{t+1} | s_t, a_t)$        // simulation
                $r_t \leftarrow \mathcal{R}(s_t^{\text{p}}, s_t^{\text{g}})$        // compute reward
                store $(s_t, z_t^{\text{omnigrasp}}, r_t, s_{t+1})$ into memory $M$
        $\pi_{\text{PHC}} \leftarrow$ PPO update using experiences collected in $M$
        $\hat{O}_{\text{hard}} \leftarrow$ Eval and pick hard object subset to train on.
    return $\pi_{\text{PHC}}$
```

algorithm 4: Learn *Omnigrasp*

exposed to diverse initial object states. Given the object dataset $\hat{O}$ and the provided initial states (either from MoCap or by dropping the object in simulation) q_0^{obj}, we perturb q_0^{obj} by adding randomly sampled yaw-direction rotation and adjusting the position component q_0^{obj}. We do not change the pitch and yaw of the object's initial pose as some poses are invalid in simulation. For the humanoid, we use the initial state from the dataset if provided (e.g.,GRAB dataset [360]), and a standing T-pose if there is no paired data.

Inference. During inference, the object latent code p_t^{obj}, a random object starting pose q_0^{obj}, and desired object trajectory $\hat{q}_{1:T}^{\text{obj}}$ is all that is required, without any dependency on pre-grasps or paired kinematic human pose.

10.5 Experiments

Datasets. We use the GRAB [360], OakInk [440], and OMOMO [198] to study grasping small and large objects. The GRAB dataset contains 1.3k paired full-body motion and object trajectories of 50 objects (we remove the doorknob as it is not movable). Since the GRAB dataset provides reference body and object motion, we use them to extract initial humanoid positions and pre-grasps. We follow prior art [36] in constructing cross-object (45 for training and 5 for testing) and cross-subject (9 subjects for training and 1 for testing) train-test sets. On GRAB, we evaluate on following MoCap object trajectories using the mean body shape humanoid. The OakInk dataset contains 1700 diverse objects of 32 categories with real-world scanned and generated object meshes. We split them into 1330 objects for training, 185 for validation, and 185 for testing. Train-test splits are conducted

Table 10.1: Quantitative results on object grasp and trajectory following on the GRAB dataset.

		GRAB-Goal-Test (Cross-Object, 140 sequences, 5 unseen objects)							GRAB-IMoS-Test (Cross-Subject, 92 sequences, 44 objects)						
Method	Traj	$\text{Succ}_{\text{grasp}} \uparrow$	$\text{Succ}_{\text{traj}} \uparrow$	TTR ↑	$E_{\text{pos}} \downarrow$	$E_{\text{rot}} \downarrow$	$\text{E}_{\text{acc}} \downarrow$	$\text{E}_{\text{vel}} \downarrow$	$\text{Succ}_{\text{grasp}} \uparrow$	$\text{Succ}_{\text{traj}} \uparrow$	TTR ↑	$E_{\text{pos}} \downarrow$	$E_{\text{rot}} \downarrow$	$\text{E}_{\text{acc}} \downarrow$	$\text{E}_{\text{vel}} \downarrow$
PPO-10B	Gen	98.4%	55.9%	97.5%	36.4	**0.4**	21.0	14.5	96.8%	53.2%	97.9%	35.6	**0.5**	19.6	13.9
PHC [233]	MoCap	3.6%	11.4%	81.1%	66.3	0.8	1.5	3.8	0%	3.3%	97.4%	56.5	0.3	1.4	2.9
AMP [286]	Gen	90.4%	46.6%	94.0 %	40.7	0.6	5.3	5.3	95.8 %	49.2%	96.5%	34.9	0.5	6.2	6.0
Braun et al.[36]	MoCap	79%	-	85%	-	-	-	-	64%	-	65%	-	-	-	-
Omnigrasp	MoCap	94.6%	84.8%	98.7%	**28.0**	**0.5**	**4.2**	**4.3**	95.8%	85.4%	99.8%	**27.5**	**0.6**	**5.0**	**5.0**
Omnigrasp	Gen	**100%**	**94.1%**	**99.6%**	30.2	0.93	5.4	4.7	**98.9%**	**90.5%**	**99.8%**	27.9	0.97	6.3	5.4

within categories, with train and test splits containing objects from all categories. Since no paired MoCap human motion or grasps exists for the OakInk dataset, we use an off-the-shelf grasp generator [440] to create pre-grasps. The OMOMO dataset contains 15 large objects (table lamps, monitors, *etc.*) with reconstructed mesh, and we pick 7 of them that have cleaner meshes. Due to the limited number of objects from OMOMO, we only test lifting on the objects used for training to verify that our pipeline can learn to move larger objects. On OMOMO and OakInk, we study vertical lifting (30cm) and holding (3s) as the trajectory for quantitative results.

Implementation Details. Simulation is conducted in Isaac Gym [242], where the policy is run at 30 Hz and the simulation at 60 Hz. For PULSE-X and PULSE-X, each policy is a 6-layer MLP. For the grasping task, we employ a GRU [67] based recurrent policy and use a GRU with a latent dimension of 512, followed by a 3-layer MLP. We train *Omnigrasp* for three days collecting around 10^9 samples on a Nvidia A100 GPU. PULSE-X and PULSE-X are trained once and frozen, which takes around 1.5 weeks and 3 days. Object density is 1000 kg/m^3. The static and dynamic friction coefficients of the object and humanoid fingers are set to 1. For reference object trajectory, we use $\phi = 20$ future frames sampled at 15Hz. For more details, please refer to Appendix 10.7.2.

Metrics. For the object trajectory following, we report the position error E_{pos} (mm), rotation error E_{rot} (radian), and physics-based metrics such as acceleration error E_{acc} (mm/frame^2) and velocity error E_{vel} (mm/frame). Following prior art in full-body simulated humanoid grasping [36], we report the grasp success rate $\text{Succ}_{\text{grasp}}$ and Trajectory Targets Reached (TTR). The grasp success rate $\text{Succ}_{\text{grasp}}$ deems a grasp successful when the object is held for at least 0.5s in the physics simulation without dropping. TTR measures the ratio of the target position (¡ 12cm away from the target position) reached over all the time steps in the trajectory and is only measured on successful trajectories. To measure the complete trajectory success rate, we also report $\text{Succ}_{\text{traj}}$, where a trajectory following is unsuccessful if, at any point in time, the object is ¿ 25cm away from the reference.

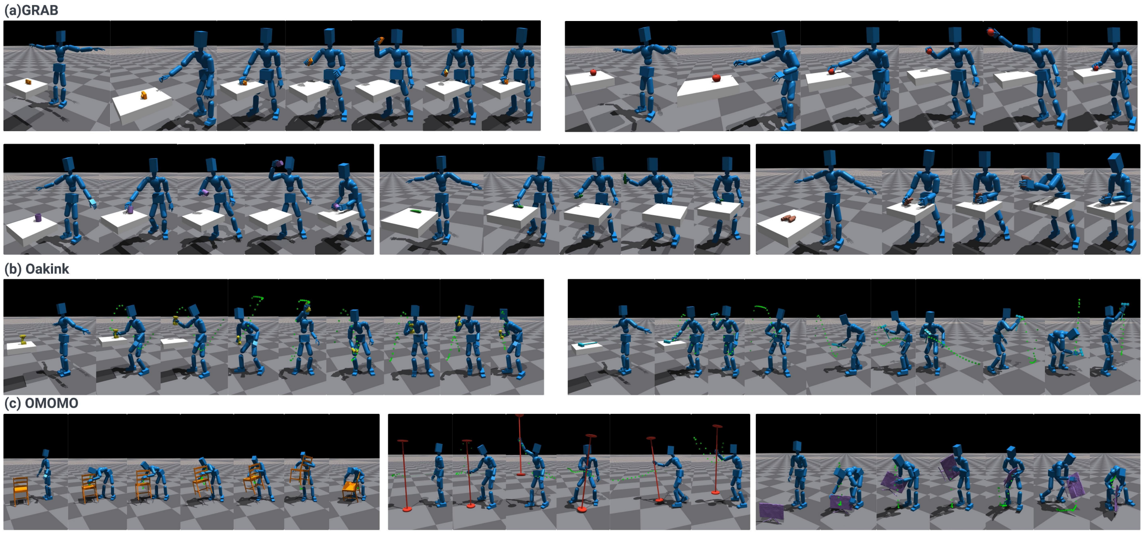


Figure 10.2: Qualitative results. Unseen objects are tested for GRAB and OakInk. Green dots: reference trajectories. Best seen in videos on our supplement site.

10.5.1 Grasping and Trajectory Following

As motion is best seen in videos, please refer to supplement site for extended evaluation on trajectory following, unseen objects, and robustness. Unless otherwise specified, all policies are trained on their respective dataset training split, and we conduct cross-dataset experiments on GRAB and OakInk. All experiments are run 10 times and averaged as the simulator yields slightly different results for each run due to e.g.,floating-point error. As full-body simulated humanoid grasping is a relatively new task with a limited number of baselines, we use Braun et [36] as our main comparison. We also implement AMP [286] and PHC [233] as baselines. We train AMP with a similar state and reward design (without using PULSE-X's latent space) and a task and discriminator reward weighting of 0.5 and 0.5. PHC refers to using an imitator for grasping, where we directly feed ground-truth kinematic body and finger motion to a pretrained imitator to grasp objects. Since PHC and PULSE-X require pre-training, we also include PPO-10B, which is trained using RL without PULSE-X for a month (∼10 billion samples).

GRAB Dataset (50 objects). Since Braun et al.do not use randomly generated trajectories, we train *Omnigrasp* using two different settings for a fair comparison: one trained with MoCap object trajectories only, and one trained using synthetic trajectories only. From Table 10.1, we can see that our method outperforms prior SOTA and baselines on all metrics, especially on success rate and trajectory following. Since all methods are simulation-based, we omit penetration/foot sliding metrics and report the precise trajectory tracking errors

Table 10.2: Quantitative results on the OMOMO dataset.

OMOMO (7 objects)						
$\text{Succ}_{\text{grasp}} \uparrow$	$\text{Succ}_{\text{traj}} \uparrow$	TTR ↑	$E_{\text{pos}} \downarrow$	$E_{\text{rot}} \downarrow$	$\text{E}_{\text{acc}} \downarrow$	$\text{E}_{\text{vel}} \downarrow$
7/7	7/7	100%	22.8	0.2	3.1	3.3

instead. Training directly using PPO without PULSE-X leads to a performance that significantly lags behind Omnigrasp, even though it has used similar aggregate samples (counting PHC-X and PULSE-X training). Compared to Braun et al., *Omnigrasp* achieves a high success rate on both object lifting and trajectory following. Directly using the motion imitator, PHC, yields a low success rate even when the ground-truth kinematic pose is provided, showing that the imitator's error (on average 30mm) is too large to overcome for precise object grasping. The body shape mismatch between MoCap and our simulated humanoid also contributes to this error. AMP leads to a low trajectory success rate, showing the importance of using a motion prior in the *actions space*. *Omnigrasp* can track the MoCap trajectory precisely with an average error of 28mm. Comparing training on MoCap trajectories and randomly generated ones, we can see that training on generated trajectories achieves better performance on success rate and position error, though worse on rotation error. This is due to our 3D trajectory generator offering good coverage on physically plausible 3D trajectories, but there is a gap between the randomly generated rotations and MoCap object rotation. This can be improved by introducing more rotation variation on the trajectory generator. The gap between trajectory $\text{Succ}_{\text{traj}}$ and grasp success $\text{Succ}_{\text{grasp}}$ shows that following the full trajectory is a much harder task than just grasping, and the object can be dropped during trajectory following. Qualitative results can be found in Fig. 10.2.

OakInk Dataset (1700 objects). On the OakInk dataset, we scale our grasping policy to ¿1000 objects and test our generalization to unseen objects. We also conduct cross-dataset experiments, where we train on the GRAB dataset and test on the OakInk dataset. Results are shown in Table 10.3. We can see that 1272 out of the 1330 objects are trained to be picked up, and the whole lifting process also has a high success rate. We observe similar results on the test split. Upon inspection, the failed objects are usually either too large or too small for the humanoid to establish a grasp. The large number of objects also places a strain on the hard-negative mining process. The policy trained on both GRAB and OakInk shows the highest success rate, as on GRAB, there are bi-manual pre-grasps, and the policy learned to use both hands.

Table 10.3: Quantitative results on OakInk with our method. We also test *Omnigrasp* cross-dataset, where a policy trained on GRAB is tested on the OakInk dataset.

	OakInk-Train (1330 objects)							OakInk-Test (185 objects)						
Training Data	Succ_grasp ↑	Succ_traj ↑	TTR ↑	E_pos ↓	E_rot ↓	E_acc ↓	E_vel ↓	Succ_grasp ↑	Succ_traj ↑	TTR ↑	E_pos ↓	E_rot ↓	E_acc ↓	E_vel ↓
OakInk	93.7%	86.2%	**100%**	21.3	**0.4**	7.7	6.0	**94.3%**	87.5%	**100%**	**21.2**	**0.4**	7.6	5.9
GRAB	84.5%	75.2%	99.9%	22.4	**0.4**	6.8	5.7	81.9%	72.1%	99.9%	22.7	**0.4**	7.1	5.8
GRAB + OakInk	**95.6%**	**92.0%**	**100%**	**21.0**	0.6	**5.4**	**4.8**	93.5%	**89.0%**	**100%**	21.3	0.6	**5.4**	**4.8**

Using both hands significantly improves the success rate on some larger objects, where the humanoid can scoop up the object with one hand and carry it with both. As OakInk only has pre-grasps using one hand, it cannot learn such a strategy. Surprisingly, training on only GRAB achieves a high success rate on OakInk, picking up more than 1000 objects without training on the dataset, showcasing the robustness of our grasping policy on unseen objects.

OMOMO Dataset (7 objects). On the OMOMO dataset, we train a policy to show that our method can learn to pick up large objects. Table 10.2 shows that our method can successfully learn to pick up all the objects, including chairs and lamps. For larger objects, the pre-grasp guidance is essential for guiding the policy to learn bi-manual manipulation skills (as is shown in Fig 10.2)

10.5.2 Ablation and Analysis

Ablation. In this section, we study the effects of different components of our framework using the cross-object split of the GRAB dataset. Results are shown in Table 10.4. First, we compare our method trained with (Row 6) or without (R1) PULSE-X's action space. Using the same reward and state design, we can see that using the universal motion prior significantly improves success rates. Upon inspection, using PULSE-X also yields human-like motion, while not using it leads to unnatural motion (see in `supplement site`). R2 vs. R6 shows that the pre-grasp guidance is essential in learning grasps that are stable for grasping objects, but without it, some objects can still be grasped successfully. The difference between R3 and R6 is whether to train using the dexterous AMASS dataset. R3 vs R6 shows that without training on a dataset that has diverse hand motion and full-body motion, the policy can learn to pick up objects (high grasp success rate), but struggles in trajectory following. This is expected as the motion prior probably lacks the motion of "holding the object while moving". R4 and R5 show that object position randomization and hard-negativing mining are crucial for learning robust and successful policies. Ablations on the object latent code, RNN policy, *etc.* can be found in the Appendix 10.7.2.

Table 10.4: Ablation on various strategies of training *Omnigrasp*. PULSE-X: whether to use the latent motion representation. pre-grasp: pre-grasp guidance reward. Dex-AMASS: whether to train PULSE-X on the dexterous AMASS dataset. Rand-pose: randomizing the object initial pose. Hard-neg: hard-negative mining.

GRAB-Goal-Test (Cross-Object, 140 sequences, 5 unseen objects)												
idx	PULSE-X	pre-grasp	Dex-AMASS	Rand-pose	Hard-neg	$\text{Succ}_{\text{grasp}} \uparrow$	$\text{Succ}_{\text{traj}} \uparrow$	TTR ↑	$E_{\text{pos}} \downarrow$	$E_{\text{rot}} \downarrow$	$\text{E}_{\text{acc}} \downarrow$	$\text{E}_{\text{vel}} \downarrow$
1	✗	✓	✓	✓	✓	97.0%	33.6%	92.8%	43.5	**0.5**	10.6	8.3
2	✓	✗	✓	✓	✓	77.1%	57.9%	97.4%	54.9	1.0	5.5	5.2
3	✓	✓	✗	✓	✓	94.4%	77.3%	99.3%	30.5	0.9	4.8	**4.4**
4	✓	✓	✓	✗	✓	92.9%	79.9%	99.2%	31.4	1.1	**4.5**	**4.4**
5	✓	✓	✓	✓	✗	94.0%	71.6%	98.4%	32.3	1.3	6.2	5.7
6	✓	✓	✓	✓	✓	**100%**	**94.1%**	**99.6%**	**30.2**	0.9	5.4	4.7

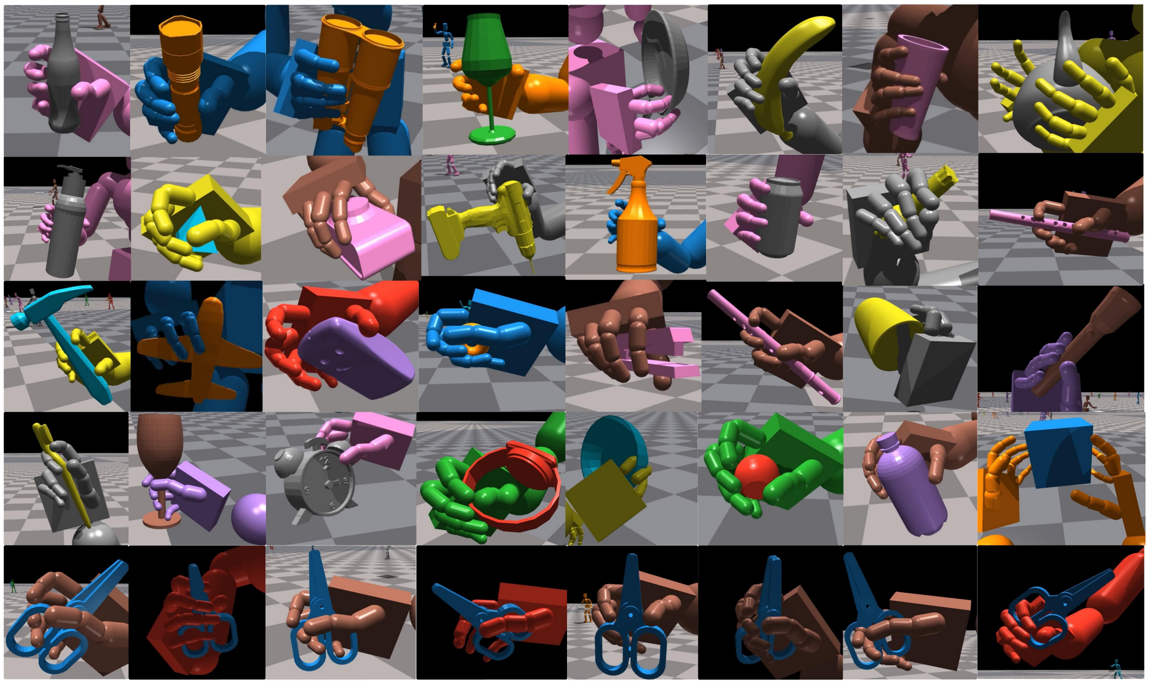


Figure 10.3: *(Top rows)*: grasping different objects using both hands. *(Bottom)* diverse grasps on the same object.

Table 10.5: Study on how noise affects pretrained Omnigrasp Policy

		GRAB-Goal-Test (Cross-Object, 140 sequences, 5 unseen objects)							GRAB-IMoS-Test (Cross-Subject, 92 sequences, 44 objects)						
Method	Noise Scale	$\text{Succ}_{\text{grasp}} \uparrow$	$\text{Succ}_{\text{traj}} \uparrow$	TTR ↑	$E_{\text{pos}} \downarrow$	$E_{\text{rot}} \downarrow$	$\text{E}_{\text{acc}} \downarrow$	$\text{E}_{\text{vel}} \downarrow$	$\text{Succ}_{\text{grasp}} \uparrow$	$\text{Succ}_{\text{traj}} \uparrow$	TTR ↑	$E_{\text{pos}} \downarrow$	$E_{\text{rot}} \downarrow$	$\text{E}_{\text{acc}} \downarrow$	$\text{E}_{\text{vel}} \downarrow$
Omnigrasp	0	**100%**	**94.1%**	**99.6%**	**30.2**	**0.93**	**5.4**	**4.7**	98.9%	**90.5%**	**99.8%**	**27.9**	**0.97**	**6.3**	**5.4**
Omnigrasp	0.01	**100%**	91.4%	99.2%	34.8	1.1	15.6	11.5	**99.5%**	86.2%	99.6%	32.5	1.0	17.9	13.2

Analysis: Diverse Grasps. In Fig. 10.3, we visualize the grasping strategy used by our method. We can see that based on the object shape, our policy uses a diverse set of grasping strategies to hold the object during the trajectory following. Based on the trajectory and object initial pose, *Omnigrasp* discovers different grasping poses for the *same* object, showcasing the advantage of using simulation and laws of physics for grasp generation. We also notice that for larger objects, our policy will resort to using two hands and a non-prehensile transport strategy. This behavior is learned from pre-grasps in GRAB, which utilize both hands for object manipulation.

