Towards 4D Perception with Foundational Priors

Tarasha Khurana

CMU-RI-TR-25-91

October 21, 2025

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

Thesis Committee:
Deva Ramanan, *chair*
Shubham Tulsiani
Katerina Fragkiadaki
Carl Vondrick, *Columbia University*
Leonidas Guibas, *Stanford University and Google*

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

To the little girl who saw the world through a peculiar lens.

Abstract

As humans, we are constantly interacting with and observing a three-dimensional dynamic world. Building this spatiotemporal or 4D understanding in vision algorithms is not straightforward as there is orders of magnitude less 4D data than 2D images and videos. This underscores the need to find meaningful ways to exploit 2D data to realize 4D tasks. Recent advancements in building "foundation models" – that have learnt generative/structural priors in a data-driven manner from internet-scale data – have allowed us access to these rich real-world priors for free. In this thesis, we investigate how one can tune these priors for 4D perception tasks like amodal tracking and completion, dynamic reconstruction and next-timestep prediction.

We pursue three complementary directions. First, in the absence of foundational priors, we build these ourselves in a self-supervised manner via the task of next-timestep prediction using sequences of 3D LiDAR sweeps of dynamic scenes. Importantly, we show that bottlenecking next-timestep prediction with a 4D representation is crucial. We find that such a forecasting model can be used for downstream motion planning for autonomous vehicles, which helps reduce collision rates to a large extent.

Second, we capitalize on foundational priors in a zero-shot manner. We turn to large reconstruction models that predict per pixel depth for images and videos. We use these to solve two underconstrained tasks – (1) tracking objects across occlusions in 2.5D, and (2) reconstructing dynamic scenes from sparse-views. In both settings, we find that one can do drastically better than prior state-of-the-art using additional scene cues in the form of data-driven depth priors.

Third, we exploit foundational priors via finetuning. We specifically look at video diffusion models and reformulate amodal perception and dynamic novel-view synthesis into self-supervised tasks that video models are good at i.e. inpainting. We find that it is surprisingly light-weight, in terms of data *and* compute, to finetune video diffusion models. This suggests that concepts similar to human visual perception are embedded in foundation models, which only have to be "controlled" to perform other tasks.

Together these contributions highlight how one can build, leverage and adapt foundational priors for spatiotemporal perception in a scalable manner – the scale is enabled by relying increasingly on internet-scale 2D data and carefully designing self-supervised objectives for learning.

Acknowledgments

First and foremost, I'm deeply grateful to my advisor, Deva Ramanan, for helping me grow into a better researcher. He unknowingly encouraged me to be curious, to dig deeper into details and *knowingly* taught me to articulate my work better. Deva is the most patient and supportive advisor anyone could hope for. My conversations with him were always intellectually stimulating, and I admired his way of understanding new literature by asking the right questions. One of the most important lessons I learnt from Deva was to "jump right into the fire" when things felt scary. I understood him more when I started mentoring other students, which made me realize how invested he was in all of his students' research. I am thankful I could get a year of learning to mentor from him (using his tactics). More than this, I am grateful to him for helping me find my voice, my beliefs and my individuality in research in the last six years.

I am thankful to Shubham Tulsiani, Katerina Fragkiadaki, Carl Vondrick and Leonidas Guibas for finding time and agreeing to serve on my committee. I strive to do good research like them and look up to their work ethics. Carl graciously found time for discussing ideas around some of my works, and Shubham threw some light on preparing for a potential academic career. They are all my role models.

I'm also indebted to my other mentors along the way. Simon Lucey served on my master's thesis committee, and since then, even five-minute chats with him at CVPRs have kept my hopes high for an entire year until the next one. James Hays was really fun to host Argoverse challenges with, and helped me in all ways he could. I was always inspired by the thought-provoking questions and suggestions David Held shared during the two years of our collaboration. I worked with Alireza Fathi and Cordelia Schmid during my internship and learned to push the boundaries of research fearlessly. Laura Herlant backed my exploration of ideas during another internship.

I am especially thankful to Chetan Arora, my undergrad research mentor, who I spent the first three years of my computer vision research journey with, from 2016 to 2019. He taught me the basics – how to read (and re-read) papers carefully and understand details that can help reproduce algorithms. He pushed me way outside my comfort zone and helped me be at CMU. I love seeing him at CVPRs.

Atul Rai was my manager and co-founder of a security and surveillance startup. He put my research skills to real-world use, fighting all odds of deploying systems in resource-constrained settings. He celebrates my success at CMU more than I do.

My high-school years at VBPS were the most formative parts of my life. My favorite

principal, late Veena Bhasin ma'am taught me the most powerful lesson when I was the head girl of primary school. She told me, "When you are at the mike, you have the authority". This fixed all of my public speaking and even before my defense, I remembered her advice for mental preparation. My Geography teacher, Vatsala Tiwari, taught me it was okay to say *no* to different things in life. Pratibha Kasturia made me love the English language, and her red pen edits on my essays help me today to write better research papers. I still use Anjali Razdan's advice on never leaving a sentence hanging at the end of a paragraph if it ends at less than half a linewidth. I have not named many other teachers at VBPS but time and again I have made sure they know they are special to me. I'm deeply thankful to all.

My Kathak and Hindustani music gurus, Dilip Saha and Kalpana Mandal gave me lifelong munitions of creativity & happiness, which I had to put to frequent use in the PhD. They have known me the longest and seen me grow since childhood.

I feel incredibly lucky to have had such wonderful friends and colleagues over the past six years — Achal Dave, Peiyun Hu, Aayush Bansal, Martin Li, Gengshan Yang, William Qi, Dinesh Reddy, Rawal Khirodkar, Adam Harley, Jason Zhang, Yufei Ye, Zhiqiu Lin, Neehar Peri, Gautam Gare, Jonathon Luiten, Aljosa Osep, Kangle Deng, Nate Chodosh, Haithem Turki, Arun Vasudevan, Anish Madan, Poorvi Hebbar, Rakshanda Hassan, Unnat Jain, Anshika Gupta, Homanga Bharadwaj, Jay Karhade, Khiem Vuong, Nikhil Keetha, Jeff Tan, Cheng-yen Hseih, Kaihua Chen, Zihan Wang, Saswat Subhajyoti Mallick, Yiming Gong, Mehar Khurana, Abhikhya Tripathi, Gaurav Parmar, Ruihan Gao, Swaminathan Gurumurthy, Erica Weng, Bart Duisterhof, Ishan Khatri, Qianqian Wang, Ayush Tewari. Those at CMU truly made Pittsburgh feel like home. I apologize for missing many names.

My girl gang – Himangi Mittal, Sally Chen, Jenny Seidenschwarz, Rashmi Salamani, Maitri Jain, Vidhi Jain, Nupur Kumari, my mom – was extremely supportive and a great relief to talk to. I am constantly in awe of how strong, smart, compassionate and emotionally intelligent women are. We can do anything.

My acknowledgment would be incomplete without the mention of my husband, Akash Sharma, who was the only person to first-hand witness every hour of every day of this journey. I am thankful for his companionship – for making the bad days bearable, and the good days better! From him, I learn to do mathematically grounded research from first principles, write better code, adopt new literature with an open mind, type faster, be opinionated and be bold in trying new things. He complements me.

My final and the most important thanks would be to my immediate and extended family who provided me with an unbelievably strong foundation of love, love and

more love. My father, Parveen Khurana, has hardwired me to "enjoy" any and every situation, and I seriously mean *any*. I am now unfazed by anything. My mother, Abha Khurana, has checked on me every day and night and made sure that everything in my life was on track. I learnt from her to stand up for the right things and not lose ground! Although I will always remember my brother, Mehar Khurana, as a seven-year old, he amazes me today with his maturity and hard work. My brother and I are an embodiment of our parents' values and strength. I also have the most loving set of in-laws anyone could ask for. My extended family is truly an irreplacable pack of relatives and I could not imagine growing up without any of them.

Contents

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

Chapter 1

Introduction

1.1 Human perception and gaps in computer vision

Humans are able to safely interact with the three-dimensional world where things are constantly in motion. Consider a real-world driving scene in Figure 1.1. When negotiating at a speed bump as shown, the ego-car has to understand the dynamic structure of its environment – frequently occluding objects that reappear, distance of other vehicles from the ego-car, estimation of where each of these vehicles will move in the future, building a local map of the environment and so on.

Perhaps unsurprisingly, humans are able to navigate this complex scene with ease as they have developed a sense of spatiotemporal reasoning over time [267]. Specifically, humans use concepts like spatial cognition [58] and predictive coding [239] where a mental model of the world is maintained in a reference frame and observations coming from the world are constantly compared with predictions made by the mind using this mental model. In contrast, even state-of-the-art vision algorithms struggle to emulate this four-dimensional (4D: 3D space + time) reasoning, especially when faced with sparse, noisy, or incomplete sensory inputs.

1.2 Towards 4D perception with foundational priors

In this thesis, we want to take a step forward in enabling this spatiotemporal or four-dimensional (4D) reasoning in present day vision algorithms. The gold standard for building such an algorithm would be to exploit full supervision from four-dimensional groundtruth data and use physical models that govern the process of this data generation. This means that for a given video observation of a scene, we would need the 3D location of every pixel at every timestep. Upcoming class of sensors like 4D LiDARs [153], synthetic data sources like simulators [59] or game engines [119, 121], and classic multi-view capture setups [133, 206], are viable means of obtaining this data.

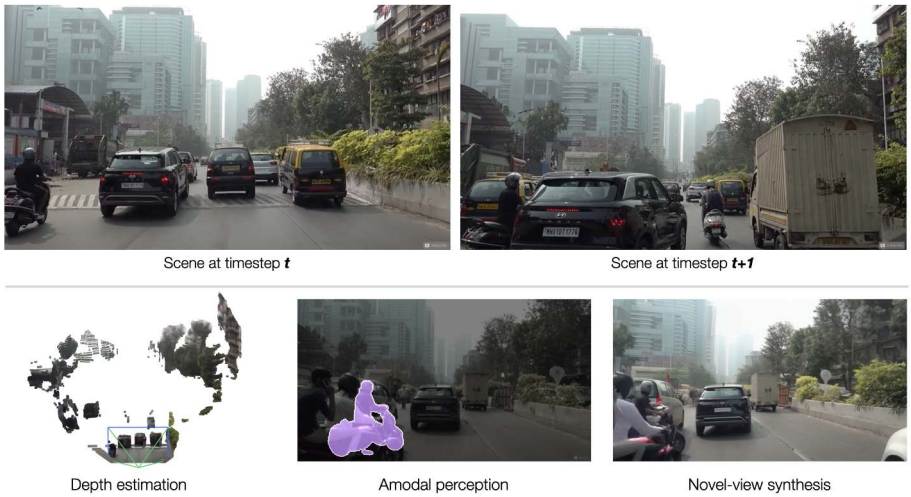


Figure 1.1. (**top**) We visualize an example of a real world driving scenario from the streets of Mumbai in India, by showing two successive frames from this scene. (**bottom**) For interacting safely with a complex dynamic world like this, we need to build the same spatiotemporal reasoning in vision algorithms that humans have developed over time – depth perception from a monocular video stream [29], gestalt psychology of observing partially visible objects but still understanding they are a whole [11, 317], and ability to map environments and understand its novel viewpoints [239].

Physical forward models like volume rendering from inverse graphics, and sensor models may be one way of also incorporating 4D priors into algorithms that learn from 4D data. Unfortunately, this approach is not scalable as 4D data is not widely available, and is rather cumbersome to curate. To put things into perspective, there is orders of magnitude difference between the 2D image/video, 3D and 4D data available today (c.f. Fig.1.2).

Foundation models In recent years, large foundation models have become popular because of how they are able to exploit and learn from internet-scale image [107] and video [21, 28] data which is the most abundatly available. As a result, these methods show that physical properties (such as 3D / 4D consistency) of the world "emerge" [28] upon foundational training, and are encoded in the models as "foundational priors". Foundation models have been developed for various discriminative and generative tasks.

For spatiotemporal perception, two classes of models are particularly relevant. *Monocular image and video depth estimation models* exploit vast unlabeled datasets and implicit geometric priors to infer dense scene structure from a single viewpoint, supporting applications in robotics, AR/VR, and autonomous driving. Likewise, *video diffusion models* extend diffusion-based

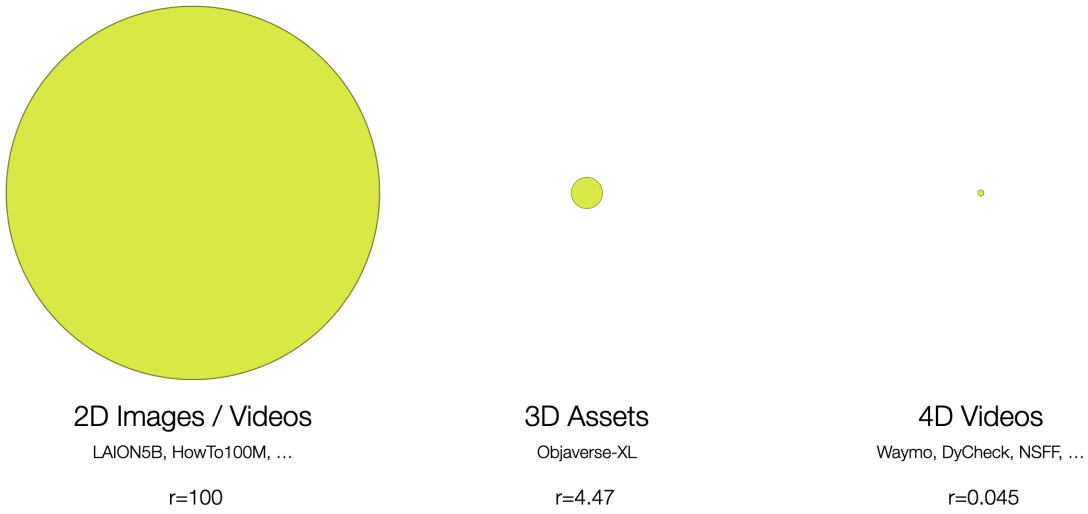


Figure 1.2. We visualize the magnitude of difference in the amount of 2D, 3D and 4D data available today, by logarithmically scaling the radius of each circle by the amount of data, normalized to a maximum range of 100. It is readily apparent that there is order of magnitudes difference in the scale of image/video, static 3D and dynamic 3D datasets that exist today. Building 4D models that learn only from 4D data is not a scalable approach. Instead, in this thesis we argue that one must utilize internet-scale 2D video data to realize 4D tasks.

generative frameworks to the temporal domain, producing high-fidelity and temporally coherent video predictions or completions from sparse or noisy inputs. Beyond these, foundation models encompass large vision–language architectures, Gaussian-splatting–based scene representations, and spatiotemporal transformers. Together, these models reveal that rich physical and structural properties of the world can emerge purely from data-driven training at internet scale, offering powerful priors that can be built, leveraged, and adapted for scalable 4D perception. My thesis builds upon this observation and is concretely organized into three main parts as shown in Fig. 1.3 which I also detail next.

1.3 Part I: Training in-house 4D models from scratch

In the absence of pretrained priors, we construct them ourselves using explicit 4D supervision from the largest such source of data – self-driving fleets.

- **Chapter 2:** *We introduce a self-supervised next-timestep prediction framework for sequences of 3D LiDAR sweeps of dynamic scenes.*

  Predicting how the world can evolve in the future is crucial for motion planning in au-

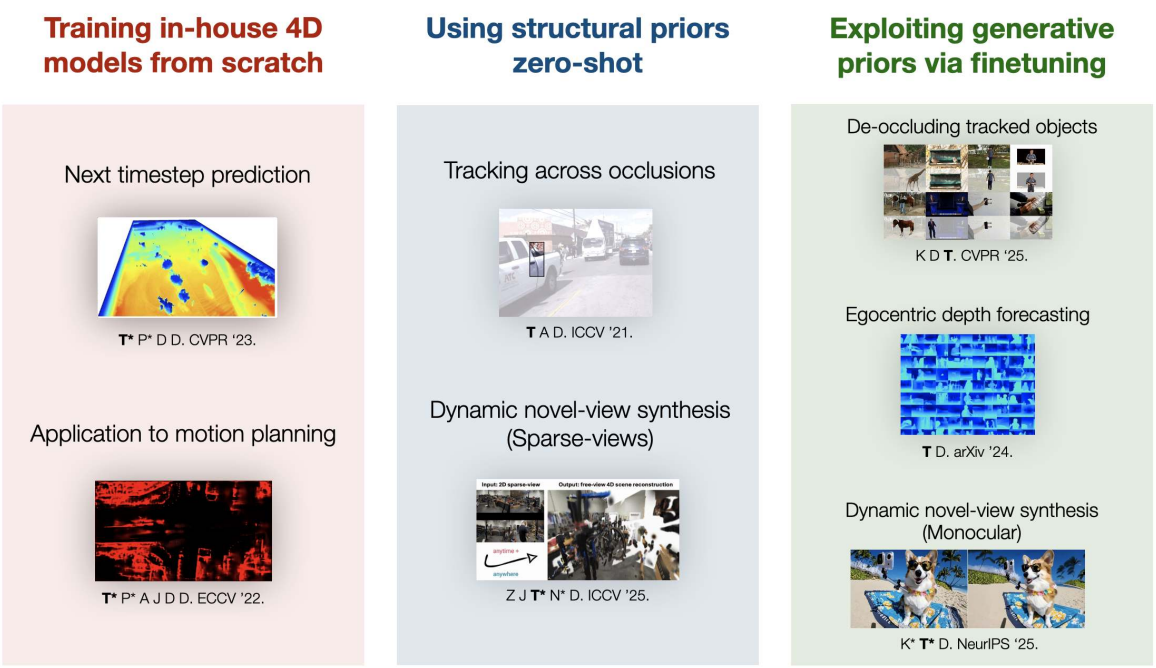


Figure 1.3. This thesis is divided into three main parts. (**left**) In the first part, we train foundation models from scratch using explicit 4D supervision. (**middle**) We then leverage large reconstruction models trained on internet-scale 2D data to solve underconstrained 4D tasks without finetuning. (**right**) Finally, we adapt foundation models more directly through lightweight finetuning.

tonomous systems. Classical methods are limited because they rely on costly human annotations in the form of semantic class labels, bounding boxes, and tracks or HD maps of cities to plan their motion — and thus are difficult to scale to large unlabeled datasets. One promising self-supervised task is 3D point cloud forecasting from unannotated LiDAR sequences. We show that this task requires algorithms to implicitly capture (1) sensor extrinsics (i.e., the egomotion of the autonomous vehicle), (2) sensor intrinsics (i.e., the sampling pattern specific to the particular LiDAR sensor), and (3) the shape and motion of other objects in the scene. But autonomous systems should make predictions about the world and not their sensors! To this end, we factor out (1) and (2) by recasting the task as one of spacetime (4D) occupancy forecasting. But because it is expensive to obtain ground-truth 4D occupancy, we "render" point cloud data from 4D occupancy predictions given sensor extrinsics and intrinsics, allowing one to train and test occupancy algorithms with unannotated LiDAR sequences. This also allows one to evaluate and compare point cloud forecasting algorithms across diverse datasets, sensors, and vehicles.

- **Chapter 3:** *We apply these forecasts to downstream motion planning for autonomous vehicles, showing that such learned representations substantially reduce collision rates.*

  Motion planning for safe autonomous driving requires learning how the environment around an ego-vehicle evolves with time. Ego-centric perception of driveable regions in a scene not only changes with the motion of actors in the environment, but also with the movement of the ego-vehicle itself. Self-supervised representations proposed for large-scale planning, such as ego-centric freespace, confound these two motions, making the representation difficult to use for downstream motion planners. In this paper, we use *geometric occupancy* as a natural alternative to view-dependent representations such as freespace. Occupancy maps naturally disentagle the motion of the environment from the motion of the ego-vehicle. However, one cannot directly observe the full 3D occupancy of a scene (due to occlusion), making it difficult to use as a signal for learning. Our key insight is to use *differentiable raycasting* to "render" future occupancy predictions into future LiDAR sweep predictions, which can be compared with ground-truth sweeps for self-supervised learning. The use of differentiable raycasting allows occupancy to *emerge* as an internal representation within the forecasting network. In the absence of groundtruth occupancy, we quantitatively evaluate the forecasting of raycasted LiDAR sweeps and show improvements of upto 15 F1 points. For downstream motion planners, where emergent occupancy can be directly used to guide non-driveable regions, this representation relatively reduces the number of collisions with objects by up to 17% as compared to freespace-centric motion planners.

Overall, we find that enforcing a 4D representation as a bottleneck that can disentangle scene motion from camera motion is essential for robust forecasting.

1.4 Part II: Using foundational priors zero-shot

Here, we leverage large reconstruction models trained on internet-scale 2D data to solve under-constrained 4D tasks without finetuning.

- **Chapter 4:** *We use monocular image and video depth estimation models to track objects across occlusions in 2.5D, exploiting their learned geometric priors.*

  Monocular object detection and tracking have improved drastically in recent years, but rely on a key assumption: that objects are visible to the camera. Many offline tracking approaches reason about occluded objects *post-hoc*, by linking together tracklets after the object re-appears, making use of reidentification (ReID). However, online tracking

in embodied robotic agents (such as a self-driving vehicle) fundamentally requires object permanence, which is the ability to reason about occluded objects *before* they re-appear. In this work, we re-purpose tracking benchmarks and propose new metrics for the task of detecting invisible objects, focusing on the illustrative case of people. We demonstrate that current detection and tracking systems perform dramatically worse on this task. We introduce two key innovations to recover much of this performance drop. We treat occluded object detection in temporal sequences as a short-term forecasting challenge, bringing to bear tools from dynamic sequence prediction. Second, we build dynamic models that explicitly reason in 3D from monocular videos without calibration, using observations produced by monocular depth estimators. To our knowledge, ours is the first work to demonstrate the effectiveness of monocular depth estimation for the task of tracking and detecting occluded objects. Our approach strongly improves by 11.4% over the baseline in ablations and by 5.0% over the state-of-the-art in F1 score.

- **Chapter 5:** *We extend this to reconstructing dynamic scenes and novel-view synthesis from sparse inputs.*

  We address the problem of dynamic scene reconstruction from sparse-view videos. Prior work often requires dense multi-view captures with hundreds of calibrated cameras (e.g. Panoptic Studio). Such multi-view setups are prohibitively expensive to build and cannot capture diverse scenes in-the-wild. In contrast, we aim to reconstruct dynamic human behaviors, such as repairing a bike or dancing, from a small set of sparse-view cameras with complete scene coverage (e.g. four equidistant inward-facing static cameras). We find that dense multi-view reconstruction methods struggle to adapt to this sparse-view setup due to limited overlap between viewpoints. To address these limitations, we carefully align independent monocular reconstructions of each camera to produce time- and view-consistent dynamic scene reconstructions. Extensive experiments on PanopticStudio and Ego-Exo4D demonstrate that our method achieves higher quality reconstructions than prior art, particularly when rendering novel views.

For both tasks, we find that one can do drastically better than prior state-of-the-art using additional scene cues in the form of data-driven depth priors.

1.5 Part III: Exploiting foundational priors via finetuning

In the final part, we adapt foundation models more directly through lightweight finetuning.

- **Chapter 6:** *We reformulate amodal perception as a video inpainting task, using video diffusion models to segment and complete about occluded dynamic objects.*

  Object permanence in humans is a fundamental cue that helps in understanding persistence of objects, even when they are fully occluded in the scene. Present day methods in object segmentation do not account for this *amodal* nature of the world, and only work for segmentation of visible or *modal* objects. Few amodal methods exist; single-image segmentation methods cannot handle high-levels of occlusions which are better inferred using temporal information, and multi-frame methods have focused solely on segmenting rigid objects. To this end, we propose to tackle video amodal segmentation by formulating it as a conditional generation task, capitalizing on the foundational knowledge in video generative models. Our method is simple; we repurpose these models to condition on a sequence of modal mask frames of an object along with contextual pseudo-depth maps, to learn which object boundary may be occluded and therefore, extended to hallucinate the complete extent of an object. This is followed by a content completion stage which is able to inpaint the occluded regions of an object. We benchmark our approach alongside a wide array of state-of-the-art methods on four datasets and show a dramatic improvement of upto 13% for amodal segmentation in an object's occluded region.

- **Chapter 7:** *We finetune a diffusion model to generate egocentric depth sequences of dynamic scenes auto-regressively.*

  Our work explores the task of generating future sensor observations conditioned on the past. We are motivated by 'predictive coding' concepts from neuroscience as well as robotic applications such as self-driving vehicles. Predictive video modeling is challenging because the future may be multi-modal and learning at scale remains computationally expensive for video processing. To address both challenges, our key insight is to leverage the large-scale pretraining of image diffusion models which can handle multi-modality. We repurpose image models for video prediction by conditioning on new frame timestamps. Such models can be trained with videos of both static and dynamic scenes. To allow them to be trained with modestly-sized datasets, we introduce invariances by factoring out illumination and texture by forcing the model to predict (pseudo) depth, readily obtained for in-the-wild videos via off-the-shelf monocular depth networks. In fact, we show that simply modifying networks to predict grayscale pixels already improves the accuracy of video prediction. Given the extra controllability with timestamp conditioning, we propose sampling schedules that work

better than the traditional autoregressive and hierarchical sampling strategies. Motivated by probabilistic metrics from the object forecasting literature, we create a benchmark for video prediction on a diverse set of videos spanning indoor and outdoor scenes and a large vocabulary of objects. Our experiments illustrate the effectiveness of learning to condition on timestamps, and show the importance of predicting the future with invariant modalities.

- **Chapter 8:** *We reformulate dynamic novel-view synthesis as a structured inpainting task, and finetune a video diffusion model for this inpainting.*

  We explore novel-view synthesis for dynamic scenes from monocular videos. Prior approaches rely on costly test-time optimization of 4D representations or do not preserve scene geometry when trained in a feed-forward manner. Our approach is based on three key insights: (1) *covisible* pixels (that are visible in both the input and target views) can be rendered by first reconstructing the dynamic 3D scene and rendering the reconstruction from the novel-views and (2) *hidden* pixels in novel views can be "inpainted" with feed-forward 2D video diffusion models. Notably, our video inpainting diffusion model (CogNVS) can be self-supervised from 2D videos, allowing us to train it on a large corpus of in-the-wild videos. This in turn allows for (3) CogNVS to be applied zero-shot to novel test videos via *test-time finetuning*. We empirically verify that CogNVS outperforms almost all prior art for novel-view synthesis of dynamic scenes from monocular videos.

In summary, we find that it is surprisingly light-weight, in terms of data *and* compute, to finetune video diffusion models. This suggests that concepts similar to human visual perception are embedded in foundation models, which only have to be "controlled" to perform other tasks.

Together these contributions highlight how one can build, leverage and adapt foundational priors for spatiotemporal perception in a scalable manner – the scale is enabled by relying increasingly on internet-scale 2D data and carefully designing self-supervised quasi-4D objectives for learning.

Finally, in the last part we discuss the findings from this thesis and related future avenues for research in 4D perception. In order to keep the thesis short and scoped, I omit my prior [10, 52, 113] work on semantic understanding of the 4D world.

Part I

Training in-house 4D models from scratch

Chapter 2

Next timestep prediction for LiDAR scans of dynamic scenes

Publication information

Khurana, T., Hu, P., Held, D. and Ramanan, D., 2023. Point cloud forecasting as a proxy for 4d occupancy forecasting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1116-1124.

2.1 Introduction

Motion planning in a dynamic environment requires autonomous agents to predict the motion of other objects. Standard solutions consist of perceptual modules such as mapping, object detection, tracking, and trajectory forecasting. Such solutions often rely on human annotations in the form of HD maps of cities, or semantic class labels, bounding boxes, and object tracks, and therefore are difficult to scale to large unlabeled datasets. One promising *self-supervised* task is 3D point cloud forecasting [202, 315, 316, 318]. Since points appear where lasers from the sensor and scene intersect, the task of forecasting point clouds requires algorithms to implicitly capture (1) sensor extrinsics (*i.e.*, the ego-motion of the autonomous vehicle), (2) sensor intrinsics (*i.e.*, the sampling pattern specific to the LiDAR sensor), and (3) the shape and motion of other objects in the scene. This task can be non-trivial even in a static scene (Fig. 2.1). We argue that autonomous systems should focus on making predictions about the world and not themselves, since an ego-vehicle has

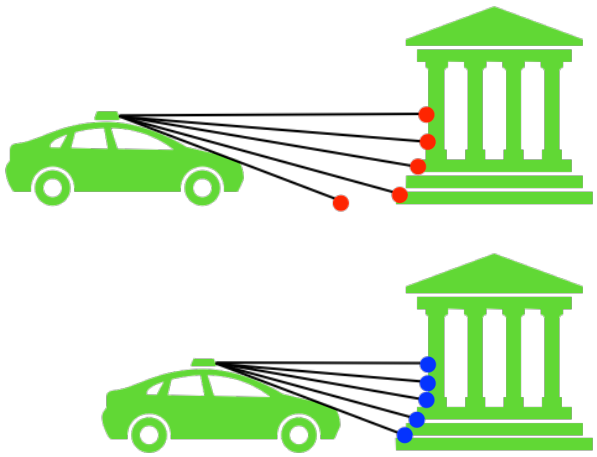


Figure 2.1. Points depend on the intersection of rays from the depth sensor and the environment. Therefore, accurately predicting points requires accurately predicting sensor extrinsics (sensor egomotion) and intrinsics (ray sampling pattern). But we want to understand dynamics of the environment, not our LiDAR sensor!

access to its future motion plans (extrinsics) and calibrated sensor parameters (intrinsics).

We factor out these (1) sensor extrinsics and (2) intrinsics by recasting the task of point cloud forecasting as one of spacetime (4D) occupancy forecasting. This disentangles and simplifies the formulation of point cloud forecasting, which now focuses solely on forecasting the central quantity of interest, the 4D occupancy. Because it is expensive to obtain ground-truth 4D occupancy, we "render" point cloud data from 4D occupancy predictions given sensor extrinsics and intrinsics. In some ways, our approach can be seen as the spacetime analog of novel-view synthesis from volumetric models such as NeRFs [206]; rather than rendering images by querying a volumetric model with rays from a known camera view, we render a LiDAR scan by querying a 4D model with rays from known sensor intrinsics and extrinsics. This allows one to train and test 4D occupancy forecasting algorithms with un-annotated LiDAR sequences. This also allows one to evaluate and compare point cloud forecasting algorithms across diverse datasets, sensors, and vehicles. We find that our approach to 4D occupancy forecasting, which can also render point clouds, performs drastically better than SOTAs in point cloud forecasting, both quantitatively (by up to 3.26m L1 error, Tab. 2.1) and qualitatively (Fig. 2.5). Our method beats prior art with zero-shot cross-sensor generalization (Tab. 2.2). To our knowledge, these are first results that generalize across train/test sensor rigs, illustrating the power of disentangling sensor motion from scene motion.

2.2 Related Work

Point Cloud Forecasting As one of the most promising self-supervised tasks that exploit unannotated LiDAR sequences, point cloud forecasting [202, 315, 316, 318] provides the algorithm past point clouds as input and asks it to predict future point clouds as output. Traditionally, both the input and the output are defined in the sensor coordinate frame, which moves with time. Although this simplifies preprocessing by eliminating the need for a local alignment, it forces the algorithm to implicitly capture (1) sensor extrinsics (i.e., the egomotion of the autonomous vehicle), (2) sensor intrinsics (i.e., the sampling pattern specific to the particular LiDAR sensor), and (3) the shape and motion of other objects in the scene. We argue that autonomous systems should make predictions about the world and not their sensors. In this paper, we reformulate point cloud forecasting by factoring out sensor extrinsics and intrinsics. Concretely, the new setup asks the algorithm to estimate the depth for rays from future timestamps. We show that one could use it as a proxy for training and testing 4D occupancy forecasting algorithms. Moreover, we demonstrate that one can evaluate existing point cloud forecasting methods under this setup, allowing 4D occupancy forecasting algorithms to be compared with point cloud forecasting algorithms.

Occupancy Forecasting Occupancy, as a predictive representation complementary to standard object-centric representations in the context of supporting downstream motion planning, has gained popularity over the last few years due to its efficiency in representing complex scenarios and interactions. Most existing works on occupancy forecasting focus on *semantic* occupancy grids from a bird's-eye view (BEV) [35, 198, 257]. They choose to focus on 2D for a good reason since most autonomous driving planners reason in a 2D BEV space. A downside is that it is expensive to obtain ground-truth *semantic* BEV occupancy for training and testing algorithms. [151] claim that if we reduce our goal from *semantic* occupancy to *geometric* occupancy, that is knowing if a location is occupied without asking which type of object is occupying it, one could learn to forecast *geometric* BEV occupancy from unannotated LiDAR sequences. In this paper, we take the idea from [151] and go beyond BEV – we propose an approach to learning to forecast 4D *geometric* occupancy from unannotated LiDAR sequences. We also propose a scalable evaluation to this task that admits standard point cloud forecasting methods.

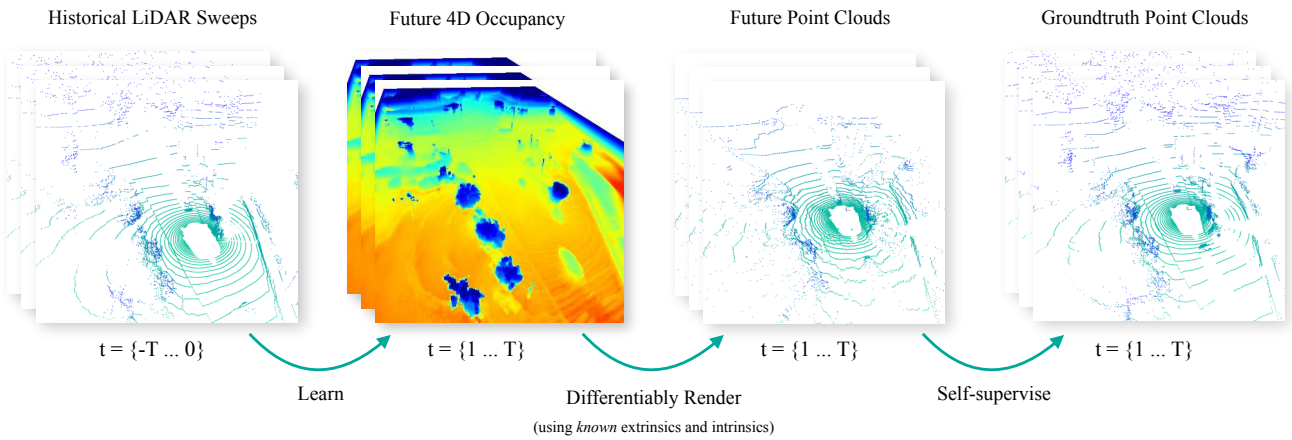


Figure 2.2. High-level overview of the approach we follow, closely inspired by a prior work [151]. Instead of directly predicting future point clouds by observing a set of historical point clouds, we take a geometric perspective on this problem and instead forecast a generic intermediate 3D occupancy-like quantity within a bounded volume. Known sensor extrinsics and intrinsics are an input to our method, which is different from how classical point cloud forecasting is formulated. We argue that this factorization is sensible as an autonomous agent plans its own motion and has access to sensor information. Please refer to our supplement for architectural details.

Novel View Synthesis We have seen tremendous progress in novel view synthesis in the last few years [190, 206, 211]. At its core, the differentiable nature of volumetric rendering allows one to optimize the underlying 3D structure of the scene by fitting samples of observations with known sensor poses without explicit 3D supervision. Our work can be thought of as novel view synthesis, where we try to synthesize depth images from novel views at future timestamps. Thanks to motion sensors (e.g., IMU), one can assume that relative LiDAR pose among frames in a log can be reliably estimated. Our work also differs from common novel view synthesis literatures in a few important aspects: (a) we use an efficient feed-forward network to predict the spacetime occupancy volume instead of applying test-time optimization; (b) we optimize an explicit volumetric scene representation (i.e., occupancy grid) instead of an implicit neural scene representation; (c) our approach relies on shape and motion prior learned across diverse scenarios in order to predict what happens next instead of reconstructing based on samples only from a specific scenario.

2.3 Method

Autonomous fleets log an abundance of unannotated sequences of LiDAR point clouds $\mathbf{X}_{-T...T}$, where we also estimate the relative sensor location for each frame $\mathbf{o}_{-T:T}$. Suppose we split such a sequence into a historic part $\mathbf{X}_{-T:0}$ and $\mathbf{o}_{-T:0}$ and a future part $\mathbf{X}_{1:T}$ and $\mathbf{o}_{1:T}$.

Standard point cloud forecasting methods, denoted by function g, take the historical sequence of point clouds $\mathbf{X}_{-T:0}$ as input and try to predict the future sequence of point clouds $\hat{\mathbf{X}}_{1:T}$.

$$\hat{\mathbf{X}}_{1:T} = g(\mathbf{X}_{-T:0}) \tag{2.1}$$

To introduce our approach, we need to first re-parametrize a point from the future LiDAR point cloud, say $\mathbf{x} \in \mathbf{X}_t$ where $t = 1 \dots T$, as a ray that starts from the sensor location $\mathbf{o}_t$, travels along the direction $\mathbf{d}$, and reaches the end point $\mathbf{x}$ after a distance of λ:

$$\mathbf{x} = \mathbf{o}_t + \lambda \mathbf{d}, \mathbf{x} \in \mathbf{X}_t \tag{2.2}$$

Conceptually, our approach, denoted by function f, takes a ray from a future timestamp t parametrized by its origin and direction $(\mathbf{o}_t, \mathbf{d})$, and tries to predict the distance $\hat{\lambda}$ the ray would travel, based on historic sequence of point clouds $\mathbf{X}_{-T:0}$ and sensor locations $\mathbf{o}_{-T:0}$.

$$\hat{\lambda} = f(\mathbf{o}_t, \mathbf{d}; \mathbf{X}_{-T:0}, \mathbf{o}_{-T:0}) \tag{2.3}$$

Intuitively, Eq. (2.3) is similar to view synthesis in NERF [206] except we are computing expected depth rather than expected color. Below, we introduce how we formulate the differentiable volumetric rendering process and use it for learning to forecast 4D occupancy.

Spacetime (4D) occupancy We define spacetime occupancy as the occupied state of a 3D location at a particular time instance. We use $\mathbf{z}$ to denote the true spacetime occupancy, which may not be directly observable due to line-of-sight visibility constraints. Consider a bounded spatialtemporal 4D volume, $\mathcal{V}$, which is discretized into spacetime voxels $\mathbf{v}$. We can use

$$\mathbf{z}[\mathbf{v}] \in \{0, 1\}, \mathbf{v} = (x, y, z, t), \mathbf{v} \in \mathcal{V} \tag{2.4}$$

to represent the occupancy of voxel $\mathbf{v}$ in the spacetime voxel grid $\mathcal{V}$, which can be *occupied* (1) or *free* (0).

In practice, we learn an occupancy predictio network h (parametrized by $\mathbf{w}$) to predict

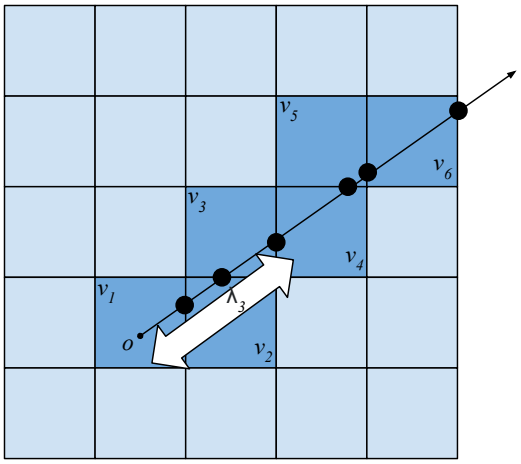


Figure 2.3. We illustrate the process of rendering depth for a given ray from the predicted occupancy grid. We assume that rays only stop at the voxel boundary, which discretizes the output space into a discrete set of events. We then compute the probability for a ray stopping at each boundary intersection. Finally, we compute the expected stopping distance.

discretized spacetime 4D occupancy given historic sequence of point clouds and sensor locations,

$$\hat{\mathbf{z}} = h(\mathbf{X}_{-T:0}, \mathbf{o}_{-T:0}; \mathbf{w}) \tag{2.5}$$

where

$$\hat{\mathbf{z}}[\mathbf{v}] \in \mathbb{R}_{[0,1]} \tag{2.6}$$

represents the predicted occupancy of voxel $\mathbf{v}$ in the spacetime voxel grid $\mathcal{V}$. Please refer to the supplementary materials for network architecture details.