Analysis: Robustness and Potential for Sim-to-real Transfer. In Table 10.5, we add uniform random noise [-0.01, 0.01] to both task observation (positions, object latent codes, etc.) and proprioception. A similar scale (0.01) of random noise is used in sim-to-real RL to tackle noisy input in real-world humanoids [143]. We see that Omnigrasp is relatively robust to input noise, even though it has not been trained with noisy input. Performance drop is more prominent in the acceleration and velocity metrics. Adding noise during training can further improve robustness. We do not claim that Omnigrasp is currently ready for real-world deployment, but we believe that a similar system design plus sim-to-real modifications (e.g. domain randomization, distilling into a vision-based policy) has the potential. We conduct more analysis on the robustness of our method with respect to initial object position, object weight, and object trajectories on our `supplement site`.

10.6 Limitations, Conclusions, and Future Work

Limitations. While *Omnigrasp* demonstrates the feasibility of controlling a simulated humanoid to grasp diverse objects and hold them to follow diverse trajectories, many limitations remain. For example, though the 6DoF input is provided in the input and reward, the rotation error remains to be further improved. *Omnigrasp* has yet to support precise in-hand manipulations. The success rate on trajectory following can be improved, as objects can be dropped or not picked up. Another area of improvement is to achieve *specific* types of grasps on the object, which may require additional input such as desired contact points and grasp. Human-level dexterity, even in simulation, remains challenging. For visualization of failure cases, see `supplement site`.

Conclusion and Future Work. In conclusion, we present *Omnigrasp*, a humanoid controller capable of grasping > 1200 objects and following trajectories while holding the object. It generalizes to unseen objects of similar sizes, utilizes bi-manual skills, and supports picking up larger objects. We demonstrate that by using a pretrained universal humanoid motion representation, grasping can be learned using simplistic reward and state designs. Future work includes improving trajectory following success rate, improving grasping diversity, and supporting more object categories. Also, improving upon the humanoid motion representation is a promising direction. While we utilize a simple yet effective unified motion latent space, separating the motion representation for hands and body [14, 36] could lead to further improvements. Effective object representation is also an important future direction. How to formulate an object representation that does not rely on canonical object pose and generalizes to vision-based systems will be valuable to help the model generalize to more objects.

10.7 Appendix

In this document, we include additional details about $PMCP$ that are omitted from the main paper due to the page limit. In Sec.10.7.1, we include additional information about training and evaluating the performance of our humanoid motion representation, PULSE-X. In Sec. 10.7.2, we include details about $PMCP$, such as the trajectory generator and training procedures.

Extensive qualitative results are provided at the project page as well as the supplementary zip files (which contain lower-resolution videos due to file size limitations). As motion is best seen in videos, we highly encourage our readers to view them to judge the capabilities of our method better. Specifically, we visualize using our controller to trace the characters "Omnigrasp" in the air while holding unseen objects during training. This complex trajectory is never seen during training. We also visualize the policy on GRAB [360], OakInk [441], and OMOMO [198] datasets, both for training and testing objects. On the GRAB dataset, we follow MoCap trajectories, while for the OakInk and OMOMO datasets, we showcase randomly generated trajectories for training. To demonstrate robustness to different object poses, weights, and directions, we also test our method by varying these variables and show that it can still pick up objects. Interestingly, we notice that our method prefers to use both hands to pick and hold the object as the weight of the object increases. We also include

Table 10.6: Imitation result on dexterous AMASS (14889 sequences).

	Dexterous AMASS-Train				
Method	Succ ↑	$E_{\text{g-mpjpe}}$ ↓	E_{mpjpe} ↓	E_{acc} ↓	E_{vel} ↓
PHC-X	99.9 %	29.4	31.0	4.1	5.1
PULSE-X	99.5 %	42.9	46.4	4.6	6.7

motion imitation and random motion sampling for PHC-X and PULSE-X. Further, we visualize our constructed dexterous AMASS dataset and the motion imitation result. Last, we include failure cases for grasping and trajectory following.

10.7.1 Details about PHC-X and PULSE-X

Data Cleaning. To train both PHC-X and PULSE-X, we follow PULSE's [232] procedure in filtering on implausible motion. This process yields 14889 motion sequences from the AMASS dataset for training our humanoid motion representation. Out of all 14889 sequences, only 9% of the sequences contain hand motion, and training on it will bias the motion imitator to have limited dexterity. Thus, we construct the dexterous AMASS dataset by pairing hand-only motion with body-only motion and demonstrate its effectiveness in learning a motion representation that enables object grasping.

Training and Architecture

The state, action, and rewards for PHC-X and PULSE-X follow the implementation choices of PULSE with the only modifications on the training data (dexterous AMASS) and humanoid (SMPL-X). PHC-X is trained for 1.5 week while PULSE-X takes 3 days. We use the same-sized networks: 6-layer MLP of units [2048, 1536, 1024, 1024, 512, 512] for PHC-X and 3-layer MLP of units [3096, 2048, 1024] for PULSE-X's encoder and decoders. We notice that due to the increase in DoF from SMPL (69) to SMPL-X (153), simulation is ∼2 times slower.

Evaluation

We evaluate PULSE-X and PHC-X on our constructed dexterous AMASS dataset. The metrics we use are the mean per-joint position error (mm) for both global $E_{\text{g-mhpe}}$ and local E_{mpjpe} (root-relative) settings. We also report acceleration and velocity errors, similar to the object trajectory following the setting but averaged across all body joints. From Table

Table 10.7: Hyperparameters for *PMCP*, PHC-X, and PULSE-X. σ: fixed variance for policy. γ: discount factor. ϵ: clip range for PPO.

Method	Batch Size	Learning Rate	σ	γ	ϵ						# of samples
PHC-X	3072	2×10^{-5}	0.05	0.99	0.2						$\sim 10^{10}$
	Batch Size	Learning Rate	Latent size								# of samples
PULSE-X	3072	5×10^{-4}	48								$\sim 10^{9}$
	Batch Size	Learning Rate	σ	γ	ϵ	w_{op}	w_{or}	w_{ov}	w_{oav}	w_{c}	# of samples
PMCP	3072	5×10^{-4}	0.36	0.99	0.2	0.5	0.3	0.05	0.05	0.1	$\sim 10^{9}$

10.6, we can see that PHC-X and PULSE-X achieve a high success rate on training data while maintaining a low per-joint error. Distilling from PHC-X to PULSE-X, we observe similar degradation in imitation performance as in PULSE, akin to the reconstruction error in training VAEs [179].

10.7.2 Details about *PMCP*

Object Processing

Since the simulator requires convex objects for simulation, we use the built-in v-hacd function to decompose the meshes into convex geometries. The parameters we use for decomposition can be found in Table 10.7. To compute object latent code, we use 512-d BPS [295] by randomly sampling 512 points on a unit sphere and calculating their distances to points on the object mesh. As some object meshes have a large number of vertices, we also perform quadratic decimation on the mesh if it contains more than 50000 vertices.

Training Details

Early Termination. During training, we terminate the episode whenever the object is more than 12cm away from its desired reference trajectory at time step t: $\|\hat{\boldsymbol{p}}_t^{\text{obj}} - \boldsymbol{p}_t^{\text{obj}}\|_2 > 0.12$.

Table Removal. Since the GRAB and OakInk datasets are table-top objects, we use a table at the beginning of the episode to support the object. However, since our randomly generated trajectory can collide with the table and the humanoid has no environmental awareness except for the object, we remove the table after certain timestamps (1.5s) during training.

Contact Detection. As IsaacGym does not provide easy access to contact labels and only provides contact forces, there is no way of differentiating between contact with the table,

Table 10.8: Additional ablations: Object-latent refers to whether to provide the object shape latent code $\boldsymbol{\sigma}^{\text{obj}}$ to the policy. RNN refers to either using an RNN-based policy or an MLP-based policy. Im-obs refers to whether to provide the policy with ground truth full-body pose $\hat{\boldsymbol{q}}_{t+1}$ as input.

GRAB-Goal-Test (Cross-Object, 140 sequences, 5 unseen objects)										
idx	Object Latent	RNN	Im-obs	Succ$_{\text{grasp}}$ ↑	Succ$_{\text{traj}}$ ↑	TTR ↑	E_{pos} ↓	E_{rot} ↓	E$_{\text{acc}}$ ↓	E$_{\text{vel}}$ ↓
1	✗	✓	✗	**100%**	93.2%	**99.8%**	**28.7**	1.3	6.1	5.1
2	✓	✗	✗	99.9%	89.6%	99.0%	33.4	1.2	4.5	4.4
3	✓	✓	✓	95.2	77.8%	97.9%	32.2	0.9	**3.2**	**3.9**
4	✓	✓	✗	**100%**	**94.1%**	99.6%	30.2	**0.9**	5.4	4.7

humanoid body, or objects. Thus, we resort to a heuristic-based way to detect contact. Specifically, if the object is within 0.2m from the hands, has non-zero contact forces, and has a non-zero velocity, we deem it to have contact with the hands.

Trajectory Generator. Randomly generated trajectories can be seen on our `supplement site` on the OakInk and OMOMO dataset, as there is no paired MoCap object motion for these datasets. We sample a random velocity and delta angle at each time step and aggregate the velocities to produce full trajectories. We bound the velocity of our randomly generated trajectories to be between [0, 2] m/s and bound the angles to be between [0, 1] radian. With a probability of 0.2, a sharp turn could happen where the angle is between [0, 2π]. As the trajectories can not be too high or low, we bound the z-direction translation to be between [0.1, 2.0]. For orientation, we sample a random ending orientation at the end of the trajectory and interpolate it between the object's initial trajectory to obtain a sequence of target rotations.

Additional Ablations

In Table 10.8, we provide additional ablations left out due to space limitations. Comparing Row 1 (R1) and R4, we can see that on the GRAB dataset cross-object test set, a policy trained without the object shape latent code $\boldsymbol{\sigma}^{\text{obj}}$ can be on par with a policy with access to it. This is because the humanoid learned a general "grasping" for small objects, and the 5 testing objects do not deviate too much from these strategies. Also, upon inspection, R1 learns to rely on bi-manual manipulation and using two hands when it cannot pick it up with one hand, at which point the object shape no longer affects the grasping pose as much. As a result, R1 suffers a higher rotation error E_{rot}. On the GRAB cross-subject test (44 objects), R1 has a trajectory success rate of Succ$_{\text{traj}}$ 84.2%, worse than R4's 90.5%. R2 vs. R4 shows that the RNN policy is more effective than the MLP-based policy, confirming our

Table 10.9: Per-object breakdown on the GRAB-Goal (cross-object) split.

Object	Braun et al.[36]			*PMCP*		
	$\text{Succ}_{\text{grasp}} \uparrow$	$\text{Succ}_{\text{traj}} \uparrow$	TTR ↑	$\text{Succ}_{\text{grasp}} \uparrow$	$\text{Succ}_{\text{traj}} \uparrow$	TTR ↑
Apple	95%	-	91%	**100%**	99.6%	**99.9%**
Binoculars	54%	-	83%	**100%**	90.5%	**99.6%**
Camera	95%	-	85%	**100%**	97.7%	**99.7%**
Mug	89%	-	74%	**100%**	97.3%	**99.8%**
Toothpaste	64%	-	94%	**100%**	80.9%	**99.0%**

intuition that some form of memory is beneficial for a sequential task, such as grasping and omnidirectional trajectory following. R3 studies the scenario where we provide ground truth full-body pose $\hat{\boldsymbol{q}}_t$ to the policy at all times, similar to the setting in PhysHOI [396] (though without the contact graph). Results show that this strategy leads to worse performance, and also prevents us from training on objects that do not have paired MoCap full-body motion. This indicates that the contact graph is needed to imitate human-object interaction precisely. *PMCP* provides a flexible interface to support learning and testing on novel objects without needing paired ground-truth full-body motion.

Per-object Successrate breakdown

In Table 10.9, we break down the per-object success rate on the cross-object split of the GRAB dataset. Of the 5 novel objects, our model finds it hardest to pick up the toothpaste, which has an elongated surface. Upon inspection, we find that *PMCP* will slip on the round edges of the toothpaste surface and fail to grasp the object. Compared to previous SOTA [36], *PMCP* outperforms in all metrics and objects.

10.7.3 Additional Discussions

Alternatives to PULSE-X

One alternative way for reusing the motor skills from a motion imitator like PHC-X is to train a kinematic motion latent space to provide reference motion to drive PHC-X. Such a general-purpose kinematic latent space has been used in physics-based control for pose estimation [391] and animation [468]. However, few have been extended to include dexterous hands. These latent spaces, like HuMoR [310], model motion transition using an encoder $\boldsymbol{q}_\phi(\boldsymbol{z}_t|\hat{\boldsymbol{q}}_t, \hat{\boldsymbol{q}}_{t-1})$ and decoder $p_\theta(\hat{\boldsymbol{q}}_t|\boldsymbol{z}_t, \hat{\boldsymbol{q}}_{t-1})$ where $\hat{\boldsymbol{q}}_t$ is the pose at time step t and $\boldsymbol{z}_t$ is the latent code. $\boldsymbol{q}_\phi$ and $\boldsymbol{p}_\theta$ are trained using supervised learning. The issue with applying such a latent space to simulated humanoid control is twofold:

- The output $\hat{\boldsymbol{q}}_t$ of the VAE model, while representing natural human motion, does not model the PD-target (action) space required to maintain balance. This is shown in prior art [391, 468], where an additional motion imitator is still needed to actuate the humanoid by imitating $\hat{\boldsymbol{q}}_t$ instead of using $\hat{\boldsymbol{q}}_t$ as policy output (PD-target).

- $\boldsymbol{q}_\phi$ and $\boldsymbol{p}_\theta$ are optimized using MoCap data, whose $\hat{\boldsymbol{q}}_t$ values are computed using ground truth motion and finite difference (for velocities). As a result, $\boldsymbol{q}_\phi$ and $\boldsymbol{p}_\theta$ handle noisy humanoid states from simulation poorly. Thus, [391] runs the kinematic latent space in an open-loop auto-regressive fashion without feedback from physics simulation (e.g.,using $\hat{\boldsymbol{q}}_{t-1}$ from the previous time step's output rather than from simulation). The lack of feedback from physics simulation leads to floating and unnatural artifacts [391], and the imitator heavily relies on residual force control to maintain stability.

10.7.4 Broader social impact.

Our method can be used to create a realistic grasping policy for humanoids, generate animation, or synthesize stable grasps. While the state designs have access to privileged information, the overall system design methodology (plus sim-to-real transfer techniques such as domain randomization) has the potential to be transferred to a real humanoid robot. Thus, it has a potential positive social impact, as it can create content or help build the next generation of home robots.

Part IV

Unified Human Motion Estimation and Generation

Chapter 11

Unified Human Motion Estimation and Generation

11.1 Introduction

Human motion modeling is a longstanding topic in computer vision and graphics, with applications in gaming, animation, and 3D content creation. These creative applications typically require precise and intuitive user control. Consider a scenario where a user aims to generate motion sequences integrating multiple modalities: starting from a video clip, transitioning to follow textual descriptions, syncing with audio cues, and aligning with another video, all while providing fine-grained control via user-defined keyframes. Such sequences must precisely replicate observed human movements, reflect intended actions described by text or music, and adhere consistently to specified keyframes. While recent advances have made significant progress in individual tasks, achieving such precision and flexibility across multiple modalities remains challenging. Specifically, motion estimation from videos typically involves deterministic predictions focused on accuracy, whereas text/music-to-motion generation requires diversity to all possible motions. Consequently, these tasks are usually treated independently despite sharing common representations like temporal dynamics and kinematic structures. This separation limits cross-task knowledge transfer and requires maintaining distinct models.

Recent studies have revealed the synergistic relationship between motion estimation and generation tasks. Generative models [138, 311, 373] have provided robust priors for motion estimation, particularly in challenging scenarios such as world-space estimation [32, 186, 202, 443]. Conversely, leveraging large-scale video data for estimation has enhanced the realism of generative models by enriching their learned motion distributions [210]. This motivates developing a unified generalist model capable of handling both tasks concurrently across multiple modalities. However, developing such a framework presents significant challenges due to the contrasting objectives of these tasks: generation requires producing diverse and plausible outputs from abstract inputs like text or audio, while estimation demands precise motion reconstruction from concrete observations such as videos and keypoints. Creating a unified architecture that effectively balances diverse generation with accurate reconstruction while leveraging shared representations remains a complex challenge.

To address these issues, we propose GENMO, a Generalist Model for Human Motion that unifies estimation and generation within a single framework. We formulate motion estimation as constrained motion generation adhering to observed signals. This unification yields synergistic benefits: generative priors enhance plausibility in challenging estimation scenarios (e.g., occlusions), while diverse video data enrich generative diversity without requiring ground-truth 3D annotations.

GENMO is built upon a diffusion model framework incorporating a novel dual-mode training paradigm: (1) *estimation mode*, where we feed the GENMO diffusion denoiser with zero-initialized noise and the largest diffusion timestep, forcing the model to produce maximum likelihood estimation (MLE) of the motion based on the conditional signals; (2) *generation mode*, follows traditional diffusion training by sampling noisy motions and timesteps according to a predefined schedule, enabling the model to learn rich generative distributions from the conditioning signals. This dual-mode approach allows GENMO to excel at both precise estimation and diverse generation tasks. We further enhance the framework with an estimation-guided training objective that effectively leverages in-the-wild videos with 2D annotations, substantially expanding the model's generative capabilities. Furthermore, our architectural innovations enable the processing of variable-length motion sequences and seamlessly integrate arbitrary combinations of multi-modal conditioning signals at different time intervals, as demonstrated in Fig. 5.1. Notably, GENMO generates multi-conditioned motions in a single feedforward diffusion pass, without requiring complex post-processing steps.

Through extensive empirical evaluation, we demonstrate GENMO's capabilities across a comprehensive suite of tasks encompassing both global and local motion estimation, as well as diverse motion generation tasks including music-to-dance synthesis, text-to-motion generation, and motion-inbetweening. Our experimental results establish that GENMO achieves state-of-the-art performance across various tasks (global motion estimation, local motion estimation, and music-to-dance generation), validating its efficacy as a unified generalist framework for human motion modeling.

Our contributions are summarized as follows:

- We propose GENMO, the first generalist model unifying state-of-the-art global motion estimation with flexible human motion generation conditioned on videos, music, text, 2D keypoints, and 3D keyframes.
- Our architecture design supports seamless generation of variable-length motions conditioned on arbitrary numbers and combinations of multimodal inputs without complex post-processing.
- We propose a novel dual-mode training paradigm to explore the synergy between regression and diffusion, and introduce an estimation-guided training objective that enables effective training on in-the-wild videos.
- We demonstrate bidirectional benefits: generative priors improve estimation under challenging conditions like occlusions; conversely, diverse video data enhances generative expressiveness.

11.2 Related Work

11.2.1 Human Motion Generation

Human motion generation has progressed significantly in recent years [17, 52, 70, 75, 80, 125, 130, 138, 145, 291, 291, 311, 339, 373, 378, 425, 474, 475, 508] leveraging a variety of conditioning signals such as text [64, 113, 124, 167], actions [121], speech [7, 507], music [204, 341, 351, 366, 376, 378], and scenes/objects [134, 190, 401, 450, 480]. Recently, multimodal motion generation has also gained attention [29, 227, 476, 506] enabling multiple input modalities. However, most existing methods focus solely on generative tasks without supporting estimation. For instance, the method [476] supports video input but treats it as a generative task, resulting in motions that loosely imitate video content rather than precisely matching it. In contrast, our method jointly handles generation and estimation tasks, yielding more precise video-conditioned results.

For long-sequence motion generation, existing works mostly rely on ad-hoc post-processing techniques to stitch separately generated fixed-length motions [12, 290, 296, 477]. In contrast, our method introduces a novel diffusion-based architecture enabling seamless generation of arbitrary-length motions conditioned on multiple modalities without complex post-processing.

Existing datasets, such as AMASS [240], are limited in size and diversity. To address the scarcity of 3D data, Motion-X [210] and MotionBank [426] augment datasets using 2D videos and 3D pose estimation models [338, 462], but the resulting motions often contain artifacts. In contrast, our method directly leverages in-the-wild videos with 2D annotations without explicit 3D reconstruction, reducing reliance on noisy data and enhancing robustness and diversity.

11.2.2 Human Motion Estimation

Human pose estimation from images [174, 200, 329], videos [68, 116, 185], or even sparse marker data [192, 293, 451] has been studied extensively in the literature. Recent works focus primarily on estimating global human motion in world-space coordinates [186, 202, 338, 399, 443, 462]. This is an inherently ill-posed problem, hence these methods leverage generative priors and SLAM methods to constrain human and camera motions, respectively. However, these methods typically involve computationally expensive optimization or separate post-processing steps.

More recent approaches aim to estimate global human motion in a feed-forward manner [336, 338, 399, 478], offering faster solutions. Our method extends this direction by jointly modeling generation and estimation within a unified diffusion framework. This integration leverages shared representations and generative priors during training to produce more plausible estimations.

11.3 Generalist Model for Human Motion

GENMO unifies motion estimation and generation by formulating both tasks as conditional motion generation. Specifically, it synthesizes a human motion sequence x of length N based on a set of condition signals $\mathcal{C}$ and a set of corresponding condition masks $\mathcal{M}$, where N can be arbitrarily large. The condition set $\mathcal{C}$ includes one or more of the following: video feature $c_{\text{video}} \in \mathbb{R}^{N\times d_{\text{video}}}$, camera motion $c_{\text{cam}} \in \mathbb{R}^{N\times d_{\text{cam}}}$, 2D skeleton $c_{\text{2d}} \in \mathbb{R}^{N\times d_{\text{2d}}}$, music clip $c_{\text{music}} \in \mathbb{R}^{N\times d_{\text{music}}}$, 2d bounding box $c_{\text{bbox}} \in \mathbb{R}^{N\times d_{\text{bbox}}}$, or natural language $c_{\text{text}} \in \mathbb{R}^{M\times d_{\text{text}}}$ that describes the motion where M is the number of text tokens. The condition mask $\mathcal{M}$ consists of the mask $m_\star \in \mathbb{R}^{N\times d_\star}$ for each condition type $c_\star$ in $\mathcal{C}$. The mask matrix is of the same size as the condition feature and its element is one if the condition feature is available and zero otherwise.

Joint Local and Global Motion Representation. We now introduce the motion representation we use for x. Most text-to-motion generation methods adopt an egocentric motion representation that encodes human motion in a heading-free local coordinate system. However, for motion estimation, human motions are typically represented in the camera coordinate system to ensure better image feature alignment that facilitates learning. In this work, to obtain a unified generation and estimation model, we adopt a general human motion representation that encodes both the egocentric and camera-space human motions, along with the camera poses. Our approach leverages the gravity-view coordinate system [336], where the global trajectory of a person at frame i includes the gravity-view orientation $\Gamma^i_{\text{gv}} \in \mathbb{R}^6$ and the local root velocities $v^i_{\text{root}} \in \mathbb{R}^3$. The local motion at the i-th frame is represented as the SMPL [221] parameters, which consists of joint angles $\theta^i \in \mathbb{R}^{24\times 6}$, shape parameters $\beta^i \in \mathbb{R}^{10}$, and root translation $t^i_{\text{root}} \in \mathbb{R}^3$. Camera pose information at frame i is encoded through the camera-to-world transformation $\pi^i = \left(\Gamma^i_{\text{cv}}, t^i_{\text{cv}}\right)$, comprising the camera-view orientation $\Gamma^i_{\text{cv}} \in \mathbb{R}^6$ and camera translation $t^i_{\text{cv}} \in \mathbb{R}^3$. Additionally, we include contact labels $p^i \in \mathbb{R}^6$ for hands and feet (heels and toes). The complete motion sequence $x = \left\{x^i\right\}_{i=1}^N$ encompasses N human poses, where each pose $x^i \in \mathbb{R}^D$ consists of global motion, local motion, and camera pose:

$$x^i = \left(\Gamma^i_{\text{gv}}, v^i_{\text{root}}, \theta^i, \beta^i, t^i_{\text{root}}, \pi^i, p^i\right). \tag{11.1}$$

11.3.1 Unified Estimation and Generation Design

In this section, we will present the architectural design of GENMO and elucidates how it unifies motion estimation and generation within a single model. The model architecture, illustrated in Figure 11.1, transforms a noisy motion sequence x_t with the conditions $\mathcal{C}$ and condtion masks $\mathcal{M}$ into a clean motion sequence x_0 through a series of carefully designed components. The initial processing stage consists of an additive fusion block that converts x_t into a sequence of per-frame motion tokens. This block utilizes dedicated multilayer perceptrons (MLPs) to process each condition type in $\mathcal{C}$ independently, combines their features through summation to create a unified condition representation, which is further fused with noisy motion x_t to produce the motion token sequence. The resulting sequence is subsequently processed through L GENMO modules, each comprising a RoPE-based Transformer block and our novel multi-text injection block. Our architecture leverages Rotary Position Embedding (RoPE) [350], which computes attention based on relative

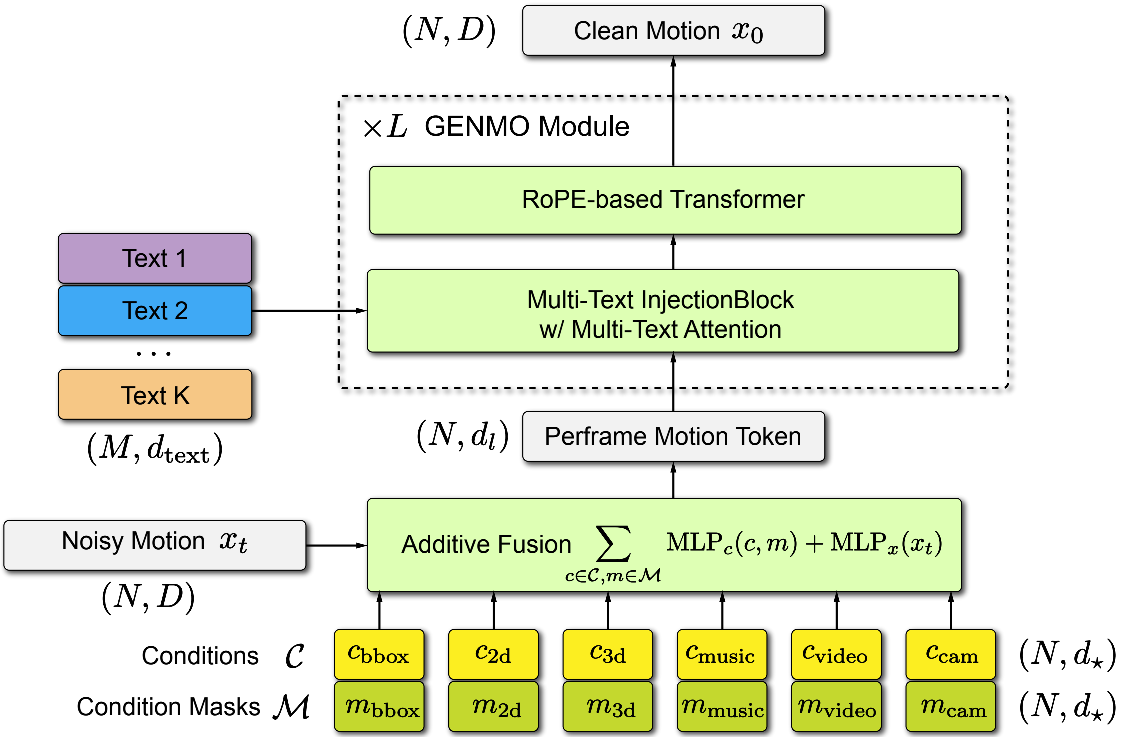


Figure 11.1: **GENMO Model Design** supports the generation of variable-length motion sequences in a single pass and enables seamless integration of multimodal conditioning signals, supporting both human motion generation and estimation.

temporal positions. This design choice enables processing of variable-length sequences and accommodates conditions lacking inherent temporal ordering, such as images and 2D skeletons.

However, text conditioning poses unique challenges. Unlike frame-aligned modalities such as video and music, text is not aligned with the motion frames. The conventional approach of concatenating text with the motion sequence is inadequate as inserting text at any positions can introduce temporal bias. To address this challenge, we propose a novel multi-text injection block that facilitates text-conditioned motion generation while accommodating multiple text inputs (K) with user-specified time windows. The multi-text injection block comprises a transformer block with our proposed multi-text attention mechanism at its core. As depicted in Figure 11.2, the multi-text attention mechanism processes K text embedding sequences $c^1_{\text{text}}, c^2_{\text{text}}, \ldots, c^K_{\text{text}}$ alongside the input motion feature sequence f_{in} to generate the output feature sequence f_{out}:

$$f_{out} = \sum_{k=1}^{K} \text{MaskedMHA}\big(f_{in}, c^k_{\text{text}}, \Omega_k\big). \tag{11.2}$$

$$\Omega_k(i,j) = \begin{cases} 1 & \text{if } i \text{ is within time window of text } k \\ 0 & \text{otherwise} \end{cases} \tag{11.3}$$

where MaskedMHA$(\cdot)$ represents a masked variant of the conventional multi-head attention mechanism. For each text input k, we employ a binary mask Ω_k that assumes a value of one when timestep i lies within the designated time window of text k, and zero otherwise. Through the multiplication of attention weights with mask Ω_k, we effectively restrict the influence of each text prompt to its corresponding time window. Although the mask introduces discontinuities at time window boundaries, GENMO successfully generates smooth motion sequences through the subsequent RoPE-based transformer block, which effectively captures and models temporal motion dynamics.

Inference with Arbitrary Motion Length. Our architecture employs relative positional embeddings rather than absolute embeddings for motion sequences, allowing GENMO to generate motions of arbitrary length in a single diffusion forward pass while naturally incorporating multiple text inputs across different time spans. During inference, we adopt a sliding window attention mechanism in the RoPE-based Transformer block, where each token attends only to tokens within a W-frame neighborhood. This design enables the generation of motion sequences longer than those seen during training while preserving computational efficiency and ensuring smooth, coherent motion transitions.

Mixed Multimodal Conditions. When conditioned on multiple modalities, our framework employs a principled approach for generation: text conditions, which lack frame-level alignment, are processed through our specialized multi-text attention mechanism, while frame-aligned modalities (e.g., video, music, 2D skeleton) are managed through a temporal masking strategy. As mentioned before, for each condition $c_\star$, we use a mask $m_\star$ of the same size to indicate whether the condition feature is (partially) present at each frame (one for present, zero otherwise). We also multiply the mask with the condition feature to nullify missing features. This simple yet effective approach enables seamless transitions between different conditioning modalities while maintaining temporal coherence.

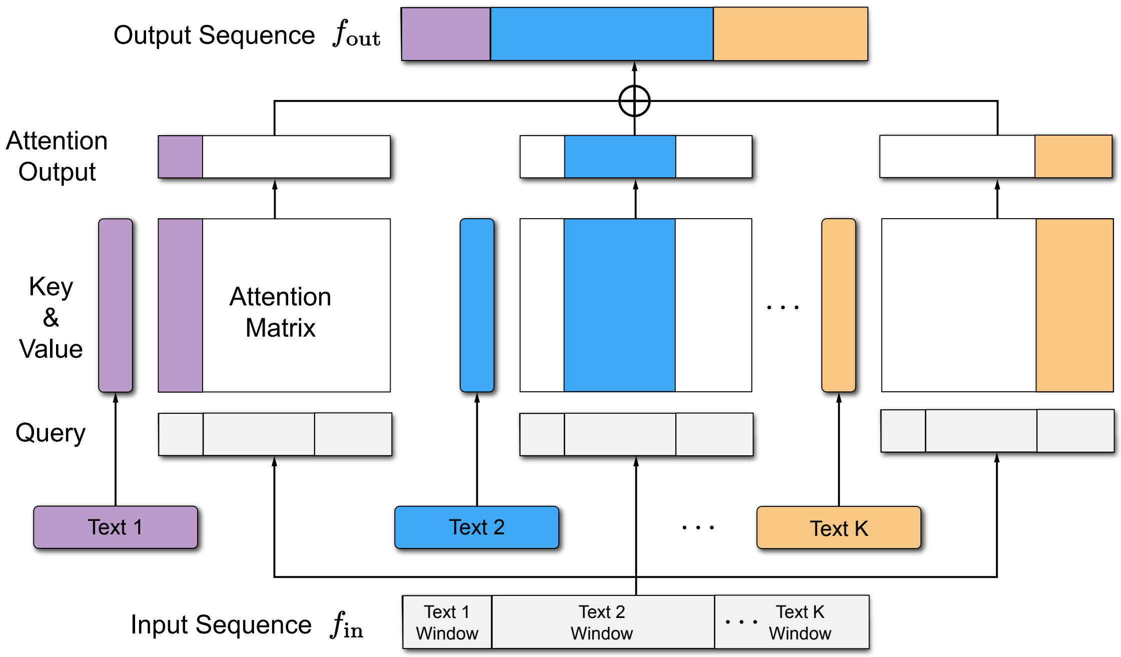


Figure 11.2: **Multi-text attention** enables flexible conditioning with multiple text inputs, each constrained to its specified time window.

11.3.2 Dual-Mode Training Paradigm

As a diffusion model, GENMO can theoretically be trained with the standard DDPM [149] objective:

$$\mathcal{L}_{\text{gen}} = \mathbb{E}_{t\sim[1,T],x_t\sim q(x_t|x_0)}\left[\left\|x_0 - \mathcal{G}(x_t, t, \mathcal{C}, \mathcal{M})\right\|^2\right], \tag{11.4}$$

where t the sampled diffusion timestep, and x_t is the noisy motion sampled from the forward diffusion process. Ideally, the model trained with this objective should be capable of generating motion sequences that satisfy the condition set $\mathcal{C}$ and mask $\mathcal{M}$, so it can be used as a motion estimation model when provided with video c_{video} or 2D skeleton c_{2d} conditions. However, we found that such a generative training objective is not enough to generate accurate motion sequences that are consistent with the input video. We observe a fundamental difference between motion estimation and text-to-motion generation tasks: motion estimation results exhibit substantially lower variability. To investigate this phenomenon, we trained separate diffusion models for text-to-motion generation and video-conditioned motion estimation, then visualized their predictions across all diffusion steps and different initial latent noises (Fig. 11.3). The results demonstrate that the video-

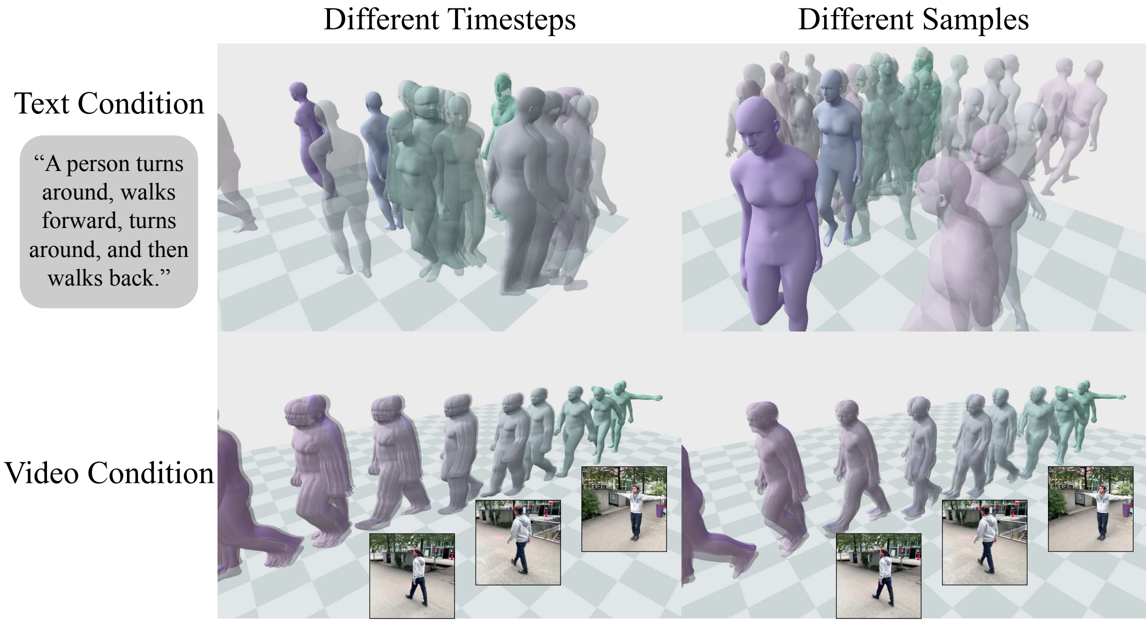

Different Timesteps
Different Samples
Text Condition
"A person turns around, walks forward, turns around, and then walks back."
Video Condition

Figure 11.3: **Variance of video/text conditioned predictions.** Left: Intermediate predictions across 50 DDIM denoising steps. Right: Predictions with 10 different initial noises (including zero-noise). Motions are transparent except the first-step and zero-noise predictions. Video conditioning yields more deterministic outputs compared to text conditioning.

conditioned model behaves more deterministically, in other words, the first-step prediction closely resembles predictions from subsequent steps with minimal variation. In contrast, the text-to-motion model exhibits significantly higher variance among steps. This observation has important implications for the estimation task: the accuracy of the first-step prediction becomes critical, as errors introduced early in the diffusion process are difficult to correct in later steps. Based on this insight, we propose a *dual-mode* training paradigm, which consists of (1) an *estimation mode* and (2) a *generation mode*. Intuitively, this dual-mode approach reinforces the quality of first-step predictions while maintaining the model's generative capabilities.

Estimation Mode. In the estimation mode, we formulate the problem as a regression task, employing maximum likelihood estimation to learn the conditional distribution $q(x|\mathcal{C}, \mathcal{M})$. This approach yields the following mean-square error (MSE) objective:

$$\mathcal{L}_{\text{est}} = \mathbb{E}_{z\sim\mathcal{N}(\mathbf{0},I)}\left[\left\|x_0 - \mathcal{G}(z, T, \mathcal{C}, \mathcal{M})\right\|^2\right]. \tag{11.5}$$

Rather than using noisy motion x_t, we utilize pure Gaussian noise $z \sim \mathcal{N}(\mathbf{0}, I)$ as input to the model, along with the corresponding maximum diffusion timestep T. This formulation ensures that the estimation mode aligns with the inherent variance characteristics of the diffusion model, thereby preventing conflicts between the generation and estimation modes.

To further enhance the quality of predicted motion sequences, we incorporate geometric regularization losses $\mathcal{L}_{\text{geo}}$ following established approaches in the literature [311, 373]. It involves decoding the predicted motion sequences into SMPL joints and vertices, followed by the application of constraints on world-space and camera-space vertices positions, world-space and camera-space joint positions, and joint contacts. In scenarios where only 2D annotations are available, we employ a 2D reprojection loss to effectively regularize the predicted motion sequences.

Generation Mode. For data with clean 3D annotations x_0, we can directly employ the standard diffusion objective in Eq. 11.4 to train the generation mode. In this section, we primarily focus on the more interesting scenario where only 2D annotations are available for the generation mode.

Unlike 3D annotations, 2D pose labels are more readily accessible through manual annotation or by applying robust 2D pose estimators on large-scale video datasets. 2D data also offers greater diversity compared to existing 3D motion capture data, which is constrained by the limited variety of subjects, motions, appearances, and environments.