Depth rendering from occupancy Given a ray query $\mathbf{x} = \mathbf{o} + \lambda\mathbf{d}$, our goal is to predict $\hat{\lambda}$ as close to λ as possible. We first compute how it intersects with the occupancy grid by voxel traversal [8] (Fig. 2.3). Suppose the ray intersects with a list of voxels $\{\mathbf{v}_1 \dots \mathbf{v}_n\}$. We discretize the ray space by assuming that a ray can only stop at voxel boundaries or infinity. We interpret occupancy of voxel $\mathbf{v}_i$ as the conditional probability that a ray leaving voxel $\mathbf{v}_{i-1}$ would stop in voxel $\mathbf{v}_i$. We can write

$$p_i = \prod_{j=1}^{i-1}(1 - \hat{\mathbf{z}}[\mathbf{v}_j])\hat{\mathbf{z}}[\mathbf{v}_i] \tag{2.7}$$

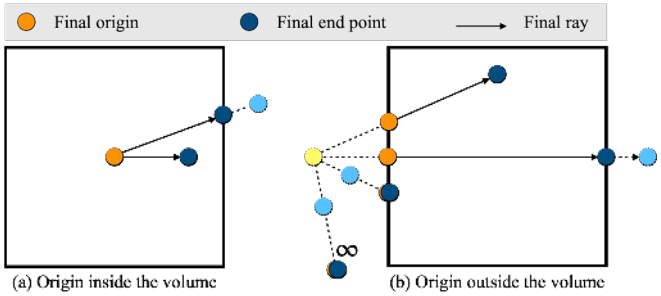


Figure 2.4. Ray Clamping. First, we move the origin towards the end point until the origin touches the volume or infinity. Then, we move the end point towards the origin until the end point touches the volume or infinity. At all times, we make sure the end point stays ahead of the origin (like two rings on a string). Being inside the volume counts as touching it.

where p_i represents the probability that a ray stops in voxel $\mathbf{v}_i$. Now we can render the distance by computing the stopping point in expectation.

$$\hat{\lambda} = f(\mathbf{o}, \mathbf{d}) = \sum_{i=1}^{n} p_i \hat{\lambda}_i \tag{2.8}$$

where $\hat{\lambda}_i$ represents the stopping distance at voxel $\mathbf{v}_i$.

You may have noticed that Eq. (2.8) does not capture the case where the ray stops outside the voxel grid, where the stopping distance is ill-defined (it will stop at infinity). During training, we allow a virtual stopping point outside the grid at the ground-truth location, i.e.,

$$\hat{\lambda} = f(\mathbf{o}, \mathbf{d}) = \sum_{i=1}^{n} p_i \hat{\lambda}_i + \prod_{i=1}^{n} (1 - p_i) \hat{\lambda}_{n+1} \tag{2.9}$$

where $\hat{\lambda}_{n+1} = \lambda$.

Loss function We can train the occupancy prediction network with a simple L1 loss between the rendered depth $\hat{\lambda}$ and the ground-truth depth λ.

$$L(\mathbf{w}) = \sum_{(\mathbf{o},\lambda,\mathbf{d}) \in (X_{1:T}, \mathbf{o}_{1:T})} |\lambda - f(\mathbf{o}, \mathbf{d}; \mathbf{X}_{-T:0}, \mathbf{o}_{-T:0}, \mathbf{w})| \tag{2.10}$$

Evaluation

The golden standard for evaluating 4D occupancy forecasting would be to compare the predicted occupancy with the ground-truth, but because it is extremely expensive to obtain ground-truth

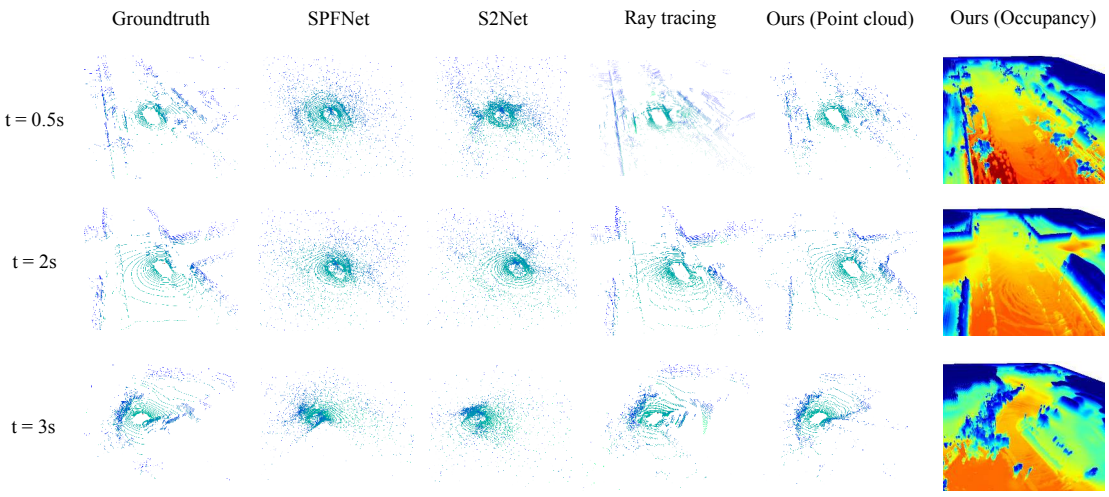


Figure 2.5. Qualitative results. We compare the point cloud forecasts of S2Net [316], SPFNet [315] and the raytracing baseline on the nuScenes dataset with our approach on three different sequences at different time horizon. Our forecasts look significantly crisper than the SOTA. This demonstrates the benefit of learning to forecast spacetime 4D occupancy with sensor intrinsics and extrinsics factored out. We also visualize the forecasted 4D occupancy at the corresponding future timestamp. As compared to simple *aggregation*-based raytracing, we are able to *spacetime-complete* 4D scenes. We highlight some potential applications in Fig. 2.6 and Fig. 2.7. We visualize a render of the predicted occupancy and the color encodes height along the z-axis.

4D occupancy, we "render" future point clouds from forecasted 4D occupancy with known sensor intrinsics and extrinsics, use the quality of rendered future point clouds as a proxy for that of forecasted 4D occupancy.

We introduce a new evaluation, where we factor out sensor intrinsics and extrinsics such that algorithms can be evaluated solely based on how well it captures how the scene unfolds. We provide future rays as queries and ask algorithms to provide a depth estimate for each query.

Given a query ray $\overrightarrow{OQ}$, there is a prediction ray $\overrightarrow{OP}$, where O represents the origin, Q represents the ground-truth end point, and P represents the predicted end point.

$$\overrightarrow{OQ} = \mathbf{o} + \lambda \mathbf{d} \tag{2.11}$$

$$\overrightarrow{OP} = \mathbf{o} + \hat{\lambda} \mathbf{d} \tag{2.12}$$

Given such a pair of rays, we define the error ε:

$$\varepsilon = |\overrightarrow{OQ} - \overrightarrow{OP}| = |\overrightarrow{PQ}| = |\lambda - \hat{\lambda}| \tag{2.13}$$

Near-field error Since LiDAR rays only travel through freespace and terminate when reaching occupied surface, there is a physical meaning behind the ε in Eq. (2.13). In practice, occupancy and freespace prediction is only relevant in regions that are reachable by the autonomous vehicle in planning's time hoziron. To reflect the focus on the reachable regions, we propose an operation to clamp any given ray $\overrightarrow{XY}$ to the fixed volume $\mathcal{V}$. We call it *ray clamping*, denoted as $\phi_{\mathcal{V}} : \overrightarrow{XY} \to \overrightarrow{X'Y'}$ and illustrated in Fig. 2.4.

We define the near-field (bounded by volume $\mathcal{V}$) prediction error $\varepsilon_{\mathcal{V}}$ as

$$\varepsilon_{\mathcal{V}} = |\phi_{\mathcal{V}}(\overrightarrow{OQ}) - \phi_{\mathcal{V}}(\overrightarrow{OP})| = |\overrightarrow{O'Q'} - \overrightarrow{O'P'}| = |\overrightarrow{P'Q'}| \tag{2.14}$$

Even though this metric penalizes disagreements of predicted depth along query rays within the bounded volume, it does not capture the severity of a prediction error. In real-world, one meter of an error close to the AV matters more. To this end, we also propose using a relative near-field prediction error $\varepsilon_{\mathcal{V}}^{rel}$ defined as,

$$\varepsilon_{\mathcal{V}}^{rel} = \frac{|\phi_{\mathcal{V}}(\overrightarrow{OQ}) - \phi_{\mathcal{V}}(\overrightarrow{OP})|}{|\overrightarrow{OQ}|} = \frac{|\overrightarrow{P'Q'}|}{|\overrightarrow{OQ}|} \tag{2.15}$$

The proposed evaluation requires one predicted ray for every ground-truth ray (query). Any algorithms that are capable of rendering depth for a given ray by design meets this requirement, including 4D occupancy forecasting from Sec. 2.3. However, for point cloud forecasting algorithms, the number of predicted points does not necessarily match the number of ground-truth rays, plus there is no one-to-one mapping between predicted and ground-truth points. To resolve this discrepancy, we propose to fit a surface to the predicted point clouds, on which we can query each ground-truth ray, find its intersection with the fitted surface, and output the (clamped) ray distance. In practice, we interpolate depth among the spherical projections of predicted rays.

We also consider vanilla chamfer distance d (2.16) and near-field chamfer distance $d_{\mathcal{V}}$ (2.17)

$$d = \frac{1}{2N} \sum_{\mathbf{x} \in \mathbf{X}} \min_{\hat{\mathbf{x}} \in \hat{\mathbf{X}}} ||\mathbf{x} - \hat{\mathbf{x}}||_2^2 + \frac{1}{2M} \sum_{\hat{\mathbf{x}} \in \hat{\mathbf{X}}} \min_{\mathbf{x} \in \mathbf{X}} ||\mathbf{x} - \hat{\mathbf{x}}||_2^2 \tag{2.16}$$

where $\mathbf{X}$, $\hat{\mathbf{X}}$ represents the ground-truth, predicted point cloud; N and M are their respective number of points.

$$d_{\mathcal{V}} = \frac{1}{2N'} \sum_{\mathbf{x} \in \hat{\mathbf{X}_{\mathcal{V}}}} \min_{\hat{\mathbf{x}} \in \mathbf{X}_{\mathcal{V}}} ||\mathbf{x} - \hat{\mathbf{x}}||_2^2 + \frac{1}{2M'} \sum_{\mathbf{x} \in \mathbf{X}_{\mathcal{V}}} \min_{\hat{\mathbf{x}} \in \hat{\mathbf{X}}_{\mathcal{V}}} ||\mathbf{x} - \hat{\mathbf{x}}||_2^2 \tag{2.17}$$

where $\mathbf{X}_{\mathcal{V}}$, $\hat{\mathbf{X}}_{\mathcal{V}}$ represents the ground-truth point cloud and predicted point cloud within the

bounding volume $\mathcal{V}$; M', N' are their respective number of points.

2.4 Experiments

Datasets We perform experiments on nuScenes [30], KITTI-Odometry [15, 91] and ArgoVerse2.0 [318]. nuScenes [30] is a full-suite autonomous driving dataset with a total of 1,000 real-world driving sequences of 15s each. KITTI [91] is also a multi-sensor dataset with 6 hours of diverse driving data across freeways and urban areas. KITTI-Odometry is a subset of this KITTI dataset where sequences have accurate sensor poses. ArgoVerse2.0 [318] contains the largest set of unannotated LiDAR sequences. Please see the supplementary material for results on ArgoVerse2.0.

Setup We consider a bounded area around the autonomous vehicle: -70m to 70m in the x-axis, -70m to 70m in the y-axis and -4.5m to 4.5m in the z-axis in the nuScenes coordinate system. This is our 4D volume $\mathcal{V}$, described in Sec. 2.3. We follow the state-of-the-art in point cloud forecasting and evaluate forecasting in a 1 second horizon and a 3 second horizon. We adopt the same setup as prior methods [315, 316]. On nuScenes, for 1s forecasting,, we take 2 frames of input and 2 frames of output at 2Hz; for 3s forecasting, we take 6 frames of input and 6 frames of output at 2Hz. For all other datasets, we always take 5 frames of input and 5 frames of output for both 1s and 3s forecasting.

Baselines First, we construct an aggregation-based raytracing baseline (similar to [202]). Specifically, we populate a binary occupancy grid given the aligned LiDAR point clouds from the past and present timesteps and use it for querying ground-truth rays. In addition to this, we compare our 4D occupancy forecasting approach to state-of-the-arts (SOTAs) in point cloud forecasting, including SPFNet [315] and S2Net [316] on the nuScenes dataset, and ST3DCNN [202] on the KITTI-Odometry dataset. For SPFNet [315] and S2Net [316], we are able to obtain the raw point cloud predictions from the authors and evaluate the results on the new metrics. For fair comparison, the S2Net results are based on a single sample from their VAE. For ST3DCNN [202], we retrain their models for 1s and 3s forecasting. In addition, the state-of-the-art approaches (barring ST3DCNN) tend to predict a confidence score for each point, indicating how valid the predicted point is; we evaluate the predicted point cloud both with and without confidence filtering, with a recommended confidence threshold at 0.05 [315, 316]. Quantitative and qualitative results with confidence filtering can be found in the supplement.

Method	Horizon	L1 (m)	AbsRel (%)	Chamfer Distance (m^2)	
				Near-field	Vanilla
S2Net [316]	1s	3.49	28.38	1.70	2.75
	3s	4.78	30.15	2.06	**3.47**
SPFNet [315]	1s	4.58	34.87	2.24	4.17
	3s	5.11	32.74	2.50	4.14
Ray tracing	1s	1.50	14.73	**0.54**	**0.90**
	3s	2.44	26.86	1.66	3.59
Ours	1s	**1.40**	**10.37**	1.41	2.81
	3s	**1.71**	**13.48**	**1.40**	4.31

Table 2.1. Results on nuScenes [30]. We see that the conclusions made from the proposed metrics are more in line with the qualitative results in Fig. 2.5. This reiterates the need for metrics that intuitively evaluate the underlying *geometry* of the scene instead of uncorrelated samples of the scene (e.g., points in space).

Qualitative results on nuScenes We compare the forecasted point clouds from our 4D occupancy forecasting approach to SOTA on point cloud forecasting in Fig. 2.5, where we see a drastic difference in how the predicted point clouds look like. Our forecasts look significantly more representative of the scene geometry compared to SOTA. This demonstrates the benefit of learning to forecast spacetime 4D occupancy with sensor intrinsics and extrinsics factored out. Surprisingly, we find that aggregation-based raytracing is a competitive baseline, qualitatively better than the SOTA. However, in addition to this *aggregation*, our approach is also able to hallucinate or *spacetime-complete* both the future motion of dynamic objects and the occluded parts of the static world. We also visualize the 3D forecasted occupancy at corresponding timestamps that our approach predicts "for free". Please refer to the caption for more details.

Results on nuScenes with new metrics We compare our 4D occupancy forecasting to SOTA on point cloud forecasting in terms of depth error along the future rays, following the evaluation protocol outlined in Sec. 2.3. We find that the 4D occupancy forecasting approach outperforms all baselines by significant margins in both 1s and 3s forecasting, reducing both the L1 and the absolute relative error by more than half, compared to the state-of-the-art methods on point cloud forecasting. The improvements here are consistent with the qualitative results in Fig. 2.5. As noted before, the raytracing baseline performs better than SOTA.

Results on nuScenes with old metrics We also evaluate by both vanilla (2.16) and near-field chamfer distance (2.17) following the protocol in Sec. 2.3. Our approach shines in terms of

Method	Train set	Horizon	L1 (m)	AbsRel (%)	Chamfer Dist. (m^2) Near-field	Vanilla
ST3DCNN [202]	KITTI-O	1s	3.13	26.94	4.11	4.51
		3s	3.25	28.58	4.19	4.83
Ours	KITTI-O	1s	**1.12**	**9.09**	**0.51**	**0.61**
		3s	**1.45**	**12.23**	**0.96**	**1.50**
Ray tracing	-	1s	**1.50**	16.15	**0.62**	**0.76**
		3s	2.82	29.67	**4.01**	5.92
Ours	AV2	1s	1.71	**14.85**	2.52	3.18
		3s	**2.52**	**23.87**	4.83	**5.79**
Ours	KITTI-O$^{20\%}$	1s	1.25	9.69	1.95	2.27
		3s	1.70	14.09	4.09	5.09
Ours	AV2 + KITTI-O$^{20\%}$	1s	**1.19**	**9.30**	**0.54**	**0.64**
		3s	**1.67**	**13.40**	**1.24**	**1.80**

Table 2.2. Performance as a function of the available target dataset (in this case, KITTI-Odometry). With access to all of KITTI-O (**top**), our method outperforms the SOTA. With no access to KITTI-O (*i.e.* zero-shot sensor generalization in the **middle**), our method trained on AV2 outperforms the ray tracing baseline at 3s, though the baseline fares well at 1s. Note that both approaches still beat the SOTA [202] by a large margin. Finally, with access to only 20% of KITTI-O (**bottom**), our method fares quite well, particularly when trained on both AV2 and KITTI-O. Cross-dataset generalization and training is made possible by disentangling sensor intrinsics/extrinsics from scene motion.

near-field chamfer distance. One contributing factor could be that our approach is specifically optimized for capturing occupancy evolution in the near field. In addition, S2Net [316] outperforms us in terms of vanilla chamfer distance, which is not surprising since we are incapable of deciding where rays end outside the predefined voxel grid.

Results on KITTI-Odometry Next, we use KITTI-Odometry to test our method in different settings with limited access to the target dataset. This mimics the setting where a next-generation sensor platform may be gradually integrated into fleet operations. Tab. 2.2 shows that with access to the full target dataset (KITTI-Odometry) for training, our method resoundingly outperforms the SOTA ST3DCNN [202]. Next, if no samples from the target dataset are available, one can employ either a non-learnable method such as our raytracing baseline, or one may pretrain on a (large) dataset with a different sensor platform. To this end, we find that our method trained on ArgoVerse2.0 outperforms the SOTA on KITTI-Odometry, while also outperforming raytracing baseline for long-horizon (3s) forecasting. Finally, with access to only 20% of KITTI-Odometry, our method pretrained on ArgoVerse2.0 and finetuned on KITTI-Odometry outperforms the

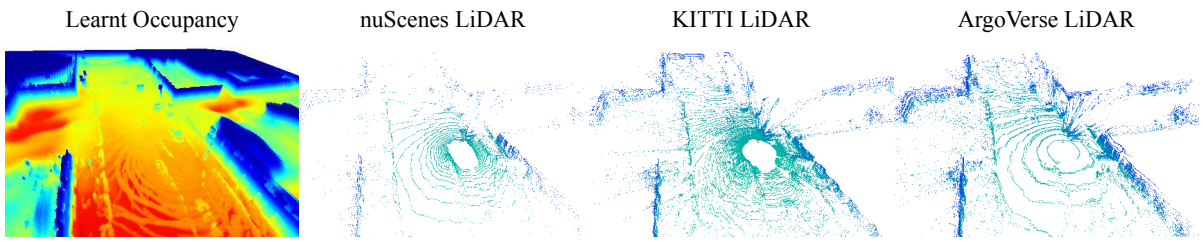


Figure 2.6. **Novel intrinsic-view synthesis** We show how to simulate different LiDAR ray patterns on top of the same learned occupancy grid. In this case, the future occupancy is predicted with historic LiDAR data scanned by nuScenes LiDAR (Velodyne HDL32E). First, we show the rendered point cloud under the native setting. Then, we show the rendered point cloud for KITTI LiDAR (Velodyne HDL64E, 2x as many beams). Finally, we have the rendered point cloud for Argoverse 2.0 LiDAR (2 VLP-32C stacked on top of each other). The fact that we can forecast occupancy on top of data captured by one type of sensor and use it to simulate future data for different sensors shows how generic the forecasted occupancy is as a representation. We support this generalization quantitatively in Tab. 2.2.

Arch.	Horizon	L1 (m)	AbsRel (%)	Chamfer Distance (m^2)	
				Near-field	Vanilla
S	1s	**1.28**	**9.27**	1.03	3.41
	3s	1.73	13.54	**1.40**	3.73
D	1s	1.40	10.37	1.41	**2.81**
	3s	**1.71**	**13.48**	**1.40**	4.31
S+R	1s	1.34	9.73	**1.00**	3.20
	3s	1.82	13.84	1.52	**3.54**

Table 2.3. We evaluate two variants of the proposed dynamic (D) architecture using the geometry forecasting metrics - static (S) and residual (S+R). We find that the static variant is a powerful baseline that beats our dynamic approach for 1s forecasting and by extension, the state-of-the-art.

alternatives. *To our knowledge, these are the first results in sensor transfer/generalization that illustrate the power of disentangling sensor extrinsics/intrinsics from scene motion.* Please see qualitative results in the supplement.

Architecture ablations

Here, we explore two other variants of our architecture: a *static* variant that predicts a single voxel grid for all future timesteps, and a *residual* variant that predicts a single static voxel grid with residual voxel grids for each output timestep. We evaluate these variants on nuScenes.

The main observation is that the static variant is a powerful baseline for short-horizon forecasting. This is because a single voxel grid serves as a dense static map of the local region, and since an extremely high majority of the world remains static, this is expected to be a reasonable

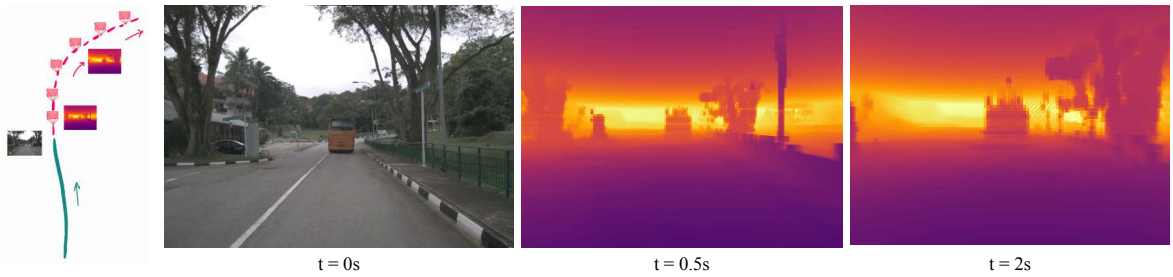


Figure 2.7. **Novel extrinsic-view synthesis** Dense depth maps rendered from the predicted future 4D occupancy from novel viewpoints. To render these depth maps, we take a novel future trajectory of the egovehicle. Placing the camera at each of these locations, always facing forward into the voxel grid (shown in the future dotted red trajectory on the left), gives us a camera coordinate system in which we can shoot rays from the camera center to every pixel in the image, and further beyond into the 4D occupancy volume. Every pixel represents the expected depth along its ray. The RGB image at $t = 0s$ is shown as reference and is not used in this rendering. For the depth maps, darker is closer, brighter is farther. Depth on sky regions is untrustworthy as no returns are received for this region from the LiDAR sensor.

baseline for short-horizon forecasting. Note that this variant is still stronger than the ray tracing baseline in Tab. 2.1 because of its ability to hallucinate occluded parts of the world. On the other hand, the proposed *dynamic* variant (which predicts one voxel grid per future timestep), performs the best at long-horizon forecasting. With the residual variant, our hope was to separate dynamic scene elements from static regions, but in practice this decomposition fails as there is not enough regularization to force motion-based separation.

Since, the static variant outperforms the state-of-the-art on 1s forecasting, we analyse these variants further in the supplement by using the segmentation annotations on nuScenes-LiDARSeg [1] and computing the proposed metrics separately on foreground and background points. This helps us understand which regions in the scene contribute the least to the performance of these variants.

Applications

Generalization across sensors In Fig. 2.6 (captioned as new intrinsic-view synthesis), we show how one can render point clouds as if they are captured by different LiDAR sensors from the same predicted future occupancy. Typically, different LiDAR sensors exhibit different ray patterns when sensing. For the case shown, the nuScenes LiDAR is an "in-domain" sensor, i.e., the occupancy grid was predicted by a network learned over LiDAR sweeps captured by a nuScenes LiDAR. The KITTI and ArgoVerse LiDARs are "out-of-domain". We hope that learning of such

a generic representation allows methods in sensor domain transfer [163, 348] to look at the task from the perspective of spacetime 4D occupancy. The formulation we have laid out also makes it easy to train across different datasets, making zero-shot cross-dataset transfer possible for LiDARs [237, 238]. In the previous section and in Tab. 2.2, we highlight the first result in this direction, where our method trained on the ArgoVerse2.0 dataset when tested on KITTI-Odometry beats the prior art [202] on KITTI-Odometry. Furthermore, our proposed disentangling also allows for multi-dataset training, for which we point the readers to the supplement.

Novel view synthesis In Fig. 2.7 (captioned as new-extrinsic view synthesis), we show dense depth maps rendered from our learnt occupancy grid using novel ego-vehicle trajectories or viewpoints. Such dense depth of a scene is not possible to get from existing LiDAR sensors that return sparse observations of the world. Although classical depth completion [290] from sparse LiDAR input exists as a single-frame (current timestep) task, here we note that with our representation, it is possible to densify sparse LiDAR point clouds from the *future*, with such rendered depth maps backprojected into 3D. This dense 360° depth is evaluated on sparse points (with the help of future LiDAR returns) by our proposed ray-based evaluation metrics.

2.5 Discussion

In this paper, we propose looking at point cloud forecasting through the lens of geometric occupancy forecasting, which is an emerging self-supervised task [151], originally set in the birds'-eye-view but extended to full 3D through this work. We advocate that this shift in viewpoint is necessary for two reasons. First, this shift helps algorithms focus on a generic intermediate representation of the world, i.e. its spatiotemporal 4D occupancy, which has great potential for downstream tasks. Second, this "renovates" how we formulate self-supervised LiDAR point cloud forecasting [202, 315, 316] by factoring out sensor extrinsics and intrinsics from the learning of shape and motion of different scene elements. In the end, we reiterate that the two tasks in discussion are surprisingly connected. We propose an evaluation protocol, that unifies the two worlds and focuses on a scalable evaluation for predicted geometry.

2.6 Appendix

In this appendix, we extend our discussion of the proposed reformulation of point cloud forecasting into 4D occupancy forecasting. Specifically, we discuss details about the network architecture of

the proposed approach in Sec. 2.6, further results on nuScenes, KITTI-Odometry and ArgoVerse2.0 in Sec. 2.6, and the quality of our forecasts separately on the foreground and background points in Sec. 2.6. The code is available on GitHub, a video summary is available on YouTube, and an overview of our work with the video versions of all figures is available at this webpage.

Network details

Architecture implementation We build on top of the encoder-decoder architecture first proposed by Zeng *et al.* [358] for neural motion planning. We extend the version of this architecture used by Khurana *et al.* [151] for forecasting occupancy in the birds'-eye-view. The only difference between our setup and that used in prior work [151], is that we treat our 4D voxel grid (X x Y x Z x T) as a reshaped 3D voxel grid (X x Y x ZT), where the Z or height dimension is incorporated into the channel dimension of the input, allowing us to still make use of 2D convolutions on a 4D voxel grid. This means that every channel in the input, represents a slice of the world through the height and time dimensions.

Differentiable renderer We extend the differentiable raycaster developed by Khurana *et al.* [151] to 3D and employ it as the differentiable voxel renderer in our approach. As in the prior work, we define our set of rays using the position of the egovehicle in the global coordinate frame as the origin, and all the LiDAR returns as the end points for the rays. The 4D voxel grid is initialized with three labels - empty, occupied and unknown based on the returns in the LiDAR sweeps. Each ray is traversed using a fast voxel traversal algorithm [8]. Given all the voxels and their occupancies along a ray, we compute the expected distance the ray travels through the voxel grid. This is same as volume rendering but in a discretized grid [206]. The gradient of the loss between this expected distance and the groundtruth distance is backpropagated to all the voxels traversed by the ray. Note that when a ray does not terminate within the voxel grid volume, we put all the probability mass of occupancy at the boundary of the voxel grid, similar to Mildenhall *et al.* [206]. This means that when a ray passes through occupancy regions that are empty (refer occupancy visuals in the main draft and supplementary video), the rays results in a point at the boundary of the voxel grid.

Dataset training and testing splits We use the official train and validation splits of nuScenes and ArgoVerse2.0. Only when comparing results on KITTI-Odometry with ST3DCNN [202],

we follow their dataset splits for training and testing. These dataset splits allow us to draw apples-to-apples comparisons with state-of-the-art approaches.

Additional results

nuScenes

Results with confidence thresholding We supplement the results in the main paper by evaluating the point cloud forecasts of SPFNet [315] and S2Net [316] by thresholding points at a recommended confidence threshold of 0.05. Qualitatively in Fig. 2.8, we observe point clouds from SOTA that only consist of high confidence LiDAR returns close to the ground plane, because of which we perform quantitatively much better than these baselines on our ray-based metrics. We summarise these results in Tab. 2.4.

Access to ground-truth egoposes during evaluation Note that in our proposed formulation of 4D occupancy forecasting, we view the LiDAR point clouds used during training as just another observation of the world, which in our case, happens to come from the view of the ego-vehicle. In reality, this LiDAR measurement of occupancy could have also come from any other observer in the world. Similarly, during evaluation, the only LiDAR measurement we have access to comes from the view of the ego-vehicle, making this the only datapoint to evaluate our occupancy forecasts against. This creates an apparent advantage for our method when comparing to point cloud forecasting approaches because they do not have access to ground-truth egoposes from the future. To alleviate this concern, first, we use the future ground-truth egoposes to align all point cloud forecasts to a global coordinate frame. Only after doing this, all the reported metrics are computed for the baselines. Second, we employ a simple motion planner based on linear dynamics, and use these planned future egoposes for evaluating our own method. We see that the metrics drop marginally, showing that the dependence of our method on ground-truth egoposes from the future is not a concern. This is also true for the ray tracing baseline, results of which are summarised in Tab. 2.5.

KITTI-Odometry

Qualitative results We supplement the quantitative results in the main paper with qualitative results in Fig. 2.9. As noted before, the trends are similar to nuScenes.

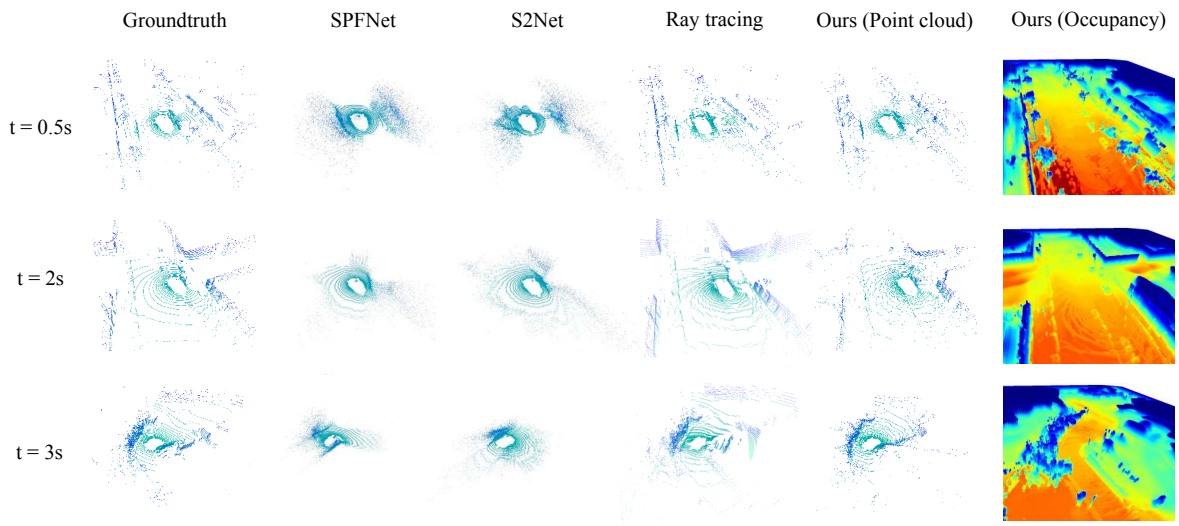


Figure 2.8. Qualitative results on the nuScenes dataset on three different sequences at different time horizons. We compare the point cloud forecasts of our approach with the aggregation-based ray tracing baseline and S2Net [316], SPFNet [315] with confidence filtering, after applying a recommended confidence threshold of 0.05 on the point clouds. Our forecasts look significantly crisper than the SOTA, however we see that the ray tracing baseline is also a strong baseline. The rendered occupancy is colored by height along z-axis.

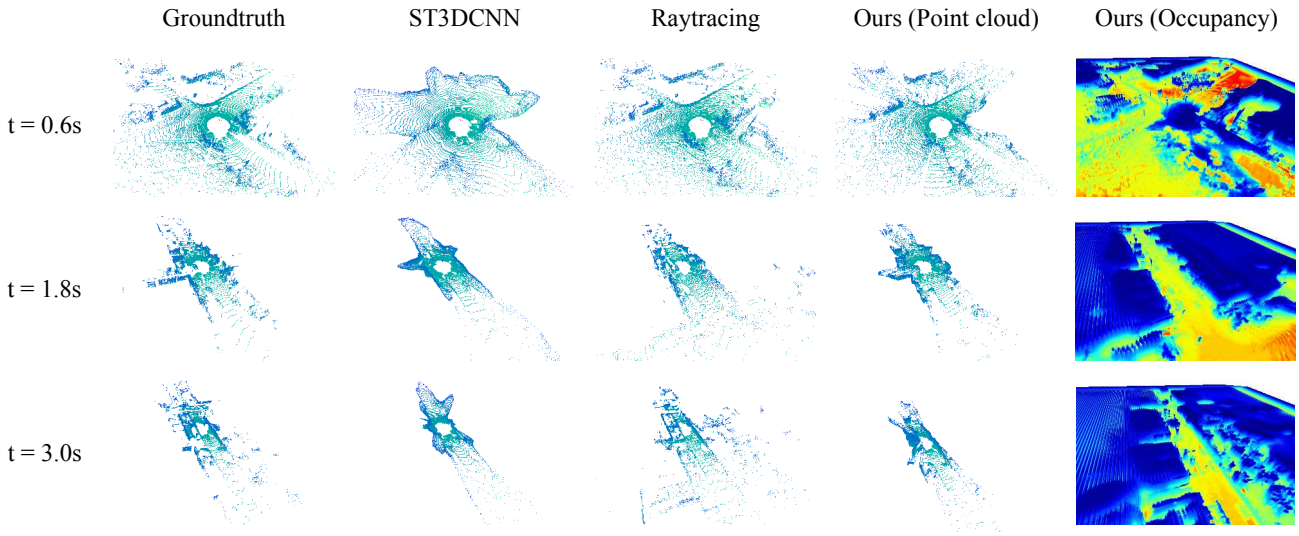


Figure 2.9. Qualitative results on KITTI-Odometry on three different sequences at different time horizons. We compare the point cloud forecasts of ST3DCNN [202] and the ray tracing baseline. We see that this SOTA is qualitatively more geometry-aware than the SOTA on nuScenes. However, our method is still more reflective of the true rigid geometry of the underlying world. The rendered occupancy is colored by height along z-axis.

Method	Horizon	L1 (m)	AbsRel (%)	Chamfer Distance (m^2)	
				Near-field	Vanilla
S2Net [316]	1s	2.88	20.57	4.61	11.77
	3s	4.97	24.79	13.10	30.95
SPFNet [315]	1s	5.30	30.12	21.24	45.12
	3s	5.70	28.65	20.99	44.71
Ray tracing	1s	1.50	14.73	0.54	**0.90**
	3s	2.44	26.86	1.66	**3.59**
Ours	1s	**1.40**	**10.37**	**1.41**	2.81
	3s	**1.71**	**13.48**	**1.40**	4.31

Table 2.4. Results on nuScenes [30] with confidence filtering on SPFNet and S2Net. As described in the main paper we threshold the points at a recommended confidence threshold of 0.05. We see that the conclusions made from the proposed metrics are more in line with the qualitative results in Fig. 2.8. This once again reiterates the need for metrics that intuitively evaluate the underlying *geometry* of the scene instead of uncorrelated samples of the scene (e.g., points in space).

Method	GT Egoposes	L1 (m)	AbsRel (%)	Chamfer Distance (m^2)	
				Near-field	Vanilla
Ray tracing	Yes	2.44	26.86	1.66	3.59
Ray tracing	No	2.50	26.35	1.60	**3.39**
Ours	Yes	**1.71**	**13.48**	**1.40**	4.31
Ours	No	1.84	13.95	1.50	4.50

Table 2.5. We experiment with using a simple linear dynamics based motion planner that can replace the ground-truth future egoposes used in our analysis. Our experiments prove that even in the absence of access to ground-truth future egoposes – which are not a concern from the viewpoint of our formulation but only a means to evaluate the occupancy predictions – simple linear dynamics models such as those based on constant velocity, suffice.

ArgoVerse2.0

We benchmark ArgoVerse2.0 in Tab. 2.6 and compare the ray tracing baseline to our method. We see that the ray tracing baseline is strong and performs better than our method in terms of the Chamfer distance. Yet, our method is able to forecast better scene geometry in the near-field, than the ray tracing baseline as suggested by the ray-based metrics.

Note that the high vanilla Chamfer distance for ArgoVerse2.0 is due to the fact that the its LiDAR is long-range (up to 200m) and we focus on points only withing the bounded volume. We also clarify that we always test cross-sensor generalization between KITTI-Odometry and ArgoVerse2.0, with training on ArgoVerse2.0 because (1) both datasets have the same number of LiDAR beams and point clouds are captured at the same frequency, and (2) ArgoVerse2.0 is a

Method	Horizon	L1 (m)	AbsRel (%)	Chamfer Distance (m^2)	
				Near-field	Vanilla
Ray tracing	1s	2.39	15.43	**0.56**	**1.90**
	3s	3.72	25.24	2.50	**11.59**
Ours	1s	**2.25**	**10.25**	1.53	60.94
	3s	**2.86**	**14.62**	**2.20**	69.81

Table 2.6. Quantitative results on the ArgoVerse2.0 [318] dataset. We compare our method trained on the ArgoVerse2.0 dataset to the ray tracing baseline and find similar trends to nuScenes and KITTI-Odometry.

Config	Horizon	Pedestrians				Vehicles				All foreground				Background			
		L1	AbsRel	Chamfer Dist. N.f.	Chamfer Dist. Vanilla	L1	AbsRel	Chamfer Dist. N.f.	Chamfer Dist. Vanilla	L1	AbsRel	Chamfer Dist. N.f.	Chamfer Dist. Vanilla	L1	AbsRel	Chamfer Dist. N.f.	Chamfer Dist. Vanilla
Ray tracing	1s	**6.09**	37.18	**61.30**	**66.79**	**3.53**	28.82	**16.92**	**21.19**	3.72	34.47	**16.09**	19.45	1.39	12.51	**0.49**	**0.85**
Ours	1s	6.43	**34.89**	79.63	68.60	3.61	**25.28**	21.47	22.59	**3.61**	**28.33**	18.19	**19.05**	**1.33**	**8.82**	1.44	3.02
Raytracing	3s	7.84	46.42	92.86	92.97	5.29	44.25	26.99	38.22	5.52	51.48	25.66	35.32	2.27	23.50	1.60	**3.48**
Ours	3s	**6.58**	**34.72**	**78.47**	**71.99**	**4.11**	**29.73**	**22.28**	**28.36**	**4.14**	**33.22**	**18.59**	**22.57**	**1.61**	**11.48**	**1.43**	4.63

Table 2.7. We extend our metrics analysis to pedestrians, vehicles, movable foreground and static background objects separately. Since, we are not able to compute these category-wise metrics for the state-of-the-art [315, 316] due to historical reasons, we do this for the ray tracing baseline. In summary, we find that our method outperforms this otherwise strong baseline at long-horizon forecasting. However, in the short-horizon the ray tracing baseline is a strong one; sometimes doing better and other times performing at par with our proposed method.

much larger and diverse dataset than KITTI-Odometry that is suitable for pretraining.

Foreground vs. background query rays

In order to further analyse the variants of our architecture, we separate the query rays as belonging to foreground or background regions, using the labels from nuScenes' LiDARSeg [1]. We evaluate both the regions using both the new and old metrics in Table 2.8 and Table 2.9.

Poor performance on foreground objects Our main observation is that all the variants perform poorly on the foreground objects (which includes moving or stationary foreground objects) as compared to the background. This is because a large number of rays and voxels (more than 90%) belong to background regions and thus, the foreground objects are downweighted during the training process. Even when the combined evaluation of foreground and background regions is considered (Table 3 in main draft), we see that the poor performance on the foreground fails to materialize in the metrics. This hints are improving the metrics and methods to focus more on

Arch.	Horizon	L1 (m)	AbsRel (%)	Chamfer Distance (m^2)	
				Near-field	Vanilla
S	1s	3.28	26.14	13.52	15.35
	3s	4.10	32.25	19.25	23.05
D	1s	3.61	28.33	18.19	19.05
	3s	4.14	33.22	18.59	22.57
S + R	1s	3.28	25.34	13.95	15.01
	3s	4.11	31.65	19.91	25.29

Table 2.8. Performance analysis on **foreground** query rays on nuScenes [30] for the different architecture variants introduced in the main draft, static (S), dynamic (D) and residual (S+R).

Arch.	Horizon	L1 (m)	AbsRel (%)	Chamfer Distance (m^2)	
				Near-field	Vanilla
S	1s	1.22	7.89	1.10	3.74
	3s	1.64	11.65	1.43	4.00
D	1s	1.33	8.82	1.44	3.02
	3s	1.61	11.48	1.43	4.63
S + R	1s	1.29	8.48	1.07	3.52
	3s	1.74	12.01	1.56	3.78

Table 2.9. Performance analysis on **background** query rays on nuScenes [30] for the different architecture variants introduced in the main draft, static (S), dynamic (D) and residual (S+R).

the forecasting of foreground objects, especially those in motion.

Strengths of each variant Another observation stemming from the above fact is that even with this disentangled evaluation on foreground, the *static* variant is the strongest baseline for short-horizon forecasting (1s). On 3s forecasting, the *dynamic* variant shines on the ray-based evaluation of background objects (some unseen background regions may only appear at future timesteps) and the *residual* variant shines on the ray-based evaluation of foreground objects (possibly decouples the foreground from background regions better).

Comparison to the ray tracing baseline Given the strength of the ray tracing baseline, we investigate its performance on foreground and background objects in comparison to our approach in Tab. 2.7. This time we further divide foreground objects into subcategories of pedestrains and vehicles, while also reporting the metrics on all foreground objects. Note that the according to the vocabulary of nuScenes, apart from different types of pedestrians and vehicles, miscellaneous movable objects like traffic cones and barriers are included in the umbrella category of foreground objects. We have the following findings:

1. For long-horizon forecasting, our method consistently does better than the ray tracing baseline, for both all types of foreground objects and background objects.
2. For short-horizon forecasting, the ray tracing baseline performs at par with our method and sometimes even better (on most types of foreground objects), hence proving to be a strong yet simple and non-learnable approach.

Chapter 3
Application to evasive motion planning

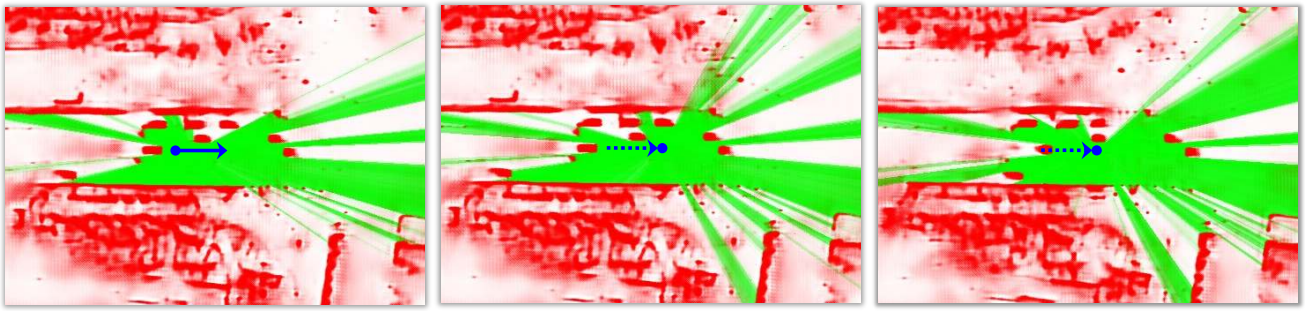
Publication information

Khurana, T., Hu, P., Dave, A., Ziglar, J., Held, D. and Ramanan, D., 2022, October. Differentiable raycasting for self-supervised occupancy forecasting. In European Conference on Computer Vision (ECCV) pp. 353-369. Cham: Springer Nature Switzerland.

3.1 Introduction

To navigate in complex and dynamic environments such as urban cores, autonomous vehicles need to perceive actors and predict their future movements. Such knowledge is often represented in some form of forecasted occupancy [257], which downstream motion planners rely on to produce safe trajectories. When tackling the tasks of perception and prediction, standard solutions consist of perceptual modules such as object detection, tracking, and trajectory forecasting, which require a massive amount of object track labels. Such solutions do not scale given the speed that log data is being collected by large fleets.

Freespace versus occupancy: To avoid the need for costly human annotations, and to enable learning at scale, self-supervised representations such as ego-centric freespace [117] have been proposed. However, such a representation couples the motion of the world with the motion of the ego-vehicle (Fig. 3.1). Our key innovation in this paper is to learn an ego-pose independent and explainable representation for safe motion planning, which we call *emergent occupancy*. Emergent occupancy decouples ego motion and scene motion using differentiable raycasting: we



(a) Sensor position y, Scene s (b) New sensor position $y + \Delta y$, Scene s (c) New sensor position $y + \Delta y$, New scene $s + \Delta s$

Figure 3.1. We propose **emergent occupancy** as a novel self-supervised representation for motion planning. Occupancy is independent of changes in sensor pose Δy, which is in contrast to prior work on self-supervised learning from LiDAR [117, 207, 313, 318], specifically, **ego-centric freespace** [117], which changes with (**a-b**) sensor pose motion Δy and (**b-c**) scene motion Δs. We use differentiable raycasting to naturally decouple ego motion from scene motion, allowing us to learn to forecast occupancy by self-supervision from pose-aligned LiDAR sweeps.

design a network that learns to "space-time complete" the future volumetric state of the world (in a world-coordinate frame) given past LiDAR observations. Consider an ego-vehicle that moves in a static scene. Here, LiDAR returns (even when aligned to a world-coordinate frame) will still *swim* along the surfaces of the fixed scene (Fig. 3.2). This implies that even when the world is static, most of what the ego-vehicle observes through the LiDAR sensor appears to move with complex nonlinear motion, but in fact those observations can be fully explained by static geometry and ego-motion (via raycasting). LiDAR forecasters need to implicitly predict this ego-motion of the car to produce accurate future returns. However, we argue that such prediction doesn't make sense for autonomous agents that *plan* their future motion. Importantly, our differentiable raycasting network has access to future camera ego-poses as *input*, both during training (since they are available in archival logs) and testing (since state-of-the-art planners explicitly search over candidate trajectories).

Self-supervision: Note that ground-truth future volumetric occupancy is largely unavailable without human supervision, because the full 3D world is rarely observed; the ego-vehicle only sees a limited number of future views as recorded in a single archival log. To this end, we apply a differentiable raycaster that projects the forecasted volumetric occupancy into a LiDAR sweep, as seen by the future ego-vehicle motion in the log. We then use the difference between the raycasted sweep and actual sweep as a signal for self-supervised learning, allowing us to train models on massive amounts of unannotated logs.

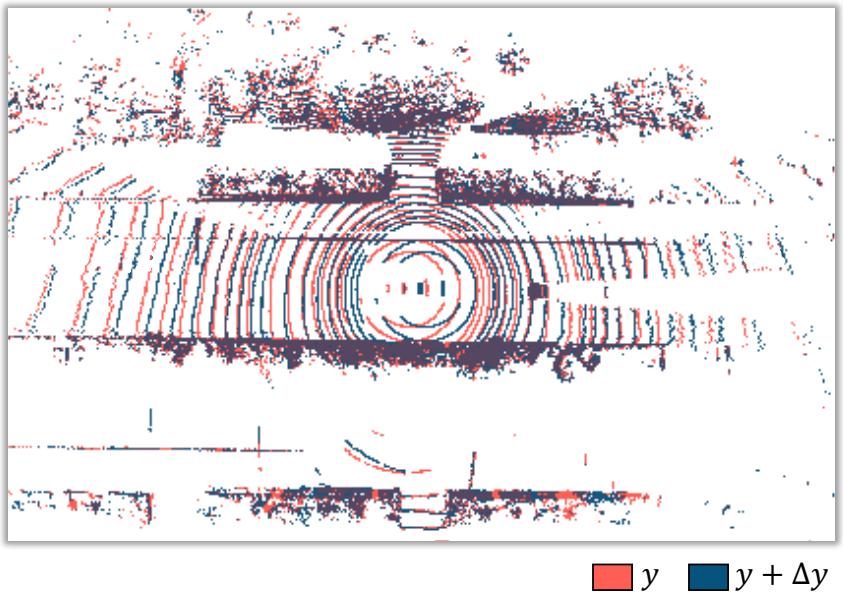


Figure 3.2. We pose-align two succesive LiDAR sweeps of a static scene s to a common world coordinate-frame (using the notation of Fig. 3.1). Even though there is zero scene motion Δs, points appear to drift or *swim* across surfaces. This is due to the fact that points are obtained by intersecting rays from a moving sensor Δy with static scene geometry. This in turn implies that points can appear to move since they are not tied to physical locations on a surface. This apparent movement ($\Delta \tilde{s}$) is in general a complex nonlinear transformation, even when the sensor motion Δy is a simple translation (as shown above). Traditional methods for self-supervised LiDAR forecasting [117, 207, 313, 318] require predicting the complex transformation $\Delta \tilde{s}$ which depends on the unknown Δy, while our differentiable-raycasting framework assumes Δy is an *input*, dramatically simplifying the task of the forecasting network. From a planning perspective, we argue that the future (planned) change-in-pose *should* be an input rather than an output.

Planning: Lastly, we show that such forecasted space-time occupancy can be jointly learned with space-time costmaps for end-to-end motion planning. Owing to LiDAR self-supervision, we are able to train on recent unsupervised LiDAR datasets [199] that are orders of magnitude larger than their annotated counterparts, resulting in significant improvement in accuracy for both forecasted occupancy and motion plans. Interestingly, as we increase the amount of archival training data at the cost of zero additional human annotation, object shape, tracks, and multiple futures "emerge" in the arbitrary quantities predicted by our model despite there being no direct supervision on ground-truth occupancy.

3.2 Related Work

Occupancy as a scene representation: Knowledge regarding what is around an autonomous vehicle (AV) and what will happen next is captured in different representations throughout the standard modular perception and prediction (P&P) pipeline [36, 162, 256, 311]. Instead of separate optimization of these modules [208, 291], Sadat et al. [257] propose bird's-eye view (BEV) *semantic occupancy* that is end-to-end optimizable. As an alternative to *semantic occupancy*, Hu et al. [116] propose BEV *ego-centric freespace* that can be self-supervised by raycasting on aligned LiDAR sweeps. However, the ego-centric freespace entangles motion from other actors, which is arguably more relevant for motion planning, with ego-motion. In this paper, we propose *emergent occupancy* to isolate motion of other actors. While we focus on self-supervised learning at scale, we acknowledge that for motion planning, some semantic labelling is required (e.g., state of a traffic light) which can be incorporated via semi-supervised learning.

Differentiable raycasting: Differentiable raycasting has shown great promise in learning the underlying scene structure given samples of observations for downstream novel view synthesis [206], pose estimation [347], etc. In contrast, our application is best described as "space-time scene completion", where we learn a network to predict an explicit space-time occupancy volume. Furthermore, our approach differs from existing approaches in the following ways. We use LiDAR sequences as input and raycast LiDAR sweeps given future occupancy and sensor pose. We work with explicit volumetric representations [190] for dynamic scenes with a feed-forward network instead of test-time optimization [218].

Self-supervision: Standard P&P solutions do not scale given how fast log data is collected by large fleets and how slow it is to curate object track labels. To enable learning on massive amount of unlabeled logs, supervision from simulation [41, 49, 50, 59], auto labeling using multi-view constraints [228], and self-supervision have been proposed. Notably, tasks that can be naturally self-supervised by LiDAR sweeps e.g., scene flow [207] have the potential to generalize better as they can leverage more data. More recently, LiDAR self-supervision has been explored in the context of point cloud forecasting [312, 313, 318]. However, when predicting future sweeps given the history, as stated before, past approaches often tend to couple motion of the world with the motion of the ego-vehicle [312].

Motion Planning: An understanding of what is around an AV and what will happen next [291] is crucial. This is typically done in the bird's eye-view (BEV) space by building a modular P&P pipeline. Although BEV motion planning does not precisely reflect planning in

the 3D world, it is widely used as the highest-resolution and computation- and memory-efficient representation [35, 257, 358]. However, training such modules often requires a massive amount of data. End-to-end learned planners requiring less human annotation have emerged, with end-to-end imitation learning (IL) methods showing particular promise [41, 49, 248]. Such methods often learn a neural network to map sensor data to either action (known as behavior cloning) or "action-ready" cost function (known as inverse optimal control) [214]. However, they are often criticized for lack of explainable intermediate representations, making them less accountable for safety-critical applications [224]. More recently, end-to-end learned but modular methods producing explainable representations, e.g., neural motion planners [35, 257, 358] have been proposed. However, these still require costly object track labels. Unlike them, our approach learns explainable intermediate representations that are explainable quantities for safety-critical motion planning without the need of track labels.

3.3 Method

Autonomous fleets provide an abundance of *aligned* sequences of LiDAR sweeps $\mathbf{x}$ and ego vehicle trajectories $\mathbf{y}$. How can we make use of such data to improve perception, prediction, and planning? In the sections to follow, we first define occupancy. Then we describe a self-supervised approach to predicting future occupancy. Finally, we describe an approach for integrating this forecasted occupancy into neural motion planners. Note that in the text that follows, we use ego-centric freespace and freespace interchangeably.

Occupancy

We define occupancy as the state of occupied space at a particular time instance. We use $\mathbf{z}$ to denote the true occupancy, which may not be directly observable due to visibility constraints. Let us write

$$\mathbf{z}[\mathbf{u}] \in \{0, 1\}, \mathbf{u} = (x, y, t), \mathbf{u} \in \mathbf{U} \tag{3.1}$$

to denote the occupancy of a voxel $\mathbf{u}$ in the space-time voxel grid $\mathbf{U}$, which can be *occupied* (1) or *free* (0). The spatial index of $\mathbf{u}$, i.e., (x, y) represents the spatial location from a bird's-eye view. Given a sequence of *aligned* sensor data and ego-vehicle trajectory $(\mathbf{x}, \mathbf{y})$, there may be multiple plausible occupancy states $\mathbf{z}$ that "explain" the sensor measurements. We denote this set of plausible occupancy states as $\mathbf{Z}$.

Forecasting Occupancy. Suppose we split an aligned sequence of LiDAR sweeps and

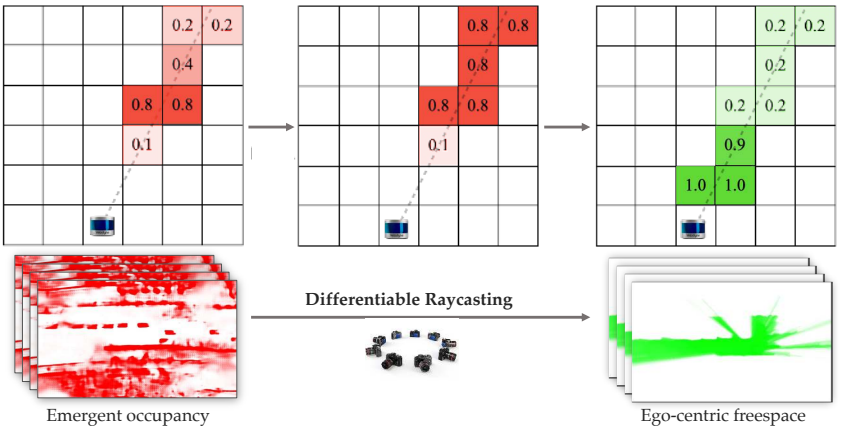


Figure 3.3. Differentiable procedure for estimating ego-centric freespace from volumetric occupancy, necessary for computing the loss from (3.3). The left image depicts predicted emergent occupancy, on which we perform a cumulative max along the LiDAR ray from known sensor poses (middle), which is differentiable because it is essentially re-indexing. The result is then inverted to produce (soft) visible ego-centric freespace estimates. To identify BEV pixels along the LiDAR ray, we perform fast voxel traversal in 2D [8].

ego-vehicle trajectory $(\mathbf{x}, \mathbf{y})$ into a historic pair $(\mathbf{x_1}, \mathbf{y_1})$ and a future pair $(\mathbf{x_2}, \mathbf{y_2})$. Our goal is to learn a function f that takes historical observations $(\mathbf{x_1}, \mathbf{y_1})$ as input and predicts emergent future occupancy $\hat{\mathbf{z}}_\mathbf{2}$. Formally,

$$\hat{\mathbf{z}}_\mathbf{2} = f(\mathbf{x_1}, \mathbf{y_1}), \tag{3.2}$$

If the true occupancy $\mathbf{z_2}$ were observable, we could directly supervise our forecaster, f. Unfortunately, in practice, we only observe LiDAR sweeps, $\mathbf{x}$. We show in the next section how to supervise f with LiDAR sweeps using differentiable raycasting techniques.

Raycasting

Given an occupancy estimate $\hat{\mathbf{z}}$, sensor origin $\mathbf{y}$ and directional unit vectors for rays $\mathbf{r}$, a differentiable raycaster $\mathcal{R}$ can raycast LiDAR sweeps $\hat{\mathbf{x}}$. We use $\hat{\mathbf{d}}$ to represent the expected distance these rays travel before hitting obstacles: $\hat{\mathbf{d}} = \mathcal{R}(\mathbf{r}; \hat{\mathbf{z}}, \mathbf{y})$. Then we can reconstruct the raycast LiDAR sweep $\hat{\mathbf{x}}$ as $\hat{\mathbf{x}} = \mathbf{y} + \hat{\mathbf{d}} * \mathbf{r}$.

Learning to Forecast Occupancy

Given the predicted occupancy $\hat{\mathbf{z}}_\mathbf{2}$ (Eq. 3.2), and the captured sensor pose $\mathbf{y_2}$, a differentiable raycaster $\mathcal{R}$ can take rays $\mathbf{r_2}$ as input and produce $\hat{\mathbf{d}}_\mathbf{2} = \mathcal{R}(\mathbf{r_2}; \hat{\mathbf{z}}_\mathbf{2}, \mathbf{y_2})$. Note that this formulation

allows us to decouple the motion of the world captured by change in occupancy, $\hat{\mathbf{z}}_2$, and the motion of the ego-vehicle captured by change in sensor origin, $\mathbf{y_2}$.

This also allows us to supervise $\hat{\mathbf{z}}_2$ using a loss function that measures the difference between the raycast distance $\hat{\mathbf{d}}_2$ and the ground-truth distance $\mathbf{d_2}$.

$$L_r = \text{loss}(\hat{\mathbf{d}}_2, \mathbf{d_2}) \tag{3.3}$$

Loss function: One natural loss function might be distance between the raycast depth and measured depth along each ray. In practice, we care most about disagreements of freespace which can inform safe motion plans. To emphasize such disagreements, we define voxels encountered along the ray as having a free versus not-free binary label, and use a binary cross-entropy loss (summed over all voxels encountered by each ray until the boundary of voxel grid, ref. Fig. 3.3). We adopt an encoder-decoder architecture that predicts future emergent occupancy given historical LiDAR sweeps, differentiably raycasts future LiDAR sweeps and self-supervises using archival sweeps (ref. highlighted branch of Fig. 3.4 (a)).

Learning to Plan

The previous section described an approach for predicting future LiDAR returns via differentiable raycasting of BEV space-time occupancy maps. We now show that such costmaps can be integrated directly into an end-to-end motion planner that makes use of space-time costmaps for scoring candidate trajectories. We follow [117], but modify their derivation to take into account emergent occupancy.

Max-margin planning: We learn a model g to predict a space-time cost map, $\mathbf{c_2}$, over future timestamps given past observations $(\mathbf{x_1}, \mathbf{y_1})$:

$$\mathbf{c_2} = g(\mathbf{x_1}, \mathbf{y_1}), \text{ where } \mathbf{c_2}[\mathbf{u}] \in \mathbb{R}, \mathbf{u} \in \mathbf{U_2} \tag{3.4}$$

where $\mathbf{U_2}$ represents the space-time voxel grid over future timestamps. We define the cost of a trajectory as the sum of costs at its space-time way-points. The best candidate future trajectory according to the cost map is the one with the lowest cost:

$$\hat{\mathbf{y}}_2^* = \arg\min_{\hat{\mathbf{y}} \in \mathbf{Y_2}} C(\hat{\mathbf{y}}; \mathbf{c_2}) = \arg\min_{\hat{\mathbf{y}} \in \mathbf{Y_2}} \sum_{\mathbf{u} \in \hat{\mathbf{y}}} \mathbf{c_2}[\mathbf{u}] \tag{3.5}$$

where $\mathbf{Y_2}$ represents the set of viable future trajectories.

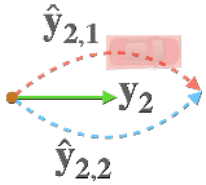
Loss function: We use a max-margin loss function, where the target cost of a candidate trajectory ($\hat{\mathbf{y}}$) is equal to the cost of the expert trajectory ($\mathbf{y_2}$) plus a margin. We can write the objective as follows:

$$L_p = \left[C\left(\mathbf{y_2}; \mathbf{c_2}\right) - \left(\min_{\hat{\mathbf{y}} \in \mathbf{Y_2}} C\left(\hat{\mathbf{y}}; \mathbf{c_2}\right) - D\left(\hat{\mathbf{y}}, \mathbf{y_2}\right) \right) \right]_+ \tag{3.6}$$

where $[\cdot]_+ = \max(\cdot, 0)$ and D is a function that quantifies the desired margin between the cost of a candidate trajectory and the cost of an expert trajectory. A common choice for D is Euclidean distance between pairs of way-points:

$$D(\hat{\mathbf{y}}_\mathbf{2}, \mathbf{y_2}) = ||\hat{\mathbf{y}}_\mathbf{2}, \mathbf{y_2}||_2 \tag{3.7}$$

Learning cost maps that reflect such cost margins only requires expert demonstrations, which are readily available in archival log data. However, sometimes candidates trajectories that are equally distant from the expert one should bear different costs. We provide an example (right) where the red trajectory should cost more than blue in the presence of an obstacle despite both being equidistant from the expert demonstration.



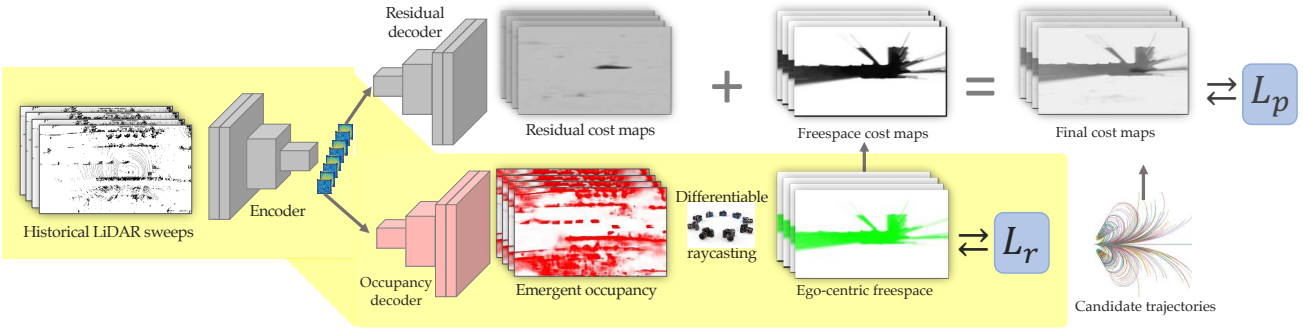
Guided planning: To further distinguish among candidate trajectories, one could introduce extra penalty terms given additional supervision.

$$D(\hat{\mathbf{y}}_\mathbf{2}, \mathbf{y_2}) = ||\hat{\mathbf{y}}_\mathbf{2}, \mathbf{y_2}||_2 + \gamma\ P(\hat{\mathbf{y}}_\mathbf{2}) \tag{3.8}$$

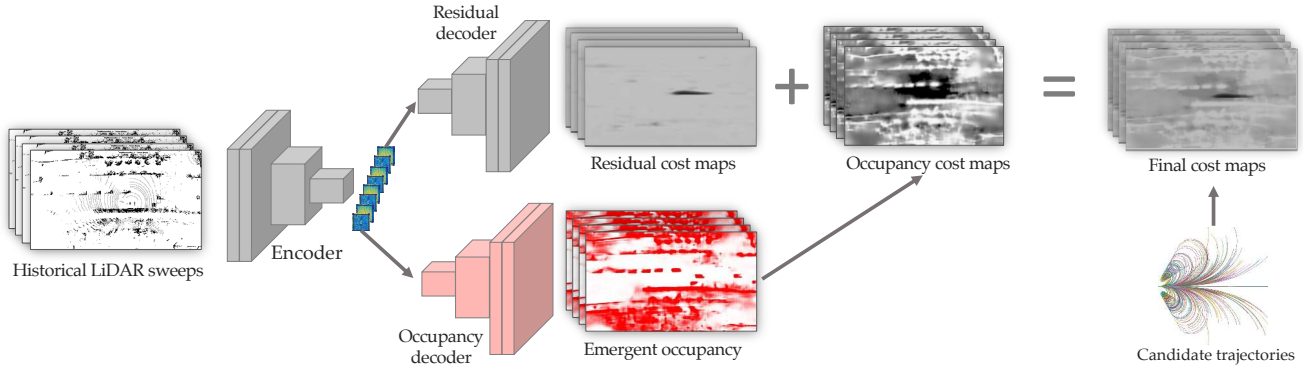
where P represents a penalty function and γ is a predefined scaling factor. Zeng et al. [358] propose to define an additional penalty such that candidate trajectories that collide with object boxes would cost an additional γ in addition to the deviation from the expert demonstration. We refer to this approach as *object-guided planning*, which is effective but costly as it requires object track labels.

More scalable alternatives to object supervision can be adopted, such as formulation of the penalty term proposed by Hu et al. [117]. Concretely, candidate trajectories that reach outside the freespace as observed by future LiDAR poses would incur an additional penalty. We refer to this as *freespace-guided planning*.

Residual costmaps: Instead of directly predicting the cost map $\mathbf{c_2}[\mathbf{u}]$, we follow prior work [117] and predict a residual cost map $\tilde{\mathbf{c}}_\mathbf{2}[\mathbf{u}]$ that is added to the cost map from freespace estimate



(a) Training architecture.



(b) Inference architecture.

Figure 3.4. Overview of our training and inference-time planning architectures. Highlighted network branch in (a) is used to learn future emergent occupancy, which is augmented by the residual branch that predicts residual cost maps, eventually used in computing a guided planning loss.

based on predicted emergent occupancy.

$$\mathbf{c_2}[\mathbf{u}] = \tilde{\mathbf{c}}_\mathbf{2}[\mathbf{u}] + \alpha \ \text{proj}(\hat{\mathbf{z}}_\mathbf{2}; \mathbf{y_2})[\mathbf{u}], \ \ \mathbf{u} \in \hat{\mathbf{y}}_\mathbf{2} \tag{3.9}$$

where α is a predefined constant and $\tilde{\mathbf{c}}_\mathbf{2}$ represents the predicted residual cost map. The operation $\text{proj}(\hat{\mathbf{z}}_\mathbf{2}; \mathbf{y_2})$ is illustrated in Fig. 3.3.

Multi-task planning (new): In addition to the raycasting loss in Fig. 3.4, we add L_p as an additional planning loss. In other words, the emergent occupancy prediction architecture is augmented with another decoder branch to predict the residual cost maps while sharing the encoder features. Because of this, emergent occupancy forecasting becomes the auxiliary task for the end-to-end motion planner. We illustrate the network architecture during training in Fig. 3.4 (a).

Test-time occupancy cost maps (new): At test time, to compute ego-centric freespace cost

maps based on predicted emergent occupancy, for each candidate sample trajectory, one would need to perform raycasting from its waypoints, which is prohibitively expensive. Fortunately, this is exactly equivalent to directly accessing emergent occupancy on the waypoints along the candidate trajectory (because of the cumulative max-operation used in deriving freespace from occupancy - see Fig. 3.3), as formally expressed in Eq. (3.10).

$$\text{proj}(\hat{\mathbf{z}}_\mathbf{2}; \hat{\mathbf{y}}_\mathbf{2})[\mathbf{u}] = \hat{\mathbf{z}}_\mathbf{2}[\mathbf{u}], \;\; \mathbf{u} \in \hat{\mathbf{y}}_\mathbf{2} \tag{3.10}$$

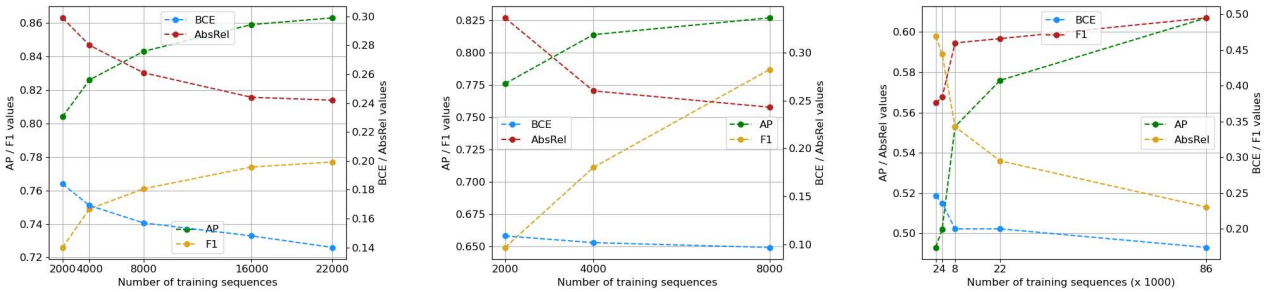
The simplified test-time architecture is illustrated in Fig. 3.4 (b). When optimizing for future trajectories, we restrict the search space of future trajectories to the ones with a smooth transition from the past trajectory [117, 358]. Please refer to the supplement for other implementation information such as detailed network architecture.

3.4 Experiments

Datasets: We evaluate occupancy forecasting and motion planning on two datasets: nuScenes [30] and ONCE [199]. nuScenes features real-world driving data with 1,000 fully annotated 15 second logs. ONCE is the largest driving dataset with 150 hours of real-world data including 1 million LiDAR sweeps, collected in a range of diverse environments such as urban and suburban areas. As annotation is expensive, only a small subset of logs in ONCE are fully annotated, making it ideal for self-supervised learning. We include comparison against state-of-the-art forecasting and planning approaches on both datasets. We also construct multiple baselines for all ablative evaluation for bird's eye-view motion planning. To understand how our occupancy forecasting and motion planning performance scales to an increasing amount of training data, we randomly curate different training sets of the datasets. Since only a small subset of 8K samples in ONCE is labeled, we do this by progressively increasing the number of training samples by adding scenes from both their labeled and unlabeled-small splits, which include 8K, and 86K training samples respectively. Some of our analysis exists only on the combined labeled and unlabeled-small split which totals to 94K samples. For nuScenes, we randomly sample scenes from their official training set. For all experiments that follow, we take in a historical LiDAR stack of 2 seconds and forecast for the next 3 seconds.

Dataset	Diff. Raycast	$\frac{\lvert \mathbf{d}-\hat{\mathbf{d}} \rvert}{\mathbf{d}}$ (↓)	BCE (↓)	F1 (↑)	AP (↑)
nuScenes	- [117]	0.297	0.221	0.665	0.769
	✓	**0.242**	**0.140**	**0.777**	**0.863**
ONCE	- [117]	0.371	0.143	0.635	0.732
	✓	**0.243**	**0.097**	**0.787**	**0.827**

Table 3.1. Indirect evaluation of emergent occupancy forecasting with respect to groundtruth LiDAR sweeps. On both nuScenes and ONCE, we significantly improve forecasting accuracy across all metrics by using differentiable raycasting for decoupling the scene and ego-motion, unlike Hu *et al.* [117].



(a) nuScenes (b) ONCE-labeled (c) ONCE-unlabeled

Figure 3.5. We highlight the merits of our self-supervised approach which can be given any amount of unlabeled LiDAR data to train on, in the form of posed archival LiDAR sweeps, thereby increasing the performance of emergent occupancy forecasting (evaluated using classification metrics such as average-precision and F1). Please refer to the supplement for corresponding tables.

Emergent Occupancy Forecasting

Metrics: Since, the groundtruth for true occupancy is unavailable, we quantitatively evaluate the LiDAR sweeps raycast from the emergent occupancy predictions. Specifically, our first evaluation computes the absolute relative error between the groundtruth distance traveled by every ray starting from the sensor origin, and the expected distance traveled by corresponding rays; where the expected distance is obtained by casting rays through the forecasted occupancy. Second, we score *every* BEV voxel traversed by a ray using its 'free' or 'not-free' state. This dense per-ray evaluation is equivalent to evaluating the per-pixel binary classification of an ego-centric freespace map with respect to its groundtruth, allowing us to compare to the baseline discussed below. We compute the dense binary cross-entropy, average precision and the F1-score. All metrics are averaged across all prediction timesteps (up to 3s).

Baseline: We re-implement the future-freespace architecture from [117] which directly forecasts

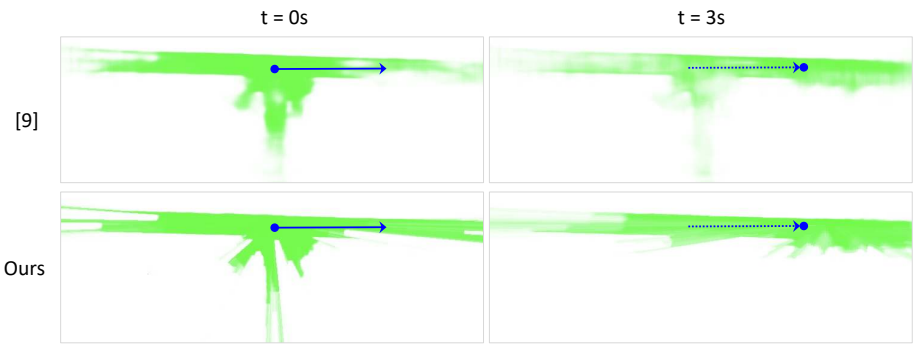


Figure 3.6. Future ego-centric freespace from [117] and our model, raycasted from predicted emergent occupancy. Note how the presence of moving and parked cars on roadsides is captured well by our approach even 3s in the future.

ego-centric freespace. For building our architecture, we adapt this network to predict an arbitrary quantity which differentiably raycasts into ego-centric freespace given a sensor location. On training this architecture in a self-supervised manner, the arbitrary quantity *emerges* into emergent occupancy, an explainable intermediate representation for downstream motion planners.

Main results: We compare the performance of both approaches in Tab. 3.1. Note the drastic improvement in all metrics on using differentiable raycasting to decouple the scene motion from the ego-motion of the sensor on both nuScenes and ONCE. With increase of up to 15% F1 points, we highlight the high-quality of our predicted occupancy and the pronounced effect of adding differentiable raycasting. Our results show that occupancy reasoning is an important intermediate task, *even* if the end-goal is simply understanding freespace: Our method, which predicts occupancy as an intermediate target, outperforms [117], which directly aims to predict freespace. Fig. 3.6 visualizes predicted ego-centric freespace for a single scenario in ONCE using [117] and our approach at $t = 0, 3s$ in the future. In Fig. 3.5, we show how adding more training samples to both datasets result in an upward trend in performance across *all* metrics. This increasing generalizability and scaling of training data comes for free with our self-supervised approach.

Motion Planning

Metrics: We follow prior works and compute three metrics for evaluating motion planning performance, including (1) L2 error; (2) point collision rate; (3) box collision rate. The L2 distance measures how close the planned trajectory follows the expert trajectory at each future timestamp. The point collision rate measures how often the planned waypoint is within the BEV

nuScenes	Box Collision (%)			L2 Error (m)		
	1s	2s	3s	1s	2s	3s
IL [240]	0.08	0.27	1.95	**0.44**	**1.15**	**2.47**
FF [117]	0.06	0.17	1.07	0.55	1.20	2.54
Ours	**0.04**	**0.09**	**0.88**	0.67	1.36	2.78
NMP [358]	0.04	0.12	*0.87*	*0.53*	*1.25*	*2.67*
P3 [257]	*0.00*	*0.05*	1.03	0.59	1.34	2.82

Table 3.2. We compare end-to-end state-of-the-art motion planners on nuScenes-val. NMP and P3 are supervised approaches that have access to object tracking labels.

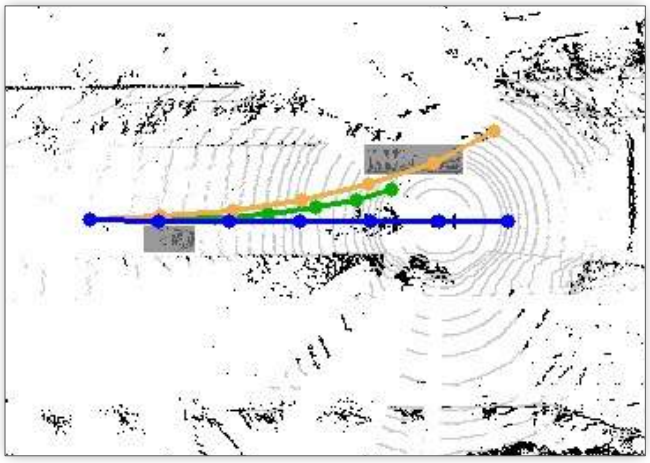
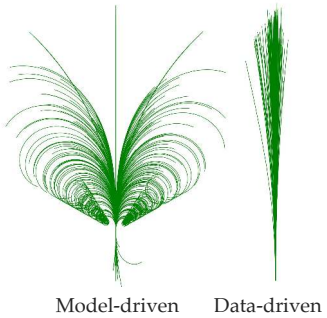


Figure 3.7. A vanilla spacetime trajectory with a lower L2 error wrt. expert, may collide into objects unlike a proposed trajectory with larger L2 error but no collision.

boxes of other objects. The box collision rate measures how often the BEV box of the ego-vehicle intersects with BEV boxes of other objects.