Due to its inherent estimation capability, GENMO can naturally leverage 2D data for training the generation mode. Specifically, we propose an estimation-guided generation training strategy. First, we generate a pseudo-clean motion from the estimation mode using video or 2D skeleton as conditions: $\hat{x}_0 = \mathcal{G}(z, T, \mathcal{C})$. Subsequently, we sample a noisy motion sequence $\hat{x}_t$ through the forward diffusion process: $q\big(\hat{x}_t|\hat{x}_0\big)$. We then apply a 2D reprojection loss on the predicted clean motion using the 2D keypoint annotations x_{2d}:

$$\mathcal{L}_{\text{gen-2D}} = \mathbb{E}_{\hat{x}_t \sim q(\hat{x}_t|\hat{x}_0), t \sim [1,T]} \Big[\big\| x_{\text{2d}} - \Pi(\mathcal{G}(\hat{x}_t, t, \mathcal{C})) \big\|^2 \Big], \tag{11.6}$$

where Π represents the 2D projection function. For the generation mode, we also apply the aforementioned geometric losses $\mathcal{L}_{\text{geo}}$ to regularize the predicted motion sequences.

Training Mode Selection. We train the model on diverse datasets with various types of modalities. When training on datasets with strong conditioning signals that render the motion distribution more deterministic, such as video or 2D skeletons, we utilize both the estimation and generation modes to train GENMO. Conversely, when training on datasets with abstract conditions that result in more generative motion distributions, such as text and music, we exclusively employ the generation mode. This mode selection principle is

Table 11.1: **World-grounded human motion estimation.** We evaluate the global motion quality on the EMDB-2 [176] dataset and RICH [155]. Parenthesis denotes the number of joints used to compute WA-MPJPE$_{100}$, W-MPJPE$_{100}$ and Jitter.

	EMDB (24)					RICH (24)				
Models	WA-MPJPE$_{100}$	W-MPJPE$_{100}$	RTE	Jitter	Foot-Sliding	WA-MPJPE$_{100}$	W-MPJPE$_{100}$	RTE	Jitter	Foot-Sliding
GLAMR [462]	280.8	726.6	11.4	46.3	20.7	129.4	236.2	3.8	49.7	18.1
TRACE [356]	529.0	1702.3	17.7	2987.6	370.7	238.1	925.4	610.4	1578.6	230.7
SLAHMR [443]	326.9	776.1	10.2	31.3	14.5	98.1	186.4	28.9	34.3	5.1
COIN [202]	152.8	407.3	3.5	-	-	-	-	-	-	-
WHAM (w/ DPVO) [338]	135.6	354.8	6.0	22.5	4.4	109.9	184.6	4.1	19.7	3.3
WHAM (w/ GT extrinsics) [338]	131.1	335.3	4.1	21.0	4.4	109.9	184.6	4.1	19.7	3.3
GVHMR (w/ DPVO) [336]	111.0	276.5	2.0	16.7	**3.5**	78.8	126.3	2.4	**12.8**	**3.0**
GVHMR (w/ GT extrinsics) [336]	109.1	274.9	1.9	**16.5**	**3.5**	78.8	126.3	2.4	**12.8**	**3.0**
TRAM (w/ DROID-SLAM) [399]	76.4	222.4	1.4	-	-	-	-	-	-	-
Ours (w/ DROID-SLAM)	74.3	202.1	1.2	17.8	8.8	**75.3**	**118.6**	**1.9**	15.0	6.7
Ours (w/ GT extrinsics)	**69.5**	**185.9**	**0.9**	17.7	8.6	**75.3**	**118.6**	**1.9**	15.0	6.7

applied to both 3D and 2D data.

11.4 Experiments

We evaluate the performance of GENMO on four different tasks including video-to-motion, music-to-dance, text-to-motion, and motion in-betweening. Note that for all experiments we use a single one-in-all checkpoint jointly trained for all tasks unless stated otherwise.

Datasets. GENMO is trained on a diverse collection of motion datasets spanning multiple tasks: (1) motion capture data from AMASS [240]; (2) motion estimation benchmarks including BEDLAM [30], Human3.6M [159], and 3DPW [384]; (3) music-to-dance data from AIST++ [204]; (4) text-to-motion data from HumanML3D [122]; (5) 2D keypoints and text descriptions from Motion-X [210]. Comprehensive details regarding the training procedure and implementation are provided in the supplementary material.

For evaluation, we use RICH [155], and EMDB [176] for global human motion estimation, 3DPW [384] for local human motion estimation, AIST++ [204] for music-to-dance generation, and HumanML3D [122] and Motion-X [210] for text-to-motion generation.

Evaluation Metrics. For the music-to-dance generation, we follow the standard evaluation metrics [204, 376] and report the FID, Diversity, PFC, and BAS. For text-to-motion generation, we follow the standard evaluation metrics in previous works [122, 373] and report the R-Precision (Top 3), FID, Diversity, and MultiModal Dist. We also test the motion in-betweening performance by reporting the WA-MPJPE and PA-MPJPE for all the keyframes.

For motion estimation, we report MPJPE, PA-MPJPE, and PVE to evaluate the local

Table 11.2: **Camera-space metrics.** We evaluate the camera-space motion quality on the 3DPW [384], RICH [155] and EMDB-1 [176] datasets. * denotes models trained with the 3DPW training set.

		3DPW (14)				RICH (24)				EMDB (24)			
	Models	PA-MPJPE	MPJPE	PVE	Accel	PA-MPJPE	MPJPE	PVE	Accel	PA-MPJPE	MPJPE	PVE	Accel
per-frame	CLIFF* [207]	43.0	69.0	81.2	22.5	56.6	102.6	115.0	22.4	68.1	103.3	128.0	24.5
	HybrIK* [200]	41.8	71.6	82.3	–	56.4	96.8	110.4	–	65.6	103.0	122.2	–
	HMR2.0 [116]	44.4	69.8	82.2	18.1	48.1	96.0	110.9	18.8	60.6	98.0	120.3	19.8
	ReFit* [398]	40.5	65.3	75.1	18.5	47.9	80.7	92.9	17.1	58.6	88.0	104.5	20.7
temporal	VIBE* [185]	51.9	82.9	98.4	18.5	68.4	120.5	140.2	21.8	81.4	125.9	146.8	26.6
	TRACE* [356]	50.9	79.1	95.4	28.6	–	–	–	–	70.9	109.9	127.4	25.5
	SLAHMR [443]	55.9	–	–	–	52.5	–	–	9.4	69.5	93.5	110.7	7.1
	PACE [186]	–	–	–	–	49.3	–	–	8.8	–	–	–	–
	WHAM* [338]	35.9	57.8	68.7	6.6	44.3	80.0	91.2	5.3	50.4	79.7	94.4	5.3
	GVHMR* [336]	36.2	55.6	67.2	5.0	39.5	**66.0**	**74.4**	4.1	42.7	**72.6**	**84.2**	**3.6**
	TRAM* [399]	35.6	59.3	69.6	4.9	-	-	-	-	45.7	74.4	86.6	4.9
	Ours∗ (w/o 2D Training)	35.2	55.4	67.0	**4.8**	40.6	66.4	75.4	**4.0**	44.3	76.0	88.9	4.3
	Ours*	**34.6**	**53.9**	**65.8**	5.2	**39.1**	66.8	75.4	4.1	**42.5**	73.0	84.8	3.8

motion. Acceleration error (Accel) is also reported to measure the smoothness of the motion. For global motion estimation, we report W-MPJPE$_{100}$ and WA-MPJPE$_{100}$. We also evaluate the error accumulation over long sequences by reporting RTE in %. Jitters and foot sliding (FS) during contacts are also reported. Details are provided in the supplementary material.

Qualitative Results. We provide extensive qualitative results in the supplementary video, demonstrating the effectiveness and versatility of GENMO.

11.4.1 Evaluation of Motion Estimation

Global Motion Estimation.. We compare GENMO with state-of-the-art (SOTA) methods for recovering global human motion from videos with dynamic cameras. To ensure fair comparison across methods that employ different SLAM techniques during inference, we also report results using ground-truth camera parameters provided by the datasets. As shown in Table 11.1, GENMO consistently outperforms specialized methods trained exclusively for human motion estimation. Notably, our approach achieves a W-MPJPE of 202.1 mm on the EMDB dataset, surpassing TRAM [399] (222.4 mm) despite both methods utilizing identical SLAM systems and backbone features for video encoding. This performance advantage stems from our unified motion generation and estimation framework, where the generative prior enhances the quality of reconstructed motions. GENMO also demonstrates superior performance on the RICH dataset compared to all existing methods. Extensive

Table 11.3: **Benchmark of Music-to-Dance Generation.** Motion quality is evaluated on the AIST++ [204] dataset.

Methods	**FID**$_k$ ↓	**FID**$_m$ ↓	**Div**$_k$ ↑	**Div**$_m$ ↑	**PFC** ↓	**BAS** ↑
FACT [204]	86.43	43.46	6.85	3.32	2.2543	0.1607
Bailando [341]	28.16	**9.62**	7.83	6.34	1.754	0.2332
EDGE [376]	42.16	22.12	3.96	4.61	1.5363	0.2334
Ours (music only)	**16.10**	13.91	8.47	7.26	0.7340	0.2282
Ours	40.91	18.51	**10.09**	**7.48**	**0.3702**	**0.2708**

qualitative results are provided in the supplementary material.

Local Motion Estimation. We evaluate GENMO against SOTA methods for local 3D human motion estimation. Quantitative results in Table 11.2 demonstrate that GENMO surpasses existing approaches across most metrics. Additionally, we present results without training on 2D-only data, where the observed performance degradation highlights the effectiveness of our estimation-guided 2D training objective. Further evaluation on the challenging 3DPW-XOCC dataset [201] reveals that our generative prior enables GENMO to maintain robust performance even under severe occlusions and truncations. Comprehensive analyses and results on 3DPW-XOCC are provided in the supplementary material.

11.4.2 Evaluation of Motion Generation

Comparison on Music-to-Dance. We evaluate music-to-dance generation performance on the AIST++ dataset [204], with results presented in Table 11.3. GENMO is benchmarked against SOTA methods and a specialized variant of our model trained exclusively on AIST++ for music-to-dance generation. Notably, our generalist model, jointly trained across multiple estimation and generation tasks, demonstrates substantially enhanced motion diversity, physical plausibility, and motion-music correlation, as evidenced by superior Div$_k$, Div$_m$, PFC, and BAS metrics. While GENMO exhibits higher FID values compared to the specialized music-only variant, this performance differential is expected given that our generalist model was trained on considerably more heterogeneous motion data spanning multiple tasks and domains.

Comparison on Text-to-Motion. We evaluate the text-to-motion generation capabilities of GENMO on both HumanML3D (Table 11.4) and Motion-X (Table 11.5) datasets. Our method demonstrates superior performance compared to the baseline model MDM [373],

Table 11.4: **Benchmark of Text-to-Motion Generation** on the HumanML3D [122] dataset. R@3 denotes R-Precision (Top 3).

Methods	Rep.	**R@3 ↑**	**FID ↓**	**MM Dist ↓**	**Diversity →**
Real	HumanML3D	0.797	0.002	2.974	9.503
T2M [123]	HumanML3D	0.740	1.067	3.340	9.188
MDM [373]	HumanML3D	0.611	0.544	5.566	9.559
M2DM [188]	HumanML3D	0.763	0.352	3.134	9.926
EMDM [496]	HumanML3D	0.786	0.112	3.110	9.551
Ours (w/o 2D Training)	SMPL	0.556	0.245	3.128	11.660
Ours	SMPL	0.632	0.216	3.466	11.342

Table 11.5: **Benchmark of Text-to-Motion Generation.** Motion quality is evaluated on the Motion-X [210] dataset.

Methods	**R@3 ↑**	**FID ↓**	**MM Dist ↓**	**Diversity →**
Real	0.791	0.001	2.823	11.702
MDM [373]	0.313	2.389	6.745	8.720
Ours (w/o 2D Training)	0.401	0.515	5.210	12.124
Ours	**0.472**	**0.207**	**4.801**	11.719

exhibiting enhanced motion fidelity and improved text-prompt correspondence across both benchmarks. To assess the impact of 2D data training, we compare GENMO with its variant without training on Motion-X's 2D data. The results indicate that incorporating 2D training substantially enhances motion generation performance across both HumanML3D and Motion-X datasets. These findings substantiate the efficacy of leveraging 2D data within GENMO's framework for text-conditioned motion generation tasks.

Discussion on HumanML3D Performance. Although GENMO exhibits worse performance compared to SOTA methods like EMDM [496], this discrepancy can be stemmed from our representation choice: GENMO utilizes SMPL parameters to represent human motion for unified estimation and generation, whereas SOTA methods employ the HumanML3D representation — the same representation used by the encoders of the FID and R-Precision metrics. *This representational mismatch introduces an inherent disadvantage for GENMO*, as it necessitates bidirectional conversion of ground-truth motions from HumanML3D to SMPL during training and conversion of our generated motions to the HumanML3D

Table 11.6: **Motion In-betweening Experiments.** The DDPM baseline is the proposed method without the estimation objective, only using the standard diffusion objective for training. "w/o Estimation." is the proposed method without training for the motion estimation task. "w/o 2D Training" is trained without $\mathcal{L}_{\text{gen-2D}}$. Results are reported using PA-MPJPE/WA-MPJPE.

	HumanML3D		Motion-X	
Models	2-Keyframe	5-Keyframe	2-Keyframe	5-Keyframe
Diffusion-only	71.6/98.8	46.3/70.4	97.6/154.9	56.3/106.9
w/o Estimation	64.9/97.5	47.5/72.6	97.9/151.0	69.6/116.4
w/o 2D Training	56.4/**85.1**	**36.7**/59.5	68.3/136.8	44.6/98.6
Ours	**53.5**/85.3	37.1/**58.5**	**58.8/122.7**	**40.5/89.5**

format during evaluation. These conversion processes inevitably introduce distribution shifts through alterations in bone lengths, joint angles, and joint velocities, consequently affecting performance metrics and limiting the upper bound of GENMO's achievable performance on these HumanML3D-specific metrics.

Experiments on Motion In-betweening. We further evaluate the performance of conditional motion generation through the motion in-betweening task, following the methodology of prior diffusion-based approaches [373] by overwriting the noisy motion with the keyframe poses before each denoising step. Experiments are conducted on both HumanML3D and Motion-X test sets under two configurations with either 2 or 5 sampled keyframes. As shown in Table 11.6, GENMO achieves superior performance through its unified estimation and generation training compared to the diffusion-only baseline. Furthermore, the incorporation of additional 2D-only data and joint training with video-conditioned motion estimation substantially enhances motion in-betweening quality.

11.4.3 Ablation Study

Impact of the Estimation Mode. To assess the efficacy of our proposed estimation mode, we evaluate a variant of our method trained exclusively with the generation mode ("Diffusion-only"). Table 11.7 presents quantitative comparisons of global human motion estimation performance on the RICH and EMDB datasets, using direct model predictions without post-processing for static joints. The results demonstrate that omitting the estimation objective significantly degrades global motion estimation performance, confirming the

Table 11.7: **Ablation studies on motion estimation.** The DDPM baseline is the proposed method without the estimation objective, only using the standard diffusion objective for training. The regression baseline is the proposed method without the generation objective.

	RICH (24)		EMDB (24)	
Models	WA-MPJPE$_{100}$	W-MPJPE$_{100}$	WA-MPJPE$_{100}$	W-MPJPE$_{100}$
Diffusion-only	88.9	143.9	128.6	307.7
Regression-only	87.0	141.0	121.1	300.1
Ours	**81.3**	**130.6**	**114.6**	**281.7**

Table 11.8: **Effect of inference steps** on motion generation and estimation performance.

	HumanML3D (Gen.)	EMDB (Est.)	
Models	FID	W-MPJPE$_{100}$	MPJPE
Step=1 (Regression)	$0.260^{\pm.101}$	280.0	73.0
Step=2	$0.242^{\pm.083}$	276.8	72.5
Step=5	$0.231^{\pm.091}$	**274.9**	**72.2**
Step=10	$0.237^{\pm.126}$	275.8	72.3
Step=50	$\mathbf{0.216}^{\pm.119}$	278.7	72.7

estimation objective's crucial role in enhancing consistency between predicted motions and input videos. This finding is further corroborated by the motion in-betweening results in Table 11.6, which similarly indicate that the estimation objective improves in-betweening performance.

Impact of the Generation Mode. We compare our unified model against a pure regression baseline (trained solely with the estimation mode, akin to SOTA human motion estimation methods) to evaluate the impact of the generation objective. Quantitative comparisons on the RICH and EMDB datasets (Table 11.7) reveal that our unified model consistently outperforms the regression baseline, suggesting that incorporating generative priors enhances motion quality in human motion estimation tasks.

Different Inference Steps. We evaluate the impact of denoising steps using the standard DDIM [345] inference pipeline. As shown in Table 11.8, motion estimation performance remains relatively stable across different step counts, while text-to-motion generation shows greater sensitivity. Notably, single-step denoising sufficiently produces video-consistent human motions, with optimal estimation performance achieved at 5 inference steps — a balance that effectively leverages generative priors without introducing excessive variance.

11.5 Conclusion

In this work, we introduced GENMO, a generalist framework for human motion modeling that bridges the gap between motion estimation and generation tasks. We showed that GENMO can effectively leverage shared representations to enable synergistic benefits: generative priors enhance motion estimation robustness under challenging conditions, while diverse video data enriches the generative capabilities. GENMO can produce variable-length motion generation in a single pass and supports flexible control using text, videos, music, 2D keypoints, and 3D keyframes. GENMO achieved state-of-the-art performance on both motion estimation and generation benchmarks, while also reducing reliance on 3D motion capture data. Extensive experiments demonstrated that GENMO is not only capable of handling multiple human motion tasks within a single framework but also achieves superior results compared to task-specific models.

As with any other work, GENMO has some limitations. Currently, it relies on off-the-shelf SLAM methods to obtain camera parameters for videos. Integrating camera estimation inside GENMO is an interesting future work. Moreover, currently, our model only supports full-body motion. We plan to enable facial expressions and hand articulation in the future.

Part V

Conclusion and Future Work

11.6 Conclusion and Lessons

In this thesis, we explored the problem of modeling human motion across multiple types of interactions. We presented multiple projects for estimating and generating human motions, and combined multiple knowledge resources for a broad range of tasks, including video, motion capture, physics simulation, etc.

We began by estimating human motion from videos involving multiple people moving together, where their interactions were implicitly present. At this stage, we focused on global position estimation without modeling local body motion or deformation. We studied the appearance matching and parametric filtering solutions for tracking humans in crowds and in uniform appearances.

After investigating video-based motion estimation, we gradually shifted toward the generation of dynamic human trajectories. In estimation tasks, the primary goal was to deterministically match the observed video data, whereas in generation tasks, we aimed to produce diverse and plausible motion sequences through multi-modal modeling. We proposed to use the multi-modal distribution describing the motion intentions as the prior for the normalizing flow model to improve the diversity and controllability of trajectory generation.

To address more complex human interactions, we incorporated both global position changes and local body deformations into the motion generation process. The objective was to generate motion that appeared visually natural and physically realistic. While generative models trained on videos or motion capture datasets effectively enforced visual realism, we found it necessary to integrate physics simulators to ensure physical plausibility, such as preventing foot sliding and ground penetration. We explored solutions involving motion imitators and developed a method for imitating kinematic human pose sequences capable of capturing a broad range of real-world motion patterns. Building on this imitator, we introduced a language-guided human motion generation framework for interactive indoor scenes.

Seeking to move beyond interactions with static environments, we extended our study to human motion involving dynamic and manipulable objects. We addressed this problem in two stages. Due to the complexity of modeling hand pose and finger articulation required for dexterous grasping, we first developed a model for static hand-object grasp generation. These generated grasps were then used to guide reinforcement learning policies for hand-object interaction. Subsequently, we proposed a physics-based whole-body motion generation model that integrated visual realism from motion capture data, physical realism from simulation, and grasp priors from static grasp generation.

After examining a wide range of motion estimation and generation tasks, we explored using a single unified model to handle both. Using sensory data such as video as an optional condition, we developed a multi-modal conditioned diffusion model capable of generating and estimating human motion. This model also supported additional conditioning inputs, including 2D/3D human skeleton sequences, natural language, and music.