Trajectory sampling: When evaluating performance on nuScenes, we follow previous state-of-the-art approaches [117, 358] and sample a combination of straight lines, circles, and clothoid curves as trajectory samples. Owing to the scene diversity in ONCE, we notice that such a sampling strategy does not capture the distribution of expert trajectories on ONCE as they range widely in their velocities and directions. Inspired by [35], we sample a data-driven trajectories to complement the model-driven samples (right). The supplement provides more details on our data-driven sampler.



	Freespace Guided	Multi Task	Diff. Raycast	Box Collision (%)			Point Collision (%)			L2 Error (m)		
				1s	2s	3s	1s	2s	3s	1s	2s	3s
(a)	-	-	-	0.08	0.27	1.95	**0.00**	**0.00**	0.35	0.44	1.15	2.47
(b)	✓	-	-	0.06	0.17	1.07	**0.00**	0.01	0.04	0.55	1.20	2.54
(c)	-	✓	-	0.08	0.17	1.29	**0.00**	0.02	0.08	**0.42**	**1.06**	**2.30**
(d)	✓	✓	-	**0.02**	0.10	1.10	**0.00**	**0.00**	0.08	0.52	1.22	2.64
(e)	✓	✓	✓	0.04	**0.09**	**0.88**	**0.00**	0.01	**0.03**	0.67	1.36	2.78

Table 3.3. Ablation studies on nuScenes-val. Note that (a) is IL, (b) is FF, and (e) is **Ours** in Tab. 3.2.

Planning on nuScenes

Baselines: We compare our proposed approach to four baseline end-to-end motion planners. First, we implement a pure imitation learning (IL) baseline, a max-margin neural motion planner self-supervised by expert trajectories, as described in Eq. (3.7). Second, we re-implement future-freespace-guided max-margin planner (FF) proposed by Hu *et al.* [117], as captured by Eq. (3.8). Third, we re-implement a simplified neural motion planner (NMP) without modeling costs related to map information and traffic light status as such information is unavailable on nuScenes. Last, we re-implement a simplified version of perceive, predict, and plan (P3) where we do not distinguish semantic occupancy of different classes. To ensure a fair comparison, we adopt the same neural net architecture for the baselines and our approach.

Main results: As Tab. 3.2 shows, in terms of collision rates, our self-supervised approach outperforms both self-supervised baselines (IL and FF) by a large margin. Moreover, our approach achieves the same collision rate at 3s as the best of supervised baselines. We also observe a commonly observed trade-off between L2 errors and collision rates [358]. For example, pure imitation learning achieves the lowest L2 errors with the highest collision rates.

Ablation studies: We perform extensive ablation studies in Tab. 3.3 to understand where improvements come from. There are three main observations:

- Differentiable raycasting reduces collision rate at further horizon (3s), as seen in (d) vs. (e), suggesting decoupling motion of the world (space-time occupancy) from ego-motion is helpful when learning long range cost maps.
- Multi-task learning further reduces collision rates, as seen in (a) vs. (c). Training max-margin planners with an auxiliary self-supervised forecasting task significantly reduces the collision rates without hurting L2.

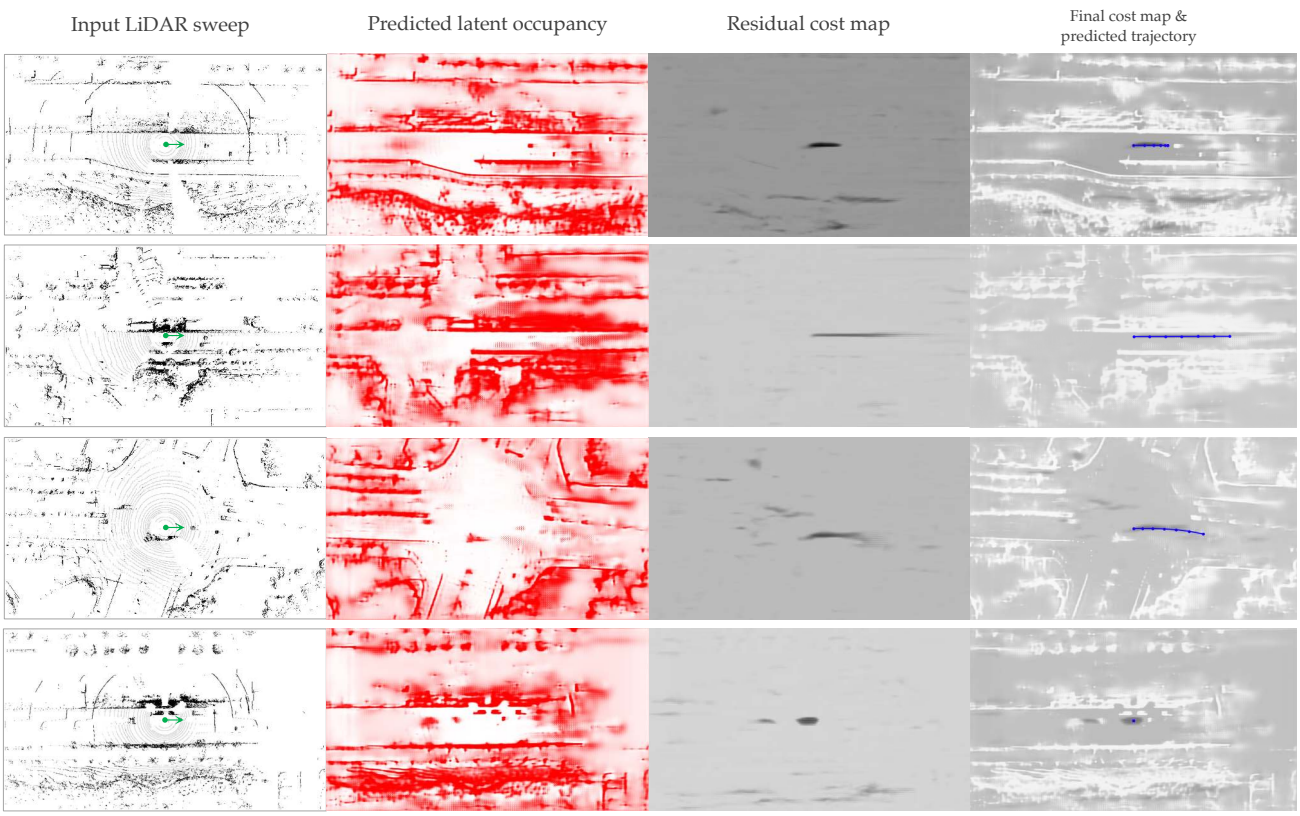
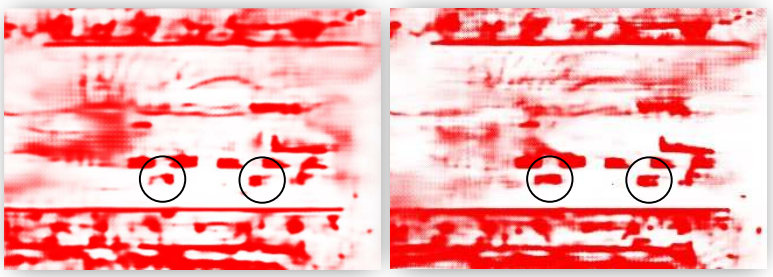


Figure 3.8. Qualitative results of our learned model. From top to bottom, we visualize various scenarios, including slowing down, speeding up, navigating an intersection and staying still. All columns after the first one are visualized at future timestamp t=0.5s. We successfully forecast the motion of surrounding objects, e.g. in third row, which results in safer planned trajectories.

- Freespace-guided cost margin is crucial to lowering collision rates, as seen in (a) vs. (b), (c) vs. (d). However, there is a trade-off: the L2 errors tend to increase as being expert-like (at all costs) is no longer the only objective. In Fig. 3.7, we show an example result describing why L2 error is a misleading metric that doesn't allow for alternate future plans that are otherwise viable. Additionally, Casas *et al.* [35] show that collision rate is a more consistent metric between evaluation in the open- and closed-loop setups.

Planning on ONCE

Baseline: ONCE offers a massive amount of unlabeled, diverse LiDAR sweeps paired with ego-vehicle trajectories and a small fully labeled subset of about 8K samples. We train a re-implemented neural motion planner as a supervised baseline on the fully labeled subset. We train



Emergent occupancy (8K) Emergent occupancy (94K)

Figure 3.9. Evolution of estimated emergent future occupancy.

our self-supervised approach over a wide range of training sizes, from 2K to 94K.

Main results: Perhaps unsurprisingly, our first observation is that the metrics on ONCE are inflated as compared to nuScenes, because of the diverse range of environments ONCE features, ranging from straight highways to complex city road structures. To show the scalability of our approach on such a diverse and large dataset, we plot the L2 error and (box) collision rate at 3s as a function of the amount of training data in Fig. 3.10. Both the L2 error and the collision rate of our approach continue to improve as we increase the size of the training set. In comparison, the supervised neural motion planner achieves an L2 error of 4.45m and a box collision rate of 2.54% at a training size of 8K.

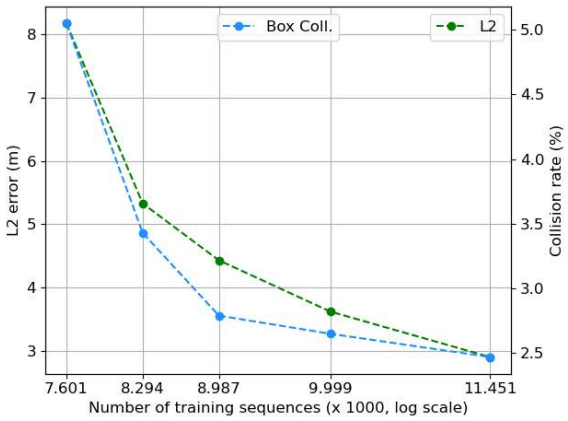


Figure 3.10. Planning performance vs. larger ONCE training set size.

At 94K training samples, our self-supervised approach achieves a dramatically lower L2 error of 2.9m and a lower collision rate of 2.47%. Importantly, such scalability for motion planning comes for free as our approach is self-supervised. We show some qualitative results on the ONCE dataset in Fig. 3.8 where our approach is able to deal with a number of varying driving scenarios; decelerate and stop when necessary, predict long trajectories when unoccupied regions are predicted ahead, avoid collisions with other vehicles while navigating an intersection, or stay stationary. Please refer to our supplement for further quantitative evaluation, visualization of future cost maps and more qualitative examples that feature failure cases (e.g., forecasted occupancy diffuses over time).

Evolution of occupancy estimates: Our model tends to produce better estimates of emergent occupancy as we increase the amount of training data. The percent of semantic object pixels recalled from the ground-truth semantic object labels in our predicted occupancy map increases from 51% to 59% at t=0s when we increase the amount of training data from 8K to 94K. Qualitatively, this can be seen in Fig. 3.9 where the shape of two cars in the right lane looks more "space-time complete" for the model trained with increased data.

3.5 Discussion

We propose *emergent occupancy* as a self-supervised and explainable representation for motion planning. Our novel differentiable raycasting procedure enables the learning of occupancy forecasting under the self-supervised task of LiDAR sweep forecasting. The raycasting setup also allows us to decouple ego motion from scene motion, making forecasting an easier task for the network to learn. Experimental results suggest that such decoupling is also helpful for downstream motion planning. Such training at scale allows object shape, tracks, and multiple futures to "emerge" in the predicted emergent occupancy.

3.6 Appendix

In this supplement, we discuss more details of our experimental setup in Sec. 3.6, discuss supplementary evaluation of occupancy forecasting in Sec. 3.6 and analyse the quantitative and qualitative performance of our motion planning architecture further in Sec. 3.6.

Experimental Setup

Network Architecture

Architecture Implementation We use the same neural network architecture as proposed by Zeng *et al.* [358] and developed on by Hu *et al.* [117]. Different from these two networks, we use two decoders, one that predicts the emergent occupancy cost maps, and one that predicts the residual cost maps. The differentiable raycaster proposed by us acts as a layer over the occupancy cost maps, that produces raycast sweeps for 7 future timesteps (accounting for three seconds in the future).

Freespace is computed from these sweeps and it is used in 3 places in the network: (1) in computing a dense per-pixel classification loss with the groundtruth freespace, (2) in computing

the final cost maps which are a sum of the freespace and the residual cost maps, and (3) in computation of the cost margin for the planning loss.

Input and output We follow the same input and output BEV data format as that used in Hu *et al.* [117] for nuScenes, except that we now take input from [-2s, 0s]. For nuScenes, this means 20 input timestamps and a stack of $704 \times 400 \times 20$ size. For ONCE, since the LiDAR sweeps are collected at 2Hz, this stack is of size $704 \times 400 \times 5$ to accomodate 2s of input data. The output of the network for both the datasets is of size $704 \times 400 \times 7$ to account for 3s of forecasts at a 0.5s interval, starting at the 0th timestep. Each pixel in the BEV map covers an area of $0.2m \times 0.2m$. To compute groundtruth freespace cost maps, we apply ground segmentation [102, 117] to the output LiDAR sweep and raycast as described in the main paper.

Differentiable raycaster First, we collect a set of rays, with origin as the position of the ego-vehicle in the world coordinate frame and endpoints as the endpoints in a groundtruth LiDAR sweep. For a given ray (origin and direction), we find the voxels that the ray travels in the BEV LiDAR scan using a fast voxel traversal algorithm proposed by Amanatides *et al.*[8]. Given all the voxels along a ray, we perform a soft raycast along the ray as follows: we sample occupancy states given the predicted occupancy probabilities, raycast to get free vs occluded space, and average the raycasts over all samples. In practice, we do this analytically by computing the expectation, as done by many prior works based on volume rendering [206].

Data-driven sampler

Following prior work that uses data-driven trajectory sampling techniques for evaluating the performance in mapless driving scenarios [35], we curate a dataset of expert trajectories by binning the trajectories in the train set by their velocity. Once this dataset is curated during preprocessing, we use retrieval based on the past timestep's speed and direction profiles to index into the appropriate bin in the dataset. When a velocity is not available in the set of data-driven trajectories, we compute the nearest speed and angle from the set, for a given sample. From this nearest bin, we randomly sample 200 valid trajectories and append them to our set of 2000 model-based trajectories.

This approach avoids arbitrary choice of steering profiles for the ONCE dataset, since this information in unknown in ONCE (note that this is available for nuScenes with the CAN bus data). This is useful because in comparison to nuScenes, the ONCE dataset is composed of

Figure 3.11. Distribution of train trajectories in nuScenes (**left**) and ONCE (**right**).

a complementary set of trajectories, as shown in Fig. 3.11. Using this data-driven trajectory sampler in conjunction with the standard trajectory sampler gives us a complete coverage of possible future trajectories, including the ones that appear the most in the ONCE dataset.

Occupancy Forecasting

We supplement the evaluation of occupancy forecasting on ONCE and nuScenes in the main paper by providing complete results of Fig. 7 in Tab. 3.4. The unlabeled subsets of ONCE do not include samples from the labeled train set. As described, increasing the amount of training data directly impacts the improvement in performance. It is worth noting the performance difference between the 8k set of ONCE-labeled and ONCE-unlabeled. The higher metrics on the labeled set indicates that the ONCE labeled set is much higher quality and falls in the same data distribution as compared to the val set. nuScenes training subsets also show increasing performance with increase in data.

Motion Planning

Planning on ONCE

Quantitative Analysis This section supplements our results on the ONCE dataset from the main paper. We show the complete results of Fig. 12 in Tab. 3.7. Note that as the amount of data is increased during training, the L2 error and box collision rate decreases dramatically.

Dataset	Size	$\frac{\lvert\mathbf{d}-\hat{\mathbf{d}}\rvert}{\mathbf{d}}$ (↓)	BCE (↓)	F1 (↑)	AP (↑)
ONCE	2,000	0.336	0.109	0.649	0.776
	4,000	0.260	0.102	0.711	0.814
	8,000	**0.243**	**0.097**	**0.787**	**0.827**
ONCE (unlabeled)	2,000	0.598	0.246	0.376	0.493
	4,000	0.589	0.236	0.384	0.502
	8,000	0.553	0.200	0.460	0.553
	22,000	0.536	0.200	0.466	0.576
	86,000	**0.513**	**0.174**	**0.495**	**0.607**
nuScenes	2,000	0.299	0.184	0.726	0.804
	4,000	0.280	0.169	0.749	0.826
	8,000	0.261	0.157	0.761	0.843
	16,000	0.244	0.148	0.774	0.859
	22,000	**0.242**	**0.140**	**0.777**	**0.863**

Table 3.4. Supplementary table for the evaluation of occupancy forecasting on ONCE-val and nuScenes-val with models trained on different subsets of the ONCE labeled, unlabeled and nuScenes train set.

Even though box collision rate is a stricter metric than point collision rate, we see a consistent trend in it at the longest horizon. Our best model beats the neural motion planner [358] baseline described in the main paper. Note that such a baseline can only be trained with the labeled training set of ONCE (with 8K samples), whereas all the raw unlabelled LiDAR logs in ONCE can be used by our method since it is self-supervised.

We also conduct an ablative study of our approach on the ONCE dataset in Tab. 3.6. Note that since the hyperparameters are not tuned for the ONCE dataset, the best performing method on the Box Collision metric at 3s horizon is by Hu *et al.* [117]. Intuitively, this difference in performance shows that the trajectories selected by our planner pass close to the objects in the environment, such that they incur a box collision but not point collision. This is expected as even though we outperform Hu *et al.* [117] on occupancy forecasting, the guided planning loss used optimizes for point collision by summing per way-point occupancy cost instead of box collision, on which we outperform all other methods at 3s horizon.

Ablations on training architecture

While we compute the predicted cost margin in the main paper by summing the egocentric-freespace cost maps with the residual cost maps for an apples-to-apples comparison with Hu *et al.*[117], a more natural training architecture for motion planning would sum up the occupancy

Training	L2 Error (m)			Point Collision (%)			Box Collision (%)		
size	1s	2s	3s	1s	2s	3s	1s	2s	3s
8,000	0.84	2.26	4.45	0.00	0.04	1.06	0.04	0.14	2.54
2,000	1.97	4.37	8.18	0.07	0.39	1.84	0.77	2.36	5.05
4,000	1.13	2.79	5.33	0.00	0.04	1.02	0.14	0.81	3.43
8,000	1.00	2.33	4.43	0.00	0.04	**0.74**	0.04	**0.28**	2.79
22,000	0.71	1.87	3.62	0.00	0.14	1.06	0.04	0.39	2.65
94,000	**0.56**	**1.49**	**2.90**	**0.00**	**0.04**	0.99	**0.00**	0.39	**2.47**

Table 3.5. Planning metrics at different amount of training data on ONCE-val. First row corresponds to our reimplementation of the neural motion planner [358] baseline described in the main paper.

	Cost	Mid	Diff.	L2 Distance (m)			Point Collision (%)			Box Collision (%)		
	Margin	Task	Raycast	1s	2s	3s	1s	2s	3s	1s	2s	3s
(a)	-	-	-	**0.61**	**1.64**	**3.33**	**0.00**	**0.00**	1.02	**0.00**	0.42	2.47
(b)	✓	-	-	0.80	2.12	4.15	**0.00**	**0.00**	0.78	**0.00**	**0.18**	**1.84**
(c)	-	✓	-	0.89	2.40	4.78	**0.00**	0.04	1.63	**0.00**	0.35	3.85
(d)	✓	✓	-	0.90	2.49	4.99	**0.00**	0.04	1.10	0.07	0.25	2.61
(e)	✓	✓	✓	1.00	2.33	4.43	**0.00**	0.04	**0.74**	0.04	0.28	2.79

Table 3.6. Ablation studies on ONCE. Note that (a) is IL, (b) is FF, and (e) is **Ours**.

and residual cost maps during training, similar to the test-time architecture. Such an architecture would compute egocentric-freespace only for self-supervision with the multi-task loss and use the occupancy cost maps for motion planning. In Tab. 3.7, we evaluate training with this ablated architecture. Note that since the occupancy cost maps are now optimized directly during training, the performance across all metrics increases.

Limitations

We highlight a few limitations of our work. First, our self-supervised emergent occupancy does not offer semantics (e.g., traffic light and lane information) that is crucial for urban navigation. Despite this, we show that learning to drive with future occupancy is a safe fallback option in industrial autonomous driving. Second, BEV occupancy by itself does not handle overhead structures (e.g., trees, overpass); this may be mitigated by learning which occupied voxels are 'passable' during differentiable raycasting. Third, we rely on open-loop evaluation, where the world (incorrectly) unfolds in the same manner as the expert trajectory. Although this can be

Dataset	Training size	L2 Error (m)			Point Collision (%)			Box Collision (%)		
		$1s$	$2s$	$3s$	$1s$	$2s$	$3s$	$1s$	$2s$	$3s$
nuScenes	-	0.76	1.61	3.23	0.00	0.00	0.15	0.04	0.15	0.98
ONCE	2,000	2.10	4.39	7.74	**0.00**	0.25	1.45	0.32	1.94	4.80
	4,000	1.09	2.73	5.15	**0.00**	0.04	0.99	0.07	0.53	3.28
	8,000	0.87	2.24	4.32	**0.00**	**0.00**	0.78	0.04	0.25	2.37
	22,000	0.71	1.92	3.74	**0.00**	0.28	1.02	0.07	0.67	2.65
	94,000	**0.50**	**1.32**	**2.61**	**0.00**	0.04	**0.71**	**0.00**	**0.21**	**1.94**

Table 3.7. Evaluation of planning metrics on nuScenes and ONCE by adding the occupancy cost maps to residual cost maps during training.

corrected in a closed-loop setup, with our work we show that optimizing for collision metrics, with or without L2 error, can act as a proxy for learning to drive safe in real-world. Fourth, our method assumes accurate ego-motion during training but does not require it at test time. Finally, as the supplementary video highlights, our occupancy estimates diffuse over time, capturing multiple futures. However, we posit that future occupancy can be made more robust by constraining it with scene flow.

Part II

Using foundational priors zero-shot

Chapter 4

Large reconstruction models for object tracking across occlusions

Publication information

Khurana, T., Dave, A. and Ramanan, D., 2021. Detecting invisible people. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) pp. 3174-3184.

4.1 Introduction

Object detection has seen immense progress, albeit under a seemingly harmless assumption: that objects are *visible to the camera* in the image. However, objects that become fully occluded (and thus, invisible) continue to exist and move in the world. Indeed, object permanence is a fundamental visual cue exhibited by infants in as early as 3 months [11, 124]. Practical autonomous systems must similarly reason about such objects that undergo complete occlusions to ensure safe operation (Figure 4.1). Interestingly, existing work on object detection and tracking tends to de-emphasize this capability, either choosing to completely ignore highly-occluded instances for evaluation [67, 181, 255, 333], or simply downweighting them because they occur so rarely that they fail to materially affect overall performance [204]. One reason that invisible-object detection may have been under-emphasized in the tracking community is that for *offline* analysis, one can post-hoc reason about the presence of an occluded object by relinking detections *after* it reappears. This approach has spawned the large subfield of reidentification (ReID). However, in

Figure 4.1. We visualize an online tracking scenario from Argoverse [39] that requires tracking a pedestrian through a complete occlusion. Such applications cannot wait for objects to re-appear (*e.g.*, as re-identification approaches do): autonomous agents must properly react *during* the occlusion. We treat online detection of occluded people as a *short-term forecasting* challenge.

an *online* setting (such as an autonomous vehicle that must make decisions given the available sensor information), intelligent agents must be able to instantaneously reason about occluded objects *before* they re-appear.

Problem formulation: We begin by introducing benchmarks and metrics for evaluating the task of detecting and tracking invisible people. To do so, we repurpose existing tracking benchmarks and introduce metrics for evaluating this task that appropriately reward detection of occluded people. To ensure benchmarks are online, we forbid algorithms from accessing future frames when reporting object states for the current frame. Although this task requires reasoning about object trajectories, it can be evaluated as both a *detection* and a *tracking* problem. For the latter, we introduce extensions to tracking metrics in the supplement. When analyzing our metrics, it becomes readily apparent that human annotation of ground-truth occluded objects is challenging. We provide pilot human vision experiments in Section 4.4 that show annotators are still consistent, but exhibit larger variation in labeling the pixel position of occluded instances. This suggests that algorithms for occluded object detection should report *distributions* over object locations rather than precise discrete (bounding box) locations. Inspired by metrics for evaluating multimodal distributions in the forecasting literature [39], we explore probabilistic algorithms that make k predictions which are evaluated by Top-k accuracy.

Analysis: Perhaps not surprisingly, our first observation is that performance of state-of-the-art detectors and trackers plummets on occluded people, from 68.5% to 28.4%; it is far easier to detect visible objects than invisible ones! This underscores the need for the community to focus on this underexplored problem. We introduce two simple but key innovations for addressing

this task, which improve performance from 28.4% to 39.8%. (a) We recast the problem of online tracking of occluded objects as a *short-term forecasting* challenge. We explore state-of-the-art deep forecasting networks, but find that classic linear dynamics models (Kalman filters) perform quite well. (b) Because modeling occlusions is of central importance, we cast the problem as one of 3D tracking given 2D image measurements.

Novelty: While there exists considerable classic work on 3D tracking from 2D [26, 48, 249, 272], much focuses on 3D modeling of tracked objects. Instead, we find that the 3D structure of scene occluders is important for understanding where tracked objects can "hide". Typically such dense 3D understanding requires calibrated multiview sensors [64, 291]. Instead, we show that recent advances in uncalibrated *monocular depth estimation* provide "good enough" estimates of relative depth that still enable dense freespace reasoning. This is crucial because monocular depth has the potential to be far more scalable [303]. To our knowledge, ours is the first work to use uncalibrated depth estimates for multi-object tracking and detection of occluded objects.

Overview: After reviewing related work, we present our core algorithmic contributions, including straightforward but crucial extensions to classic linear dynamics models to (a) incorporate putative depth observations from a monocular network and (b) forecast object state even during occlusions. We conclude with extensive evaluations on three datasets [55, 204, 302] repurposed for detecting occluded objects.

4.2 Related Work

Amodal object detection aims to segment the full extent of objects that may be partially (but not *fully*) occluded. [374] introduces this task with a dataset labeled by multiple annotators, which is later expanded by [375]. More recently, [229] introduces a larger dataset of amodal annotations on the KITTI [91] dataset. Approaches in this setting largely rely on training variants of standard detectors (*e.g.* [100]) on amodal annotations generated synthetically from modal datasets [62, 170, 336, 367]. As this line of work addresses detection from a single image, it requires objects to be at least *partially visible*. By contrast, we target fully occluded people, which cannot be recovered from a single frame.

Multi-object tracking requires tracking across partial and full occlusions. Approaches for this task address occlusions post-hoc in an *offline* manner, using appearance-based re-identification models to identify occluded objects after they become visible. These appearance-based models can be incorporated into tracking approaches, as part of a graph optimization problem [16, 222, 355]

or online linking [17, 319]. In this work, we point out that some approaches *internally* maintain online estimates of the position of occluded people [17, 19, 319], but explicitly choose not to report these internal predictions, as they tend to be noisy and, thus, are penalized heavily by current benchmarks. We provide two simple extensions to these internal predictions that significantly improve detection of occluded people while preserving accuracy on visible people. [93] tracks occluded objects using contextual 'supporters', but requires a user to initialize a single object to track in uncluttered scenes; by contrast, we simultaneously detect and track people in large crowds.

Other work shares our motivation of tracking in 3D but relies on additional depth sensors [88] or stereo setups [38, 128]. Finally, many surveillance-based tracking systems explicity reason about object occupancy and occlusion, but require calibrated cameras to compute ground plane coordinates [2, 74, 126, 146, 154]. By contrast, our work emphasizes detection of *occluded* people in *uncalibrated, monocular* videos. To do so, we use monocular depth estimators via technical innovations that address noise in predicted depth estimates. Our method generalizes to arbitrary videos, since estimating monocular depth is far more scalable than retrieving additional sensor information for any video.

Forecasting approaches predict pedestrian trajectories in future, unobserved frames. These approaches leverage social cues from nearby pedestrians or semantic scene information to better model person trajectories [158, 165, 197, 219, 263, 334]. Recently, data-driven approaches have also been proposed for learning social cues [7, 250]. We note that detection of fully occluded people can be formulated as forecasting the trajectory of a visible person in future frames, where the positions of the occluded person are unobserved, but the rest of the frame *can* be observed. Some approaches do use forecasting to track objects [73, 196] but we use a constant-velocity model to forecast trajectories *along* with depth cues from the observed frames, to improve detection of occluded people. In Section 4.4, we show that while this approach can use a more powerful forecasting model, the constant-velocity approximation is sufficient in our setting.

4.3 Method

We build an online approach for detecting invisible people starting with a simple tracker, using estimated trajectories of visible people to forecast their location during occlusions. We describe our tracking mechanism, building upon [320]. While such trackers *internally* forecast the location of occluded people for improved tracking, these forecasts tend to be noisy and cannot directly

localize occluded people. To address this, we incorporate depth cues from a monocular depth estimator to reason about occlusions in 3D.

Background To detect people during occlusions, we build on a simple online tracker [320] that estimates the trajectories of visible people. We briefly describe aspects relevant to our approach, but refer the reader to [320] for a more detailed explanation. In the first frame, this tracker instantiates a track for each detected person. The tracker adds each track to its "active" set, representing people that have been seen so far. Each track maintains a Kalman Filter whose state space encodes the position (x, y), aspect ratio (a), height (h), and corresponding velocities ($\dot{x}, \dot{y}, \dot{a}, \dot{h}$) of the person.The filter's process model assumes a constant velocity model with gaussian noise (i.e., $x_t = x_{t-1} + \dot{x_{t-1}} + \epsilon_x$). At each successive frame, the tracker first runs the *predict* step of the filter, using the process model to forecast the location of the track in the new frame. Next, each detection in the current frame is matched to this set of active tracks based on appearance features, and distance to the tracks' forecasted location (as estimated by the filter). A new track is created for all detections that are unmatched. If a track is matched to a detection, the detection is used as a new observation to update the track's filter, and the detection is reported as part of the track. Importantly, if a track does not match to any detection, its forecasted box is *not* reported. When a track is not matched to a detection for more than N_{age} frames, it is deleted.

Short-term forecasting across occlusions Although this tracker *internally* forecasts the positions of all tracks at each step, its estimates are used only to improve the association of tracks to detections, and are not reported externally. However, these internally forecasted track locations are crucial as they may correspond to an occluded person. We show that naively reporting these track locations leads to significant *recall* of occluded people, but the noise in these estimates results in poor precision. Further, these noisy estimates lead to a small decrease in *overall* accuracy, as standard benchmarks largely focus on visible people. We improve these estimates by augmenting them with 3D information. Specifically, we use a monocular depth estimator [173] to get per pixel depth estimates of the scene. We then augment our Kalman Filter state space with the *inverse* depth. Inverse depth is a commonly used representation predicted by depth estimators [164, 173] due to important benefits, including the ability to represent points at infinity and ability to model uncertainty in pixel disparity space (commonly used for stereo-based depth estimation [212]). Our state space thus additionally includes $1/z$ variable.

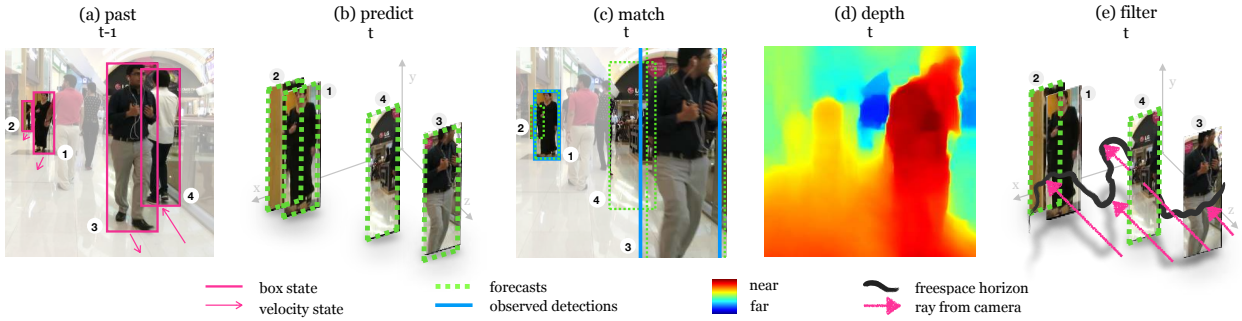


Figure 4.2. (a) Frame $t-1$ has active tracks $\{1, 2, 3, 4\}$, each with an internal state of its 2D position, size, velocity, and *depth* (see text). (b) We forecast tracks in 3D for frame t. (c) Tracks are matched to observed detections at t using spatial and appearance cues. Matched tracks are considered visible (*e.g.* 1, 3). Tracks which don't match to a visible detection (*e.g.* 2, 4) may be occluded, or simply incorrectly forecasted. (d) To resolve this ambiguity, we leverage depth cues from a monocular depth estimator, to compute (e) the *freespace horizon*. The region between the camera and the horizon must be freespace, while the area beyond it is unobserved, and so may contain *occluded* objects. Tracks lying beyond the freespace horizon are reported as occluded (*e.g.* 2). Tracks *within* freespace (*e.g.* 4) should have been visible, but did not match to any visible detections. Hence, we assume these tracks are incorrectly forecasted, and we delete them.

Tracking in 3D camera coordinates using 2D image coordinates Equipped with depth estimates, we formulate tracking with a constant velocity model in 3D using 2D measurements. Unlike prior work which assumes linear dynamics in (projected) 2D image measurements, our dynamics model operates in 3D using depth cues, resulting in far more realistic person trajectories. We derive our uncalibrated tracker by demonstrating that the unknown camera focal length f can be folded into a motion noise parameter that can be easily tuned on a training set. Hence our final method runs without calibration on arbitrary videos.

Let us model objects as cylinders with centroids (X_t, Y_t, Z_t), height H and aspect ratio A_t. We model object height as constant, but allow for varying aspect ratios because people are non-rigid. We can then compute image-measured bounding boxes with centroid (x_t, y_t) and dimensions (h_t, a_t) as follows:

$$x_t = f\frac{X_t}{Z_t}, \quad y_t = f\frac{Y_t}{Z_t}, \quad h_t = f\frac{H}{Z_t}, \quad a_t = A_t \tag{4.1}$$

We extend the commonly used constant velocity model with Gaussian noise from 2D [19, 319] to 3D:

$$X_t = X_{t-1} + \dot{X}_{t-1} + \epsilon_X, \quad \epsilon_X \sim \mathcal{N}(0, \sigma_X), \tag{4.2}$$

where similar equations hold for Y_t, Z_t and A_t. Let the observed (inverse) depth from a depth estimator associated with an object be $1/z_t$. Since image measurements are given by perspective projection of real world coordinates, we have the following equations (assuming Gaussian image noise):

$$x_t = f\frac{X_t}{Z_t} + \epsilon_x, \quad \epsilon_x \sim \mathcal{N}(0, \sigma_x) \tag{4.3}$$

$$\frac{1}{z_t} = \frac{1}{Z_t} + \epsilon_z, \quad \epsilon_z \sim \mathcal{N}(0, \sigma_z) \tag{4.4}$$

with similar equations for y_t, h_t, and a_t. Note that inverse depth naturally assumes a large uncertainty in far away regions, and a small uncertainty in nearby regions. Defining a 3D state space leads us to a modified formulation, written as $\left(f\frac{X_t}{Z_t}, f\frac{Y_t}{Z_t}, \frac{1}{Z_t}, A_t, f\frac{H}{Z_t}, f\frac{\dot{X}_t}{Z_t}, f\frac{\dot{Y}_t}{Z_t}, \dot{A}_t\right)$. We can therefore rewrite Equation (4.2) as:

$$f\frac{X_t}{Z_t} \approx f\frac{X_t}{Z_{t-1}} = f\frac{X_{t-1}}{Z_{t-1}} + f\frac{\dot{X}_{t-1}}{Z_{t-1}} + f\frac{\epsilon_X}{Z_{t-1}} \tag{4.5}$$

$$x_t \approx x_{t-1} + \dot{x}_{t-1} + f\frac{\epsilon_X}{Z_{t-1}} \tag{4.6}$$

where the approximation holds if depths are smooth over time ($Z_t \approx Z_{t-1}$). Technically, the above is no longer a linear dynamics model since the noise depends on the state. But the equation suggests that *one can approximately apply a Kalman filter on 2D image measurements augmented with a temporal noise model that is scaled by the estimated inverse-depth of the object.* Intuitively, this suggests that one should enforce smoother tracks for objects far away. Our approach thus scales the process noise (ϵ_X) for far away objects, leading to more accurate predictions. Algorithmically, [320] by default scales process and observation noise covariances according to the person's height; our approach instead multiplies the process covariance by the person's estimated depth, computed by aggregating past monocular depth observations and state estimates over time.

Assumptions. Because we do not assume calibrated cameras, we do not know f. Rather, we make use of training videos provided in standard tracking benchmarks and simply tune scaled variances $\sigma'_X = f\sigma_X$ directly on the training set. We make two additional assumptions: that people move with constant velocity in 3D, and that depth estimates are smooth over time. Although these do not always hold in real world scenarios, we empirically find that our method generalizes to diverse scenarios.

Filtering estimates lying in freespace. Equipping our state space with depth information

allows us to forecast 3D trajectories. Meanwhile, applying a monocular depth estimator allows us to determine regions in 3D space that are occluded to the camera without requiring calibration. Specifically, if our approach forecasts a person at a point $P_f = (x_f, y_f, z_f)$, we can determine whether P_f should be visible to the camera by estimating whether P_f lies in the freespace [64] between the camera and its nearest occluder. In the filter stage in Figure 4.2, we visualize one slice of the "freespace horizon": points beyond this horizon are occluded, while points between the camera and horizon are visible.

Concretely, let z_o be the (observed) depth of the horizon at (x_f, y_f). If the forecasted depth (z_f) lies closer to the camera than the horizon depth (z_o), as with person "4" in Figure 4.2 (e), then the person must be in the *freespace* between the camera and its closest object, and therefore visible. If we *do not* detect this person, then we assume the forecast is an error, and either suppress the forecasted box for the current frame (in the case of small errors, when $z_f < \alpha_{\text{supp}} z_o$) or delete the track entirely (for large errors, when $z_f < \alpha_{\text{delete}} z_o$). A key advantage of this approach is the ability to reason about occlusions arising not only from interactions between tracked people, but also from natural occluders such as trees or cars. Section 4.4 shows that this modification is critical for improving the precision of our trajectory forecasts.

Camera motion. Camera motion is challenging, as our approach assumes linear dynamics for trajectories. To address this, we follow prior work (e.g., [17]) in estimating a non-linear pixel warp W between neighboring frames which maps pixel coordinates (x_{t-1}, y_{t-1}) in one frame to the next (x_t, y_t). This warp is then used to align boxes forecasted using frames up to $t-1$ with frame t. Note that this alignment assumes the motion of dynamic objects is small relative to the scene motion, allowing for the use of an image registration algorithm [66]. Despite the simplicity of this modification, we show in the supplement that it helps considerably for the moving camera sequences. We also detail our algorithm with pseudo-code in the supplement. We proceed to an empirical analysis of the task and prior methods, showing the benefits of each component of our proposed approach.

4.4 Experiments

We first describe our proposed benchmarks, including the datasets and our proposed metrics for evaluating the task of detecting occluded people. Next, we conduct an oracle study in Section 4.4 to analyze how well existing approaches can detect occluded people. We then compare our proposed approach to these state-of-the-art approaches in multiple settings in Section 4.4. Finally,

Figure 4.3. We visualize bounding boxes labeled by multiple (4) in-house annotators (**left**). During small occlusions, annotators strongly agree. During large occlusions (less than 10% visible, last frame), annotators still agree to a fair extent (average IoU overlap of 60%, **right**), but require temporal video context. We use these to justify our Top-k evaluation and motivate our probabilistic tracking approach.

we analyze each component of our approach with a detailed ablation study in Section 4.4.

Dataset. Evaluating our approach is challenging, as most datasets do not annotate occluded objects. The MOT-17 [204], MOT-20 [55] and PANDA [302] datasets are key exceptions which label both visible and occluded people, along with a *visibility* field indicating what portion of the person is visible to the camera. We find that a majority of the annotations in these datasets (over 85% in each dataset) are people that are at least partially visible, leading standard evaluations on these datasets to underemphasize occluded people. To address this, we separately evaluate accuracy on the subset of fully *occluded* people (indicated by $< 10\%$ visibility). MOT-17 contains 7 sequences with publicly available groundtruth, and 7 test sequences with held-out groundtruth. We evaluate on these 14 sequences. MOT-20 contains 8 sequences, of which 4 have held-out groundtruth. PANDA officially releases a high-resolution 2FPS groundtruth for its 10 train and 5 test sequences. Because tracking and forecasting is challenging at such low frame rates, we reached out to the authors who provided a high-frame rate (30FPS), low-resolution groundtruth for 9 train videos. We report results on MOT-20 and PANDA train set without tuning our pipeline on any of the videos in these datasets. From visual inspection, we found that visibility labels in PANDA tend to be noisy (see the supplement), and so we define objects with up to 33% visibility as occluded. We carry out the analysis including oracle and ablation study on MOT-17 train and report the final results on MOT-17 test, MOT-20 and PANDA datasets. In all, these three datasets target a diverse set of application scenarios – static surveillance cameras, car-mounted cameras, and hand-held cameras.

Metric. As most benchmarks consist primarily of visible people, existing metrics which measure performance across all people underemphasize the accuracy of detecting occluded people. We propose detection and tracking metrics (see supplement for latter) which evaluate accuracy

Detections	Tracks	Occl Strat	Online?	Top-5				Top-1 F1	
				Occl F1	Occl Prec	Occl Rec	All F1	Occl	All
Groundtruth (vis.)	Groundtruth	Interpolate	✗	87.3 ±0.1	83.8 ±0.2	91.1 ±0.1	98.0 ±0.0	79.8	96.8
Faster R-CNN	Groundtruth	Interpolate	✗	46.4 ±0.1	65.5 ±0.1	35.9 ±0.1	70.5 ±0.0	34.4	68.1
Groundtruth (vis.)	DeepSORT	Interpolate	✗	53.3 ±0.2	86.7 ±0.1	38.5 ±0.2	92.3 ±0.0	44.4	92.0
Faster R-CNN	DeepSORT	Interpolate	✗	32.2 ±0.0	60.8 ±0.2	21.9 ±0.0	69.9 ±0.0	23.2	68.4
Faster R-CNN	DeepSORT	Forecast	✓	29.8 ±0.2	29.5 ±0.4	30.2 ±0.1	69.4 ±0.0	20.9	66.5

Table 4.1. Oracle ablations on MOT-17 train reporting Top-5 F1 and Top-1 F1 for occluded and all people, using Faster R-CNN detections. 'Occl strat' stands for Occlusion Strategy. We report the Top-5 mean and standard deviation for 3 runs.

on occluded people, as indicated by visibility $< 10\%$ and on all (visible and invisible) people. Since localizing fully-occluded people involves higher positional uncertainty than visible people, we allow algorithms to predict k potential locations for each person.

Top-k F1: We start by modifying the standard detection evaluation protocol [67, 181]. For every person, we allow methods to report k predictions, $P = \{p_1, p_2, \ldots, p_k\}$. We match these predictions to all groundtruth boxes based on intersection-over-union (IoU). We define the overlap between a groundtruth g and P as the maximum overlap with the predictions p_i in P — , $\text{IoU}(g, P) = \max_i \text{IoU}(g, p_i)$. We use this overlap definition and perform standard matching between predictions and groundtruth, with a minimum overlap threshold of α_{IoU}.

When evaluating accuracy across all people, matched groundtruth boxes are true positives (TP), all unmatched groundtruth are false negatives (FNs, or misses), and unmatched detections are false positives (FP). When evaluating accuracy on occluded people, only matched *occluded* groundtruth boxes count as TPs, only unmatched *occluded* groundtruth boxes count as FNs, and all unmatched detections count as FPs. Intuitively, when evaluating metrics for occluded people, we do not penalize a detector for correctly detecting a visible person, but we *do* penalize it for false positives that do not match any visible or occluded person.

We now describe how the k-vector of predictions is obtained: in addition to a state mean (first sample), our probabilistic method maintains covariances for x and z state variables which result in a 2D gaussian. Since these gaussians may extend incorrectly into freespace, we perform rejection sampling to accumulate k-1 predictions which respect freespace constraints. This gives us P. For baseline methods that are not probabilistic or do not have access to a depth map, we artificially simulate this distribution by tuning two scale factors that control the size of gaussians as a function of a bounding box's height. We tune these scale factors on MOT-17 train and use

them throughout experiments.

Top-1 F1: When $k = 1$, this metric is simply the standard F1 metric. We additionally report this Top-1 F1 for occluded and *all* people. We do not use the standard 'average precision' (AP) metric as most detectors and trackers on the MOT and PANDA datasets do not report confidences.

IDF1: To evaluate tracking, we report the standard IDF1 metric and also modify it for evaluating occluded people. Specifically, we divide the groundtruth tracks into visible and occluded segments, and perform matching only on the occluded segments. Once the tracks are matched, we compute IDTP as the number of matched occluded boxes, IDFP as the number of unmatched occluded *or* visible predictions, and IDFN as the number of unmatched occluded groundtruth boxes. We similarly modify MOTA in the supplement.

To guide evaluation, we conduct a human vision experiment with 10 in-house annotators who annotate 59 tracks with occlusions. Figure 4.3 shows that annotators have lower consistency when labeling occluded people than visible people. To address this ambiguity in localizing occluded people, we choose a low $\alpha_{IoU} = 0.5$ and $k = 5$ in our experiments.

Implementation details. We empirically set parameters in our approach on MOT-17 train with Faster R-CNN [246] detections. The optimal thresholds for filtering forecasts on the train set are $\alpha_{\text{delete}} = 0.88, \alpha_{\text{supp}} = 1.06$[1]. During occlusion we treat a person as a point, freezing its aspect ratio and height. We fix N_{age} to 30. The supplement presents further details of our method, parameters and their tuning protocol, including improvements by tuning N_{age}. We tune on MOT-17 train and apply these tuned parameters on MOT-17 test, MOT-20, and PANDA. We find that our method and its hyperparameters tuned on the train set generalize well to the test set. We use [173] for monocular depth estimates, which has been shown to work well in the wild. While these estimates can be noisy, we qualitatively find that the *relative* depth orderings used in our approach are fairly robust.

What is the impact of *visible* detection on occluded detection? We first evaluate an offline approach which uses groundtruth detections and tracks for visible people to (linearly) interpolate detections for occluded people in Table 4.10. As this method perfectly localizes visible people, and most people in this benchmark are visible, it achieves a high overall Top-5 F1 of 98.0 (Table 4.10, row 1). Additionally, despite using simple linear interpolation, this oracle also achieves a high Top-5 F1 of 87.3 for *invisible* people. This result indicates that although

[1]Note that $\alpha_{\text{supp}} > 1$ allows the forecasted depth to be closer to the camera than the observed depth, accounting for potential noise in the depth estimator to reduce the number of forecasts that are suppressed.

long-term forecasting of pedestrian trajectories may require higher-level reasoning [165, 197, 263], short-term occlusions may be modeled linearly.

Next, we evaluate the same approach with detections from a Faster R-CNN [246] model in place of groundtruth (Table 4.10, row 2). This leads to a significant drop in both overall and occluded accuracy, indicating that improvements in *visible* person detection can improve detection for invisible people. Finally, although Occluded Top-5 F1 drops, it is significantly above chance, suggesting that current detectors equipped with appropriate trackers can detect invisible people.

What is the impact of *tracking* on occluded detection? So far, we have assumed oracle linking of detections, allowing for linear interpolation of bounding boxes to detect people through occlusion. We now evaluate the impact of using an online tracker, equipped with re-identification, on detecting occluded people. Removing the oracle results in a drastic drop in accuracy: the Top-5 F1 score for occluded people drops by over 30 points (87.3 to 53.3, Table 4.10 row 3) using groundtruth detections, and 14 points with Faster R-CNN detections (46.4 to 32.2, Table 4.10 row 4). Despite this significant drop in Occluded Top-5 F1, the overall Top-5 F1 is significantly more stable (from 98.0 to 92.3 for groundtruth detections and 70.5 to 69.9 for Faster R-CNN), showing that *overall* person detection and tracking underemphasizes the importance of detecting occluded people.

Can online approaches work? These results indicate that in the offline setting, existing visible-person detection and tracking approaches can detect invisible people via interpolation. We now evaluate a simple *online* approach, which uses an off-the-shelf visible person detector (Faster R-CNN), equipped with a tracker (DeepSORT) and linear (constant velocity) forecasting for detecting invisible people (Table 4.10, row 5). Moving to an online setting results in a similar Top-5 F1 score but significantly reduces the precision for occluded persons, from 60.8 to 29.5. This is expected as even though linear forecasting recalls slightly more number of boxes than offline interpolation (recall from 21.9 to 30.2), its naive nature results in many more false positives resulting in a much lower precision and therefore, a similar F1 score. In Section 4.4, we present simple modifications to this approach that recover much of this performance gap.

Comparison to Prior Work Next, we apply our approach to the output of existing methods to evaluate its improvement over prior work. Table 4.11 shows results on the MOT-17 train set, showing our approach improves significantly in Occluded Top-5 F1 ranging from 6.0 to 13.0 points, while maintaining the overall F1. Detecting invisible people requires reliable amodal detectors for visible people (ref. Section 4.4). For this reason, we use *visible* groundtruth detections from

		Top-5 F1		Top-1 F1	
		Occl	All	Occl	All
MOT-17	DPM [71]	17.2	46.7	13.2	46.5
	+ Ours	24.6 (+7.4)	49.3 (+2.6)	17.4	48.4
	FRCNN [246]	28.4	68.5	20.1	67.4
	+ Ours	39.8 (+11.4)	70.5 (+2.0)	26.7	68.5
	SDP [338]	45.2	80.5	35.8	79.8
	+ Ours	51.2 (+6.0)	80.8 (+0.3)	38.5	79.4
	Tracktor++ [17]	32.4	77.0	22.7	76.8
	+ Ours	45.4 (+13.0)	77.2 (+0.2)	33.2	76.5
	MIFT [123]	37.8	75.9	29.9	75.1
	+ Ours	44.9 (+7.1)	75.6 (-0.3)	33.8	74.3
	CTrack [372]	38.7	84.8	29.4	84.2
	+ Ours	47.9 (+9.2)	84.4 (-0.4)	36.4	83.4
MOT-20	FRCNN	42.5	71.2	27.5	70.7
	+ Ours	46.1 (+3.6)	71.5 (+0.3)	28.6	70.9
PANDA	GT (visible)	45.5	90.6	30.5	90.5
	+ Ours	49.5 (+4.0)	90.5 (-0.1)	34.1	90.3

Table 4.2. Results on MOT-17 [204], MOT-20 [55] and PANDA [302] train. We evaluate on public detections provided with MOT-17 (DPM, FRCNN, SDP), two trackers that operate on public detections (Tracktor++, MIFT), and CenterTrack which does not use public detections. We use (public FRCNN, *visible* groundtruth) detections for (MOT-20, PANDA). Our method improves on occluded people across all trackers.

PANDA, similar to the oracle experiments in Section 4.4, as no public set of amodal detections come with PANDA (unlike MOT-17 or MOT-20). Table 4.11 shows that our method improves the detection of occluded people by 4.0% on PANDA using groundtruth visible detections and by 3.6% on MOT-20 using the Faster-RCNN public detections. We explicitly do not tune our hyperparameters for these two datasets, showing that our method is robust to changes in video data distribution. MOT-20 and PANDA contain a few sequences with top-down views, where occlusions are rare. We disable our depth and occlusion reasoning on such sequences; please see supplement.

As MOT-17 and MOT-20 test labels are held out, we worked with the MOTChallenge authors to implement our metrics on the test server. Table 4.12 shows that MIFT[2][123] and Tracktor++ [17] achieve the highest Occluded Top-5 F1 amongst prior online approaches on MOT-17 and MOT-20 test respectively. Applying our approach on top of these methods improves results

[2]MIFT is referred to as ISE_MOT17R on the MOT leaderboards

		Top-5 F1		Top-1 F1	
		Occl	All	Occl	All
MOT-17	Ours	43.4	76.8	31.4	75.6
	MIFT [123]	38.4	77.3	29.7	76.7
	UnsupTrack [137]	35.9	78.1	26.6	77.4
	GNNMatch [217]	35.2	74.3	26.3	73.7
	GSM_Tracktor [188]	35.4	73.8	26.2	73.2
	Tracktor++ [17]	33.3	73.3	24.8	73.0
MOT-20	Ours	46.9	76.7	33.3	75.2
	Tracktor++ [17]	44.2	76.0	34.2	75.3
	UnsupTrack [137]	41.7	71.4	30.9	70.8
	SORT20 [320]	38.5	65.2	27.3	63.6

Table 4.3. Results on MOT-17 and MOT-20 test set. The best, second-best and third-best methods are highlighted.

significantly by 5.0% to 43.4 F1 and by 2.7% to 46.9 F1, leading to a new state-of-the-art for occluded person detection on MOT-17 and MOT-20 test.

Table 4.11 shows that our method consistently improves occluded F1. However, it sometimes results in a drop in overall accuracy. We attribute this to the increased number of false positives introduced while tackling the challenging task of detecting invisible people. These false positives for invisible people are counted as false positives for *all* people, whether visible or invisible. This causes existing metrics to penalize methods for even *trying* to detect invisible people. In safety critical applications, where worst-case accuracy may be more appropriate, our approach significantly improves during complete occlusions by up to 13.0% on MOT-17, while mildly decreasing average accuracy by 0.4%.

Ablation Study We now study the impact of each component of our approach in Table 4.4, focusing on the Occluded Top-5 F1 metric using Faster R-CNN detections on the MOT-17 train set. First, we show that the DeepSORT tracker, upon which our approach is built, results in a 28.4 Occluded Top-5 F1. Reporting the internal, linear forecasts from the tracker increases the score to 29.8, driven primarily by a 12.5% improvement in recall. Compensating for camera motion provides another 2.4% improvement. Next, leveraging depth cues to incorporate freespace constraints, as detailed in Section 4.3, improves accuracy by 3.5%, driven primarily by a 14.6% jump in precision, indicating that this component drastically reduces false positives. Finally,

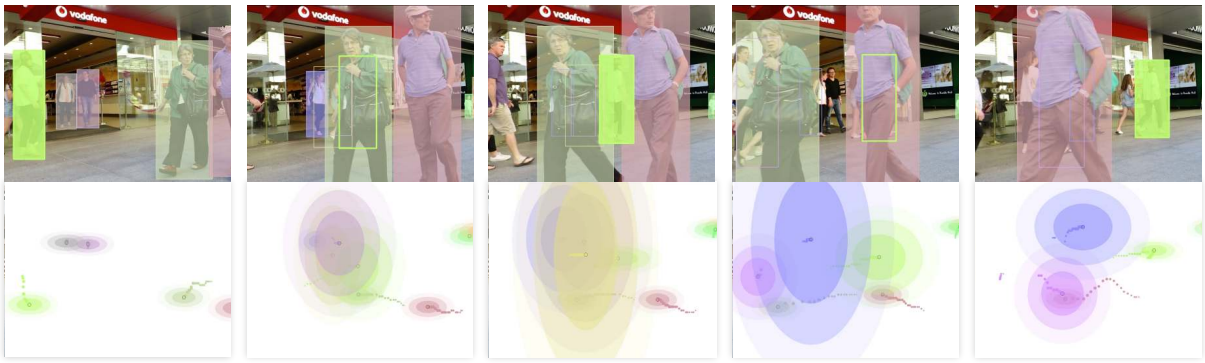


Figure 4.4. Our probabilistic model reports a *distribution* over 3D location during occlusions. We visualize (occluded, visible) detection with (outlined, filled-in) bounding boxes (**top**). We provide "birds-eye-view" top-down visualizations of Gaussian distributions over 3D object centroids with covariance ellipses (**bottom**). During occlusion, variance grows roughly linearly with the number of consecutively-occluded frames. We are also able to correctly predict depth of occluded people in the top down view, e.g. in the second last frame, which would not be possible with single-frame monocular depth estimates. During evaluation, we truncate the uncertainty using our freespace estimates (not visualized). Please refer to the supplement video.

we add depth-aware process noise to handle perspective transformations between 2D and 3D coordinates, which leads to an improvement of 4.1%, resulting in a final score of 39.8. Only a 1.0% improvement in F1 as compared to 4.1% with Top-5 F1 suggests that our uncertainty estimates are significantly improved by the depth-aware process noise scaling. In all, our approach leads to an improvement of 11.4% over the baseline. Figure 4.4 presents a sample result from our approach, where the person in the green bounding box is detected throughout two full occlusion phases, marked with an unfilled box.

One concern with our approach might be that the average depth inside a person's bounding box may contain pixels from the background or an occluder. To verify the impact of this, we evaluate a variant where we use segmentation masks for all the bounding boxes in MOT-17's FRCNN public detections using MaskRCNN [100]. We initialize the z state variable in the model with the average depth inside this mask. On doing so, the Top-1 occluded F1 increases from 26.7 to 27.3, indicating that masks can help with estimating the person's depth, but boxes are a reasonable approximation. We kindly refer the reader to our supplement for further ablative analysis, including an analysis of more recent depth estimators, ablations on moving stationary sequences, and failure cases (in supplementary video).

Forecasting: We evaluate replacing our linear forecaster with state-of-the-art forecasters. We

	Top-5				Top-1 F1	
	Occl F1	Occl Prec	Occl Rec	All F1	Occl	All
DeepSORT	28.4 ±0.1	71.9 ±0.2	17.7 ±0.1	68.5 ±0.0	20.1	67.4
+ Forecast	29.8 ±0.2	29.5 ±0.4	30.2 ±0.1	69.4 ±0.0	20.9	66.5
+ Egomotion	32.2 ±0.2	33.1 ±0.3	31.3 ±0.1	70.4 ±0.0	23.2	67.9
+ Freespace	35.7 ±0.0	47.7 ±0.1	28.6 ±0.0	70.4 ±0.0	25.7	68.4
+ Dep. noise	39.8 ±0.2	52.6 ±0.6	32.0 ±0.0	70.5 ±0.1	26.7	68.5

Table 4.4. MOT-17 train ablations. Each row adds a component to the row above. 'Dep. noise' is depth-aware noise.

supply these forecasters with a birds-eye-view representation of visible person trajectories. As these forecasters forecast only the birds-eye-view (x, z) coordinates, we rely on our approach's estimates of the height, width, and y coordinate. We evaluate two trajectory forecasting approaches for crowded scenes, Social GAN (SGAN) [98] and STGAT [125]. SGAN and STGAT result in Occluded Top-5 F1 scores of 36.0 and 36.4 respectively. While this improves over the baseline at 28.4, it underperforms our linear forecaster at 39.8. This suggests that simple linear models suffice for short, frequent occlusions. We refer the reader to the supplement for more details and analysis.

4.5 Discussion

We propose the task of detecting fully-occluded objects from uncalibrated monocular cameras in an online manner. Our experiments show that current detection and tracking approaches struggle to find occluded people, dropping in accuracy from 68% to 28% F1. Our oracle experiments reveal that interpolating across tracklets in an offline setting noticeably improves F1, but the task remains difficult because of large occlusions. We propose an online approach that forecasts the trajectories of occluded people, exploiting depth estimates from a monocular depth estimator to better reason about potential occlusions. Our approach can be applied to the output of existing detectors and trackers, leading to significant accuracy gains of 11% over the baseline, and 5% over state-of-the-art. We hope our problem definition and initial exploration of this safety-critical task encourages others to do so as well.

4.6 Appendix

In this supplement, we provide further analysis of our method and implementation details of our experiments. We present additional ablation studies of our method and extend our detection evaluation to tracking to use the popular IDF1 and MOTA (multi-object tracking accuracy) metrics. We also provide details regarding our human vision experiment, which analyzes people's ability to detect and localize highly occluded objects and discuss the experimental setup for PANDA and MOT-20 datasets, and Section 4.6 reports the runtime and presents pseudocode of our final depth-aware tracking algorithm.

Ablation Study In this section, we analyze the impact of using different depth estimators (Section 4.6), segmentation masks in place of bounding boxes for estimating average depth (Section 4.6), more sophisticated forecasters (Section 4.6), the performance of our method on moving stationary cameras (Section 4.6), and, finally, the importance of different hyperparameters (Section 4.6).

Monocular Depth Estimators Our method relies on an off-the-shelf monocular depth estimator to enable occlusion reasoning in 3D. In our main paper, we used the MegaDepth [173] estimator throughout our experiments. Here, we evaluate whether recent advances in monocular depth estimation provide more reliable *relative* depth estimates of people as used by our method. Specifically, we replace the MegaDepth estimator with the MannequinChallenge [174] and MIDAS [164] depth estimators in our method. We evaluate on MOT-17 using the Faster-RCNN set of public detections, and set all hyperparameters in our pipeline to their default values and disable the depth-aware noise scaling. This simple variant of our pipeline allows us to evaluate the quality of depth estimates from each of the three methods. Table 4.5 shows that the per frame depth estimator from Mannequin Challenge [174] does worse than MegaDepth [173] by 1.2 Top-5 F1 for invisible people and MIDAS [164] similarly does worse by 1.0 point. By the standard Top-1 F1 metric, these estimators degrade accuracy by 1.2 and 0.2 points respectively. As this simple variant of our pipeline is aimed at evaluating the relative depth orderings output from the depth estimators, these results suggest that while these depth estimators have become more accurate and generalizable over the years, the relative depth orderings of objects has not significantly improved.

	Top-5 F1		Top-1 F1		IDF1	
Depth est.	Occl	All	Occl	All	Occl	All
MegaDepth [173]	35.4±0.2	69.8±0.0	26.7	68.4	9.5	53.3
Mannequin [174]	34.2±0.2	69.4±0.0	25.5	68.0	8.5	53.3
MIDAS [164]	34.4±0.1	69.5±0.0	26.5	68.2	9.1	53.8

Table 4.5. Comparison of monocular depth estimators used in our pipeline. More recent depth estimators do not seem to provide more reliable *relative* depth orderings, which are used by our method.

		Top-5 F1		Top-1 F1		IDF1	
Depth	Res.	Occl	All	Occl	All	Occl	All
MIDAS	1x	34.4±0.1	69.5±0.0	26.5	68.2	9.1	53.8
MIDAS	2x	35.5±0.2	70.0±0.0	27.0	68.5	9.8	53.9
MIDAS	3x	37.5±0.2	69.9±0.0	27.0	68.2	10.8	53.9

Table 4.6. We evaluate a recent depth estimator, MIDAS [164], at varying input resolutions. At higher resolutions (3x), the estimator improves Top-5 F1 by 3.1 points, suggesting higher resolutions can improve depth estimates, likely by providing more reliable relative depths for faraway pedestrians.

Since monocular depth estimators can take as input images of varying sizes, we evaluate the effect of using higher resolution images as input to the estimator. Using a higher resolution input can increase the size of smaller objects in the scene (e.g., people far away), potentially allowing depth estimators to output more precise depth estimates. We evaluate using higher resolutions as input with the MIDAS [164] estimator in Table 4.6. By default, we resize images to a resolution of 512×384 pixels ('1x', the resolution MIDAS is trained with) from their original resolution of 1920×1080. We evaluate MIDAS [164] at 2× and 3× this default resolution and find in that doing so improves the Top-5 F1 for invisible peopleby 3.1%. We note here that this is not the case with the other two depth estimators [173, 174] whose performance decreases or stagnates with higher resolutions (not shown).

Boxes vs Masks

Our method estimates a person's depth by taking the average of the depth estimates within the person's bounding box. However, these pixels may contain background regions, leading to incorrect depth estimates. To address this, we evaluate a variant which uses an off-the-shelf instance segmentation method to only compute the average depth within a predicted person mask.

	Top-5 F1		Top-1 F1		IDF1	
	Occl	All	Occl	All	Occl	All
Boxes	39.8 ±0.2	70.5 ±0.1	26.7	68.5	10.5	54.8
Masks	40.6±0.3	71.3±0.0	27.3	69.1	11.0	54.7

Table 4.7. Replacing boxes by masks for getting mean depth of a person only helps by a small amount suggesting that boxes can reasonably replace masks.

To do this, we pass the Faster R-CNN public detections from MOT-17 as proposals into the mask head of Mask R-CNN [100]. Occasionally, this instance segmentation method may fail to produce a reasonable mask for a person. We design a simple strategy for detecting a common failure case: if the output segmentation mask covers less than 25% of the bounding box (in cases where the people are too small or out-of-distribution), we discard the predicted mask and treat the full bounding box as the mask. We do not use masks for the forecasted boxes of occluded people, as these boxes cover unknown occluders. In Table 4.7, we find that masks modestly help our method, increasing Top-5 and Top-1 F1 by 0.6 and 0.8 points for occluded people. Interestingly, we also see an increase in overall F1 by the same amount.

Forecasting Approaches

As described in the main paper, we use a constant velocity forecaster in our probabilistic approach. In Sec 4.3, we showed that replacing our our simple linear forecaster with more sophisticated state-of-the-art forecasters that exploit social cues did not improve performance. Here, we provide implementation details for these experiments, and analyze different variants. The approaches discussed in the main paper, SGAN [98] and STGAT [125] are supplied the top-down views from our algorithm. Both SGAN and STGAT forecast 20 samples and then choose the closest trajectory to the groundtruth from these 20. This advantage is not feasible for an online approach where groundtruth cannot be supplied to the algorithm. To simulate the online setting, we sample the mean trajectory from these approaches by requesting the trajectory corresponding to the zero noise vector. We calculate an approximate average scale factor of 20.0 between the trajectory values learnt by these models and the trajectory values available for input from our method, which we use to scale down our input values. Additionally, each of these methods has an 8- and 12-timestep forecasting model. In the main paper, we report the best of these models for both approaches and report other models in Table 4.8. For STGAT, the 8- and 12-timestep models

		Top-5 F1		Top-1 F1		IDF1	
		Occl	All	Occl	All	Occl	All
Single	SGAN-8	35.4±0.2	70.2±0.0	24.6	67.8	8.9	54.3
	SGAN-12	35.0±0.1	70.1±0.0	24.2	67.7	8.7	54.2
	STGAT-8	35.1±0.1	70.1±0.0	24.5	67.6	8.6	54.3
	STGAT-12	35.6±0.2	70.3±0.0	24.7	67.9	9.1	54.4
Multi	SGAN-8	36.0±0.2	70.3±0.0	24.8	67.9	9.2	54.4
	SGAN-12	36.0±0.3	70.3±0.0	24.9	67.9	9.3	54.4
	STGAT-8	36.2±0.3	70.3±0.0	24.5	67.8	8.8	54.3
	STGAT-12	36.4±0.1	70.4±0.0	24.8	67.9	9.2	54.4

Table 4.8. MOT-17 train forecasting ablations with state-of-the-art social forecasting models.

used are trained on the ETH [219] dataset and for SGAN, the 8- and 12-timestep models are trained on the ZARA1 [169] dataset. Each of these models is made to predict for 30-timesteps by supplying the last 8 forecasted timesteps iteratively. The occlusion phase may not last 30 timesteps for all people. We therefore use the information from our pipeline about the number of occluded timesteps and replace the x and z values from the output of our pipeline with SGAN and STGAT's forecasted x and z values. In Table 4.8, we additionally report the performance of the methods when we provide past trajectories of *multiple* people as input, allowing the method to leverage social cues. For the Top-5 evaluation, we use the blind baseline described in Sec. 4 of our main paper. The conclusion remains that simple linear models suffice for short, frequent occlusions as our approach always performs better than any of the social forecasting settings of SGAN and STGAT.

Moving vs Stationary Camera Sequences

In the MOT-17 dataset, 3 camera sequences are stationary and 4 are captured from a moving camera. We separately study the effect of using different components of our pipeline on these sets of camera sequences. Table 4.9 shows that compensating for camera egomotion and filtering estimates lying in freespace helps the moving camera sequences by 4.5% and 4.0% Occluded Top-5 F1 respectively while for the stationary camera sequences, enforcing smoother tracks for faraway objects and filtering freespace estimates helps by 3.6% and 2.0% F1 respectively.

	Top-5				Top-1 F1		IDF1	
	Occl F1	Occl Prec	Occl Rec	All F1	Occl	All	Occl	All
Moving sequences								
DeepSORT	27.3 ±0.3	49.7	18.8	72.4 ±0.0	17.3	67.0	2.2	56.5
+ Forecast	21.3 ±0.1	15.4	34.6	68.4 ±0.1	13.3	63.6	5.6	50.2
+ Egomotion	25.8 ±0.0	19.4	38.7	71.3 ±0.0	17.1	66.9	8.7	53.2
+ Freespace	29.8 ±0.3	28.0	31.8	72.8 ±0.0	19.9	69.2	9.4	55.2
+ Dep. noise	34.3 ±0.1	32.8	35.9	73.3 ±0.1	20.2	69.4	9.8	55.9
Stationary sequences								
DeepSORT	29.2 ±0.1	94.0	17.3	66.2 ±0.0	21.7	65.9	1.1	55.0
+ Forecast	39.1 ±0.4	62.2	28.5	70.2 ±0.0	28.7	68.6	10.1	55.4
+ Egomotion	38.0 ±0.1	60.2	27.8	69.8 ±0.0	28.5	68.5	9.6	55.3
+ Freespace	40.0 ±0.0	76.1	27.1	68.9 ±0.0	30.3	67.9	10.0	54.9
+ Dep. noise	43.6 ±0.3	78.7	30.2	68.8 ±0.0	31.4	67.9	11.2	54.1

Table 4.9. MOT-17 train ablations for moving stationary camera sequences.

Hyperparameter tuning We describe a few parameters of our approach and how to tune them, in addition to the ones described in the paper. The N_{age} parameter in our pipeline controls the number of frames that an occluded track is forecasted for before it is deleted. We show in Figure 4.5 that the DeepSORT baseline is largely invariant to this parameter, as it does not report its internal forecasts. Reporting these estimates, whether directly (corresponding to 'DeepSORT+Forecast') or with our approach (corresponding to 'Our Pipeline'), highlights the impact of the parameter. This behaviour results in a precision-recall 'curvelet' which shows that by increasing N_{age}, we can trade-off the precision and recall for invisible people detection. The difficulty of this task can be highlighted by the trend that increasing N_{age} hardly increases recall beyond a point but instead decreases precision dramatically because of the introduction of many false positive boxes in the scene. We use the number of frames as a surrogate for uncertainty, as we find that this correlates well with the uncertainty estimated by the Kalman Filter, as shown in Figure 4 in the main paper.

We use a hyperparameter $f_{process}$ to scale the process noise covariance (refer Section 3.3 in the main paper). We additionally scale the observation noise covariance by $f_{observation}$ to account for the removal of default scaling by height of [320]. In our algorithm, we use $f_{process} = 900$ and $f_{observation} = 600$.

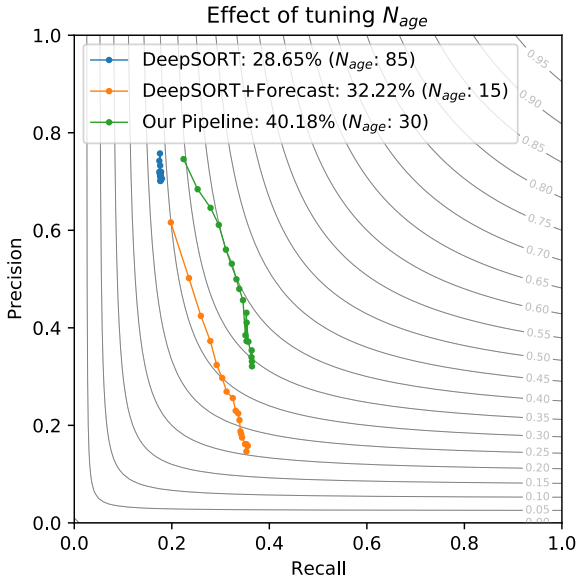


Figure 4.5. Detecting occluded people is sensitive to the threshold used to declare a detection-under-high-occlusion. We fix the number of N_{age} frames that a track is allowed to be in an occluded state. By increasing N_{age}, we can tradeoff precision and recall in invisible-people-detection which results in a "PR-curvelet". The curvelets represent the experiments in rows 1, 2 and 5 of Table 4 in the main paper.

IDF1-Occluded & MOTA-Occluded In the main paper, we report detection results using the probabilistic and standard F1 metrics. Here, we supplement these results with the IDF1 and MOTA (Multi-Object Tracking Accuracy) tracking metrics [18]. To do this, we follow the strategy in the main paper: We do not penalize tracks that match to visible people, but we reward only tracks that match to occluded people.

IDF1: To evaluate tracking, we report the standard IDF1 metric and also modify it for evaluating occluded people. Specifically, we divide the groundtruth tracks into visible and occluded segments, and perform matching only on the occluded segments. Once the tracks are matched, we compute IDTP as the number of matched occluded boxes, IDFP as the number of unmatched occluded *or* visible predictions, and IDFN as the number of unmatched occluded groundtruth boxes. In Tables 4.10, 4.11, 4.12, 4.13, we show that we improve the tracking of occluded people by a large margin (upto 14.3%) while maintaining the overall tracking performance. The conclusions in all cases remain the same as the detection metrics, except for the peculiar case of PANDA where we see an 8.1% drop in the overall IDF1 metric. We attribute this to the small size of people in PANDA and the top-down camera viewpoint which changes the distribution of the depth estimates returned by the monocular depth estimator. By tuning noise parameters to

Detections	Tracks	Occl Strat	Online?	IDF1	
				Occl	All
Groundtruth (vis.)	Groundtruth	Interpolate	✗	77.8	96.7
Faster R-CNN	Groundtruth	Interpolate	✗	20.5	67.4
Groundtruth (vis.)	DeepSORT	Interpolate	✗	21.3	81.0
Faster R-CNN	DeepSORT	Interpolate	✗	6.4	53.3
Faster R-CNN	DeepSORT	Forecast	✓	7.6	53.3

Table 4.10. Supplementary oracle ablations on MOT-17 train.

adapt to this new distribution, we can recover 6.9% of this drop.

MOTA: In addition to reporting standard MOTA, we modify it for occluded tracks by counting detections matched to occluded groundtruth as true positives (TP), unmatched detections as false positives (FP), and unmatched groundtruth as false negatives (FN), and only count ID-switches (IDS) for tracks corresponding to occluded groundtruth. Perhaps surprisingly, we find in Table 4.13 that the MOTA metric is negative for all ablations. To better understand this, we note that MOTA is a simple combination of TP, FP, IDS, divided by the total number of groundtruth boxes (GT):

$$\text{MOTA} = 1 - \frac{\sum_t \text{FP}_t + \text{FN}_t + \text{IDS}_t}{\sum_t \text{GT}_t}$$

Thus, a method which simply reports no tracks will achieve a MOTA of 0 (as $\text{FP} = 0, \text{FN} = \text{GT}, \text{IDS} = 0$), seemingly outperforming all approaches in Table 4.13. This suggests MOTA penalizes methods for even *trying* to detect occluded people. In general, if a tracker produces more false positives than true positives, MOTA will always be negative! This indicates that MOTA is not an appropriate metric for challenging tasks, such as detecting occluded people.

Human Vision Experiment In the main paper, we briefly described our human vision experiment to understand the challenges in detecting occluded people, and to motivate our evaluation and probabilistic approach. We provide further details here. We ask 10 in-house annotators to label fully occluded people in the MOT-17 [204] training set. To focus annotation effort on occluded people, we sampled track segments (1) containing at least 10 contiguous occluded frames, preceded by (2) 10 frames where the person is visible (and at least one where the person has $> 70\%$ visibility). Additionally, we avoid annotating small people (< 20 pixels on

Table 4.11. Supplementary tracking results on MOT-17 [204], MOT-20 [55] and PANDA [302] train.

		IDF1	
		Occl	All
MOT-17	DPM	2.9	36.9
	+ Ours	7.2	36.8
	FRCNN	1.5	55.6
	+ Ours	10.5	54.8
	SDP	10.9	64.6
	+ Ours	17.0	64.7
	Tracktor++	1.3	65.1
	+ Ours	15.6	66.8
	MIFT	9.4	61.7
	+ Ours	16.5	62.6
	CTrack	5.4	65.0
	+ Ours	16.2	70.2
MOT-20	FRCNN	2.9	42.2
	+ Ours	5.0	42.0
PANDA	GT (visible)	2.5	70.2
	+ Ours	4.6	62.1

either side), and limit the number of total frames in a segment to 50.

Annotators labeled at 10 fps (every 3rd frame in a 30fps video) in a simulated *online* setup. When an annotator is asked to label frame t, she has access to past frames (before t), but *not* future frames $> t$. Once the annotator submits a label for t, she is shown the next frame to label, and is no longer allowed to edit the annotation for frame t.

Overall, 10 people labeled a total of 113 tracks, 46 of which were unique. This resulted in a total of 991 annotated boxes. Our key finding was that even for complete occlusions (less than 10% visibility), annotators still agreed to a fair extent (60% IoU-agreement), making the problem harder than localizing visible people, but still feasible for humans. To account for these observations, we evaluate with our invisible-people detection metric at an IoU of 0.5.

PANDA and MOT-20 We first discuss the quality of visibility labels in PANDA followed by the criteria we follow for disabling the depth and freespace reasoning in our method for a subset of videos in PANDA [302] and MOT-20 [55].

PANDA classifies the visibility of people into 4 discrete classes – 'without occlusion', 'partial occlusion', 'heavy occlusion' and 'disappearing'. According to the dataset authors, these corre-

		IDF1	
		Occl	All
MOT-17	Ours	14.7	58.7
	MIFT [123]	10.4	56.4
	UnsupTrack [137]	9.7	62.6
	GNNMatch [217]	6.9	56.1
	GSM_Tracktor [188]	7.4	57.8
	Tracktor++ [17]	5.2	55.1
MOT-20	Ours	11.2	51.1
	Tracktor++ [17]	10.2	48.8
	UnsupTrack [137]	9.6	50.6
	SORT20 [320]	8.8	45.1

Table 4.12. Supplementary tracking results on MOT-17 and MOT-20 test set. The best, second-best and third-best methods are highlighted.

spond to 100%, 66%, 33% and 0% visibility labels on a continuous 0-100 scale. On qualitative inspection, we find that most 33% visible people in PANDA are fully-occluded (by our definition of $< 10\%$ visibility). Though the visibility annotation protocol is not detailed in the paper, we hypothesize that this anomaly exists because only those people are marked with 0% visibility which strictly have 0 visible pixels. Some examples are shown in Figure 4.6. Owing to this, we set the threshold of calling a person invisible in the PANDA dataset as 33% visibility.

Some sequences in PANDA and MOT-20 are top-down view videos where occlusions are unlikely to occur. In such sequences, we revert to using the standard DeepSORT tracker. For MOT-20, we disable our method on two sequences captured from a camera mounted at a high height based on visual inspection. For the PANDA dataset, which specifies the building floor on which the camera is mounted, we use DeepSORT for cameras mounted on or above the 8th floor. We note that this decision can be easily made in the real world by practitioners based on the height of the camera.

Runtime & Pseudo-code We precompute depth maps and detection features at 4.5 FPS and 11 FPS respectively. These are used in an online manner by our pipeline that runs at 4 FPS without explicit optimization. In Algorithm 1, we present the pseudocode of our approach.