Here, we summarize some lessons learned during the presented works and some excluded but relevant works during the author's Ph.D. study:

1. **Parametric Models and Data-Based Models.** Deep learning-based models have become mainstream in many areas. For example, deep feature-based appearance matching is now the standard for target association in multi-object tracking. Similarly, training end-to-end generative models on motion capture data is considered the main approach for human motion generation. However, as demonstrated in multiple projects (Chapter 5, Chapter 6, Chapter 7), parametric modeling, such as linear motion filtering, mixed Gaussian priors, or physical rules from simulation, remains a key tool for improving the quality and efficiency of motion estimation and generation. Given that certain data are underrepresented in available datasets and that some modalities are difficult to collect or represent, we believe combining task-specific inductive biases from parametric models with data-based learnable models offers significant advantages.

2. **Physics Simulation and Generative Model.** Our generation works use two arts of methodologies, generative models (Chapter 6, Chapter 9, Chapter 11) and physics simulation (Chapter 7, Chapter 8, Chapter 10). Following the popular paradigm of end-to-end training, generative models benefit from large-scale and diverse annotated kinematic datasets but fall short in preventing unrealistic physics and body shapes. On the other hand, a physical simulator has significant advantages for motion generation with explicitly enforced physical rules, while it potentially suffers from human-like demonstrations and visual plausibility. We learned the significance of combining the two streams as in Chapter 10. We believe this can be extended to more complicated and high-fidelity human motion modeling in the future.

3. **Training Unified Models with Heterogeneous Data.** End-to-end training with uniformly annotated, single-modality data has long been the norm. However, due to the limitations in data availability and modality coverage, we encountered challenges in training generative models with homogeneous data sources. To address

this, we adopted a strategy of leveraging heterogeneous, multi-source datasets to train unified models. By encoding different data modalities into a shared latent space and conducting joint training therein, we achieved promising results (Chapter 6, Chapter 10, Chapter 11). This approach mirrors practices in contemporary multimodal large language models, and our findings reinforce the value of this paradigm for complex motion understanding and generation tasks.

Beyond the work presented in this thesis, there remains substantial room for advancement in human motion estimation and generation. Several limitations in our current work point to promising directions for future research.

First, we only model human-human interaction implicitly and do not account for local body deformation during social interactions. Moreover, to manage the complexity of multi-agent interactions common in daily life, we separately studied motion in interactions with other humans, scenes, and objects. However, this simplification limits realism. In particular, our treatment of human-scene and human-object interactions considers only rigid objects, whereas articulable and non-rigid objects play a critical role in shaping motion patterns—especially for dexterous hand operations.

These limitations highlight that human motion modeling is still in a relatively early stage compared to other computer vision tasks. Looking forward, our overarching goal is to build a generalist motion model capable of estimating and generating visually and physically realistic motion across all forms of interactive human behavior, as illustrated in Figure 11.4. To move toward this ambitious goal, we outline the following future directions:

1. We aim to develop models that can capture detailed human body motion and deformation when multiple individuals interact in close proximity. This is a particularly challenging problem because standard parametric models, such as SMPL [220], often fail under these conditions. Specifically, the joint-to-mesh pipeline tends to produce severe mesh penetrations when people are in physical contact. Furthermore, modeling the fine-grained social dynamics in such interactions adds another layer of complexity.

2. We seek to build unified estimation and generation models that can jointly reason about multiple elements—scene layouts, dynamic and manipulable objects, and other humans—within a single setup, whether in real-world 3D environments or simulation. With both static assets and active human agents embedded in a physics-aware environment, we are also interested in developing tools capable of replaying and analyzing sophisticated real-world dynamics in a controlled, reproducible way.

3. We plan to model the dexterous operation of both human hands to enable human-level manipulation capabilities with embodiment. This is a highly challenging problem that requires joint advances in computer vision and robotics, and potentially relies on richer sensory data beyond kinematic and physical inputs. As an initial step, we propose building a strong hand motion prior capable of estimating plausible hand movements from monocular in-the-wild video, while also remaining physically feasible during downstream demonstrations.

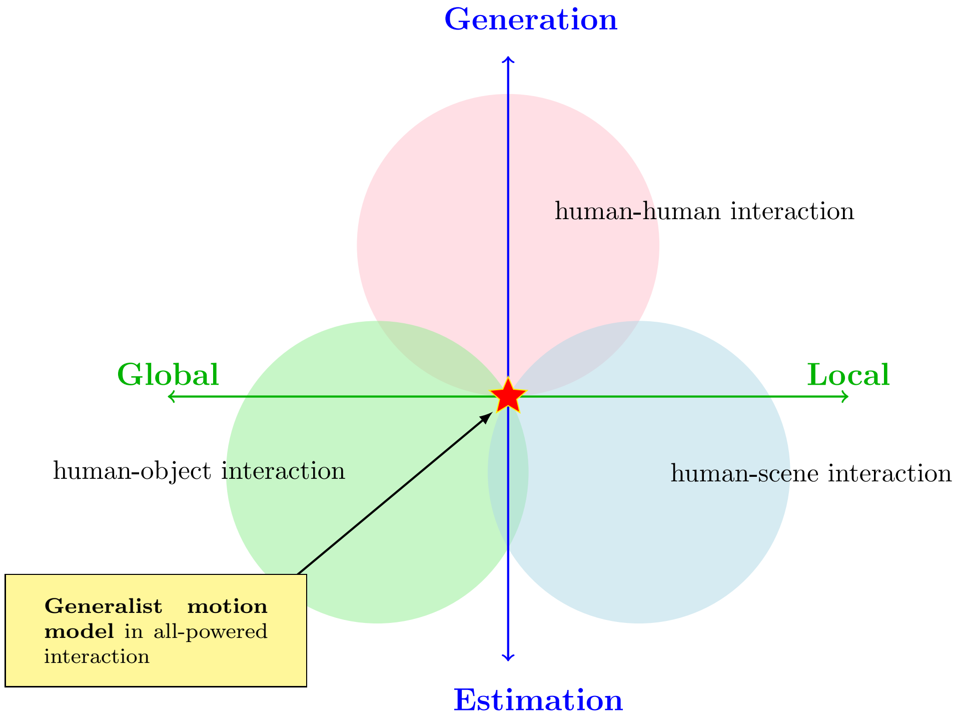


Figure 11.4: The quadrant to summarize projects we present in this thesis and the future goal of building a generalist motion model in all-powered interaction.

Bibliography

[1] Dexterous hand series. URL https://www.shadowrobot.com/dexterous-hand-series/.

[2] Dexterous hand series. https://www.shadowrobot.com/dexterous-hand-series/, 19 September 2023. Accessed: 2024-5-13.

[3] Adobe. Firefly, 2024. https://www.adobe.com/products/firefly.html.

[4] Martin Ahrnbom, Mikael G Nilsson, and Håkan Ardö. Real-time and online segmentation multi-target tracking with track revival re-identification. In *VISIGRAPP (5: VISAPP)*, pages 777–784, 2021.

[5] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[6] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016.

[7] Simon Alexanderson, Rajmund Nagy, Jonas Beskow, and Gustav Eje Henter. Listen, denoise, action! audio-driven motion synthesis with diffusion models. *ACM Transactions on Graphics (TOG)*, 2023.

[8] Hong-Yuan Mark Liao Alexey Bochkovskiy, Chien-Yao Wang. Yolov4: Yolov4: Optimal speed and accuracy of object detection. *arXiv*, 2020.

[9] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5167–5176, 2018.

[10] Joao Pedro Araújo, Jiaman Li, Karthik Vetrivel, Rishi Agarwal, Jiajun Wu, Deepak Gopinath, Alexander William Clegg, and Karen Liu. Circle: Capture in rich contextual environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21211–21221, 2023.

[11] Amir Atapour-Abarghouei and Toby P Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2800–2810, 2018.

[12] Nikos Athanasiou, Mathis Petrovich, Michael J. Black, and Gül Varol. TEACH: Temporal action composition for 3D humans. In *International Conference on 3D Vision (3DV)*, 2022.

[13] Nikos Athanasiou, Mathis Petrovich, Michael J Black, and Gül Varol. Sinc: Spatial composition of 3d human motions for simultaneous action generation. *arXiv preprint arXiv:2304.10417*, 2023.

[14] Jinseok Bae, Jungdam Won, Donggeun Lim, Cheol-Hui Min, and Young Min Kim. Pmp: Learning to physically interact with environments using part-wise motion priors. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–10, 2023.

[15] Fan Bao, Shen Nie, Kaiwen Xue, Chongxuan Li, Shi Pu, Yaole Wang, Gang Yue, Yue Cao, Hang Su, and Jun Zhu. One transformer fits all distributions in multi-modal diffusion at scale. *arXiv preprint arXiv:2303.06555*, 2023.

[16] Yaakov Bar-Shalom, Fred Daum, and Jim Huang. The probabilistic data association filter. *IEEE Control Systems Magazine*, 29(6):82–100, 2009.

[17] Emad Barsoum, John Kender, and Zicheng Liu. HP-GAN: Probabilistic 3D human motion prediction via GAN. In *CVPR Workshops*, 2018.

[18] Emad Barsoum, John Kender, and Zicheng Liu. Hp-gan: Probabilistic 3d human motion prediction via gan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1418–1427, 2018.

[19] Sumit Basu, Irfan Essa, and Alex Pentland. Motion regularization for model-based head tracking. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 3, pages 611–616. IEEE, 1996.

[20] Matan Ben-Yosef and Daphna Weinshall. Gaussian mixture generative adversarial networks for diverse datasets, and the unsupervised clustering of images, 2018.

[21] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 41–48, New York, NY, USA, June 2009. Association for Computing Machinery.

[22] Kevin Bergamin, Simon Clavet, Daniel Holden, and James Richard Forbes. DReCon: Data-driven responsive control of physics-based characters. *ACM Trans. Graph.*, 38 (6):11, 2019.

[23] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *ICCV*, pages 941–951, 2019.

[24] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

[25] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking perfor-

mance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

[26] Glen Berseth, Cheng Xie, Paul Cernek, and Michiel Van de Panne. Progressive reinforcement learning with distillation for multi-skilled motion control. *arXiv preprint arXiv:1802.04765*, 2018.

[27] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468. IEEE, 2016.

[28] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468. IEEE, 2016.

[29] Yuxuan Bian, Ailing Zeng, Xuan Ju, Xian Liu, Zhaoyang Zhang, Wei Liu, and Qiang Xu. Motioncraft: Crafting whole-body motion with plug-and-play multimodal controls. *arXiv preprint arXiv:2407.21136*, 2024.

[30] Michael J. Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 8726–8737, June 2023.

[31] Michael J Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. Bedlam: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8726–8737, 2023.

[32] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *ECCV*, 2016.

[33] Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G Willcocks. Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. *IEEE transactions on pattern analysis and machine intelligence*, 44(11):7327–7347, 2021.

[34] Samarth Brahmbhatt, Cusuh Ham, Charles C. Kemp, and James Hays. ContactDB: Analyzing and predicting grasp contact via thermal imaging. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. URL https://contactdb.cc.gatech.edu.

[35] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6247–6257, 2020.

[36] Jona Braun, Sammy Christen, Muhammed Kocabas, Emre Aksan, and Otmar Hilliges. Physically plausible full-body hand-object interaction synthesis. *Int. Conf. on 3D Vis.*, 2024.

[37] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis,

Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[38] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[39] Vittorio Caggiano, Sudeep Dasari, and Vikash Kumar. Myodex: Generalizable representations for dexterous physiological manipulation. *ICIP*, 2022.

[40] Jiarui Cai, Mingze Xu, Wei Li, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. Memot: Multi-object tracking with memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8090–8100, 2022.

[41] Jiarui Cai, Mingze Xu, Wei Li, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. Memot: Multi-object tracking with memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8090–8100, 2022.

[42] Jinkun Cao, Hongyang Tang, Hao-Shu Fang, Xiaoyong Shen, Cewu Lu, and Yu-Wing Tai. Cross-domain adaptation for animal pose estimation. *iccv*, August 2019.

[43] Jinkun Cao, Hongyang Tang, Hao-Shu Fang, Xiaoyong Shen, Cewu Lu, and Yu-Wing Tai. Cross-domain adaptation for animal pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9498–9507, 2019.

[44] Jinkun Cao, Xin Wang, Trevor Darrell, and Fisher Yu. Instance-aware predictive navigation in multi-agent environments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5096–5102. IEEE, 2021.

[45] Jinkun Cao, Xinshuo Weng, Rawal Khirodkar, Jiangmiao Pang, and Kris Kitani. Observation-centric sort: Rethinking sort for robust multi-object tracking. *arXiv preprint arXiv:2203.14360*, 2022.

[46] Jinkun Cao, Xinshuo Weng, Rawal Khirodkar, Jiangmiao Pang, and Kris Kitani. Observation-centric sort: Rethinking sort for robust multi-object tracking. *arXiv preprint arXiv:2203.14360*, 2022.

[47] Jinkun Cao, Hao Wu, and Kris Kitani. Track targets by dense spatio-temporal position encoding. *arXiv preprint arXiv:2210.09455*, 2022.

[48] Jinkun Cao, Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris M Kitani. Object tracking by hierarchical part-whole attention. *ICRA*, 2024.

[49] Zhe Cao, Ilija Radosavovic, Angjoo Kanazawa, and Jitendra Malik. Reconstructing hand-object interactions in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12417–12426, 2021.

[50] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[51] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[52] Pablo Cervantes, Yusuke Sekikawa, Ikuro Sato, and Koichi Shinoda. Implicit neural representations for variable length human motion generation. In *ECCV*, 2022.

[53] Orcun Cetintas, Guillem Brasó, and Laura Leal-Taixé. Unifying short and long-term tracking with graph hierarchies. *arXiv preprint arXiv:2212.03038*, 2022.

[54] Mohamed Chaabane, Peter Zhang, Ross Beveridge, and Stephen O'Hara. Deft: Detection embeddings for tracking. *arXiv preprint arXiv:2102.02267*, 2021.

[55] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[56] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019.

[57] Yin-Wen Chang, Cho-Jui Hsieh, Kai-Wei Chang, Michael Ringgaard, and Chih-Jen Lin. Training and testing low-degree polynomial data mappings via linear svm. *Journal of Machine Learning Research*, 11(4), 2010.

[58] Tatjana Chavdarova, Pierre Baqué, Stéphane Bouquet, Andrii Maksai, Cijo Jose, Timur Bagautdinov, Louis Lettry, Pascal Fua, Luc Van Gool, and François Fleuret. Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5030–5039, 2018.

[59] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 627–636, 2019.

[60] Long Chen, Haizhou Ai, Zijie Zhuang, and Chong Shang. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *2018 IEEE international conference on multimedia and expo (ICME)*, pages 1–6. IEEE, 2018.

[61] Tao Chen, Jie Xu, and Pulkit Agrawal. A system for general in-hand object reorientation. *Conference on Robot Learning*, 2021.

[62] Tao Chen, Megha Tippur, Siyang Wu, Vikash Kumar, Edward Adelson, and Pulkit Agrawal. Visual dexterity: In-hand reorientation of novel and complex object shapes. *Science Robotics*, 8(84):eadc9244, 2023. doi: 10.1126/scirobotics.adc9244. URL https://www.science.org/doi/abs/10.1126/scirobotics.adc9244.

[63] Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 18000–18010, 2023.

[64] Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, Jingyi Yu, and Gang Yu. Executing your commands via motion diffusion in latent space. In *CVPR*, 2023.

[65] Nuttapong Chentanez, Matthias Müller, Miles Macklin, Viktor Makoviychuk, and Stefan Jeschke. Physics-based motion capture imitation with deep reinforcement learning. In *Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games*, pages 1–10, 2018.

[66] Nuttapong Chentanez, Matthias Müller, Miles Macklin, Viktor Makoviychuk, and Stefan Jeschke. Physics-based motion capture imitation with deep reinforcement learning. *Proceedings - MIG 2018: ACM SIGGRAPH Conference on Motion, Interaction, and Games*, 2018.

[67] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*, 2014.

[68] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Beyond static features for temporally consistent 3d human pose and shape from a video. In *CVPR*, 2021.

[69] Wongun Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *Proceedings of the IEEE international conference on computer vision*, pages 3029–3037, 2015.

[70] B. Chopin, N. Otberdout, M. Daoudi, and A. Bartolo. Human motion prediction using manifold-aware wasserstein gan. In *FG*, 2021.

[71] Sammy Christen, Muhammed Kocabas, Emre Aksan, Jemin Hwangbo, Jie Song, and Otmar Hilliges. D-grasp: Physically plausible dynamic grasp synthesis for hand-object interactions. https://ait.ethz.ch/projects/2022/d-grasp/downloads/d-grasp.pdf, 2021. Accessed: 2021-12-7.

[72] Sammy Christen, Lan Feng, Wei Yang, Yu-Wei Chao, Otmar Hilliges, and Jie Song. Synh2r: Synthesizing hand-object motions for learning human-to-robot handovers. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3168–3175. IEEE, 2024.

[73] Sammy Christen, Shreyas Hampali, Fadime Sener, Edoardo Remelli, Tomas Hodan, Eric Sauser, Shugao Ma, and Bugra Tekin. Diffh2o: Diffusion-based synthesis of hand-object interactions from textual descriptions. *arXiv preprint arXiv:2403.17827*, 2024.

[74] Peng Chu, Jiang Wang, Quanzeng You, Haibin Ling, and Zicheng Liu. Transmot: Spatial-temporal graph transformer for multiple object tracking. *arXiv preprint arXiv:2104.00194*, 2021.

[75] Setareh Cohan, Guy Tevet, Daniele Reda, Xue Bin Peng, and Michiel van de Panne.

Flexible motion in-betweening with diffusion models. *SIGGRAPH*, 2024.

[76] MMTracking Contributors. MMTracking: OpenMMLab video perception toolbox and benchmark. https://github.com/open-mmlab/mmtracking, 2020.

[77] Rob Cornish, Anthony Caterini, George Deligiannidis, and Arnaud Doucet. Relaxing bijectivity constraints with continuously indexed normalising flows. In *International conference on machine learning*, pages 2133–2143. PMLR, 2020.

[78] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[79] Peng Dai, Renliang Weng, Wongun Choi, Changshui Zhang, Zhangping He, and Wei Ding. Learning a proposal classifier for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2443–2452, 2021.

[80] Wenxun Dai, Ling-Hao Chen, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. Motionlcm: Real-time controllable motion generation via latent consistency model. In *ECCV*, 2024.

[81] Sudeep Dasari, Abhinav Gupta, and Vikash Kumar. Learning dexterous manipulation from exemplar object trajectories and pre-grasps. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3889–3896. IEEE, 2023.

[82] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. Tao: A large-scale benchmark for tracking any object. In *European conference on computer vision*, pages 436–454. Springer, 2020.

[83] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. Tao: A large-scale benchmark for tracking any object. In *European conference on computer vision*, pages 436–454. Springer, 2020.

[84] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(7):3366–3385, July 2022.

[85] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.

[86] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv:2003.09003[cs]*, March 2020. URL http://arxiv.org/abs/1906.04567. arXiv: 2003.09003.

[87] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*, 2020.

[88] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[89] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[90] Nat Dilokthanakul, Pedro A. M. Mediano, Marta Garnelo, Matthew C. H. Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders, 2017.

[91] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[92] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[93] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. Strongsort: Make deepsort great again. *IEEE Transactions on Multimedia*, 2023.

[94] David Duvenaud. *Automatic model construction with Gaussian processes.* PhD thesis, University of Cambridge, 2014.

[95] Herbert Edelsbrunner, David Kirkpatrick, and Raimund Seidel. On the shape of a set of points in the plane. *IEEE Transactions on information theory*, 29(4):551–559, 1983.

[96] Zicong Fan, Maria Parelli, Maria Eleni Kadoglou, Muhammed Kocabas, Xu Chen, Michael J Black, and Otmar Hilliges. Hold: Category-agnostic 3d reconstruction of interacting hands and objects from video. *arXiv preprint arXiv:2311.18448*, 2023.

[97] Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J Black, and Otmar Hilliges. Arctic: A dataset for dexterous bimanual hand-object manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12943–12954, 2023.

[98] Hao-Shu Fang, Jinkun Cao, Yu-Wing Tai, and Cewu Lu. Pairwise body-part attention for recognizing human-object interactions. In *Proceedings of the European conference on computer vision (ECCV)*, pages 51–67, 2018.

[99] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 2023.

[100] James Ferryman and Ali Shahrokni. Pets2009: Dataset and challenge. In *2009 Twelfth IEEE international workshop on performance evaluation of tracking and surveillance*, pages 1–6. IEEE, 2009.

[101] Jerry A Fodor and Zenon W Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71, 1988.

[102] Robert M French and Nick Chater. Using noise to compute error surfaces in connec-

tionist networks: a novel means of reducing catastrophic forgetting. *Neural Comput.*, 14(7):1755–1769, July 2002.

[103] Yang Fu, Yunchao Wei, Yuqian Zhou, Honghui Shi, Gao Huang, Xinchao Wang, Zhiqiang Yao, and Thomas Huang. Horizontal pyramid matching for person re-identification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8295–8302, 2019.

[104] Zipeng Fu, Xuxin Cheng, and Deepak Pathak. Deep whole-body control: Learning a unified policy for manipulation and locomotion. *CoRL*, October 2022.

[105] Levi Fussell, Kevin Bergamin, and Daniel Holden. SuperTrack: motion tracking for physically simulated characters using supervised learning. *ACM Trans. Graph.*, 40(6): 1–13, December 2021.

[106] Juergen Gall, Angela Yao, Nima Razavi, Luc Van Gool, and Victor Lempitsky. Hough forests for object detection, tracking, and action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 33(11):2188–2202, 2011.

[107] Jiawei Gao, Ziqin Wang, Zeqi Xiao, Jingbo Wang, Tai Wang, Jinkun Cao, Xiaolin Hu, Si Liu, Jifeng Dai, and Jiangmiao Pang. Coohoi: Learning cooperative human-object interaction with manipulated object dynamics. *Advances in Neural Information Processing Systems*, 37:79741–79763, 2024.

[108] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 409–419, 2018.

[109] Damien Garreau, Wittawat Jitkrittum, and Motonobu Kanagawa. Large sample analysis of the median heuristic. *arXiv preprint arXiv:1707.07269*, 2017.

[110] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

[111] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

[112] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237, 2013.

[113] Anindita Ghosh, Noshaba Cheema, Cennet Oguz, Christian Theobalt, and Philipp Slusallek. Synthesis of compositional animations from textual descriptions. In *ICCV*, 2021.

[114] Anindita Ghosh, Rishabh Dabral, Vladislav Golyanik, Christian Theobalt, and Philipp Slusallek. Imos: Intent-driven full-body motion synthesis for human-object interactions. In *Eurographics*, 2023.

[115] Spyros Gidaris and Nikos Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *Proceedings of the IEEE international*

conference on computer vision, pages 1134–1142, 2015.

[116] Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa, and Jitendra Malik. Humans in 4d: Reconstructing and tracking humans with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14783–14794, 2023.

[117] Kehong Gong, Bingbing Li, Jianfeng Zhang, Tao Wang, Jing Huang, Michael Bi Mi, Jiashi Feng, and Xinchao Wang. PoseTriplet: Co-evolving 3D human pose estimation, imitation, and hallucination under self-supervision. *CVPR*, March 2022.

[118] Nicolas Franco Gonzalez, Andres Ospina, and Philippe Calvez. Smat: Smart multiple affinity metrics for multiple object tracking. In *International Conference on Image Analysis and Recognition*, pages 48–62. Springer, 2020.

[119] Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17113–17122, 2022.

[120] Jiaqi Guan, Ye Yuan, Kris M Kitani, and Nicholas Rhinehart. Generative hybrid representations for activity forecasting with no-regret learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 173–182, 2020.

[121] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2Motion: Conditioned generation of 3D human motions. In *ACM International Conference on Multimedia (ACMMM)*, 2020.

[122] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5152–5161, June 2022.

[123] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5152–5161, 2022.

[124] Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. TM2T: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *ECCV*, 2022.

[125] Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. 2023.

[126] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2255–2264, 2018.

[127] Fredrik Gustafsson. Particle filter theory and practice with positioning applications.

IEEE Aerospace and Electronic Systems Magazine, 25(7):53–82, 2010.

[128] Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, and P-J Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on signal processing*, 50(2):425–437, 2002.

[129] Tuomas Haarnoja, Kristian Hartikainen, Pieter Abbeel, and Sergey Levine. Latent space policies for hierarchical reinforcement learning. *arXiv preprint arXiv:1804.02808*, 2018.

[130] Ikhsanul Habibie, Daniel Holden, Jonathan Schwarz, Joe Yearsley, and Taku Komura. A recurrent variational autoencoder for human motion synthesis. In *BMVC*, 2017.

[131] Félix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)*, 39(4):60–1, 2020.

[132] Leonard Hasenclever, Fabio Pardo, Raia Hadsell, Nicolas Heess, and Josh Merel. CoMic: Complementary task learning & mimicry for reusable skills. http://proceedings.mlr.press/v119/hasenclever20a/hasenclever20a.pdf. Accessed: 2023-2-13.

[133] Leonard Hasenclever, Fabio Pardo, Raia Hadsell, Nicolas Heess, and Josh Merel. CoMic: Complementary task learning & mimicry for reusable skills. In Hal Daumé Iii and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4105–4115. PMLR, 2020.

[134] Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael J Black. Stochastic scene-aware motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11374–11384, 2021.

[135] Mohamed Hassan, Partha Ghosh, Joachim Tesch, Dimitrios Tzionas, and Michael J Black. Populating 3d scenes by learning human-scene interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14708–14718, 2021.

[136] Mohamed Hassan, Yunrong Guo, Tingwu Wang, Michael Black, Sanja Fidler, and Xue Bin Peng. Synthesizing physical character-scene interactions. *arXiv preprint arXiv:2302.00883*, 2023.

[137] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11807–11816, 2019.

[138] Chengan He, Jun Saito, James Zachary, Holly Rushmeier, and Yi Zhou. Nemf: Neural motion fields for kinematic animation. In *NeurIPS*, 2022.

[139] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[140] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[141] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[142] Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. In *arXiv*, 2024.

[143] Tairan He, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Learning human-to-humanoid real-time whole-body teleoperation, 2024.

[144] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.

[145] Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. MoGlow: Probabilistic and controllable motion synthesis using normalising flows. *ACM Transactions on Graphics (TOG)*, 2020.

[146] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[147] David V Hinkley. On the ratio of two correlated normal random variables. *Biometrika*, 56(3):635–639, 1969.

[148] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[149] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[150] Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017.

[151] Andrea Hornakova, Roberto Henschel, Bodo Rosenhahn, and Paul Swoboda. Lifted disjoint paths with application in multiple object tracking. In *International Conference on Machine Learning*, pages 4364–4375. PMLR, 2020.

[152] Andrea Hornakova, Timo Kaiser, Paul Swoboda, Michal Rolinek, Bodo Rosenhahn, and Roberto Henschel. Making higher order mot scalable: An efficient approximate solver for lifted disjoint paths. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6330–6340, 2021.

[153] Taylor Howell, Nimrod Gileadi, Saran Tunyasuvunakool, Kevin Zakka, Tom Erez, and Yuval Tassa. Predictive Sampling: Real-time Behaviour Synthesis with MuJoCo. dec 2022. doi: 10.48550/arXiv.2212.00541. URL https://arxiv.org/abs/2212.00541.

[154] Buzhen Huang, Liang Pan, Yuan Yang, Jingyi Ju, and Yangang Wang. Neural MoCon: Neural motion control for physically plausible human motion capture. March 2022.

[155] Chun-Hao P. Huang, Hongwei Yi, Markus Höschle, Matvey Safroshkin, Tsvetelina Alexiadis, Senya Polikovsky, Daniel Scharstein, and Michael J Black. Capturing and

inferring dense full-body human-scene contact. In *CVPR*, 2022.

[156] Siyuan Huang, Zan Wang, Puhao Li, Baoxiong Jia, Tengyu Liu, Yixin Zhu, Wei Liang, and Song-Chun Zhu. Diffusion-based generation, optimization, and planning in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16750–16761, 2023.

[157] Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6272–6281, 2019.

[158] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(7):1325–1339, 2014.

[159] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *TPAMI*, 36(7):1325–1339, 2014.

[160] Pavel Izmailov, Polina Kirichenko, Marc Finzi, and Andrew Gordon Wilson. Semi-supervised learning with normalizing flows, 2019.

[161] Alejandro Jaimes and Nicu Sebe. Multimodal human–computer interaction: A survey. *Computer vision and image understanding*, 108(1-2):116–134, 2007.

[162] Andrew H Jazwinski. *Stochastic processes and filtering theory.* Courier Corporation, 2007.

[163] Zhiwei Jia, Xuanlin Li, Zhan Ling, Shuang Liu, Yiran Wu, and Hao Su. Improving policy optimization with generalist-specialist learning. June 2022.

[164] Juntao Jian, Xiuping Liu, Manyi Li, Ruizhen Hu, and Jian Liu. Affordpose: A large-scale dataset of hand-object interactions with affordance-driven hand pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14713–14724, 2023.

[165] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *arXiv preprint arXiv:2306.14795*, 2023.

[166] Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation. In *ICCV*, 2021.

[167] Peng Jin, Yang Wu, Yanbo Fan, Zhongqian Sun, Yang Wei, and Li Yuan. Act as you wish: Fine-grained control of motion diffusion model with hierarchical semantic graphs. In *NeurIPs*, 2023.

[168] Rooji Jinan and Tara Raveendran. Particle filters for multiple target tracking. *Procedia Technology*, 24:980–987, 2016.

[169] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pages 182–193. International Society for Optics and Photonics, 1997.

[170] Simon J Julier and Jeffrey K Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[171] Jordan Juravsky, Yunrong Guo, Sanja Fidler, and Xue Bin Peng. Padl: Language-directed physics-based character control. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022.

[172] Rudolf Emil Kalman et al. Contributions to the theory of optimal control. *Bol. soc. mat. mexicana*, 5(2):102–119, 1960.

[173] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[174] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018.

[175] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J Black, Krikamol Muandet, and Siyu Tang. Grasping field: Learning implicit representations for human grasps. In *2020 International Conference on 3D Vision (3DV)*, pages 333–344. IEEE, 2020.

[176] Manuel Kaufmann, Jie Song, Chen Guo, Kaiyue Shen, Tianjian Jiang, Chengcheng Tang, Juan José Zárate, and Otmar Hilliges. EMDB: The Electromagnetic Database of Global 3D Human Pose and Shape in the Wild. In *ICCV*, 2023.

[177] Rawal Khirodkar, Jyun-Ting Song, Jinkun Cao, Zhengyi Luo, and Kris Kitani. Harmony4d: A video dataset for in-the-wild close human interactions. *Advances in Neural Information Processing Systems*, 37:107270–107285, 2024.

[178] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[179] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, pages 1–14, 2014.

[180] Polina Kirichenko, Pavel Izmailov, and Andrew G Wilson. Why normalizing flows fail to detect out-of-distribution data. *Advances in neural information processing systems*, 33:20578–20589, 2020.

[181] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. U. S. A.*, 114(13):3521–3526, 2017.

[182] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *European conference on computer vision*, pages 201–214. Springer, 2012.

[183] Jonathan Ko and Dieter Fox. Gp-bayesfilters: Bayesian filtering using gaussian process prediction and observation models. *Autonomous Robots*, 27(1):75–90, 2009.

[184] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3964–3979, 2020.

[185] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. VIBE: Video inference for human body pose and shape estimation. In *CVPR*, 2020.

[186] Muhammed Kocabas, Ye Yuan, Pavlo Molchanov, Yunrong Guo, Michael J. Black, Otmar Hilliges, Jan Kautz, and Umar Iqbal. PACE: Human and motion estimation from in-the-wild videos. In *3DV*, 2024.

[187] Frederic Koehler, Viraj Mehta, and Andrej Risteski. Representational aspects of depth and conditioning in normalizing flows. In *International Conference on Machine Learning*, pages 5628–5636. PMLR, 2021.

[188] Hanyang Kong, Kehong Gong, Dongze Lian, Michael Bi Mi, and Xinchao Wang. Priority-centric human motion generation in discrete latent space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14806–14816, 2023.

[189] Parth Kothari, Sven Kreiss, and Alexandre Alahi. Human trajectory forecasting in crowds: A deep learning perspective. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[190] Nilesh Kulkarni, Davis Rempe, Kyle Genova, Abhijit Kundu, Justin Johnson, David Fouhey, and Leonidas Guibas. NIFTY: Neural object interaction fields for guided human motion synthesis. *arXiv:2307.07511*, 2023.

[191] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. MOTChallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942 [cs]*, April 2015. URL http://arxiv.org/abs/1504.01942. arXiv: 1504.01942.

[192] Jiye Lee and Hanbyul Joo. Mocap everyone everywhere: Lightweight motion capture with smartwatches and a head-mounted camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[193] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 336–345, 2017.

[194] Sunmin Lee, Sebastian Starke, Yuting Ye, Jungdam Won, and Alexander Winkler. Questenvsim: Environment-aware simulated motion tracking from sparse sensors. *arXiv preprint arXiv:2306.05666*, 2023.

[195] Erich L Lehmann and George Casella. *Theory of point estimation*. Springer Science & Business Media, 2006.

[196] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, volume 26, pages 655–664. Wiley Online Library, 2007.

[197] Jiachen Li, Fan Yang, Hengbo Ma, Srikanth Malla, Masayoshi Tomizuka, and Chiho

Choi. Rain: Reinforced hybrid attention inference network for motion forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16096–16106, 2021.

[198] Jiaman Li, Jiajun Wu, and C Karen Liu. Object motion guided human motion synthesis. *ACM Transactions on Graphics (TOG)*, 42(6):1–11, 2023.

[199] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. HybrIK: A hybrid analytical-neural inverse kinematics solution for 3D human pose and shape estimation. November 2020.

[200] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In *CVPR*, 2021.

[201] Jiefeng Li, Siyuan Bian, Qi Liu, Jiasheng Tang, Fan Wang, and Cewu Lu. Niki: Neural inverse kinematics with invertible neural networks for 3d human pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12933–12942, 2023.

[202] Jiefeng Li, Ye Yuan, Davis Rempe, Haotian Zhang, Pavlo Molchanov, Cewu Lu, Jan Kautz, and Umar Iqbal. Coin: Control-inpainting diffusion prior for human and camera motion estimation. In *ECCV*, pages 426–446. Springer, 2024.

[203] Quanzhou Li, Jingbo Wang, Chen Change Loy, and Bo Dai. Task-oriented human-object interactions generation with implicit neural representations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3035–3044, 2024.

[204] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *ICCV*, 2021.

[205] Wei Li, Yuanjun Xiong, Shuo Yang, Mingze Xu, Yongxin Wang, and Wei Xia. Semi-tcl: Semi-supervised track contrastive representation learning. *arXiv preprint arXiv:2107.02396*, 2021.

[206] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6936–6945, 2019.

[207] Zhihao Li, Jianzhuang Liu, Zhensong Zhang, Songcen Xu, and Youliang Yan. Cliff: Carrying location information in full frames into human pose and shape estimation. In *ECCV*, 2022.

[208] Chao Liang, Zhipeng Zhang, Yi Lu, Xue Zhou, Bing Li, Xiyong Ye, and Jianxiao Zou. Rethinking the competition between detection and reid in multi-object tracking. *arXiv preprint arXiv:2010.12138*, 2020.

[209] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *arXiv.org*, 2109.13410, 2021.

[210] Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and

Lei Zhang. Motion-x: A large-scale 3d expressive whole-body human motion dataset. In *NeurIPS*, 2023.

[211] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[212] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[213] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[214] Libin Liu and Jessica Hodgins. Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018.

[215] Peiqi Liu, Yaswanth Orru, Chris Paxton, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Ok-robot: What really matters in integrating open-knowledge models for robotics. *arXiv preprint arXiv:2401.12202*, 2024.

[216] Shaowei Liu, Subarna Tripathi, Somdeb Majumdar, and Xiaolong Wang. Joint hand motion and interaction hotspots prediction from egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3282–3292, 2022.

[217] Siqi Liu, Guy Lever, Zhe Wang, Josh Merel, S M Ali Eslami, Daniel Hennes, Wojciech M Czarnecki, Yuval Tassa, Shayegan Omidshafiei, Abbas Abdolmaleki, Noah Y Siegel, Leonard Hasenclever, Luke Marris, Saran Tunyasuvunakool, H Francis Song, Markus Wulfmeier, Paul Muller, Tuomas Haarnoja, Brendan D Tracey, Karl Tuyls, Thore Graepel, and Nicolas Heess. From motor control to team play in simulated humanoid football. May 2021.

[218] Xueyi Liu and Li Yi. Geneoh diffusion: Towards generalizable hand-object interaction denoising via denoising diffusion. In *ICLR*, 2024.

[219] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graph.*, 34(6), 2015.

[220] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *SIGGRAPH Asia*, 34(6):248:1–248:16, 2015.

[221] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: a skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6):1–16, 2015.

[222] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[223] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[224] Jiaxin Lu, Hao Kang, Haoxiang Li, Bo Liu, Yiding Yang, Qixing Huang, and Gang Hua. Ugg: Unified generative grasping. *arXiv preprint arXiv:2311.16917*, 2023.

[225] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International journal of computer vision*, 129(2):548–578, 2021.

[226] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International journal of computer vision*, 129(2):548–578, 2021.

[227] Mingshuang Luo, Ruibing Hou, Zhuo Li, Hong Chang, Zimo Liu, Yaowei Wang, and Shiguang Shan. M^{3}gpt: An advanced multimodal, multitask framework for motion comprehension and generation. In *NeurIPs*, 2024.

[228] Zhengyi Luo, Jiashun Wang, Kangni Liu, Haotian Zhang, Chen Tessler, Jingbo Wang, Ye Yuan, Jinkun Cao, Zihui Lin, Fengyi Wang, et al. Humanoidolympics: Sports environments for physically simulated humanoids.

[229] Zhengyi Luo, Ryo Hachiuma, Ye Yuan, and Kris Kitani. Dynamics-regulated kinematic policy for egocentric pose estimation. *NeurIPS*, 34:25019–25032, 2021.

[230] Zhengyi Luo, Shun Iwase, Ye Yuan, and Kris Kitani. Embodied scene-aware human pose estimation. *NeurIPS*, June 2022.

[231] Zhengyi Luo, Ye Yuan, and Kris M Kitani. From universal humanoid control to automatic physically valid character creation. June 2022.

[232] Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris Kitani, and Weipeng Xu. Universal humanoid motion representations for physics-based control. *arXiv preprint arXiv:2310.04582*, 2023.

[233] Zhengyi Luo, Jinkun Cao, Alexander W. Winkler, Kris Kitani, and Weipeng Xu. Perpetual humanoid control for real-time simulated avatars. In *International Conference on Computer Vision (ICCV)*, 2023.

[234] Zhengyi Luo, Jinkun Cao, Sammy Christen, Alexander Winkler, Kris Kitani, and Weipeng Xu. Grasping diverse objects with simulated humanoids. *arXiv preprint arXiv:2407.11385*, 2024.

[235] Zhengyi Luo, Jinkun Cao, Rawal Khirodkar, Alexander Winkler, Kris Kitani, and Weipeng Xu. Real-time simulated avatar from head-mounted sensors. *arXiv preprint arXiv:2403.06862*, 2024.

[236] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Robust visual tracking via hierarchical convolutional features. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2709–2723, 2018.

[237] Takahiro Maeda and Norimichi Ukita. Fast inference and update of probabilistic density estimation on trajectory prediction. In *Proceedings of the IEEE/CVF*

International Conference on Computer Vision, pages 9795–9805, 2023.

[238] Gerard Maggiolino, Adnan Ahmad, Jinkun Cao, and Kris Kitani. Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification. *arXiv preprint arXiv:2302.11813*, 2023.

[239] Gerard Maggiolino, Adnan Ahmad, Jinkun Cao, and Kris Kitani. Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification. *arXiv preprint arXiv:2302.11813*, 2023.

[240] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *ICCV*, 2019.

[241] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. AMASS: Archive of motion capture as surface shapes. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-Octob:5441–5450, 2019.

[242] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance GPU-based physics simulation for robot learning. *tech report*, August 2021.

[243] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[244] Priyanka Mandikal and Kristen Grauman. Dexvip: Learning dexterous grasping with human hand pose priors from video. In *Conference on Robot Learning*, pages 651–661. PMLR, 2022.