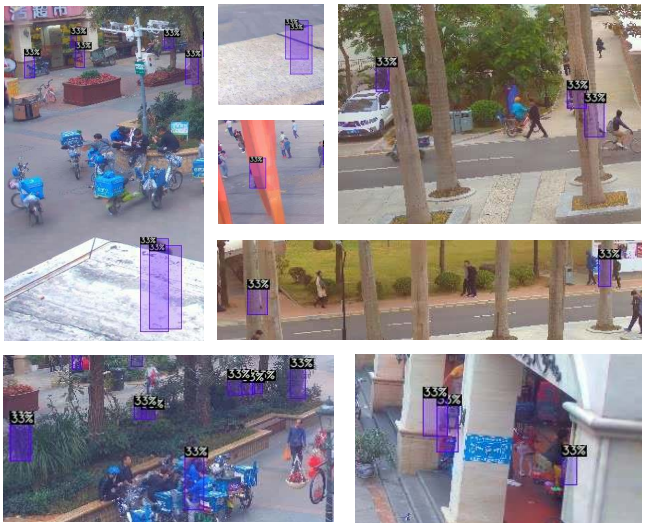


Figure 4.6. 'Heavy occlusion' or 33% visibility labels in PANDA are closer to the $< 10\%$ visibility labels in the MOT-17 and MOT-20 datasets. For this reason, we set the visibility threshold in the PANDA dataset to 33%.

	IDF1		MOTA	
	Occl	All	Occl	All
DeepSORT	1.5	55.6	-11.9	49.4
+ Forecast	7.6	53.3	-85.7	42.0
+ Egomotion	9.1	54.5	-72.1	44.6
+ Freespace	9.7	55.0	-35.2	48.1
+ Dep. noise	10.5	54.8	-31.5	48.5

Table 4.13. Analysis of IDF1- and MOTA-occluded for the MOT-17 train ablations. Note that MOTA is not useful for distinguishing trackers for difficult tasks, as it leads to negative values (while an approach which reports no detections would achieve MOTA of 0).

Algorithm 1 Invisible-people Kalman Tracker

Data: Detections $\mathcal{D}$ in current frame, $f_i \in \mathcal{F}$, the set of all frames
Result: Set of active tracks, $\mathcal{T} = \{t_1, \ldots, t_k\}$ s.t. $t_j \in \{\mathcal{T}_{occluded}, \mathcal{T}_{visible}\}$

def *update()*:
  X, Y1, Y2, Z = *match()*;
  Update the tracks with the KF Update step for all pairs in X;
  Initialise new tracks for Z;
  Increase age of all tracks in Y1;
  Add Y2 to $\mathcal{T}_{occluded}$;
def *match()*:
  Compare forecasted depth, z_f with horizon depth, z_o;
  If $z_f < \alpha_{supp} z_o$, keep track in $\mathcal{T}_{visible}$ but don't output;
  Else, trigger occluded state logic by adding track to $\mathcal{T}_{occluded}$;
  Bipartite-match detections to active tracks to based on last-known appearance;
  Match unclaimed visible tracks to unclaimed detections using IoU;
  Let X be matched tracks and detection;
  Let Y be unclaimed tracks;
  Let Z be unclaimed detections;
  Separate Y into visible (Y1) and occluded (Y2) tracks;
  for *all tracks in Y2* **do**
    If $z_f < \alpha_{delete} z_o$, delete track;
  end
  Return X,Y1,Y2,Z;
def *predict()*:
  Find warp marix W between current and past frame;
  for *all active tracks* **do**
    Warp the mean of current tracker state with the warp matrix;
    Assume a Constant Velocity Model;
    If track is occluded, assume no velocity for a and h;
    Else, assume constant velocity for a and h;
    Assume temporal process noise for all state variables (e.g., process noise $f\frac{\epsilon_X}{Z}$ for x);
    Carry out the KF Predict step to get a new state from the warped state;
  end
def *main()*:
  for *every incoming frame* **do**
    predict new states for all tracks using *predict()*;
    update all tracks with detections from the current frame using *update()*;
    output all active tracks that are either currently occluded or visible;
  end

Chapter 5

Large reconstruction models for novel-view synthesis from sparse-views

Publication information

Wang, Z., Tan, J., Khurana, T., Peri, N. and Ramanan, D., 2025. MonoFusion: Sparse-View 4D Reconstruction via Monocular Fusion. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).

5.1 Introduction

Accurately reconstructing dynamic 3D scenes from multi-view videos is of great interest to the vision community, with applications in AR/VR [182, 269], autonomous driving [191], and robotics [305, 370]. Prior work often studies this problem in the context of dense multi-view videos, which require dedicated capture studios that are prohibitively expensive to build and are difficult to scale to diverse scenes in-the-wild. In this paper, we aim to strike a balance between the ease and informativeness of multi-view data collection by reconstructing skilled human behaviors (e.g., playing a piano and performing CPR) from four equidistant inward-facing static cameras (Fig. 5.1).

Problem setup. Despite recent advances in dynamic scene reconstruction [42, 81, 82, 83], current approaches often require dozens of calibrated cameras [134, 194], are category specific [339], or struggle to generate multi-view consistent geometry [177]. We study the problem of recon-

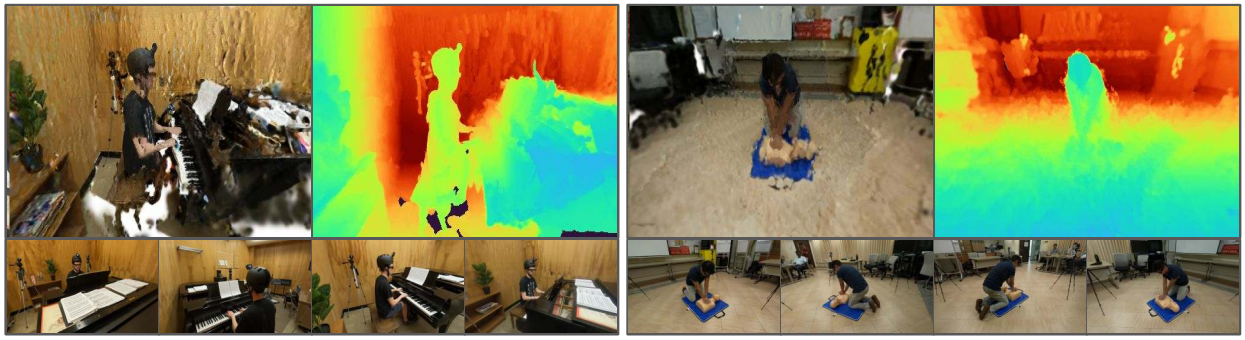


Figure 5.1. **Dynamic Scene Reconstruction from Sparse Views**. MonoFusion reconstructs dynamic human behaviors, such as playing the piano or performing CPR, from four equidistant inward-facing static cameras. We visualize the RGB and depth renderings of a 45° novel view between two training views. Training views are shown below for reference.

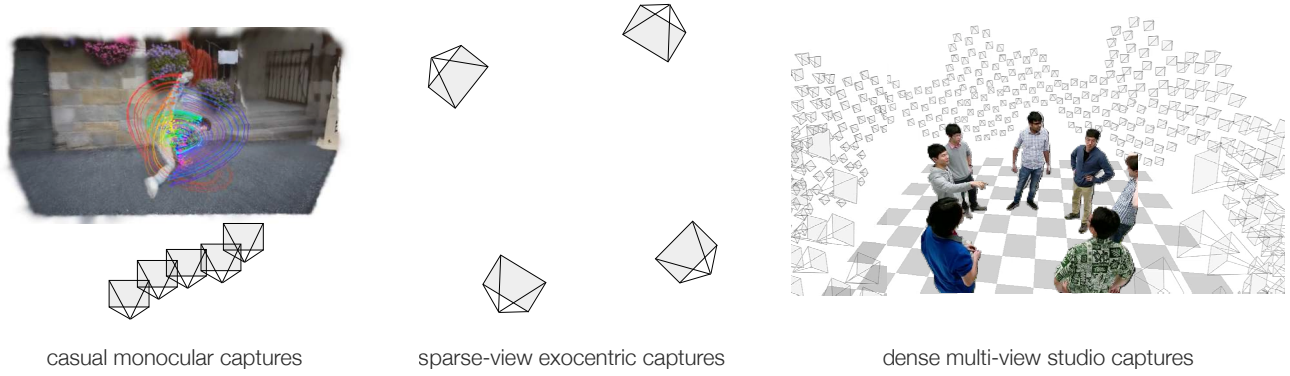


Figure 5.2. **Problem Setup.** Our sparse-view setup (**middle**) strikes a balance between ill-posed reconstructions from casual monocular captures [83, 225] and well-constrained reconstructions from dense multi-view studio captures [134]. Unlike existing "sparse-view" datasets like DTU [129] and LLFF [205], our setup is more challenging because input views are 90° apart with limited cross-view correspondences.

structing dynamic human behaviors from an *in-the-wild capture studio*: a small set of (4) portable cameras with limited overlap but complete scene coverage, such as in the large-scale Ego-Exo4D dataset [94]. We argue that sparse-view limited-overlap reconstruction presents unique challenges not found in dense multi-view setups and typical "sparse view" captures with large covisibility (Fig. 5.2). For dense multi-view captures, it is often sufficient to rely solely on geometric and photometric cues for reconstruction, often making use of classic techniques from (non-rigid) structure from motion [76]. As a result, these methods fail in sparse-view settings with limited cross-view correspondences.

Key insights. We find that initializing sparse-view reconstructions with monocular geometry estimators like MoGe [300] produces higher quality results. However, naively merging independent monocular geometry estimates often yields inconsistent geometry across views (e.g. duplicate structures), resulting in a local minima during 3D optimization. Instead, we carefully align monocular reconstructions (that are *independently* predicted for each view and time) to a global reference frame that is learned from a *static* multi-view reconstructor (like DUSt3R [301]). Furthermore, many challenges in inferring view-consistent and time-consistent depth become dramatically simplified when working with *fixed cameras with known poses* (inherent to the in-the-wild capture setup that we target). For example, temporally consistent background geometry can be enforced by simply averaging predictions over time.

Contributions. We present three major contributions.

- We highlight the challenge of reconstructing skilled human behaviors in dynamic environments from sparse-view cameras in-the-wild.
- We demonstrate that monocular reconstruction methods can be extended to the sparse-view setting by carefully incorporating monocular depth and foundational priors.
- We extensively ablate our design choices and show that we achieve state-of-the-art performance on PanopticStudio and challenging sequences from Ego-Exo4D.

5.2 Related Work

Dynamics scene reconstruction. Dynamic scene reconstruction [42] has received significant interest in recent years. While classical work [60, 210] often relies on RGB-D sensors, or strong domain knowledge [33, 54], recent approaches [177] based on neural radiance fields [206] have progressed towards reconstructing dynamic scenes in-the-wild from RGB video alone. However, such methods are computationally heavy, can only reconstruct short video clips with limited dynamic movement, and struggle with extreme novel view synthesis. Recently, 3D Gaussian Splatting [143, 194] has accelerated radiance field training and rendering via an efficient rasterization process. Follow-up works [179, 321, 345] repurpose 3DGS to reconstruct dynamic scenes, often by optimizing a fixed set of Gaussians in canonical space and modeling their motion with deformation fields. However, as Gao et al. [83] points out, such methods often struggle to reconstruct realistic videos. Many works address this shortcoming by relying on 2D point tracking priors [299], fusing Gaussians from many timesteps [168], modeling isotropic Gaussians [274],

or exploiting domain knowledge such as human body priors [279]. However, these approaches study the reconstruction problem in the monocular setting. As 4D reconstruction from a single viewpoint is under-constrained, practical robotics setups for manipulation [147] and hand-object interaction [70, 160, 296] adopt camera rigs where a sparse set of cameras capture the scene of interest. Similarly, datasets like Ego-Exo4D [94], DROID [147] and H2O [160] explore sparse-view capture for dynamic scenes in-the-wild.

Novel-view synthesis from sparse views. Both NeRF and 3D Gaussian Splatting require dense input view coverage, which hinders their real-world applicability. Recent works aim to reduce the number of required input views by adding additional supervision and regularization, such as depth [56] or semantics [351]. FSGS [376] builds on Gaussian splatting by producing faithful static geometry from as few as three views by unpooling existing Gaussians and adopting extra depth supervision. Recent studies such as [337], on the other hand, adds noise to Gaussian attributes and relies on a pre-trained ControlNet [366] to repair low-quality rendered images. Other works such as MVSplat [46] build a cost volume representation and predict Gaussian attributes in a feed-forward manner. However, they can only synthesize novel views with small deviations from the nearest training view. For methods that rely on learned priors, high-quality novel view synthesis is often limited to images within the training distribution. Such methods cannot handle diverse real-world geometry. Diffusion-based reconstruction methods [87, 325, 368] try to generate additional views consistent with the sparse input views, but often produce artifacts. In our case, four sparse view cameras are separated around 90° apart, posing unique challenges.

Feed-forward geometry estimation. Learning-based methods, such as monocular depth networks, are able to reconstruct 3D objects and scenes by learning strong priors from training data. While early works [63, 77] focus on in-domain depth estimation, recent works build foundational depth models by scaling up training data [238, 300, 341], resolving metric ambiguity from various camera models [115, 221], or relying on priors such as Stable Diffusion [78, 139, 251]. Unfortunately, monocular depth networks are not scale or view consistent, and often require extensive alignment against ground-truth to produce meaningful metric outputs. To address these shortcomings, DUSt3R [301] and MonST3R [363] propose the task of point map estimation, which aims to recover scene geometry as well as camera intrinsics and extrinsics given a pair of input images. These methods unify single-view and multi-view geometry estimation, and enable consistent depth estimation across either time or space.

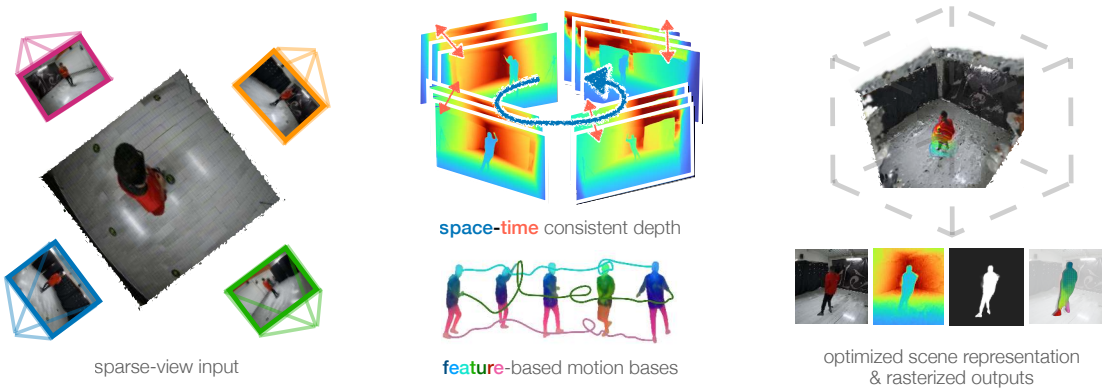


Figure 5.3. **Approach.** Given sparse-view video sequences of a scene (left), we aim to optimize a 3D gaussian representation over time. We begin by running DUSt3R [301], a *static* multi-view reconstruction method, on the sparse views of a given reference timestamp. This generates a global reference frame that connects all views. Next, we use MoGe [300] to independently predict depth maps for each camera. Since these depth predictions are only defined up to an *affine transformation*, we must estimate a scale and shift for each predicted depth map across all views and time instants. To achieve this, we leverage the fact that background pixels remain static over time. Specifically, for each time instant and each view, we align the background regions of each camera's depth map to the global reference frame by adjusting the scale and shift parameters accordingly (middle, top). This process requires a foreground-background mask for all input videos (which can be obtained using off-the-shelf tools like SAM 2 [241]). To reduce occlusions and noisy depth predictions, we concatenate all aligned background depth points, and average corresponding background points (where correspondence across time is trivially given by the 2D pixel index of the unprojected pointmap) across time. Lastly, we find that motion bases constructed from feature-clustering form a more geometrically consistent set of bases (middle, bottom), than those initialized by noisy 3D tracks [299]. Our optimization yields a 4D scene representation from which we can rasterize RGB frames, depth maps, a foreground silhouette, and object features from novel views (right).

5.3 Method

Given sparse-view (i.e. 3 – 4) videos from stationary cameras as input, our method recovers the geometry and motion of a dynamic 3D scene (Fig. 5.3). We model the scene as canonical 3D Gaussians (Sec. 5.3), which translate and rotate via a linear combination of motion bases. We initialize consistent scene geometry by carefully aligning geometry predictions from multiple views (Sec. 5.3), and initialize motion trajectories by clustering per-point 3D semantic features distilled from 2D foundation models (Sec. 5.3). We formulate a joint optimization which simultaneously recovers geometry and motion (Sec. 5.3).

3D Gaussian Scene Representation

We represent the geometry and appearance of dynamic 3D scenes using 3D Gaussian Splatting [143], due to its efficient optimization and rendering. Each Gaussian in the canonical frame t_0 is parameterized by $(\mathbf{x}_0, \mathbf{R}_0, \mathbf{s}, \alpha, \mathbf{c})$, where $\mathbf{x}_0 \in \mathbb{R}^3$ is the Gaussian's position in canonical frame, $\mathbf{R}_0 \in \mathbb{SO}(3)$ is the orientation, $\mathbf{s} \in \mathbb{R}^3$ is the scale, $\alpha \in \mathbb{R}$ is the opacity, and $\mathbf{c} \in \mathbb{R}^3$ is the color. The position and orientation are time-dependent, while the scale, opacity, and color are persistent over time. We additionally assign a semantic feature $\mathbf{f} \in \mathbb{R}^N$ to each Gaussian (Sec. 5.3), where $N = 32$ is an arbitrary number representing the embedding dimension of the feature. Empirically, we find that fixing the color and opacity of Gaussians results in a better performance. In summary, for the i-th 3D Gaussian, the optimizable attributes are given by $\Theta^{(i)} = \{\mathbf{x}_0^{(i)}, \mathbf{R}_0^{(i)}, \mathbf{s}^{(i)}, \mathbf{f}^{(i)}\}$. Following [371], the optimized Gaussians are rendered from a given camera to produce an RGB image and a feature map using a tile-based rasterization procedure.

Space-Time Consistent Depth Initialization

Similar to recent methods [278, 299], we rely on data-driven monocular depth priors to initialize the position and appearance of 3D Gaussians over time. Given the success of initializing 3DGS with monocular depth in single-view settings [299], one might think to naturally extend this to multi-view settings by independently initializing from monocular depth for each view. However, this yields conflicting geometry signals, as monocular depth estimators commonly predict up to an unknown scale and shift factor. Thus, the unprojected monocular depths from separate views are often inconsistent, resulting in duplicated object parts.

Multi-view pointmap prediction. DUSt3R [301] predicts multi-view consistent pointmaps across K input images by first inferring pairwise pointmaps, followed by a global 3D optimization that searches for per-image pointmaps and pairwise similarity transforms (rotation, translation, and scale) that best aligns all pointmaps with each other.

We run DUST3R on the multiview images at time t, but constrain the global optimization to be consistent with the K known stationary camera extrinsics $\{\mathbf{P}_k\}$ and intrinsics $\{\mathbf{K}_k\}$. This produces per-image global pointmaps $\{\chi_k^t\}$ in metric coordinates. One can then compute a depth map by simply projecting each pointmap back to each image with the known cameras

$$d_k^t(u, v) \begin{bmatrix} u & v & 1 \end{bmatrix}^T = \mathbf{K}_k \mathbf{P}_k \chi_k^t(u, v) \tag{5.1}$$

This produces metric-scale multi-view consistent depth maps $d_k^t(u, v)$, which are still not consistent

over time.

Spatio-temporal alignment of monocular depth with multi-view consistent pointmaps. In fact, even beyond temporally inconsistency, such multiview predictors tend to underperform on humans since they are trained on multiview data where dynamic humans are treated as outliers. Instead, we find monocular depth estimators such as MoGe [300] to be far more accurate, but such predictions are not metric (since they are accurate only up to an affine transformation) and are not guaranteed to be consistent across views *or* times. Instead, our strategy is to use the multiview depth maps from DUST3R as a metric target to *align* monocular depth predictions, which we write as $m_k^t(u, v)$. Specifically, we search for scale and shift factors a_k^t and b_k^t that minimize the following error:

$$\underset{\{a_k^t, b_k^t\}}{\operatorname{argmin}} \sum_{t=1}^{T} \sum_{k=1}^{K} \sum_{u,v \in \mathrm{BG}_k^t} \left\| (a_k^t m_k^t(u, v) + b_k^t) - d_k^t(u, v) \right\|^2 \tag{5.2}$$

where BG_k^t refers to a pixelwise background mask for camera k at frame t. The above uses metric background points as a target for aligning all monodepth predictions. The above optimization can be solved quite efficiently since each time t and view k can be optimized independently with a simple least-squares solver (implying our approach will easily scale to long videos). However, the above optimization will still produce scale factors that are not temporally consistent since the targets are temporally inconsistent as well. But we can exploit the constraint that background points should be *static* across time for stationary cameras. To do so, we replace $d_k^t(u, v)$ with a static target $d_k(u, v)$ obtained by averaging depth maps over time or selecting a canonical reference timestamp. The final set of scaled time- and view-consistent depthmaps are then unprojected back to 3D pointmaps. Note that this tends to produce accurate predictions for static background points, but the dynamic foreground may remain noisy because they cannot be naively denoised by simple temporal averaging. Rather, we rely on motion-based 3DGS optimization to enforce smoothness of the foreground, described next.

During our experiments, we identified two additional limitations that significantly impact visual quality.

(1) Scale initialization: We observed that initializing 3D Gaussian scales with k-nearest neighbors often results in poor appearance, such as extremely large Gaussians filling empty space and blurring the background. To address this, we follow SplaTAM [142] and initialize each Gaussian scale based on its projected pixel area: scale $= \frac{d}{0.5(f_x + f_y)}$, where d is a pixel's depth and f_x, f_y are

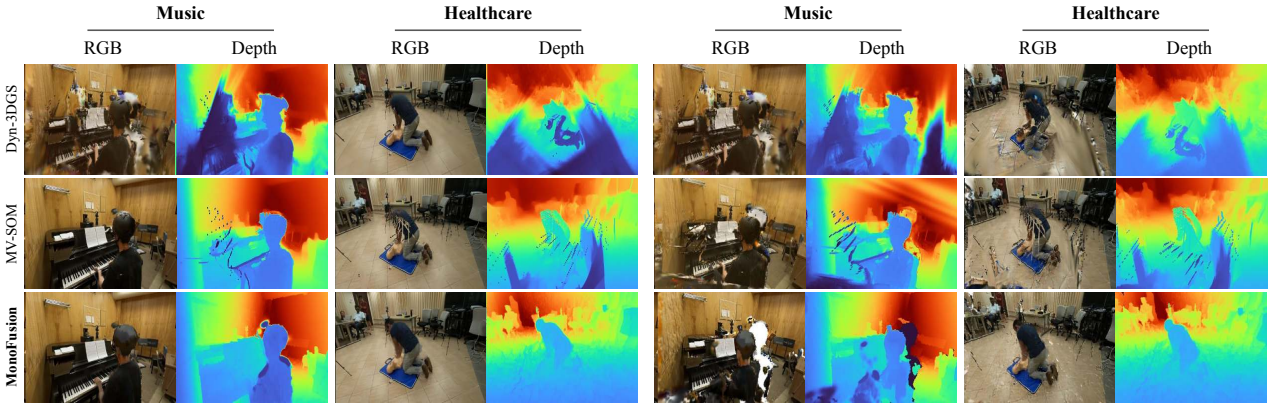


Figure 5.4. **Qualitative analysis of held-out view synthesis on ExoRecon.** We show qualitative results of held-out view synthesis (**left**) and a 5° deviation from the static camera position at the held-out timestamp (**right**). As compared to other multi-view baselines, our method does dramatically better at interpolating the motion of dynamic foreground (left), even from new camera views (right). We posit that Dynamic 3DGS suffers because of lack of geometric constraints and MV-SOM has duplicate foreground artifacts because of conflicting depth initialization from the four views.

focal lengths.

(2) Insufficient Gaussian density: Using only one Gaussian per input pixel fails to adequately capture fine details. We instead initialize 5 Gaussians per input pixel, providing better representation of fine details.

Grouping-based Motion Initialization

Beyond initializing time- and view-consistent geometry in the canonical frame, we also aim to initialize reasonable estimates of the scene motion. We model a dynamic 3D scene as a set of $\mathcal{N}$ canonical 3D Gaussians, along with time-varying rigid transformations $\mathbf{T}_{0\to t} = [\mathbf{R}_{0\to t}\mathbf{t}_{0\to t}] \in \mathbb{SE}(3)$ that warp from canonical space to time t:

$$\mathbf{x}_t = \mathbf{R}_{0\to t}\mathbf{x}_0 + \mathbf{t}_{0\to t} \quad \mathbf{R}_t = \mathbf{R}_{0\to t}\mathbf{R}_0 \tag{5.3}$$

Motion bases. Similar to Shape of Motion [299], we make the observation that in most dynamic scenes, the underlying 3D motion is often low-dimensional, and composed of simpler units of rigid motion. For example, the forearms tend to move together as one rigid unit, despite being composed of thousands of distinct 3D Gaussians. Rather than storing independent 3D motion trajectories for each 3D Gaussian (i), we define a set of B learnable basis trajectories $\{\mathbf{T}_{0\to t}^{(i,b)}\}_{b=1}^{B}$. The time-varying rigid transforms are written as a weighted combination of basis trajectories,

using fixed per-point basis coefficients $\{w^{(i,b)}\}_{b=1}^{B}$:

$$\mathbf{T}_{0\to t}^{(i)} = \sum_{b=1}^{B} \mathbf{w}^{(i,b)} \mathbf{T}_{0\to t}^{(i,b)} \tag{5.4}$$

Motion bases via feature clustering. Unlike Shape of Motion which initializes motion bases by clustering 3D tracks, our key insight is that semantically grouping similar scene parts together can help regularize dynamic scene motion, without ever initializing trajectories from noisy 3D track predictions. Inspired by the success of robust and universal feature descriptors [213], we obtain pixel-level features for each input image by evaluating DINOv2 on an image pyramid. We average features across pyramid levels and reduce the dimension to 32 via PCA [9]. We choose the small DINOv2 model with registers, as it produces fewer peaky feature artifacts [51].

Given the consistent pixel-aligned pointmaps $\chi_{t,k}^{(\text{time+view})}$, we associate each pointmap with the 32-dim feature map $\mathbf{f}_{t,k}$ computed from the corresponding image. We perform k-means clustering on per-point features $\mathbf{f}$ to produce b initial clusters of 3D points. After initializing 3D Gaussians from pointmaps, we set the motion basis weight $\mathbf{w}^{(i,b)}$ to be the L2 distance between the cluster center and 3D Gaussian center. We initialize the basis trajectories $\mathbf{T}_{0\to t}^{(b)}$ to be identity, and optimize them via differentiable rendering.

Optimization

As observed in prior work [82, 175], using photometric supervision alone is insufficient to avoid bad local minima in a sparse-view setting. Our final optimization procedure is a combination of photometric losses, data-driven priors, and regularizations on the learned geometry and motions.

During each training step, we sample a random timestep t and camera k. We render the image $\hat{\mathbf{I}}_{t,k}$, mask $\hat{\mathbf{M}}_{t,k}$, features $\hat{\mathbf{F}}_{t,k}$, and depth $\hat{\mathbf{D}}_{t,k}$. We compute reconstruction loss by comparing to off-the-shelf estimates:

$$\mathcal{L}_{\text{recon}} = \left\|\hat{\mathbf{I}} - \mathbf{I}\right\|_1 + \lambda_{\text{m}} \left\|\hat{\mathbf{M}} - \mathbf{M}\right\|_1 + \lambda_{\text{f}} \left\|\hat{\mathbf{F}} - \mathbf{F}\right\|_1 + \lambda_{\text{d}} \left\|\hat{\mathbf{D}} - \mathbf{D}\right\|_1 \tag{5.5}$$

We additionally enforce a rigidity loss between randomly sampled dynamic Gaussians and their k nearest neighbors. Let $\hat{\mathbf{X}}_t$ denote the location of a 3D Gaussian at time t, and let $\hat{\mathbf{X}}_{t'}$ denote its location at time t'. Over neighboring 3D Gaussians i, we define:

$$\mathcal{L}_{\text{rigid}} = \sum_{\text{neighbors } i} \left\|\hat{\mathbf{X}}_t - \hat{\mathbf{X}}_t^{(i)}\right\|_2^2 - \left\|\hat{\mathbf{X}}_{t'} - \hat{\mathbf{X}}_{t'}^{(i)}\right\|_2^2 \tag{5.6}$$

Dataset	Method	Full Frame				Dynamic Only			
		PSNR ↑	SSIM ↑	LPIPS ↓	AbsRel ↓	PSNR ↑	SSIM ↑	LPIPS ↓	IOU ↑
Panoptic Studio	SOM [299]	17.86	0.687	0.460	0.491	18.75	0.701	0.236	0.358
	Dyn3D-GS [194]	25.37	0.831	0.266	0.207	26.11	0.862	0.129	—
	MV-SOM [299]	26.28	0.858	0.241	0.331	26.80	0.883	0.161	0.886
	MonoFusion	**28.01**	**0.899**	**0.117**	**0.149**	**27.52**	**0.944**	**0.022**	**0.965**
ExoRecon	SOM [299]	14.73	0.535	0.482	0.843	15.63	0.559	0.450	0.294
	Dyn3D-GS [194]	24.28	0.692	0.539	0.612	24.61	0.673	0.384	—
	MV-SOM-DS [299]	28.37	0.906	0.079	0.398	28.23	0.931	0.063	0.872
	MV-SOM [299]	26.91	0.890	0.138	0.474	27.31	0.919	0.078	0.845
	MonoFusion	**30.43**	**0.927**	**0.061**	**0.290**	**29.71**	**0.947**	**0.017**	**0.963**

Table 5.1. **Quantitative analysis of held-out view synthesis.** We benchmark our method against state-of-the-art approaches by evaluating the novel-view rendering and geometric quality on both the dynamic foreground region and the entire scene, across the held-out frames from input videos. MV-SOM is a multi-view version of Shape-of-Motion [299] that we construct by instantiating four different instances of single-view shape of motion, and optimize them together. On Panoptic Studio, groundtruth depth for computing the AbsRel metric is obtained from 27-view optimization of the original Dynamic 3DGS, and for ExoRecon, we project the released point clouds obtained via SLAM from Aria glasses. When evaluating single-view baselines, SOM [299], we naively aggregate their predictions from the four views and evaluate this aggregated prediction against the evaluation cameras.

5.4 Experiments

Implementation details. We optimize our representation with Adam [155]. We use 18k gaussians for the foreground and 1.2M for the background. We fix the number of $\mathbb{SE}(3)$ motion bases to 28 and obtain these from feature clustering (Sec. 5.3). For the depth alignment, we use points above the confidence threshold of 95%. We show results on 7 10-sec long sequences at 30fps with a resolution of 512×288. Training takes about 30 minutes on a single NVIDIA A6000 GPU. Our rendering speed is about 30fps.

Datasets. We conduct qualitative and numerical evaluation on Panoptic Studio [134] and a subset of Ego-Exo4D [94] which we call ExoRecon.

Panoptic Studio is a massively multi-view capture system which consists of 480 video streams of humans performing skilled activities. Out of these 480 views, we manually select 4 camera views, 90° apart to simulate the same exocentric camera setup as Ego-Exo4D. Given these 4 training view cameras, we find 4 other intermediate cameras 45° apart from the training views, and use these for evaluating novel view synthesis from 45° camera views.

For in-the-wild evaluation of sparse-view reconstruction, we repurpose Ego-Exo4D [94], which

Method	PSNR ↑	SSIM ↑	LPIPS ↓	IOU ↑	AbsRel (↓)
SOM	16.73	0.554	0.491	0.287	0.578
Dyn3D-GS	23.31	0.776	0.316	—	0.273
MV-SOM	21.56	0.541	0.433	0.482	0.413
MonoFusion	**25.73**	**0.847**	**0.158**	**0.943**	**0.188**

Table 5.2. **Quantitative analysis of 45° novel-view synthesis on Panoptic Studio.** We benchmark our method against state-of-the-art approaches by evaluating both the dynamic foreground region and the entire scene. Notably, the evaluation is conducted on novel views where the cameras are at least 45° apart from all training views. We additionally evaluate the geometric reconstruction quality with absolute relative (AbsRel) error in rendered depth.

includes sparse-view videos of skilled human activities. While many Ego-Exo4D scenarios are out of scope for dynamic reconstruction with existing methods (due to fine-grained object motion, specular surfaces, or excessive scene clutter), we find one scene each from the 6 different scenarios in Ego-Exo4D with considerable object motion: *dance*, *sports*, *bike repair*, *cooking*, *music*, *healthcare*. For each scene, we extract 300 frames of synchronized RGB video streams, captured from 4 different cameras with known parameters. We remove fisheye distortions from all RGB videos and assume a simple pinhole camera model after undistortion. We call this subset ExoRecon, and show results on these sequences. Please see the appendix for more visuals.

Metrics. We follow prior work [194, 340] in evaluating the perceptual and geometric quality of our reconstructions using PSNR, SSIM, LPIPS and absolute relative (AbsRel) error in depth. We compute these metrics on the entire image, and also on only the foreground region of interest. We additionally evaluate the quality of the dynamic foreground silhouette by reporting mask IoU, computed as $(\hat{\mathbf{M}}\&\mathbf{M})/(\hat{\mathbf{M}}||\mathbf{M})$. Similar to prior work [340], our evaluation views are a set of held-out frames, subsampled from the input videos from 4 exocentric cameras, in both Panoptic Studio and ExoRecon.

Note that since the cameras in our setup are stationary, above evaluation only analyses the *interpolation* quality of different methods. More explicitly, we also benchmark novel-view synthesis on Panoptic Studio with an evaluation camera placed 45° away from the training view cameras. Since such a ground-truth evaluation camera is not available in ExoRecon, we only show qualitative results.

Baselines. We compare our method with prior work on dynamic scene reconstruction from single or multiple views. Among methods that operate on monocular videos, we run Shape of

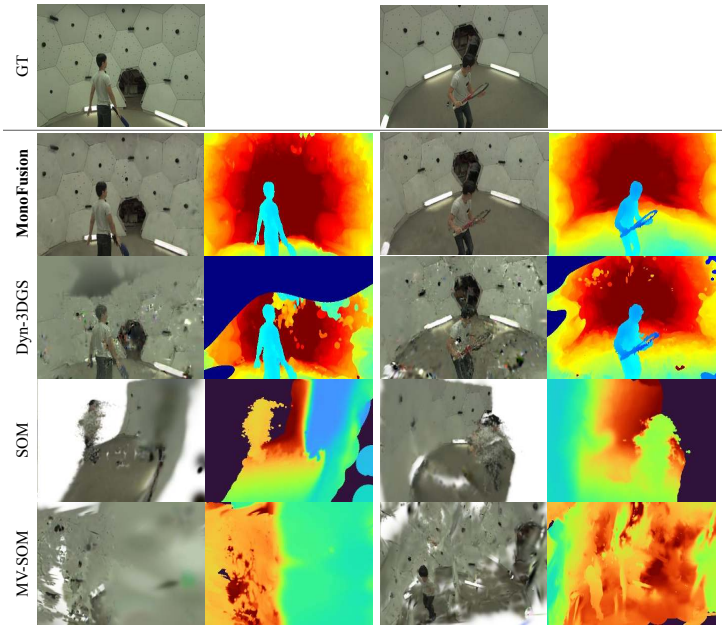


Figure 5.5. **Qualitative results of 45° novel-view synthesis results on Panoptic Studio.** We show qualitative novel-view synthesis results of our method compared to baselines on the softball (left) and tennis (right) sequences. We visualize the groundtruth RGB image for the 45° at the top. Our rendered extreme novel-view RGB image closely matches ground truth. We find that all other baselines struggle to generalize to extreme novel views.

Motion [299] on 8 scenes from Panoptic Studio following the setup of Dynamic 3D Gaussians [194] and our curated dataset ExoRecon that covers 6 diverse scenes. Finally, we consider two multi-view dynamic reconstruction baselines, Dynamic 3D Gaussians [194], and a naive multi-view extension of Shape of Motion (MV-SOM). To construct the latter baseline, we simply concatenate the Gaussians and motion bases from four independently-initialized instances of single-view SOM, and optimize all four instances jointly. We also evaluate a variant of MV-SOM with globally-consistent depth (denoted MV-SOM-DS), obtained by running per-frame DUSt3R on the 4 input views and fixing camera poses to ground-truth during DUSt3R's global alignment. Despite using our same hyperparameters, MV-SOM-DS has more visual artifacts due to reduced depth quality, suggesting the importance of our DUSt3R+MoGe design. In the appendix, we verify that all baselines reconstruct reasonable training views.

Comparison to State-of-the-Art

Evaluation on held-out views. In Tab. 5.1, we compare our method to recent dynamic scene reconstruction baselines [194, 299, 363], following evaluation protocols from prior work [299, 340].

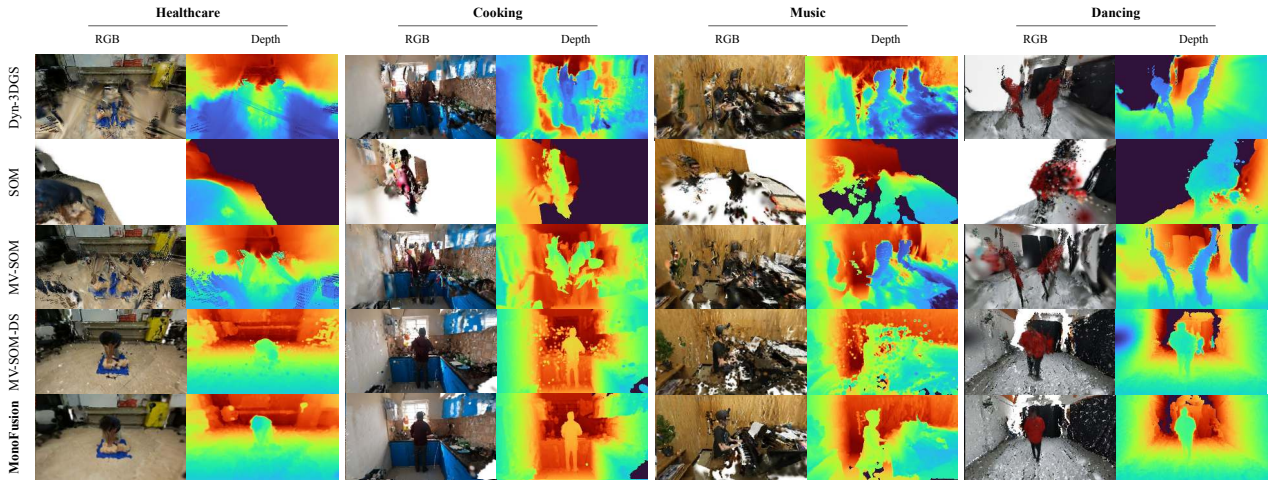


Figure 5.6. **Qualitative results of 45° extreme novel view synthesis results on ExoRecon (1/2).** We visualize the rasterized RGB image and depth map from each method for 4 diverse EgoExo sequences. Existing monocular methods (Row 2, "SOM") and their extension to multi-view (Row 3, "MV-SOM") produce poor results rendered from a drastically different novel view. MV-SOM improves upon SOM by optimizing a 4D scene representation with four view constraints, but it still suffers from duplication artifacts. Our method's careful point cloud initialization and feature-based motion bases further improve on MV-SOM. Even after running MV-SOM with multi-view-consistent depth from DUSt3R (Row 4, "MV-SOM-DS"), we find that it still fails due to reduced depth quality, often caused by suboptimal pairwise depth predictions on humans. Please see the appendix for more baseline comparisons: we find that multi-view diffusion methods contain additional hallucinations and imperfect alignment between different input views, and per-frame sparse-view 3D reconstruction methods suffer from temporal inconsistency, blurry reconstructions and missing details.