[245] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 759–776. Springer, 2020.

[246] Weibo Mao, Chenxin Xu, Qi Zhu, Siheng Chen, and Yanfeng Wang. Leapfrog diffusion model for stochastic trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5517–5526, 2023.

[247] David Marr. *Vision: A computational investigation into the human representation and processing of visual information.* MIT press, 2010.

[248] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In Gordon H Bower, editor, *Psychology of Learning and Motivation*, volume 24, pages 109–165. Academic Press, January 1989.

[249] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint*

arXiv:2101.02702, 2021.

[250] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021.

[251] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8844–8854, 2022.

[252] Stan Melax, Leonid Keselman, and Sterling Orsten. Dynamics based 3d skeletal hand tracking. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 184–184, 2013.

[253] Josh Merel, Leonard Hasenclever, Alexandre Galashov, Arun Ahuja, Vu Pham, Greg Wayne, Yee Whye Teh, and Nicolas Heess. Neural probabilistic motor primitives for humanoid control, 2018. ISSN 2331-8422.

[254] Josh Merel, Saran Tunyasuvunakool, Arun Ahuja, Yuval Tassa, Leonard Hasenclever, Vu Pham, Tom Erez, Greg Wayne, and Nicolas Heess. Catch and carry: Reusable neural controllers for vision-guided whole-body tasks. *ACM Trans. Graph.*, 39(4), 2020.

[255] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831 [cs]*, March 2016. URL http://arxiv.org/abs/1603.00831. arXiv: 1603.00831.

[256] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[257] Andrew T Miller and Peter K Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine*, 11(4):110–122, 2004.

[258] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918, 2019.

[259] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14424–14432, 2020.

[260] Abduallah Mohamed, Deyao Zhu, Warren Vu, Mohamed Elhoseiny, and Christian Claudel. Social-implicit: Rethinking trajectory prediction evaluation and the effectiveness of implicit maximum likelihood estimation. In *European Conference on Computer Vision*, pages 463–479. Springer, 2022.

[261] Gyeongsik Moon, Shunsuke Saito, Weipeng Xu, Rohan Joshi, Julia Buffalini, Harley Bellan, Nicholas Rosen, Jesse Richardson, Mize Mallorie, Philippe Bree, Tomas Simon, Bo Peng, Shubham Garg, Kevyn McPhail, and Takaaki Shiratori. A dataset

of relighted 3D interacting hands. In *NeurIPS Track on Datasets and Benchmarks*, 2023.

[262] Jeremy Morton, Tim A Wheeler, and Mykel J Kochenderfer. Analysis of recurrent neural networks for probabilistic modeling of driver behavior. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1289–1298, 2016.

[263] Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. Grounded human-object interaction hotspots from video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8688–8697, 2019.

[264] Supreeth Narasimhaswamy, Uttaran Bhattacharya, Xiang Chen, Ishita Dasgupta, Saayan Mitra, and Minh Hoai. Handiffuser: Text-to-image generation with realistic hand appearances. *arXiv preprint arXiv:2403.01693*, 2024.

[265] OpenAI. Gpt-3: Generative pre-trained transformer 3. https://openai.com/research/gpt-3, 2020.

[266] OpenAI. Gpt-4 technical report, 2023.

[267] Liang Pan, Jingbo Wang, Buzhen Huang, Junyu Zhang, Haofan Wang, Xu Tang, and Yangang Wang. Synthesizing physically plausible human motions in 3d scenes. *arXiv preprint arXiv:2308.09036*, 2023.

[268] Bo Pang, Yizhuo Li, Yifan Zhang, Muchen Li, and Cewu Lu. Tubetk: Adopting tubes to track multi-object in a one-step training model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6308–6318, 2020.

[269] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 164–173, 2021.

[270] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 164–173, 2021.

[271] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680, 2021.

[272] Hwangpil Park, Ri Yu, and Jehee Lee. Multi-segment foot modeling for human animation. In *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games*, number Article 16 in MIG '18, pages 1–10, New York, NY, USA, November 2018. Association for Computing Machinery.

[273] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

[274] Seong Hyeon Park, Gyubok Lee, Jimin Seo, Manoj Bhat, Minseok Kang, Jonathan Francis, Ashwin Jadhav, Paul Pu Liang, and Louis-Philippe Morency. Diverse and admissible trajectory forecasting through multimodal context understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 282–298. Springer, 2020.

[275] Soohwan Park, Hoseok Ryu, Seyoung Lee, Sunmin Lee, and Jehee Lee. Learning predict-and-simulate policies from unorganized human motion data. *ACM Trans. Graph.*, 38(6):1–11, November 2019.

[276] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A A Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:10967–10977, 2019. ISSN 1063-6919.

[277] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[278] Dario Pavllo, David Grangier, and Michael Auli. Quaternet: A quaternion-based recurrent model for human motion. *arXiv preprint arXiv:1805.06485*, 2018.

[279] Ziqiang Pei. Deepsort pytorch. https://github.com/ZQPei/deep_sort_pytorch, 2019.

[280] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th international conference on computer vision*, pages 261–268. IEEE, 2009.

[281] Dezhi Peng, Zikai Sun, Zirong Chen, Zirui Cai, Lele Xie, and Lianwen Jin. Detecting heads using feature refine net and cascaded multi-scale architecture. *arXiv preprint arXiv:1803.09256*, 2018.

[282] Songyou Peng, Chiyu Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. Shape as points: A differentiable poisson solver. *Advances in Neural Information Processing Systems*, 34:13032–13044, 2021.

[283] Xue Bin Peng, Glen Berseth, Kangkang Yin, and Michiel Van De Panne. DeepLoco: dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Trans. Graph.*, 36(4):1–13, July 2017.

[284] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. DeepMimic. *ACM Trans. Graph.*, 37(4):1–14, 2018.

[285] Xue Bin Peng, Michael Chang, Grace Zhang, Pieter Abbeel, and Sergey Levine. MCP: Learning composable hierarchical control with multiplicative compositional policies. May 2019.

[286] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. AMP: Adversarial motion priors for stylized physics-based character control. *ACM Trans.*

Graph., (4):1–20, April 2021.

[287] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4):1–20, 2021.

[288] Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)*, 41(4):1–17, 2022.

[289] Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)*, 41(4):1–17, 2022.

[290] Mathis Petrovich, Or Litany, Umar Iqbal, Michael J. Black, Gül Varol, Xue Bin Peng, and Davis Rempe. Multi-track timeline control for text-driven 3d human motion generation. In *CVPR Workshop on Human Motion Generation*, 2024.

[291] Ekkasit Pinyoanuntapong, Pu Wang, Minwoo Lee, and Chen Chen. Mmm: Generative masked motion model. In *CVPR*, 2024.

[292] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[293] Jose Luis Ponton, Haoran Yun, Andreas Aristidou, Carlos Andujar, and Nuria Pelechano. Sparseposer: Real-time full-body motion reconstruction from sparse data. *ACM Transactions on Graphics*, 2023.

[294] Andres Potapczynski, Gabriel Loaiza-Ganem, and John P Cunningham. Invertible gaussian reparameterization: Revisiting the gumbel-softmax. *Advances in Neural Information Processing Systems*, 33:12311–12321, 2020.

[295] Sergey Prokudin, Christoph Lassner, and Javier Romero. Efficient learning on point clouds with basis point sets. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4332–4341, 2019.

[296] Yijun Qian, Jack Urbanek, Alexander G. Hauptmann, and Jungdam Won. Breaking the limits of text-conditioned 3D motion synthesis with elaborative descriptions. In *ICCV*, 2023.

[297] Zheng Qin, Sanping Zhou, Le Wang, Jinghai Duan, Gang Hua, and Wei Tang. Motiontrack: Learning robust short-term and long-term motions for multi-object tracking. *arXiv preprint arXiv:2303.10404*, 2023.

[298] Lawrence Rabiner and Biinghwang Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

[299] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Real-world humanoid locomotion with reinforcement learning. *Science Robotics*, 9(89):eadi9579, 2024.

[300] Ilija Radosavovic, Bike Zhang, Baifeng Shi, Jathushan Rajasegaran, Sarthak Kamat,

Trevor Darrell, Koushil Sreenath, and Jitendra Malik. Humanoid locomotion as next token prediction. *arXiv preprint arXiv:2402.19469*, 2024.

[301] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

[302] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

[303] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

[304] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[305] Akshay Rangesh and Mohan Manubhai Trivedi. No blind spots: Full-surround multi-object tracking for autonomous vehicles using cameras and lidars. *IEEE Transactions on Intelligent Vehicles*, 4(4):588–599, 2019.

[306] Akshay Rangesh, Pranav Maheshwari, Mez Gebre, Siddhesh Mhatre, Vahid Ramezani, and Mohan M Trivedi. Trackmpnn: A message passing graph neural architecture for multi-object tracking. *arXiv preprint arXiv:2101.04206*, 2021.

[307] Dushyant Rao, Fereshteh Sadeghi, Leonard Hasenclever, Markus Wulfmeier, Martina Zambelli, Giulia Vezzani, Dhruva Tirumala, Yusuf Aytar, Josh Merel, Nicolas Heess, and Raia Hadsell. Learning transferable motor skills with hierarchical latent mixture policies. *arXiv preprint arXiv:2112.05062*, 2021.

[308] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

[309] Steven Reece and Stephen Roberts. An introduction to gaussian processes for the kalman filter expert. In *2010 13th International Conference on Information Fusion*, pages 1–9. IEEE, 2010.

[310] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. Humor: 3d human motion model for robust pose estimation. *arXiv preprint arXiv:2105.04668*, 2021.

[311] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J. Guibas. Humor: 3d human motion model for robust pose estimation. In *ICCV*, 2021.

[312] Davis Rempe, Zhengyi Luo, Xue Bin Peng, Ye Yuan, Kris Kitani, Karsten Kreis, Sanja Fidler, and Or Litany. Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. *arXiv preprint arXiv:2304.01893*, 2023.

[313] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[314] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019.

[315] Nicholas Rhinehart, Kris M Kitani, and Paul Vernaza. R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 772–788, 2018.

[316] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. Precog: Prediction conditioned on goals in visual multi-agent settings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2821–2830, 2019.

[317] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European conference on computer vision*, pages 17–35. Springer, 2016.

[318] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 549–565. Springer, 2016.

[319] Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. Vision-language models are zero-shot reward models for reinforcement learning. *arXiv preprint arXiv:2310.12921*, 2023.

[320] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[321] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*, 2022.

[322] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*, 2022.

[323] Stephane Ross, Geoffrey J Gordon, and J Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. *arXiv preprint arXiv:1011.0686*, 2010.

[324] Jonas Rothfuss, Fabio Ferreira, Simon Boehm, Simon Walther, Maxim Ulrich, Tamim Asfour, and Andreas Krause. Noise regularization for conditional density estimation. *arXiv preprint arXiv:1907.08982*, 2019.

[325] Ludan Ruan, Yiyang Ma, Huan Yang, Huiguo He, Bei Liu, Jianlong Fu, Nicholas Jing Yuan, Qin Jin, and Baining Guo. Mm-diffusion: Learning multi-modal diffusion models for joint audio and video generation. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 10219–10228, 2023.

[326] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv [cs.LG]*, June 2016.

[327] Amir Sadeghian, Vineet Kosaraju, Agrim Gupta, Silvio Savarese, and Alexandre Alahi. Trajnet: Towards a benchmark for human trajectory prediction. *arXiv preprint*, 2018.

[328] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 683–700. Springer, 2020.

[329] István Sárándi and Gerard Pons-Moll. Neural localizer fields for continuous 3d human pose and shape estimation. 2024.

[330] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. Technical report, 2017.

[331] Matthias Schwab, Agnes Mayr, and Markus Haltmeier. Deep gaussian mixture model for unsupervised image segmentation. *arXiv preprint arXiv:2404.12252*, 2024.

[332] Novin Shahroudi, Mihkel Lepson, and Meelis Kull. Evaluation of trajectory distribution predictions with energy score. In *Forty-first International Conference on Machine Learning*.

[333] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018.

[334] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, et al. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 3633–3642, 2015.

[335] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv [cs.LG]*, January 2017.

[336] Zehong Shen, Huaijin Pi, Yan Xia, Zhi Cen, Sida Peng, Zechen Hu, Hujun Bao, Ruizhen Hu, and Xiaowei Zhou. World-grounded human motion recovery via gravity-view coordinates. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

[337] Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt. PhysCap: Physically plausible monocular 3D motion capture in real time. (1), August 2020.

[338] Soyong Shin, Juyong Kim, Eni Halilaj, and Michael J Black. Wham: Reconstructing world-grounded humans with accurate 3d motion. In *CVPR*, 2024.

[339] Ayumi Shiobara and Makoto Murakami. Human motion generation using wasserstein GAN. In *International Conference on Digital Signal Processing (ICDSP)*, 2021.

[340] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[341] Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and Ziwei Liu. Bailando: 3d dance generation by actor-critic gpt with choreographic memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11050–11059, 2022.

[342] Gerald L Smith, Stanley F Schmidt, and Leonard A McGee. *Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle.* National Aeronautics and Space Administration, 1962.

[343] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[344] Vladimir Somers, Christophe De Vleeschouwer, and Alexandre Alahi. Body part-based representation learning for occluded person re-identification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1613–1623, 2023.

[345] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020.

[346] Daniel Stadler and Jurgen Beyerer. Improving multiple pedestrian tracking by track management and occlusion handling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10958–10967, 2021.

[347] Daniel Stadler and Jürgen Beyerer. Modelling ambiguous assignments for multi-person tracking in crowds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 133–142, 2022.

[348] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Trans. Graph.*, 38(6):209–1, 2019.

[349] Sebastian Starke, Yiwei Zhao, Taku Komura, and Kazi Zaman. Local motion phases for learning multi-contact character movements. *ACM Transactions on Graphics (TOG)*, 39(4):54–1, 2020.

[350] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[351] Jiangxin Sun, Chunyu Wang, Huang Hu, Hanjiang Lai, Zhi Jin, and Jian-Fang Hu. You never stop dancing: Non-freezing dance generation via bank-constrained manifold projection. In *NeurIPS*, 2022.

[352] Peize Sun, Jinkun Cao, Yi Jiang, Rufeng Zhang, Enze Xie, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple object tracking with transformer. *arXiv preprint arXiv:2012.15460*, 2020.

[353] Peize Sun, Jinkun Cao, Yi Jiang, Rufeng Zhang, Enze Xie, Zehuan Yuan, Changhu

Wang, and Ping Luo. Transtrack: Multiple object tracking with transformer. *arXiv preprint arXiv:2012.15460*, 2020.

[354] Peize Sun, Jinkun Cao, Yi Jiang, Zehuan Yuan, Song Bai, Kris Kitani, and Ping Luo. Dancetrack: Multi-object tracking in uniform appearance and diverse motion. *arXiv preprint arXiv:2111.14690*, 2021.

[355] Peize Sun, Jinkun Cao, Yi Jiang, Zehuan Yuan, Song Bai, Kris Kitani, and Ping Luo. Dancetrack: Multi-object tracking in uniform appearance and diverse motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20993–21002, 2022.

[356] Yu Sun, Qian Bao, Wu Liu, Tao Mei, and Michael J Black. Trace: 5d temporal regression of avatars with dynamic cameras in 3d environments. In *CVPR*, 2023.

[357] Ramana Sundararaman, Cedric De Almeida Braga, Eric Marchand, and Julien Pettre. Tracking pedestrian heads in dense crowd. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3865–3875, 2021.

[358] István Sárándi, Timm Linder, Kai O Arras, and Bastian Leibe. MeTRAbs: Metric-scale truncation-robust heatmaps for absolute 3D human pose estimation. *arXiv*, pages 1–14, 2020.

[359] István Sárándi, Alexander Hermans, and Bastian Leibe. Learning 3D human pose estimation from dozens of datasets using a geometry-aware autoencoder to bridge between skeleton formats. December 2022.

[360] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. Grab: A dataset of whole-body human grasping of objects. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 581–600. Springer, 2020.

[361] Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *ECCV*, 2020. URL https://grab.is.tue.mpg.de.

[362] Omid Taheri, Vasileios Choutas, Michael J Black, and Dimitrios Tzionas. Goal: Generating 4d whole-body motion for hand-object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13263–13273, 2022.

[363] Omid Taheri, Yi Zhou, Dimitrios Tzionas, Yang Zhou, Duygu Ceylan, Soren Pirk, and Michael J Black. Grip: Generating interaction poses using spatial cues and latent consistency. In *International conference on 3D vision (3DV)*, 2024.

[364] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[365] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional

domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020.

[366] Taoran Tang, Jia Jia, and Hanyang Mao. Dance with melody: An LSTM-autoencoder approach to music-oriented dance synthesis. In *ACM International Conference on Multimedia (ACMMM)*, 2018.

[367] Tianxin Tao, Matthew Wilson, Ruiyu Gou, and Michiel van de Panne. Learning to get up. *arXiv [cs.GR]*, April 2022.

[368] Purva Tendulkar, Dídac Surís, and Carl Vondrick. Flex: Full-body grasping without full-body grasps. In *CVPR*, 2023.

[369] Chen Tessler, Israel Yoni Kasten, Israel Yunrong Guo, and Canada Nvidia. Calm: Conditional adversarial latent models for directable virtual characters.

[370] Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 358–374. Springer, 2022.

[371] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Amit H Bermano, and Daniel Cohen-Or. Human motion diffusion model. *arXiv [cs.CV]*, September 2022.

[372] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022.

[373] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. In *ICLR*, 2023.

[374] Zhi Tian, Chunhua Shen, Xinlong Wang, and Hao Chen. Boxinst: High-performance instance segmentation with box annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5443–5452, 2021.

[375] Pavel Tokmakov, Jie Li, Wolfram Burgard, and Adrien Gaidon. Learning to track with object permanence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10860–10869, 2021.

[376] Jonathan Tseng, Rodrigo Castellon, and Karen Liu. Edge: Editable dance generation from music. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 448–458, 2023.

[377] Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, Karsten Kreis, et al. Lion: Latent point diffusion models for 3d shape generation. *Advances in Neural Information Processing Systems*, 35:10021–10039, 2022.

[378] Guillermo Valle-Pérez, Gustav Eje Henter, Jonas Beskow, André Holzapfel, Pierre-Yves Oudeyer, and Simon Alexanderson. Transflower: probabilistic autoregressive dance generation with multimodal attention. *ACM Transactions on Graphics (TOG)*, 2021.

[379] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[380] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N

Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[381] Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds. In *2018 IEEE international Conference on Robotics and Automation (ICRA)*, pages 4601–4607. IEEE, 2018.

[382] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. *arXiv:1902.03604[cs]*, 2019. URL http://arxiv.org/abs/1902.03604. arXiv: 1902.03604.

[383] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 7942–7951, 2019.

[384] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV*, 2018.

[385] Nolan Wagener, Andrey Kolobov, Felipe Vieira Frujeri, Ricky Loynd, Ching-An Cheng, and Matthew Hausknecht. MoCapAct: A multi-task dataset for simulated humanoid control. August 2022.

[386] Weikang Wan, Haoran Geng, Yun Liu, Zikang Shan, Yaodong Yang, Li Yi, and He Wang. Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3891–3902, 2023.

[387] Chaoyang Wang, Chen Kong, and Simon Lucey. Distill knowledge from nrsfm for weakly supervised 3d pose learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 743–752, 2019.

[388] Gaoang Wang, Renshu Gu, Zuozhu Liu, Weijie Hu, Mingli Song, and Jenq-Neng Hwang. Track without appearance: Learn box and tracklet embedding with local and global motion patterns for vehicle tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9876–9886, 2021.

[389] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):283–298, 2007.

[390] Jingbo Wang, Yu Rong, Jingyuan Liu, Sijie Yan, Dahua Lin, and Bo Dai. Towards diverse and natural scene-aware 3d human motion synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20460–20469, 2022.

[391] Jingbo Wang, Ye Yuan, Zhengyi Luo, Kevin Xie, Dahua Lin, Umar Iqbal, Sanja Fidler, Sameh Khamis, Hong Kong, and Mellon University, Carnegie. Learning human

dynamics in autonomous driving scenarios. International Conference on Computer Vision, 2023, 2023.