Our method beats prior art on both Panoptic Studio and ExoRecon (Fig. 5.4) datasets, when evaluated on held-out views across photometric (PSNR, SSIM, LPIPS) and geometric error (AbsRel) metrics. Note that when initializing Dynamic 3DGS [194] with 4 views we find that COLMAP fails, and so the point cloud initialization for this baseline is from a 27-view COLMAP optimization.

Interestingly, we find that although the monocular 4D reconstruction method Shape of Motion (SOM) [299] often fails to output accurate metric depth, it is robust to a limited camera shift. We hypothesize that the foundational priors of Shape of Motion allow it to produce reasonable results in under-constrained scenarios, while test-time optimization methods, especially ones that do not always rely on data-driven priors [194], can more easily fall into local optima (e.g. those caused by poor initialization) which are difficult to optimize out of via rendering losses alone.

Evaluation on a 45° novel-view. On Panoptic Studio, we use the four evaluation cameras

Method	$\mathcal{L}_{\text{feat}}$	$\mathbf{d}_n$	$\mathbf{T}_{0\to t}^{(b)}$	↑PSNR	↑SSIM	↓LPIPS	↑IoU
Baseline	✗	✗	✗	26.19	0.915	0.077	0.60
+ $\mathcal{L}_{\text{feat}}$	✓	✗	✗	25.39	0.933	0.087	0.63
+ Our depth / no $\mathcal{L}_{\text{feat}}$	✗	✓	✗	29.55	0.944	0.037	0.73
+ Our depth / $\mathcal{L}_{\text{feat}}$	✓	✓	✗	29.31	0.941	0.041	0.75
+ Motion bases (**Ours**)	✓	✓	✓	**30.40**	**0.947**	**0.037**	**0.81**

Table 5.3. **Ablation study of pipeline components.** We ablate our choice of feature-metric loss, spacetime consistent depth, and feature-based motion bases. While the proposed depth and feature-based motion bases considerably improve 4D reconstruction (evaluated by photometric errors), we find that our feature loss helps learn better motion masks (evaluated by IoU).

(placed 45° apart from the training views) to evaluate our method's novel-view rendering capability. We also evaluate the novel-view rendered depth against a 'pseudo-groundtruth' depth obtained from optimizing Dynamic 3DGS [194] with all 24 training views. In Tab. 5.2 and Fig. 5.5, we find that our method outperforms all baselines, achieving state-of-the-art 45° novel-view synthesis. Qualitative results on ExoRecon are in Fig. 5.6 & 5.7.

Ablation Study

We ablate the design decisions in our pipeline in Tab. 5.3. Our proposed space-time consistent depth plays a crucial role in learning accurate scene geometry and appearance (yielding a 3.4 PSNR improvement, Row 1 vs 3). Next, we find that the feature-metric loss $\mathcal{L}_{\text{feat}} = \left\|\hat{\mathbf{F}} - \mathbf{F}\right\|$ provides a trade-off between learning photometric properties vs.learning foreground motion and silhouette. Although the PSNR decreases, we see an increase in mask IoU (Row 1 vs 2 and Row 3 vs 4). Freezing the color of all Gaussians across frames aids learning the motion mask, as measured by mask IoU. Finally, our motion bases from feature-clustering improve overall scene optimization (final row).

Velocity-based vs. feature-based motion bases In the monocular setting, we empirically found that both designs performed equally well. However, in our 4 camera sparse view setting, we found that feature-based motion bases perform much better than velocity-based motion bases. The reason is that for velocity-based motion bases, we infer 3D velocity by querying the 2D tracking results plus depth per frame following Shape-of-Motion[299]. Thus, noisy foreground depth estimates where the estimated depth of the person flickers between foreground and backward will negatively influence the quality of velocity-based motion bases, causing rigid body parts to move erratically. In contrast, feature-based motion bases, where features are initialized from

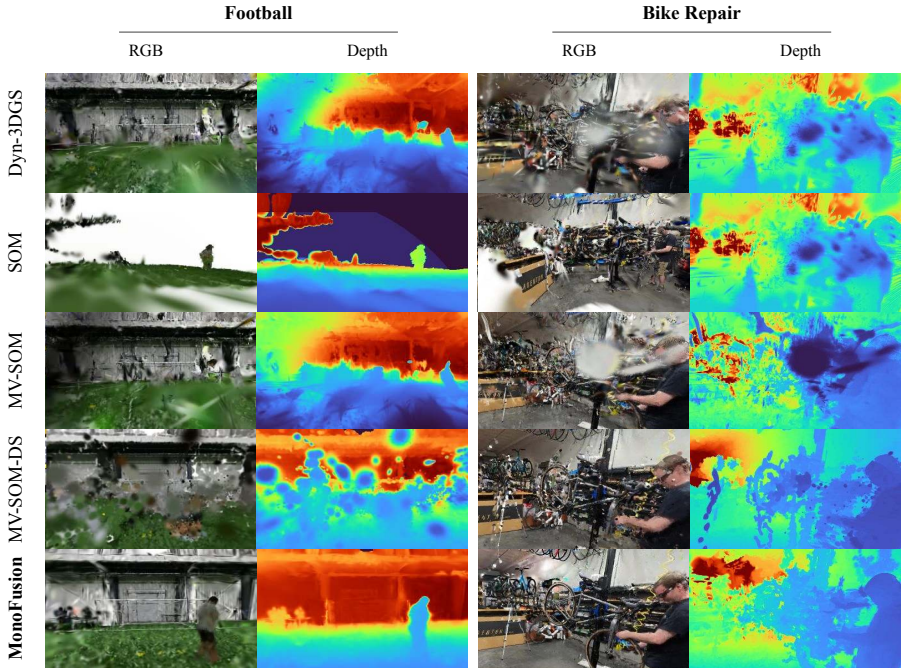


Figure 5.7. **Qualitative results of 45° extreme novel view synthesis results on ExoRecon (2/2).** We show qualitative novel-view synthesis results of our method compared to baselines on challenging sequence on ExoRecon: highly-dynamic, large scene with small foreground *football* (left) and complex, highly-occluded scene *bike repair* (right). Notably MonoFusion significantly beats other baselines in terms of quality.

more reliable image-level observations, are more robust to noisy 3D initialization and force semantically-similar parts to move in similar ways. To validate our points, in Fig. 5.8 we use PCA analysis to visualize the inferred features and find that they are consistent not only on temporal axis but also across cameras.

Effect of different number of motion bases. When the number of motion bases is not expressive enough (in our experience when the number of motion bases < 20), there are often obvious flaws in the reconstruction, such as missing arms or the two legs joining together into a single leg. In reality, we do not observe that increasing the number of motion bases further hurts the performance. Empirically, the capacity of our design (which is **28** motion bases) can effectively handle different scene dynamics.

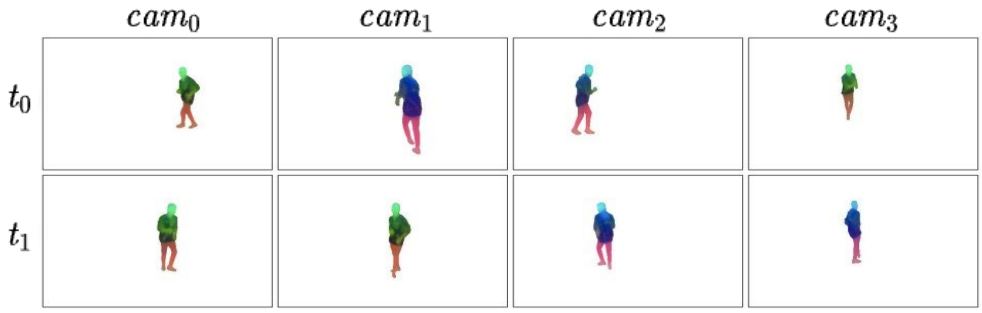


Figure 5.8. **Spatial-Temporal Visualization of feature PCA.** We perform PCA analysis and transform the 32-dim features from section 5.3 down to 3 dimensions for visualization purposes. We find that the features are consistent across views and across time. Notably, when the person turns around between t_0 and t_1 in observations from cam_1 and cam_2, the feature remains robust and consistent. The semantic consistency of features aids explainability, provides a strong visual clue for tracking, and gives confidence in our feature-guided motion bases.

5.5 Discussion

We address the problem of sparse-view 4D reconstruction of dynamic scenes. Existing multi-view 4D reconstruction methods are designed for dense multi-view setups (e.g. Panoptic Studio). In contrast, we aim to strike a balance between the ease and informativeness of multi-view data capture by reconstructing skilled human behaviors from four equidistant inward-facing static cameras. Our key insight is that carefully incorporating *priors*, in the form of monocular depth and feature-based motion clustering, is crucial. Our empirical analysis shows that on challenging scenes with object dynamics, we achieve state-of-the-art performance on novel space-time synthesis compared to prior art.

Part III

Exploiting foundational priors via finetuning

Chapter 6

Starting point: 2.1D reasoning of dynamic objects under occlusion

Publication information

Chen, K., Ramanan, D. and Khurana, T., 2025. Using Diffusion Priors for Video Amodal Segmentation. In Proceedings of the Computer Vision and Pattern Recognition (CVPR) pp. 22890-22900.

6.1 Introduction

Gestalt psychology [159] suggests that human perception inherently organizes visual elements into cohesive wholes. When an object is occluded, humans can often infer the complete outline of the object – an ability that is developed in humans in their early years [138, 215]. Additionally, object permanence [11] suggests that with some temporal context, humans can perceive objects to *persist* even when they undergo complete occlusions. Replicating these phenomena of gestalt psychology and object permanence in object segmentation has traditionally been ignored, as the community has focused largely on segmenting the visible or *modal* regions of objects (as exemplified by models like SAM [157, 242]). Recent focus has shifted to include amodal segmentation [170, 375], which involves segmenting an object's full shape, including both visible and occluded parts. This task has broad real-world applications, including safe navigation in robotic manipulation and autonomous driving [230, 243], understanding occluder-occludee relationships in complex scenes [362], and enhancing advanced image and video editing tools [216].

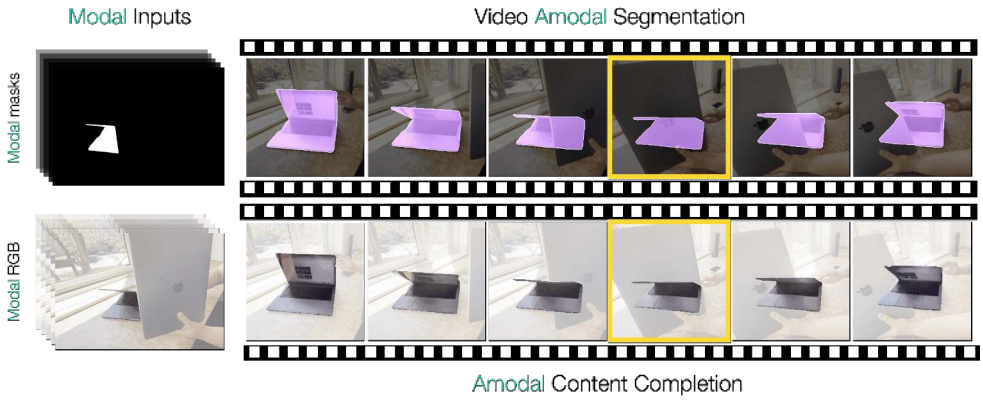


Figure 6.1. In this work, we tackle the problem of video amodal segmentation and inpainting: given a modal (visible) object sequence in a video, we develop a two-stage method that generates its amodal (visible + invisible) masks and RGB content. We capitalize on the shape and temporal consistency priors baked into video foundation models because of their large-scale pretraining. Finetuning these models enables us to infer complete shapes and RGB details of objects that undergo occlusion. Our method is effectively able to handle severe occlusions and generalizes across diverse object categories, achieving state-of-the-art results on synthetic and real-world datasets. We show one such example of an unseen deformable object category 'laptop' that undergoes a complete occlusion in the highlighted frame.

Why is this hard? In a monocular setup, amodal perception is an ill-posed problem because there are multiple plausible explanations for how an object boundary should be extended in an occluded region. Recent innovations for amodal segmentation [216, 361] and inpainting [193, 223] use diffusion frameworks for learning this multi-modal distribution, but they are not able to handle scenarios where an object could be fully-occluded. This issue is exacerbated by the lack of real-world datasets that have groundtruths for both amodal masks of objects, and their RGB content.

Status quo. Despite this, current image-based amodal segmentation algorithms [85, 141, 216, 285, 286, 361, 362] have shown impressive performance. However, these approaches are set in the single-frame setting, where they struggle with cases where objects are heavily or completely occluded. A potential solution is to approach amodal segmentation in a multi-frame setting [150] so as to infer complete occlusions with temporal context. However, existing *video* amodal segmentation algorithms [68, 346] are typically limited to rigid objects, and are dependent on additional inputs (like camera poses or optical flow) which hinders their scalability and therefore, generalization to unseen data.

Key insight. To address these challenges, we propose repurposing a video diffusion model, Stable Video Diffusion (SVD) [22], to achieve highly accurate and generalizable video amodal

segmentation. One key insight is that foundational diffusion models trained to generate pixels also bake-in strong priors on object shape. Such priors have been expoited by conditional image generation [236, 259, 366] methods that condition on semantic maps and object boundaries. We similarly exploit these priors for our task. But crucially, our multi-frame video setup allows us to propagate object shape and content across time; e.g., one can infer the shape of a fully occluded by object by looking at *other* frames where it is visible (Fig. 6.1).

Our proposed model achieves state-of-the-art performance across four synthetic and real-world video datasets, compared to a wide-variety of single-frame and multi-frame amodal segmentation baselines. We train on only synthetic data, but demonstrate strong zero-shot generalization to real-world data.

Thanks to the multi-modal generation capability of diffusion models, our approach can provide multiple plausible interpretations for the completion of occluded objects. We show that the outputs of our approach can be used for downstream applications like 4D reconstruction, scene manipulation, and pseudo-groundtruth generation.

6.2 Related Work

Image amodal segmentation. Most previous amodal segmentation research has concentrated on image-based approaches. Some methods [75, 141, 170, 230, 285, 286, 327] adopt a similar strategy to modal segmentation, where models are trained to take RGB images as input and directly output amodal masks for all objects in the scene. Another line of methods [62, 183, 216, 332, 361, 362] leverages existing modal masks, generated by modal segmentation models, to predict amodal masks based on these and additional inputs like image frames. Besides inferring the complete object shape, some approaches also *hallucinate* the RGB content in the occluded regions. Generic inpainting methods [193, 223] often fail at this task, as they rely on surrounding context, which often includes occluders. In contrast, content completion methods explicitly condition on the modal content, either by directly generating the amodal content based on modal information [216, 332] or by inpainting within the predicted amodal segmentation area [183, 362]. Due to the availability of high-quality real-world amodal image datasets [200, 230, 375], image amodal segmentation and content completion methods have shown strong performance by learning robust shape priors. However, these methods frequently struggle with cases of significant occlusion and fail entirely for fully occluded objects because the amodal cues cannot be inferred in a single-frame setting.

Video amodal segmentation. Recently, *video* amodal segmentation methods have emerged [68,

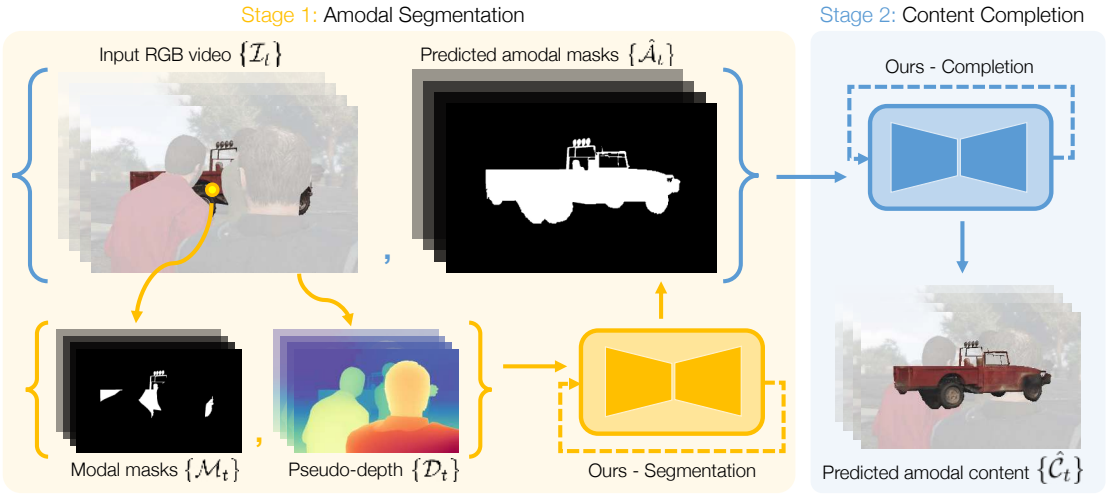


Figure 6.2. **Model pipeline** for amodal segmentation and content completion. The first stage of our pipeline generates amodal masks $\{\hat{\mathcal{A}}_t\}$ for an object, given its modal masks $\{\mathcal{M}_t\}$ and pseudo-depth of the scene $\{\mathcal{D}_t\}$ (which is obtained by running a monocular depth estimator on RGB video sequence $\{\mathcal{I}_t\}$). The predicted amodal masks from the first stage are then sent as input to the second stage, along with the modal RGB content of the occluded object in consideration. The second stage then inpaints the occluded region and outputs the amodal RGB content $\{\hat{\mathcal{C}}_t\}$ for the occluded object. Both stages employ a conditional latent diffusion framework with a 3D UNet backbone [22]. Conditionings are encoded via a VAE encoder into latent space, concatenated, and processed by a 3D UNet with interleaved spatial and temporal blocks. CLIP embeddings of $\{\mathcal{M}_t\}$ and the modal RGB content provide cross-attention cues for the first and second stage respectively. Finally, the VAE decoder translates outputs back to pixel space.

85, 346]. These approaches integrate information from preceding and succeeding frames in a video sequence, enabling temporally consistent predictions. However, the training and evaluation of most of these algorithms are limited to synthetic datasets with rigid objects of similar scale [90, 96, 230, 282]. Although these algorithms outperform image-based amodal segmentation methods within synthetic datasets, their practical applications remain limited. In contrast, we utilize both synthetic [96, 120] and real-world datasets [10, 52, 112], which include deformable objects with diverse motions and scales, often mixed with complex camera movements. Moreover, to our knowledge, this work is the first to explore video-level amodal *content completion*.

Real-world priors from diffusion models. Diffusion models have achieved significant success in generative tasks within computer vision. Initially developed for unconditional image generation [105], the scope of diffusion models has expanded in multiple directions. These advancements include, but are not limited to, implementing conditional techniques for tasks like style transfer [27, 366] and text-to-image synthesis [236], transitioning from pixel-space noise to

latent-space noise [23, 253], developing various training and sampling strategies [105, 136, 270], and extending their application from realistic image generation to video generation [22, 108, 201]. In addition, the successful adaptation of diffusion models for multiple downstream tasks, including depth estimation [140], multi-view synthesis [189, 261], and scene reconstruction [184], underscores their ability to capture object shape priors and understand potential 3D information [360]. While recent image amodal segmentation methods have demonstrated initial success in incorporating diffusion models [216, 286, 332, 361], our approach advances this progress by applying video diffusion techniques to the domain of video amodal segmentation.

6.3 Method

Consider a video sequence $\{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_T\}$ with modal (or visible) segmentation masks $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_T\}$ for a target object. Such masks can be readily obtained by conventional modal segmentors, such as Segment Anything v2 [242]. We first describe a (diffusion-based) model to generating amodal masks $\{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_T\}$ that capture the full extent of the target object, including occluded portions. We then train a second stage (diffusion-based) model that uses the input video and amodal masks to fill in (or inpaint) the RGB content of the occluded areas $\{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_T\}$.

Preliminary: diffusion framework

We make use of an open-source video latent diffusion model [23, 253] (Stable Video Diffusion (SVD) [22]) and use the EDM framework [136] for both training and inference. Compared to pixel-space diffusion, latent diffusion models reduce computational and memory demands by encoding frames into compact latent representations while preserving both perceptual and region-based alignment. The EDM framework further accelerates training convergence and reduces the required number of denoising steps during inference without compromising generation quality.

Our diffusion model takes as input the latent representation $\mathbf{z_0}$, additional conditioning $\mathbf{c}$, a noise scale σ following $\log \sigma \sim \mathcal{N}(P_{mean}, P_{std})$, and Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. The training objective is defined as:

$$\min_{\theta} \mathbb{E}_{\sigma, \mathbf{z_0}, \mathbf{c}, \epsilon} \left[\lambda || D_\theta(\mathbf{z_0} + \epsilon; \sigma, \mathbf{c}) - \mathbf{z_0} ||_2^2 \right] \tag{6.1}$$

Here, λ is a scalar related to σ, and $D_\theta = c_1(\mathbf{z_0} + \epsilon) + c_2 F_\theta(\mathbf{z_0} + \epsilon; \sigma, \mathbf{c})$ represents the predicted latent representation, which combines the noisy latent input with the v-prediction [260] output of the diffusion backbone F_θ, using additional scalars c_1 and c_2 that also depends on σ.

Modal masks in, amodal masks out

To train a high-quality amodal segmentor with limited data, one strategy is to leverage the shape and content priors of video foundation models pretrained on large-scale datasets. For this, we lean on the foundational knowledge in SVD, learnt by pretraining on the extensive LDM-F dataset [22] with 152 million examples. However, as the vanilla SVD was designed for image-to-video tasks, we adapt its structure and conditioning to suit our modal-to-amodal sequence generation task. We describe this in detail in the supplement.

Additionally, we use CLIP embeddings [231] for the modal masks, and inject them into the transformer layers for cross-attention. This provides temporal information about the visibility of objects in surrounding frames.

Different from image-level amodal completion methods that cannot handle full occlusions or maintain temporal consistency, our video diffusion model conditioned on modal masks overcomes these challenges by leveraging temporal information through memory-efficient interleaving of spatial and temporal layers. Specifically, temporal convolutions help in learning local features across video frames, while temporal attention can capture long-range dependencies between frames, *e.g.*, propagate modal masks from frames where an object is visible to where it is occluded.

Conditioning on pseudo-depth

Till now, we described how SVD is modified to enable predicting amodal masks from modal masks. We find that one can add more contextual cues about the object and scene in consideration through different data modalities. A natural choice for conditioning is RGB frames, as used in previous work [216, 362]. However, since occlusions of the target object are typically caused by objects closer to the camera, we empirically find that pseudo-depth maps provide more implicit clues about potential occluders than RGB frames, making them a more effective indicator for determining regions to complete. We demonstrate the advantages of pseudo-depth over RGB conditioning in our ablation study. To integrate this, we utilize the Depth Anything V2 monocular depth estimator [341] to convert RGB images into pseudo-depth maps, which are then incorporated into our video diffusion model as additional channels concatenated to the aforementioned input.

With the addition of pseudo-depth conditioning, the input latents for our 3D U-Net backbone have the shape $\mathcal{R}^{T\times 3C\times \frac{H}{F}\times \frac{W}{F}}$, requiring a new first convolutional layer in the 3D U-Net to accommodate the increased channels. Rather than finetuning our model with both modal masks and pseudo-depth conditionings directly, we find that it is more efficient to do a staged strategy,

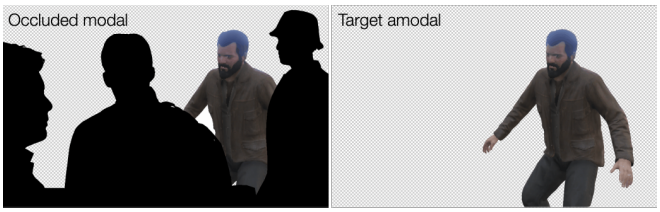


Figure 6.3. **Modal-amodal RGB training pair** for content completion. The left frame displays the partially occluded modal RGB content, generated by overlaying amodal masks (black regions) onto the amodal object to disrupt its visual integrity. The right frame shows the original, unoccluded amodal RGB object.

where we finetune our mask conditioned model first and then use it to initialize the finetuning of the mask-and-depth conditioned model. We call this approach *two-stage finetuning*, allowing the model to adapt gradually to the new conditions.

Inspired by ControlNet [366], we retain the parameters of the first channels $2C$ in the input layer from the previously trained model and initialize the newly added channels C to zero. This *zero convolution* approach ensures the model retains its initial predictive capability during the first few fine-tuning steps with the added pseudo-depth conditioning. We demonstrate the importance of these training strategies in the ablation study.

Amodal content completion

Till now, we discussed the first stage of our pipeline which outputs amodal masks for occluded objects. However, the RGB content in the occluded region is unknown. To *inpaint* these occluded areas, we use a second SVD model with the same architecture but with different conditionings; the first conditioning is the RGB content from an object's modal region, and the second conditioning is the predicted amodal mask from the first stage. We train this model to generate RGB content across the entire amodal region.

Synthetic data curation A key challenge with this approach is the lack of ground-truth RGB content in occluded regions, even in synthetic datasets like SAIL-VOS [120]. Inspired by self-supervised training-pair construction used extensively in image amodal tasks [216, 362], we extend this approach to video sequences. Figure 6.3 illustrates an example of a modal-amodal RGB content training pair. To construct such a pair, we first select an object from the dataset with near-complete visibility (above 95%). We then sequentially overlay random amodal mask sequences onto this fully visible object until its visibility falls below a set threshold, thereby simulating occlusion. This effectively generates ground-truth RGB data for the occluded regions.

6.4 Experiments

Setup

Implementation details. For training, we load the official SVD-xt 1.1 pretrained checkpoint and use the AdamW optimizer with β_1=0.9, β_2=0.999. The learning rates for the two-stage fine-tuning are set to $3 \cdot 10^{-5}$ and $3 \cdot 10^{-6}$, for training without and with additional pseudo-depth conditioning, respectively. In the case of SAIL-VOS, due to computational limitations, we set the batch size to 8 and the frame size to 128×256. Training takes approximately 30 hours on 8 Nvidia RTX 3090 GPUs. During inference, we set the EDM denoising step to 25, the guidance scale to 1.5, and use a higher frame size of 256×512 to ensure more accurate pixel-level predictions. We cover more implementation details in the appendix.

Datasets. Since amodal mask can be reliably annotated only in synthetic datasets or game engines, our model is primarily trained and evaluated on synthetic datasets. We include a zero-shot evaluation on a real-world dataset to assess its generalization ability. Among synthetic datasets, **SAIL-VOS** [120] includes 210 long video sequences with 162 common object classes generated from the photo-realistic game GTA-V, featuring frequent and significant occlusions. We use PySceneDetect [25] to identify shot transitions within these long videos, selecting only continuous scenes and segmenting them into 21,237 25-frame object sequences. **MOVi-B** and **MOVi-D**, generated by Kubrics [96], feature rich annotations of simulated environments, rigid objects, and camera motions. These datasets have been adapted as video amodal segmentation benchmarks by previous studies [68, 85] and contain 13,997 and 12,010 sequences, each with an approximate length of 25 frames. For real-world evaluation, we use **TAO-Amodal** [112], a high-quality amodal tracking dataset comprising 993 video sequences in its validation set. Unlike synthetic datasets, TAO-Amodal provides only amodal bounding box annotations, as annotating amodal masks by humans is challenging. Similar to SAIL-VOS, we segment these videos into 1,392 object sequences.

Baselines. We compare our method against recent baselines for both image and video amodal segmentation. For image-based amodal segmentation, our baselines include creating a convex hull around a given modal mask [362], AISFormer [285], PCNet-M [362], and pix2gestalt [216]. For video-based amodal segmentation, we evaluate against SaVos [346], Bi-LSTM [68, 95], EoRaS [68], and C2F-Seg [85]. We discuss more details about these baselines in the appendix. Additionally, to benchmark against regression approaches, we include transformer-based VideoMAE [284] and SVD's backbone 3D U-Net. We also evaluate the ground-truth modal masks.

Table 6.1. **Quantitative comparison on SAIL-VOS and TAO-Amodal.** We compare our method with image-based methods (top) and video-based methods (bottom). Our method outperforms all methods on the synthetic SAIL-VOS dataset, achieving nearly a 13% improvement in Top-1 mIoU$_{occ}$. Additionally, when trained on SAIL-VOS, our method demonstrates strong generalization, outperforming others in zero-shot evaluations on the real-world TAO-Amodal dataset. Bold values indicate the best method, and underlined values indicate the second best.

Method	SAIL-VOS		TAO-Amodal		
	mIoU	mIoU$_{occ}$	AP$_{25}$	AP$_{50}$	AP$_{75}$
Modal	67.89	-	93.73	82.22	63.12
Convex [362]	63.18	27.54	93.73	82.22	63.12
Convex^{R} [362]	71.21	34.27	93.73	82.22	63.12
PCNet-M [362]	74.2	42.52	94.89	85.11	65.97
AISFormer [285]	73.51	39.16	95.45	81.93	59.84
SDAmodal [361]	72.73	41.26	94.43	83.60	63.06
pix2gestalt (Top-1) [216]	54.83	26.59	80.73	57.50	28.95
pix2gestalt (Top-3) [216]	60.79	33.76	91.80	71.19	38.80
VideoMAE [284]	69.67	29.39	69.14	56.71	41.19
3D-UNet	72.79	39.54	94.59	83.83	64.33
Ours (Top-1)	<u>77.07</u>	<u>55.12</u>	<u>97.28</u>	<u>89.25</u>	<u>71.99</u>
Ours (Top-3)	**79.23**	**59.69**	**98.31**	**92.46**	**77.48**

Table 6.2. **Quantitative Comparison on MOVi-B/D.** Due to strong camera motion and higher occlusions in these datasets, multi-frame methods generally outperform single-frame methods. Our method surpasses all prior state-of-the-art, achieving over a 4% improvement in Top-1 mIoU$_{occ}$ across both datasets.

Method	MOVi-B		MOVi-D	
	mIoU	mIoU$_{occ}$	mIoU	mIoU$_{occ}$
Modal	59.19	-	56.92	-
Convex [362]	64.21	18.42	60.18	16.48
PCNet-M [362]	65.79	24.02	64.35	27.31
AISFormer [285]	77.34	43.53	67.72	33.65
SaVos [346]	70.72	33.61	60.61	22.64
Bi-LSTM [68, 95]	77.93	46.21	68.43	36.00
EoRaS [68]	81.76	49.39	74.1	38.33
C2F-Seg [85]	-	-	71.67	36.13
VideoMAE [284]	78.74	42.86	70.93	32.78
3D-UNet	82.16	49.81	75.65	40.86
Ours (Top-1)	<u>83.51</u>	<u>53.75</u>	<u>77.03</u>	<u>44.23</u>
Ours (Top-3)	**83.93**	**54.56**	**77.76**	**45.6**

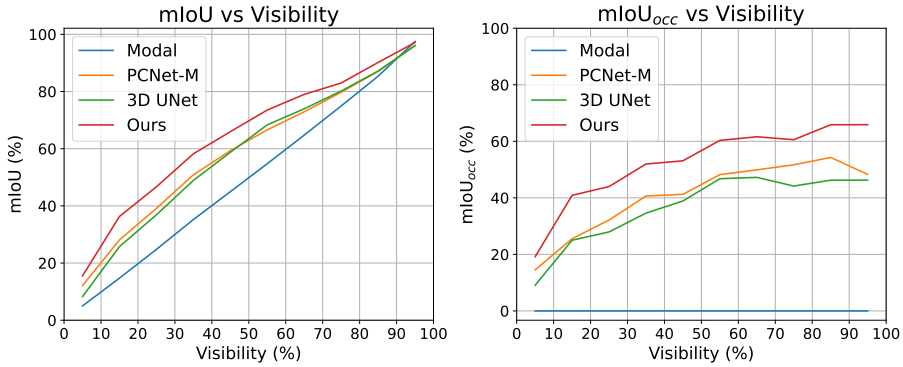


Figure 6.4. **Comparison across visibility levels** on SAIL-VOS. Our method outperforms the second-best image and video amodal segmentation methods across all visibility ranges (we use Top-1 metrics). This highlights the ability of our method to handle heavy occlusions, *and* understand when an object is not occluded.

Metrics. Following common practice in amodal segmentation [68, 85, 346], we use mIoU and mIoU$_{occ}$ as evaluation metrics. Given a modal-amodal sequence pair in each frame, where the ground-truth modal mask is $\mathcal{M}_i$, and the predicted and ground-truth amodal masks are $\hat{\mathcal{A}}_i$ and $\mathcal{A}_i$, respectively, we define IoU as $\frac{\hat{\mathcal{A}}_i \cap \mathcal{A}_i}{\hat{\mathcal{A}}_i \cup \mathcal{A}_i}$ and mIoU$_{occ}$ as $\frac{(\hat{\mathcal{A}}_i - \mathcal{M}_i) \cap (\mathcal{A}_i - \mathcal{M}_i)}{(\hat{\mathcal{A}}_i - \mathcal{M}_i) \cup (\mathcal{A}_i - \mathcal{M}_i)}$. We report the mean values across all frames in the dataset as mIoU and mIoU$_{occ}$. For TAO-Amodal, which uses bounding box evaluation instead of masks, we adopt average precision metrics used in a recent amodal tracking work [112] – AP$_{25}$, AP$_{50}$, and AP$_{75}$, based on varying IoU thresholds calculated over bounding box areas. Additionally, to account for the multimodal generation capability of diffusion-based methods, we adopt a probabilistic evaluation with Top-K metrics [150], selecting the best IoU or AP score in each frame from K predictions.

Comparison to state-of-the-art

Table 6.1 shows the quantitative comparisons on SAIL-VOS and TAO-Amodal, where our method surpasses all baselines. Notably, it achieves nearly 13% improvement over the second-best method, PCNet-M [362], in terms of mIoU$_{occ}$, highlighting effective completion of occluded object regions. Despite being trained exclusively on synthetic SAIL-VOS, a zero-shot evaluation on TAO-Amodal highlights the strong generalization of our model. We posit that, in addition to leveraging foundational knowledge and rich priors from the large-scale pretraining of SVD, our model is able to learn temporal cues that help it amodally complete any unseen object classes from neighboring frames. Figure 6.4 further illustrates our method's consistent performance across all visibility

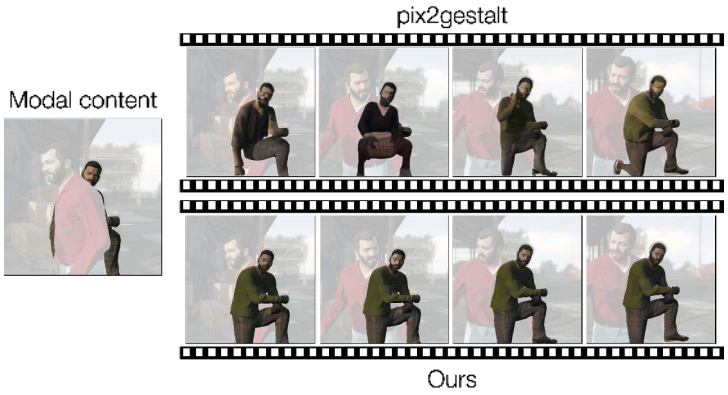


Figure 6.5. **Temporal consistency comparison** with an image amodal segmentation method. We highlight the lack of temporal coherence in a single-frame diffusion based method, pix2gestalt, for both the predicted amodal segmentation mask and the RGB content for the occluded person in the example shown. By leveraging temporal priors, our approach achieves significantly higher temporal consistency across occlusions.

ranges on SAIL-VOS [120], indicating that our method can realistically hallucinate masks in occluded regions, across the entire range of visibility levels.

We also compare with a single-frame diffusion-based method, pix2gestalt [216], in Table 6.1.

Perhaps unsurprisingly, pix2gestalt performs poorly on these video benchmarks, likely because many objects undergo *high* degrees of occlusion. Since, pix2gestalt is a single-frame method, we also find that its predictions vary significantly across frames and lack temporal coherence (c.f. Figure 6.5).

In contrast, our method does drastically better because it can handle both, high occlusions and temporal coherence across frames.

Table 6.2 provides quantitative comparisons on the MOVi-B/D datasets, where our method beats the prior state-of-the-art. Despite strong camera motion in MOVi-B/D, our model adapts well *without* access to camera extrinsics or optical flow (unlike some baselines [68, 346]). We posit that our method is able to use the 3D priors from Stable Video Diffusion and is therefore, successfully able to maintain consistent object shapes from different view-points. Notably, prior works on MOVi-B/D are evaluated using a cropped modal bounding box enlarged by 2 times as input; we adopt the same setting here for a fair comparison. However, we observed that using the full, uncropped image as input can significantly enhance model performance, and we include these results in the appendix.

Human evaluation. For content completion, due to the lack of ground truth and standardized

Table 6.3. **Ablation study for input conditioning.** We study the effect of conditioning our model on different input modalities. Results show that pseudo-depth conditioning yields greater performance improvements than RGB conditioning across almost all metrics, as it provides an actionable cue for the relative ordering of objects in the scene which helps decide which modal boundary to extend in order to predict an amodal mask. We therefore drop RGB conditioning in the final method.

Conditions			SAIL-VOS		TAO-Amodal		
mask	RGB	depth	mIoU	mIoU_{occ}	AP_{25}	AP_{50}	AP_{75}
✓	✗	✗	75.17	51.28	94.89	85.03	66.87
✓	✓	✗	76.59	53.3	95.86	86.59	**70.12**
✓	✗	✓	77.07	**55.12**	**97.28**	**89.25**	69.65
✓	✓	✓	**77.19**	54.59	96.6	87.16	69.64

metrics, we conducted a user study on 20 randomly selected sequences from SAIL-VOS and TAO-Amodal. In this user study, we did A/B testing and forced participants to choose between our method and pix2gestalt. We found that users showed a preference of 85.6% for our method over pix2gestalt.

Conditioning. Here we ablate our choice of modal mask and pseudo-depth conditioning. We also examine the effect of additionally using RGB video frames as conditioning. As shown in Table 6.3, adding RGB or pseudo-depth information improves model performance, with pseudo-depth providing a more substantial enhancement. Although combining both RGB and pseudo-depth yields a higher mIoU on SAIL-VOS, conditioning on pseudo-depth alone outperforms across other metrics. This supports our claim that pseudo-depth is a more generalizable modality that helps in deciding which modal boundary to extend in order to predict the amodal mask (see supplement for more results), and the dependence on texture and appearance cues in fact hinders generalization of our model to TAO-Amodal.

Training strategies. Table 6.4 shows the impact of our two-stage fine-tuning strategy and use of zero convolutions. Compared to randomly initializing the new input convolution layer in the 3D U-Net, we find that zero convolution significantly improves model performance. Additionally, compared to training with both modal mask and pseudo-depth conditionings from scratch, two-stage fine-tuning (where we train with modal masks first and then add pseudo-depth) leads to further quantitative improvements. We also include an ablation study on the weights initialization in the appendix.