[392] Jingbo Wang, Zhengyi Luo, Ye Yuan, Yixuan Li, and Bo Dai. Pacer+: On-demand pedestrian animation controller in driving scenarios. *arXiv preprint arXiv:2404.19722*, 2024.

[393] Ruicheng Wang, Jialiang Zhang, Jiayi Chen, Yinzhen Xu, Puhao Li, Tengyu Liu, and He Wang. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11359–11366. IEEE, 2023.

[394] Shuai Wang, Hao Sheng, Yang Zhang, Yubin Wu, and Zhang Xiong. A general recurrent tracking framework without real data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13219–13228, 2021.

[395] Tingwu Wang, Yunrong Guo, Maria Shugrina, and Sanja Fidler. UniCon: Universal neural controller for physics-based character motion. *arXiv*, 2020.

[396] Yinhuai Wang, Jing Lin, Ailing Zeng, Zhengyi Luo, Jian Zhang, and Lei Zhang. Physhoi: Physics-based imitation of dynamic human-object interaction. *arXiv preprint arXiv:2312.04393*, 2023.

[397] Yongxin Wang, Kris Kitani, and Xinshuo Weng. Joint object detection and multi-object tracking with graph neural networks. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13708–13715. IEEE, 2021.

[398] Yufu Wang and Kostas Daniilidis. Refit: Recurrent fitting network for 3d human recovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14644–14654, 2023.

[399] Yufu Wang, Ziyun Wang, Lingjie Liu, and Kostas Daniilidis. Tram: Global trajectory and motion of 3d humans from in-the-wild videos. In *European Conference on Computer Vision*, pages 467–487. Springer, 2024.

[400] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8741–8750, 2021.

[401] Zan Wang, Yixin Chen, Tengyu Liu, Yixin Zhu, Wei Liang, and Siyuan Huang. Humanise: Language-conditioned human motion generation in 3d scenes. *Advances in Neural Information Processing Systems*, 35:14959–14971, 2022.

[402] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 107–122. Springer, 2020.

[403] Waymo. Waymo open dataset: An autonomous driving dataset, 2019-2025.

[404] Mark Weber, Jun Xie, Maxwell Collins, Yukun Zhu, Paul Voigtlaender, Hartwig

Adam, Bradley Green, Andreas Geiger, Bastian Leibe, Daniel Cremers, et al. Step: Segmenting and tracking every pixel. *arXiv preprint arXiv:2102.11859*, 2021.

[405] Zehang Weng, Haofei Lu, Danica Kragic, and Jens Lundell. Dexdiffuser: Generating dexterous grasps with diffusion models. *arXiv preprint arXiv:2402.02989*, 2024.

[406] Christopher Williams and Carl Rasmussen. Gaussian processes for regression. *Advances in neural information processing systems*, 8, 1995.

[407] Alexander Winkler, Jungdam Won, and Yuting Ye. QuestSim: Human motion tracking from sparse sensors with simulated avatars. September 2022.

[408] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 748–756. IEEE, 2018. doi: 10.1109/WACV.2018.00087.

[409] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[410] Maciej Wołczyk, Magdalena Proszewska, Łukasz Maziarka, Maciej Zieba, Patryk Wielopolski, Rafał Kurczab, and Marek Smieja. Plugen: Multi-label conditional generation from pre-trained models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8647–8656, 2022.

[411] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. A scalable approach to control diverse behaviors for physically simulated characters. *ACM Trans. Graph.*, 39 (4), 2020.

[412] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. Control strategies for physically simulated characters performing two-player competitive sports. *ACM Trans. Graph.*, 40(4):1–11, July 2021.

[413] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. Physics-based character controllers using conditional VAEs. *ACM Trans. Graph.*, 41(4):1–12, July 2022.

[414] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. Physics-based character controllers using conditional vaes. *ACM Transactions on Graphics (TOG)*, 41(4): 1–12, 2022.

[415] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12352–12361, 2021.

[416] Yan Wu, Jiahao Wang, Yan Zhang, Siwei Zhang, Otmar Hilliges, Fisher Yu, and Siyu Tang. Saga: Stochastic whole-body grasping with contact. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[417] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[418] Yu Xiang, Alexandre Alahi, and Silvio Savarese. Learning to track: Online multi-object

tracking by decision making. In *Proceedings of the IEEE international conference on computer vision*, pages 4705–4713, 2015.

[419] Kevin Xie, Tingwu Wang, Umar Iqbal, Yunrong Guo, Sanja Fidler, and Florian Shkurti. Physics-based human motion estimation and synthesis from videos. September 2021.

[420] Xianghui Xie, Bharat Lal Bhatnagar, and Gerard Pons-Moll. Visibility aware human-object interaction tracking from single rgb camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4757–4768, 2023.

[421] Yuliang Xiu, Jiefeng Li, Haoyu Wang, Yinghong Fang, and Cewu Lu. Pose flow: Efficient online pose tracking. *arXiv preprint arXiv:1802.00977*, 2018.

[422] Chenxin Xu, Maosen Li, Zhenyang Ni, Ya Zhang, and Siheng Chen. Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6507, 2022.

[423] Chenxin Xu, Weibo Mao, Wenjun Zhang, and Siheng Chen. Remember intentions: Retrospective-memory-based trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6488–6497, 2022.

[424] Chenxin Xu, Robby T Tan, Yuhong Tan, Siheng Chen, Yu Guang Wang, Xinchao Wang, and Yanfeng Wang. Eqmotion: Equivariant multi-agent motion prediction with invariant interaction reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1410–1420, 2023.

[425] Liang Xu, Ziyang Song, Dongliang Wang, Jing Su, Zhicheng Fang, Chenjing Ding, Weihao Gan, Yichao Yan, Xin Jin, Xiaokang Yang, Wenjun Zeng, and Wei Wu. ActFormer: A gan-based transformer towards general action-conditioned 3d human motion generation. In *ICCV*, 2023.

[426] Liang Xu, Shaoyang Hua, Zili Lin, Yifan Liu, Feipeng Ma, Yichao Yan, Xin Jin, Xiaokang Yang, and Wenjun Zeng. Motionbank: A large-scale video motion benchmark with disentangled rule-based annotations, 2024.

[427] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018.

[428] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018.

[429] Sirui Xu, Zhengyuan Li, Yu-Xiong Wang, and Liang-Yan Gui. Interdiff: Generating 3d human-object interactions with physics-informed diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14928–14940, 2023.

[430] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. Transcenter: Transformers with dense queries for multiple-object tracking. *arXiv preprint arXiv:2103.15145*, 2021.

[431] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. Transcenter: Transformers with dense queries for multiple-object tracking. *arXiv preprint arXiv:2103.15145*, 2021.

[432] Yinzhen Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, et al. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4737–4746, 2023.

[433] Guorong Xuan, Wei Zhang, and Peiqi Chai. Em algorithms of gaussian mixture model and hidden markov model. In *Proceedings 2001 international conference on image processing (Cat. No. 01CH37205)*, volume 1, pages 145–148. IEEE, 2001.

[434] Bin Yan, Yi Jiang, Peize Sun, Dong Wang, Zehuan Yuan, Ping Luo, and Huchuan Lu. Towards grand unification of object tracking. In *ECCV*, 2022.

[435] Bin Yan, Yi Jiang, Peize Sun, Dong Wang, Zehuan Yuan, Ping Luo, and Huchuan Lu. Towards grand unification of object tracking. In *ECCV*. Springer, 2022.

[436] Sijie Yan, Zhizhong Li, Yuanjun Xiong, Huahan Yan, and Dahua Lin. Convolutional sequence generation for skeleton-based action synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4394–4402, 2019.

[437] Fan Yang, Xin Chang, Chenyu Dang, Ziqiang Zheng, Sakriani Sakti, Satoshi Nakamura, and Yang Wu. Remots: Self-supervised refining multi-object tracking and segmentation. *arXiv preprint arXiv:2007.03200*, 2020.

[438] Fan Yang, Xin Chang, Sakriani Sakti, Yang Wu, and Satoshi Nakamura. Remot: A model-agnostic refinement for multiple object tracking. *Image and Vision Computing*, 106:104091, 2021.

[439] Lixin Yang, Xinyu Zhan, Kailin Li, Wenqiang Xu, Jiefeng Li, and Cewu Lu. Cpf: Learning a contact potential field to model the hand-object interaction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11097–11106, 2021.

[440] Lixin Yang, Kailin Li, Xinyu Zhan, Fei Wu, Anran Xu, Liu Liu, and Cewu Lu. OakInk: A large-scale knowledge repository for understanding hand-object interaction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[441] Lixin Yang, Kailin Li, Xinyu Zhan, Fei Wu, Anran Xu, Liu Liu, and Cewu Lu. Oakink: A large-scale knowledge repository for understanding hand-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20953–20962, 2022.

[442] Heyuan Yao, Zhenhua Song, Baoquan Chen, and Libin Liu. Controlvae: Model-based learning of generative controllers for physics-based characters. *ACM Transactions on Graphics (TOG)*, 41(6):1–16, 2022.

[443] Vickie Ye, Georgios Pavlakos, Jitendra Malik, and Angjoo Kanazawa. Decoupling human and camera motion from videos in the wild. In *CVPR*, 2023.

[444] Yufei Ye, Abhinav Gupta, and Shubham Tulsiani. What's in your hands? 3d reconstruction of generic objects in hands. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3895–3905, 2022.

[445] Yufei Ye, Poorvi Hebbar, Abhinav Gupta, and Shubham Tulsiani. Diffusion-guided reconstruction of everyday hand-object interaction clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19717–19728, 2023.

[446] Yufei Ye, Xueting Li, Abhinav Gupta, Shalini De Mello, Stan Birchfield, Jiaming Song, Shubham Tulsiani, and Sifei Liu. Affordance diffusion: Synthesizing hand-object interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22479–22489, 2023.

[447] Yufei Ye, Abhinav Gupta, Kris Kitani, and Shubham Tulsiani. G-hop: Generative hand-object prior for interaction reconstruction and grasp synthesis. In *CVPR*, 2024.

[448] Yufei Ye, Abhinav Gupta, Kris Kitani, and Shubham Tulsiani. G-hop: Generative hand-object prior for interaction reconstruction and grasp synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1911–1920, 2024.

[449] Yuting Ye and C Karen Liu. Synthesis of detailed hand manipulations using contact sampling. *ACM TOG*, 31(4):1–10, 2012.

[450] Hongwei Yi, Justus Thies, Michael J Black, Xue Bin Peng, and Davis Rempe. Generating human interaction motions in scenes with text control. In *European Conference on Computer Vision*, pages 246–263. Springer, 2024.

[451] Xinyu Yi, Yuxiao Zhou, and Feng Xu. Transpose: Real-time 3d human translation and pose estimation with six inertial sensors. *ACM Transactions on Graphics*, 2021.

[452] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13–es, 2006.

[453] En Yu, Zhuoling Li, Shoudong Han, and Hongwei Wang. Relationtrack: Relation-aware multiple object tracking with decoupled representation. *arXiv preprint arXiv:2105.04322*, 2021.

[454] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2(5):6, 2018.

[455] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 1821–1830, 2017.

[456] Ye Yuan and Kris Kitani. 3D ego-pose estimation via imitation learning. In *Computer Vision – ECCV 2018*, volume 11220 LNCS, pages 763–778. Springer International Publishing, 2018.

[457] Ye Yuan and Kris Kitani. Ego-pose estimation and forecasting as real-time PD control. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-Octob:

10081–10091, 2019.

[458] Ye Yuan and Kris Kitani. Residual force control for agile human behavior imitation and extended motion synthesis. (NeurIPS), June 2020.

[459] Ye Yuan and Kris Kitani. Dlow: Diversifying latent flows for diverse human motion prediction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 346–364. Springer, 2020.

[460] Ye Yuan, Shih-En Wei, Tomas Simon, Kris Kitani, and Jason Saragih. SimPoE: Simulated character control for 3D human pose estimation. *CVPR*, April 2021.

[461] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9813–9823, 2021.

[462] Ye Yuan, Umar Iqbal, Pavlo Molchanov, Kris Kitani, and Jan Kautz. Glamr: Global occlusion-aware human mesh recovery with dynamic cameras. In *CVPR*, 2022.

[463] Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. PhysDiff: Physics-guided human motion diffusion model. *arXiv [cs.CV]*, December 2022.

[464] Andy Zeng, Shuran Song, Kuan-Ting Yu, Elliott Donlon, Francois R Hogan, Maria Bauza, Daolin Ma, Orion Taylor, Melody Liu, Eudald Romo, et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. *The International Journal of Robotics Research*, 41(7):690–705, 2022.

[465] Fangao Zeng and et al. Motr: End-to-end multiple-object tracking with transformer. In *ECCV*. Springer, 2022.

[466] Fangao Zeng, Bin Dong, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. *arXiv preprint arXiv:2105.03247*, 2021.

[467] Fangao Zeng, Bin Dong, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. *arXiv preprint arXiv:2105.03247*, 2021.

[468] Haotian Zhang, Ye Yuan, Viktor Makoviychuk, Yunrong Guo, Sanja Fidler, Xue Bin Peng, and Kayvon Fatahalian. Learning physically simulated tennis skills from broadcast videos. *ACM Trans. Graph.*, 42:1–14, 2023. ISSN 0730-0301,1557-7368.

[469] He Zhang, Yuting Ye, Takaaki Shiratori, and Taku Komura. Manipnet: neural manipulation synthesis with a hand-object spatial representation. *ACM TOG*, 40(4): 1–14, 2021.

[470] Hui Zhang, Sammy Christen, Zicong Fan, Otmar Hilliges, and Jie Song. Graspxl: Generating grasping motions for diverse objects at scale. *arXiv preprint arXiv:2403.19649*, 2024.

[471] Hui Zhang, Sammy Christen, Zicong Fan, Luocheng Zheng, Jemin Hwangbo, Jie Song, and Otmar Hilliges. ArtiGrasp: Physically plausible synthesis of bi-manual

dexterous grasping and articulation. In *Int. Conf. on 3D Vis.*, 2024.

[472] Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. T2m-gpt: Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[473] Mengqi Zhang, Yang Fu, Zheng Ding, Sifei Liu, Zhuowen Tu, and Xiaolong Wang. Hoidiffusion: Generating realistic 3d hand-object interaction data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8521–8531, 2024.

[474] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022.

[475] Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. In *arXiv preprint arXiv:2304.01116*, 2023.

[476] Mingyuan Zhang, Daisheng Jin, Chenyang Gu, Fangzhou Hong, Zhongang Cai, Jingfang Huang, Chongzhi Zhang, Xinying Guo, Lei Yang, Ying He, et al. Large motion model for unified multi-modal motion generation. In *ECCV*, 2024.

[477] Qinsheng Zhang, Jiaming Song, Xun Huang, Yongxin Chen, and Ming yu Liu. Diffcollage: Parallel generation of large content with diffusion models. In *CVPR*, 2023.

[478] Siwei Zhang, Bharat Lal Bhatnagar, Yuanlu Xu, Alexander Winkler, Petr Kadlecek, Siyu Tang, and Federica Bogo. Rohm: Robust human motion reconstruction via diffusion. In *CVPR*, 2024.

[479] Xiaohan Zhang, Bharat Lal Bhatnagar, Sebastian Starke, Vladimir Guzov, and Gerard Pons-Moll. Couch: Towards controllable human-chair interactions. In *European Conference on Computer Vision*, pages 518–535. Springer, 2022.

[480] Xiaohan Zhang, Bharat Lal Bhatnagar, Sebastian Starke, Ilya A. Petrov, Vladimir Guzov, Helisa Dhamo, Eduardo Pérez Pellitero, and Gerard Pons-Moll. Force: Dataset and method for intuitive physics guided human-object interaction. In *International Conference on 3D Vision (3DV)*, March 2025.

[481] Yaqi Zhang, Di Huang, Bin Liu, Shixiang Tang, Yan Lu, Lu Chen, Lei Bai, Qi Chu, Nenghai Yu, and Wanli Ouyang. Motiongpt: Finetuned llms are general-purpose motion generators. *arXiv preprint arXiv:2306.10900*, 2023.

[482] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *arXiv preprint arXiv:2004.01888*, 2020.

[483] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. *arXiv preprint arXiv:2110.06864*, 2021.

[484] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. *arXiv preprint arXiv:2110.06864*, 2021.

[485] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129(11):3069–3087, 2021.

[486] Yuang Zhang, Tiancai Wang, and Xiangyu Zhang. Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22056–22065, 2023.

[487] Yunbo Zhang, Alexander Clegg, Sehoon Ha, Greg Turk, and Yuting Ye. Learning to transfer in-hand manipulations using a greedy shape curriculum. In *Computer Graphics Forum*, volume 42, pages 25–36. Wiley Online Library, 2023.

[488] Yunbo Zhang, Deepak Gopinath, Yuting Ye, Jessica Hodgins, Greg Turk, and Jungdam Won. Simulation and retargeting of complex multi-character interactions. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023.

[489] Kaifeng Zhao, Shaofei Wang, Yan Zhang, Thabo Beeler, and Siyu Tang. Compositional human-scene interaction synthesis with semantic control. In *European Conference on Computer Vision*, pages 311–327. Springer, 2022.

[490] Kaifeng Zhao, Yan Zhang, Shaofei Wang, Thabo Beeler, and Siyu Tang. Synthesizing diverse human motions in 3d indoor scenes. *arXiv preprint arXiv:2305.12411*, 2023.

[491] Jianqiao Zheng, Sameera Ramasinghe, and Simon Lucey. Rethinking positional encoding. *arXiv preprint arXiv:2107.02561*, 2021.

[492] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12993–13000, 2020.

[493] Ellen D Zhong, Tristan Bepler, Joseph H Davis, and Bonnie Berger. Reconstructing continuous distributions of 3d protein structure from cryo-em images. *arXiv preprint arXiv:1909.05215*, 2019.

[494] Keyang Zhou, Bharat Lal Bhatnagar, Jan Eric Lenssen, and Gerard Pons-Moll. Toch: Spatio-temporal object-to-hand correspondence for motion refinement. In *ECCV*. Springer, October 2022.

[495] Keyang Zhou, Bharat Lal Bhatnagar, Jan Eric Lenssen, and Gerard Pons-Moll. Gears: Local geometry-aware hand-object interaction synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20634–20643, 2024.

[496] Wenyang Zhou, Zhiyang Dou, Zeyu Cao, Zhouyingcheng Liao, Jingbo Wang, Wenjia Wang, Yuan Liu, Taku Komura, Wenping Wang, and Lingjie Liu. Emdm: Efficient motion diffusion model for fast and high-quality motion generation. In *European Conference on Computer Vision*, pages 18–38. Springer, 2024.

[497] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 398–407, 2017.

[498] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[499] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[500] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision*, pages 474–490. Springer, 2020.

[501] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision*, pages 474–490. Springer, 2020.

[502] Xingyi Zhou, Tianwei Yin, Vladlen Koltun, and Philipp Krähenbühl. Global tracking transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8771–8780, 2022.

[503] Xingyi Zhou, Tianwei Yin, Vladlen Koltun, and Philipp Krähenbühl. Global tracking transformers. In *CVPR*, 2022.

[504] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew Dai, Zhifeng Chen, Quoc Le, and James Laudon. Mixture-of-experts with expert choice routing. February 2022.

[505] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:5738–5746, 2019.

[506] Zixiang Zhou and Baoyuan Wang. Ude: A unified driving engine for human motion generation. In *CVPR*, 2023.

[507] Lingting Zhu, Xian Liu, Xuanyu Liu, Rui Qian, Ziwei Liu, and Lequan Yu. Taming diffusion models for audio-driven co-speech gesture generation. In *CVPR*, 2023.

[508] Wentao Zhu, Xiaoxuan Ma, Dongwoo Ro, Hai Ci, Jinlu Zhang, Jiaxin Shi, Feng Gao, Qi Tian, and Yizhou Wang. Human motion generation: A survey. *TPAMI*, 2023.

[509] Yuliang Zou, Jimei Yang, Duygu Ceylan, Jianming Zhang, Federico Perazzi, and Jia-Bin Huang. Reducing footskate in human motion reconstruction with ground contact constraints. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, March 2020.