Top-k evaluation. Like other diffusion models, ours also supports multimodal generation. For instance, when parts of an object remain consistently occluded in a video (e.g., a person's legs), multiple plausible interpretations of the occluded area (e.g., standing, sitting) may exist,

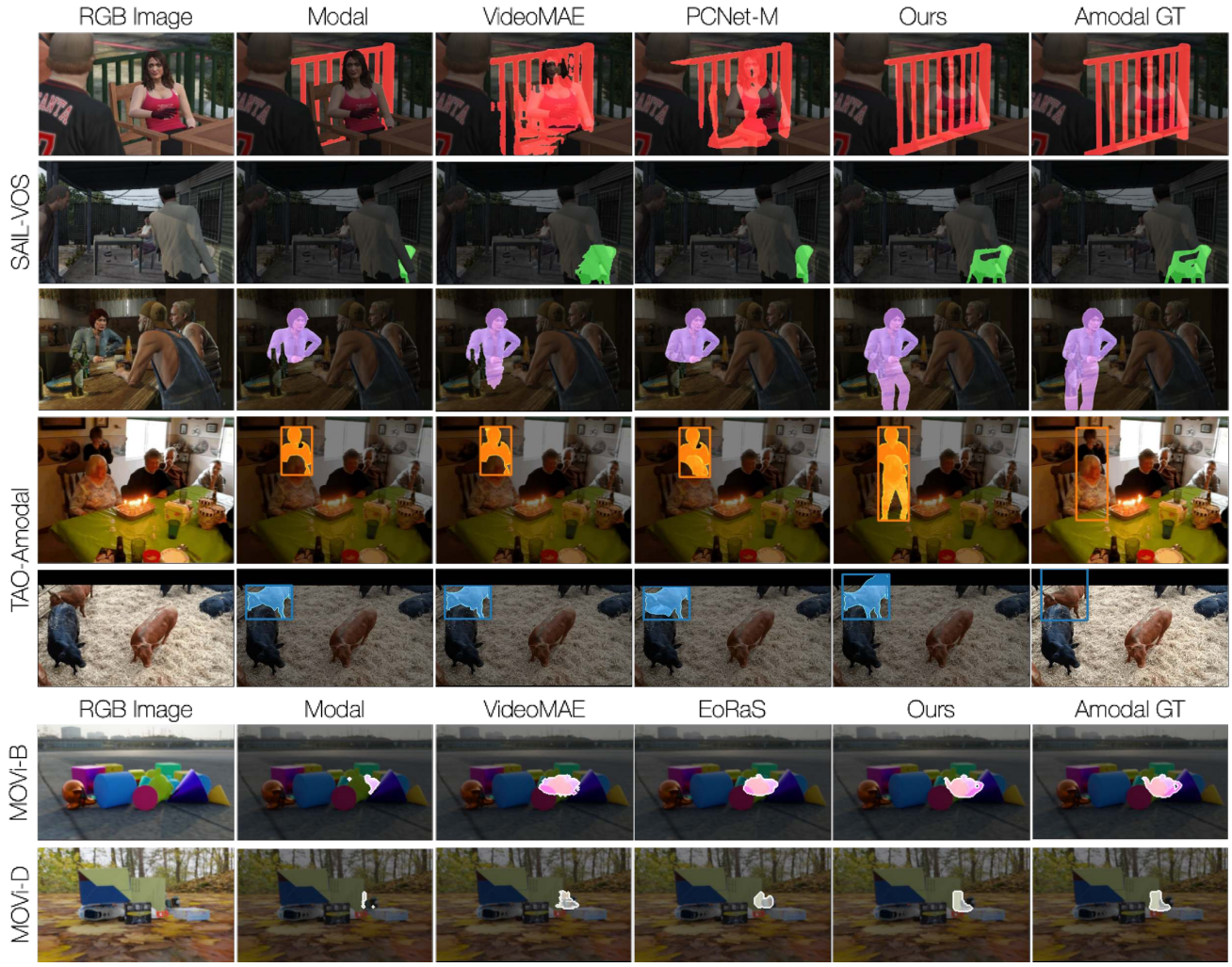


Figure 6.6. **Qualitative comparison** of amodal segmentation methods across diverse datasets. Our method leverages strong shape priors, such as for humans, chairs, and teapots, to generate clean and realistic object shapes. It also excels in handling heavy occlusions; even when objects are nearly fully occluded (e.g., "chair" in the second row of SAIL-VOS), our method achieves high-fidelity shape completion by utilizing temporal priors. Note that TAO-Amodal contains out-of-frame occlusions which none of the methods are trained for, but our method is able to handle such cases.

Figure 6.7. **Qualitative results for content completion.** Although our content completion module, initialized from pretrained SVD weights, is finetuned solely on synthetic SAIL-VOS, it achieves photorealistic, high-fidelity object inpainting even in real-world scenarios. Furthermore, our method can complete *unseen* categories, such as giraffes and plastic bottle, likely due to its ability to transfer styles and patterns from the visible parts of objects to occluded areas in the current or neighboring frames. We show examples from TAO-Amodal (top) and in-the-wild YouTube videos (bottom).

Table 6.4. **Ablation study for training strategies.** We study the effect of two-stage finetuning for segmentation. We find that zero convolution helps significantly, while two-stage fine-tuning gives us an additional, moderate improvement.

Training strategies		SAIL-VOS		TAO-Amodal		
2-stage ft.	zero-conv	mIoU	mIoU_{occ}	AP_{25}	AP_{50}	AP_{75}
✗	✗	73.73	41.35	96.27	85.93	66.45
✓	✗	72.72	32.23	95.38	86.1	68.74
✗	✓	76.92	54.25	96.58	87.64	69.34
✓	✓	**77.07**	**55.12**	**97.28**	**89.25**	**71.99**

Figure 6.8. We show an example of **multi-modal generation** from our diffusion model. Since there are multiple plausible explanations for the shape of the person in his occluded region, our model predicts two such plausible amodal masks (with the person's occluded legs in two different orientations).

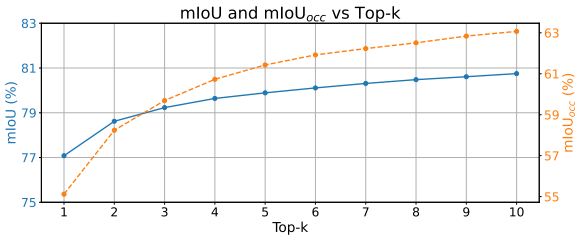


Figure 6.9. **Ablation of Top-K** on SAIL-VOS. We find that increasing the number of output samples from our method, K, leads to improvements in both mIoU and mIoU$_{occ}$; however, these improvements gradually plateau as we increase K.

as illustrated in Figure 6.8. By setting different random seeds, our model generates varying predictions for the same input, leading to different IoU values against the ground truth. Figure 6.9 reports the Top-10 mIoU and mIoU$_{occ}$ results for our model. As expected, performance improves with the number of outputs, although the gains gradually diminish.

Our amodal segmentation masks and content completions can enable multiple downstream applications. We touch on three such applications in the appendix – 4D reconstruction, scene manipulation, and pseudo-groundtruth generation. First, we find that one can use monocular video to multi-view generation methods like SV4D [330] to reconstruct dynamic objects across space and time, even when they get fully occluded. Second, we show that once all objects in a scene are de-occluded, they can be rearranged in the scene to simulate different object interactions and realities. Third, we can generate amodal segmentation pseudo-groundtruth on real-world datasets to fill in the gap for lack of real-world training data for video amodal segmentation.

6.5 Discussion

In this work, we focus on the problem of video amodal segmentation – segmenting objects to their full extent even when they may be partially or fully occluded in videos. We lean on the

large-scale pretraining of video foundation models and adapt Stable Video Diffusion [22] for the task of video amodal segmentation. Given an object's modal mask sequence, and pseudo-depth maps of the scene, we aim to predict amodal masks of the occluded object. This amodal mask is used by another model to inpaint the RGB content in the object's occluded region. One of the key insights of our work is that one can use the shape and temporal priors learnt by video foundation models. More crucially, our multi-frame setup allows us to propagate mask and RGB content from the frames where an object maybe fully visible to the frames of high occlusion. We find that, our models can perform exceedingly well even for unseen categories, likely because of their pretraining on foundational data.

Acknowledgments We thank Ege Ozguroglu, Achal Dave and Carl Vondrick for insightful discussions and clarifications. Mosam Dabhi helped with demonstrating the application of our work to 4D reconstruction.

In this appendix, we extend the discussion of our approach on video amodal segmentation. We first discuss additional setup details for our method, and then cover more experimental analysis, followed by examples of our method's potential applications. We also show more qualitative results from our method. Please see the project page for a video version of all figures.

6.6 Appendix

Stable Video Diffusion modifications

First, we replace the input conditioning $\mathbf{c}$, originally an RGB image, with binary modal masks of shape $\mathcal{R}^{T\times 1\times H\times W}$. By default, the variational autoencoder (VAE) [156] in SVD requires a 3-channel input. To address this mismatch in the number of channels, we replicate the binary mask three times, following the approach for single-channel VAE inputs in a recent work [140]. After encoding each (replicated) mask seperately, we obtain a latent tensor of shape $\mathcal{R}^{T\times C\times \frac{H}{F}\times \frac{W}{F}}$. This latent representation, concatenated with a noise image of the same shape, forms the input to our backbone which is a spatio-temporal 3D U-Net [23, 254]. The final shape of this input becomes $\mathcal{R}^{T\times 2C\times \frac{H}{F}\times \frac{W}{F}}$. In contrast to the vanilla SVD, where the latent space of a single image is duplicated T times to align with the 3D U-Net's input requirements, our 3D U-Net gets as input T *unique* frames of the modal mask sequence being used as conditioning.

Inference details

During inference with our video diffusion model, we follow common practices [22] by employing the stochastic sampler from EDM [136]. We simplify this process by omitting the second-order correction and keeping the explicit Langevin-like "churn" factors constant. The denoising process is performed over 25 steps. Specifically, when denoising the latents from z_t to z_0 for $i \in \{t, \cdots, 1\}$, each denoising step can be expressed as:

$$\hat{\mathbf{z}}_{i-1} \leftarrow \hat{\mathbf{z}}_i + (\sigma_{i-1} - \sigma_i)\frac{(\hat{\mathbf{z}}_i - D_\theta(\hat{\mathbf{z}}_i; \sigma_i))}{\sigma_i} \tag{6.2}$$

Furthermore, we employ classifier-free guidance (CFG) [103] to balance the quality and diversity of the generated samples. During training, we randomly set the conditioning to zero with a probability of $\rho = 0.1$ to simulate the unconditional case. During inference, we combine the conditional and unconditional predictions using a guidance scale of $s = 1.5$, as defined as:

$$\tilde{F}_\theta(\mathbf{z}, \mathbf{c}) = F_\theta(\mathbf{z}, \emptyset) + s(F_\theta(\mathbf{z}, \mathbf{c}) - F_\theta(\mathbf{z}, \emptyset)) \tag{6.3}$$

After denoising, the latent predictions are projected back into pixel space using the VAE decoder, which yields three-channel representations. To convert these into single-channel binary masks in the amodal segmentation stage, we sum the channel values (from 0 to 255) and binarize the predictions by thresholding. The threshold is chosen as a per channel pixel-value of 200. Finally, we take the union of the prediction with the input modal masks, ensuring modal masks remain a subset of amodal masks and are properly reflected in the output.

Regarding inference time, our method takes approximately 0.95 seconds per frame on a single RTX 3090 GPU, using around 8GB of VRAM with FP16 precision.

Baselines

In this section, we provide additional details of the image- and video-level amodal segmentation methods used for comparison.

For image-level amodal segmentation, 'Convex' [362] generates the geometric convex hull of modal masks, while 'Convex^R' [362] refines this by including only the convex hull within occluded regions predicted by 'PCNet-M'. 'PCNet-M' [362] is a self-supervised regression method that recovers amodal masks within occluder areas based on frame-level object ordering recovery. 'AISFormer' [285] employs a transformer-based head appended to a modal segmentation backbone to directly predict all amodal bounding boxes and masks within an image. 'pix2gestalt,' [216] is

an image diffusion-based method that generates amodal content conditioned on the RGB image and modal masks of the objects.

For video-level amodal segmentation, 'SaVos' [346] employs a CNN-LSTM architecture that processes RGB and modal mask patches, along with optical flow, to predict amodal masks and motions. 'EoRaS' [68] proposes an object-attention encoder that incorporates Bird's-Eye View (BEV) 3D information, relying on having access to groundtruth camera parameters. 'C2F-Seg' [85] leverages a vector-quantized latent space for coarse feature learning, refined with a convolutional module; though designed for image-level tasks, it extends to video segmentation using a spatial-temporal transformer block.

For generic video regression approaches, 'VideoMAE' [284] is a transformer-based autoencoder that we adapt for our task by setting the masking ratio to zero, applying supervised training, and using the decoder during inference. '3D-UNet' [23], the backbone of our video diffusion model, contains interleaved residual and transformer blocks with spatial and temporal modules but is trained to perform one-step generation without any iterative denoising.

Additional experiments

Note that the video versions of all qualitative results in this and the following sections can be found directly on the project page.

Improved results on MOVi-B/D. All results reported on MOVi-B/D follow prior work in segmenting objects in a region which is defined as a 100% extension of the region enclosed by the input modal mask. Therefore, all images are cropped to this region before being sent as input to any method. This is different from the standard protocol used in other datasets, where the *entire* image is sent as input (without any cropping). Here, we include results from training our model with the entire image as input on the MOVi-B/D datasets. As shown in Table 6.5, this fix significantly improves metrics, with our method achieving 4% and 6% gains in mIoU on MOVi-B and MOVi-D, respectively. Regression methods also benefit notably from this setting. We conclude that this is because MOVi-B/D include many instances of complete occlusions, for which segmentation in a cropped region is not enough for predicting amodal mask.

Qualitative evidence for pseudo-depth conditioning. The quantitative advantage of pseudo-depth conditioning was demonstrated in Table 3 of the main paper. Here, we provide qualitative evidence to illustrate the source of this improvement. As shown in Figure 6.10, pseudo-

Table 6.5. **Quantitative results on MOVi-B/D with uncropped input.** Enlarged modal region-cropped input limits the model's ability to predict an amodal mask when an object is fully occluded. Using the entire image as input restores the model's ability to complete amodal masks fully, especially when the modal area is small. This results in substantial metric improvements compared to Table 2 in the main paper. We copy over the results here for reference.

Input	Method	MOVi-B		MOVi-D	
		mIoU	mIoU_{occ}	mIoU	mIoU_{occ}
Modal cropped	VideoMAE [284]	78.74	42.86	70.93	32.78
	3D-UNet	82.16	49.81	75.65	40.86
	Ours (Top-1)	83.51	53.75	77.03	44.23
	Ours (Top-3)	83.93	54.56	77.76	45.6
Uncropped	VideoMAE [284]	85.35	49.53	79.13	42.41
	3D-UNet	84.24	46.17	76.90	36.69
	Ours (Top-1)	87.8	53.69	82.97	47.86
	Ours (Top-3)	**88.43**	**54.64**	**84.04**	**49.43**

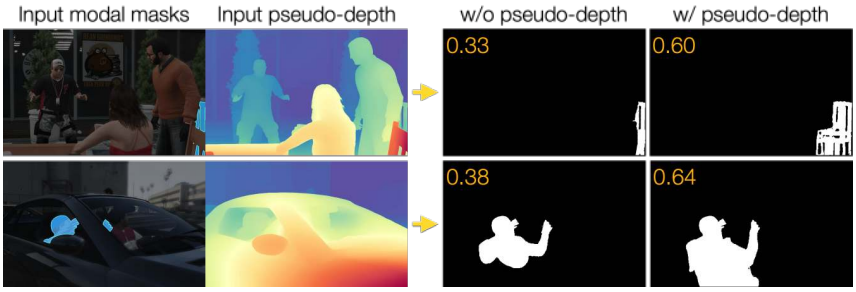


Figure 6.10. We show **how pseudo-depth aids amodal segmentation**. Object's surrounding regions with lower depth values, i.e., closer to the camera, act as potential occluders. In the top row, the occluders are the person and chair to the left of the object; in the bottom row, the occluder is the car door below the person. Depth information implicitly guides our method to complete these occluded regions.

depth conditioning encourages our method to segment areas *closer* to the camera, suggesting that depth serves as an implicit indicator of potential occluders and therefore, gives information about which occluded boundary to extend in order to predict the amodal mask.

Ablation on weights initialization. We leverage the real-world priors learnt by large-scale diffusion models by utilizing pretrained SVD checkpoints [22]. Here, we evaluate the importance of this initialization. In Table 6.6, we compare the performance of our model and the 3D U-Net with and without pretrained weights. Results show that excluding the checkpoint leads to a performance drop for both models, with a more pronounced decline for ours. These results underscore the importance of the SVD priors.

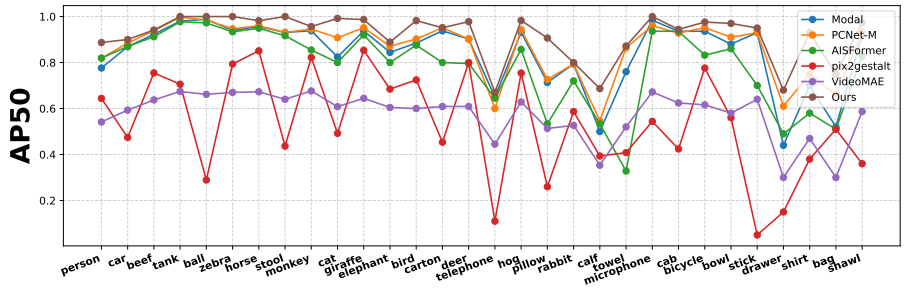


Figure 6.11. **Comparison across diverse categories** on TAO-Amodal. On a subset of the most frequent super-categories, our method consistently outperforms others under the AP50 metric. The overall trend is aligned with the quantitative results in Table 6.1. We attribute the strong generalization ability of our model to the SVD priors and its effective utilization of temporal information.

Building an end-to-end segmentation and completion model. Unlike our two-stage method, which first performs amodal segmentation and then inpaints content, the image diffusion-based method pix2gestalt [216] adopts a one-stage approach to directly generate amodal content and derive masks. A similar one-stage approach can be extended for our video setting. However, as shown in Table 6.7, our two-stage method demonstrates clear advantages over the one-stage approach. We attribute this low performance of the end-to-end method to the lack of data available for training such a single-stage method. In contrast, the two-stage method benefits from breaking down the pipeline into video amodal segmentation and content completion. For the former, it is easy to find large-scale training data of modal-amodal mask pairs from synthetic datasets. For the latter, since the content completion task reduces to video inpainting, less amount of training data is sufficient for finetuning.

Generalizability across diverse categories. Our model demonstrates strong generalization ability in a zero-shot setting on the real-world TAO-Amodal dataset, which includes many previously unseen categories. TAO-Amodal is a collection of 7 different datasets, covering a wide range of in-the-wild scenarios. Specifically, it consists of 833 object categories, out of which only 20 categories are seen during training in SAIL-VOS. For more clarity, we include a performance breakdown on a subset of the most frequent super-categories in TAO-Amodal, as shown in Figure 6.11 . The results further highlight our model's capacity to generalize across diverse categories.

Application to 4D reconstruction. Our method enables 4D reconstruction for occluded objects when used in conjunction with off-the-shelf SV4D [330]. In Figure 6.12, we compare

Table 6.6. **Ablation of SVD priors.** We study the effect of using pretrained SVD weights as initialization for our training. We find that leveraging priors from large-scale pretraining of SVD enhances both our method and the 3D UNet baseline, with particularly substantial improvements observed for our method.

Method	pretrained ckpt?	SAIL-VOS		TAO-Amodal		
		mIoU	mIoU_{occ}	AP_{25}	AP_{50}	AP_{75}
Ours	✗	68.89	26.96	93.73	79.45	57.87
Ours	✓	**75.17**	**51.28**	**94.89**	**85.03**	**66.87**
3D UNet	✗	70.85	32.66	**94.88**	83.81	59.75
3D UNet	✓	**72.79**	**39.54**	94.59	**83.83**	**64.33**

Table 6.7. **Ablation study on end-to-end amodal content completion.** We train an end-to-end version of our two-stage pipeline with a dataset of curated modal-amodal RGB training pairs from SAIL-VOS, in a similar fashion to pix2gestalt [216]. Compared to the two-stage results in Table 1 of the main paper, this approach shows a significant performance drop in both in-domain and zero-shot evaluations, highlighting the superiority of the two-stage method.

Method	SAIL-VOS		TAO-Amodal		
	mIoU	mIoU_{occ}	AP_{25}	AP_{50}	AP_{75}
Two-stage	**77.07**	**55.12**	**97.28**	**89.25**	**71.99**
One-stage	66.15	40.31	70.65	57.51	37.22

reconstructions with and without completion. Without completion, blank regions appear in occluded areas, making it more difficult to hallucinate reasonable re-projections across different views. In contrast, our method allows SV4D to produce consistent and clearer 4D reconstructions.

Application to Scene manipulation. With amodally completed objects in the scene, we can change their orderings and positions without exposing previously occluded regions. Figure 6.13 shows examples of scene manipulation, where our method facilitates manual re-composition of scenes by inpainting the occluded content of objects.

Pseudo-groudtruth for TAO-Amodal masks. TAO-Amodal [112] provides ground truth for amodal bounding boxes but lacks annotations for amodal *masks* due to the challenges of manual labeling of occluded objects in videos. We show that our method can be used to generate high-quality *pseudo*-ground truth masks for this dataset by using the information about ground-truth amodal bounding boxes, which define the extent of the amodal shape. We find that using the

Figure 6.12. **4D reconstruction results**. Without amodal completion by our method, the 4D reconstruction exhibits blank regions and unrealistic artifacts in occluded areas, such as the person's back and leg. The varying occluded portions over time confuse SV4D, disrupting its understanding of the object's 3D structure. In contrast, using completed objects from our method significantly improves the reconstruction quality, producing more consistent and clear novel-views.

amodal bounding boxes to crop the input modal mask sequences, one can train a more accurate video amodal segmentation method exclusively on SAIL-VOS. This way, our approach significantly improves evaluation metrics and aligns precisely with the amodal bounding box extent, as shown in Table 6.8. Figure 6.14 further illustrates the qualitative results of the pseudo-ground truth masks which are high-fidelity across diverse object categories. Quantitatively, we find that using the pseudo-groundtruths for finetuning baselines like VideoMAE (which have already been pre-trained on SAIL-VOS), improves their performance on the TAO-Amodal dataset by around 25%, 25%, and 20% on AP_{25}, AP_{50}, and AP_{75} respectively. Apart from this, the generated pseudo-groundtruths can be used to semi-automate the amodal mask annotation process as this is a challenging and inherently ill-posed problem.

Note that we do not include this data point in the main paper as at inference we cannot expect to have access to amodal bounding boxes but in order to produce pseudo-groundtruth, one can adopt this approach.

We now present qualitative results from all datasets and additional, in-the-wild scenarios. Figures 6.16, 6.17, 6.18, 6.19 and 6.20 compare our amodal segmentation method with more baselines on SAIL-VOS, TAO-Amodal, and MOVi-B/D. Our method demonstrates superior performance in generating high-fidelity shapes in the occluded regions of objects. Figure 6.21

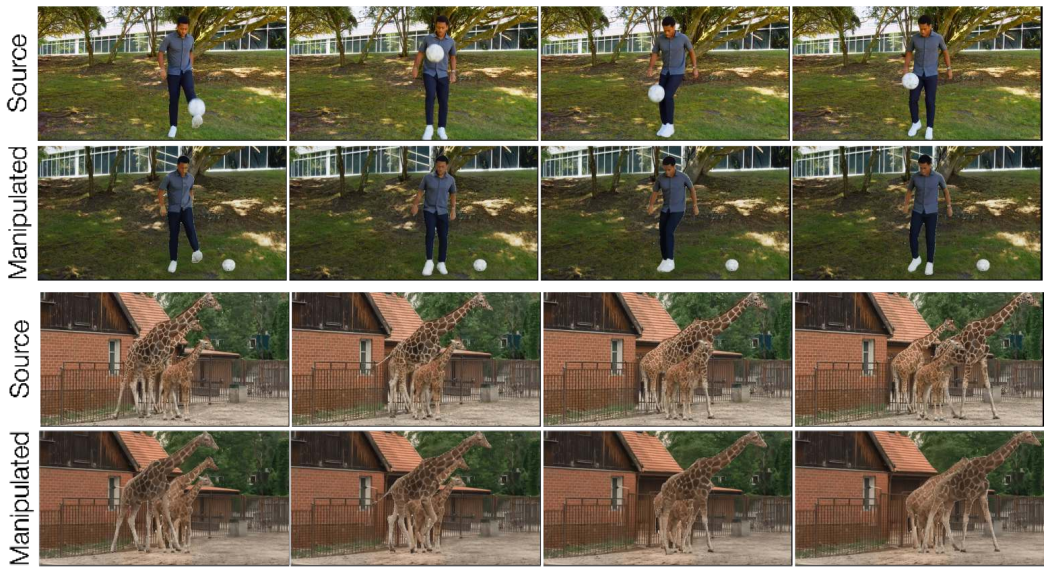


Figure 6.13. **Scene manipulation examples**. Using de-occluded objects from our method, we can reposition and reorder them to create new scenes. In the top rows, the relationship between the person and the soccer ball is altered, changing the scene from "the person is juggling" to "the person places the soccer ball aside and practices a juggling posture." In the bottom rows, the middle giraffe is moved to the front and its position is adjusted.

showcases additional in-the-wild content completion results, highlighting the photo-realistic quality and strong generalization capability of our method.

Failure cases. In Figure 6.15, we show four different kinds of failure cases. In the first case with a person swimming, our method does not successfully complete the person's amodal region. This happens often if the object of interest is occluded throughout the extent of the video; our model is not able to understand if this is a completely visible object or a consistently occluded object. In the second case, the occluded object is a bow, which has never been seen before and is completely out-of-distribution from the set of objects in SAIL-VOS. Our method fails in this case. In the third and fourth case, our method incorrectly assumes the height of a completely visible man to be greater than what it is, and predicts a sitting person to be standing. Therefore, our method lacks contextual cues about what the scene is and how the modal region looks like in the first-stage.

Table 6.8. **Pseudo-groundtruths on TAO-Amodal.** We show that using the amodal bounding box prior from the TAO-Amodal dataset to specify the extent of the output amodal segmentation mask, can help improve the quality of video amodal segmentation. We use this version of our method to produce 'pseudo-groundtruths' for TAO-Amodal. We find that these pseudo-annotations can help improve the quantitative performance of baselines like VideoMAE. See text for more details

Input setting	SAIL-VOS		TAO-Amodal		
	mIoU	mIoU_{occ}	AP_{25}	AP_{50}	AP_{75}
Uncropped	77.07	55.12	97.28	89.25	71.99
Amodal cropped	**87.44**	**69.81**	**99.59**	**99.59**	**99.48**

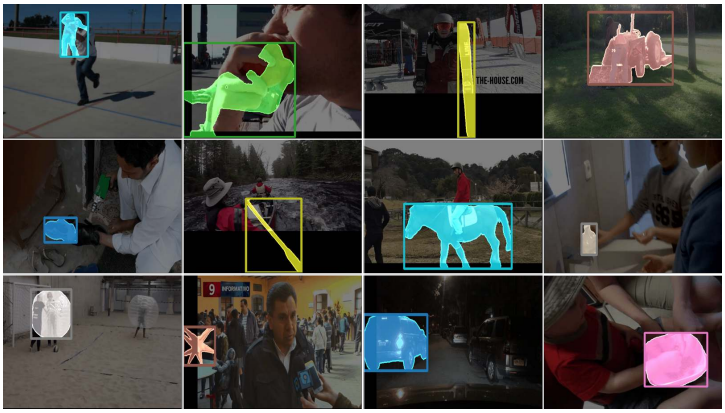


Figure 6.14. **Qualitative results for pseudo-ground truth of TAO-Amodal masks**. Leveraging the amodal bounding box as a strong prior, our method demonstrates versatility across diverse categories, such as person, tractor, and bottles, and generalizes well to unseen categories like snowboards and horses. This high-quality pseudo-ground truth can semi-automate the manual annotation of amodal masks in real-world videos.

Figure 6.15. Qualitative analysis of failure cases of our method. See text for more details.

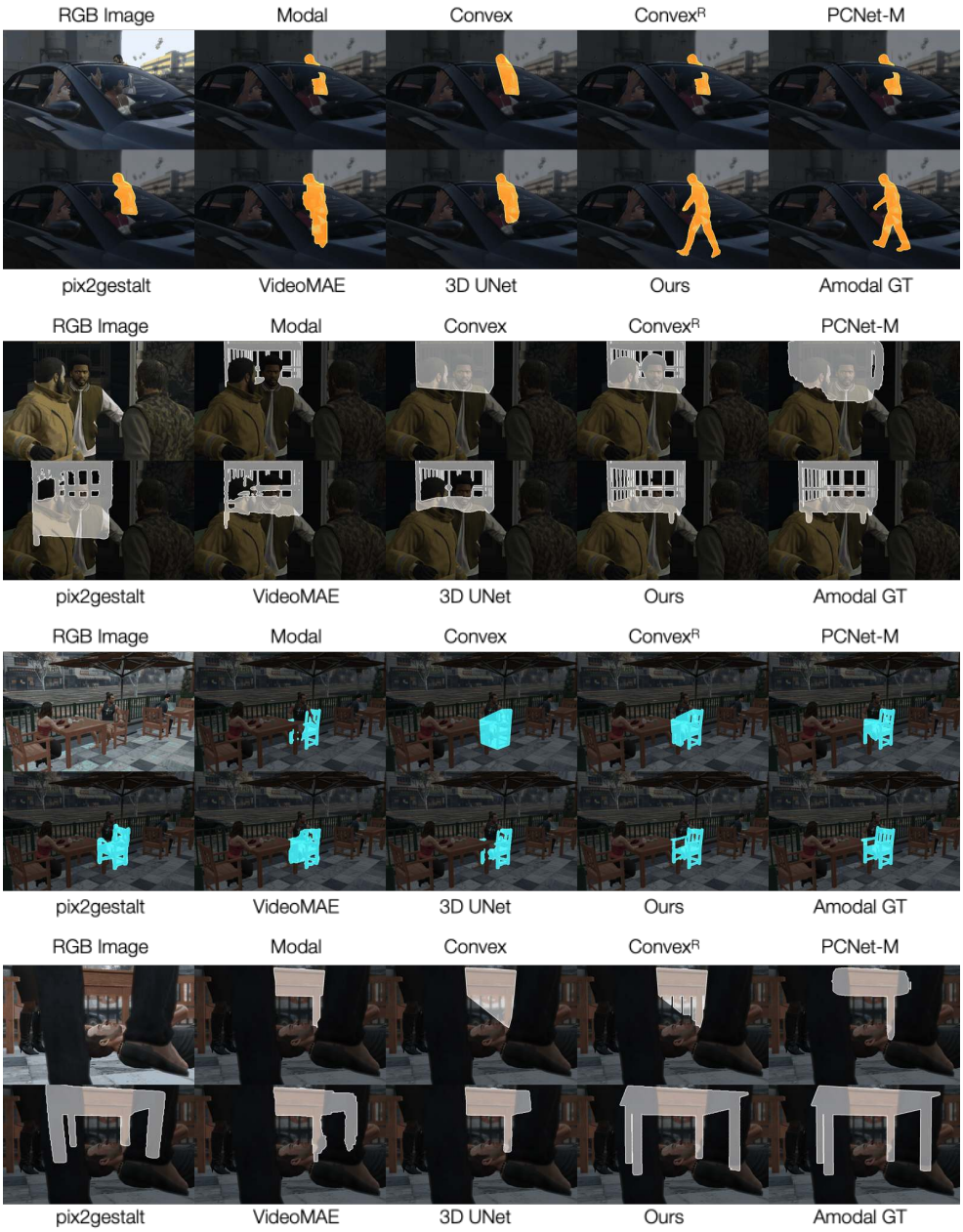


Figure 6.16. Qualitative results on SAIL-VOS. (1/2)

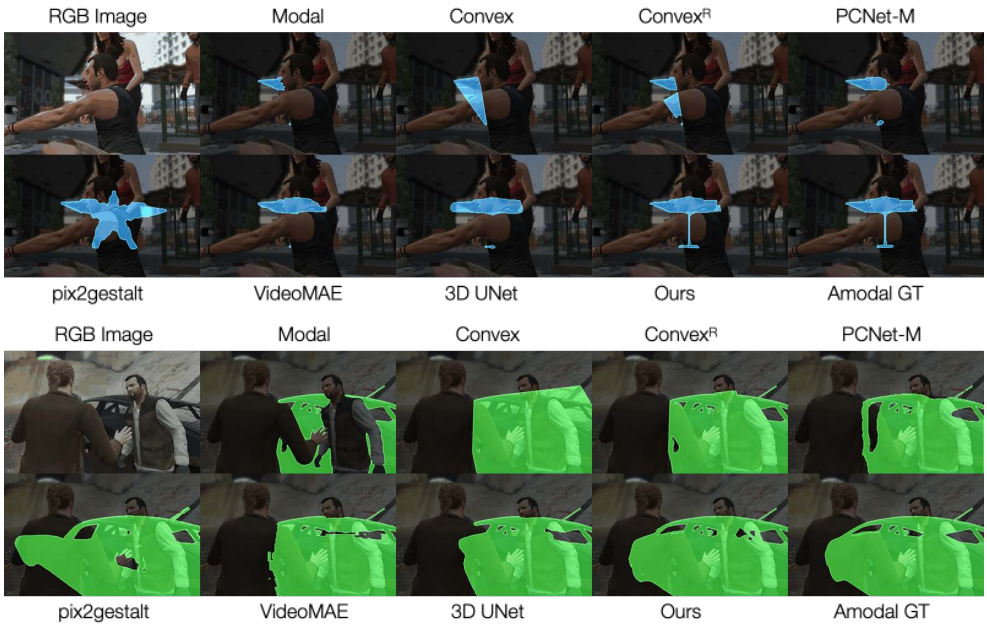


Figure 6.17. Qualitative results on SAIL-VOS. (2/2)

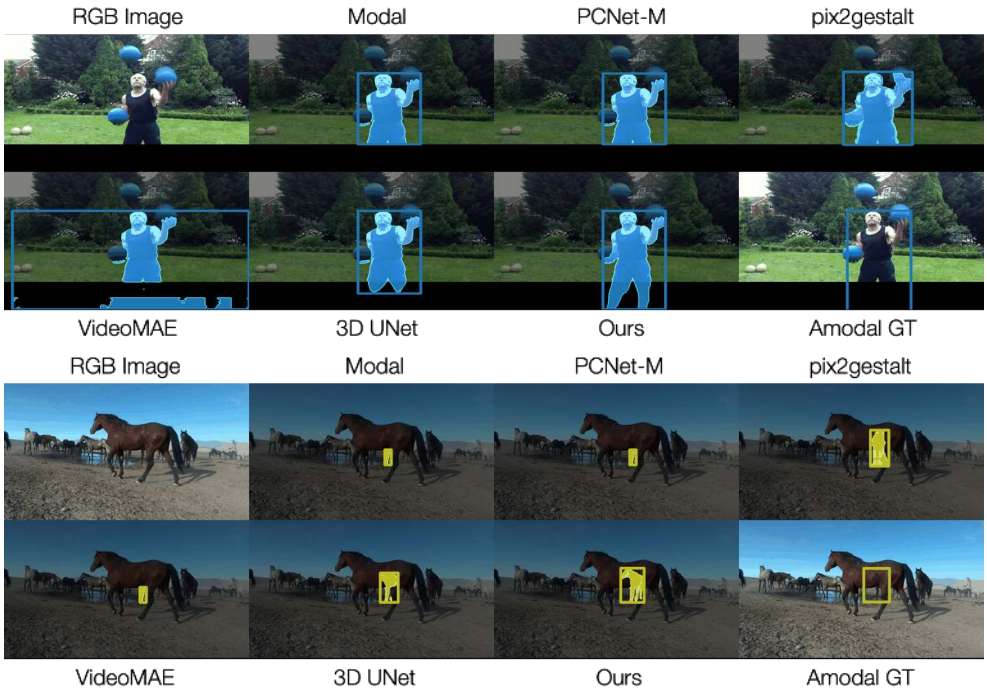


Figure 6.18. Qualitative results on TAO-Amodal. (1/2)

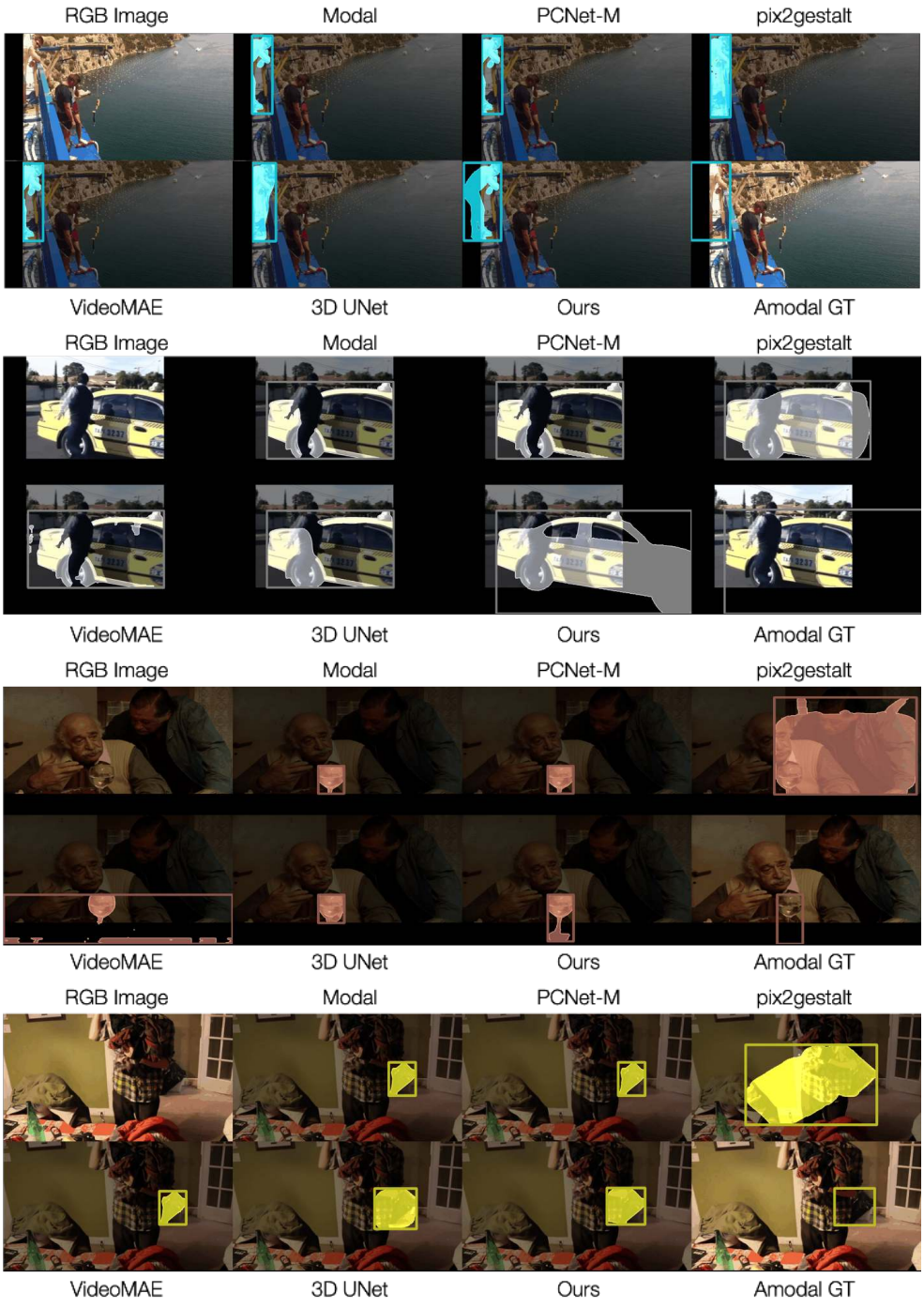


Figure 6.19. Qualitative results on TAO-Amodal. (2/2)

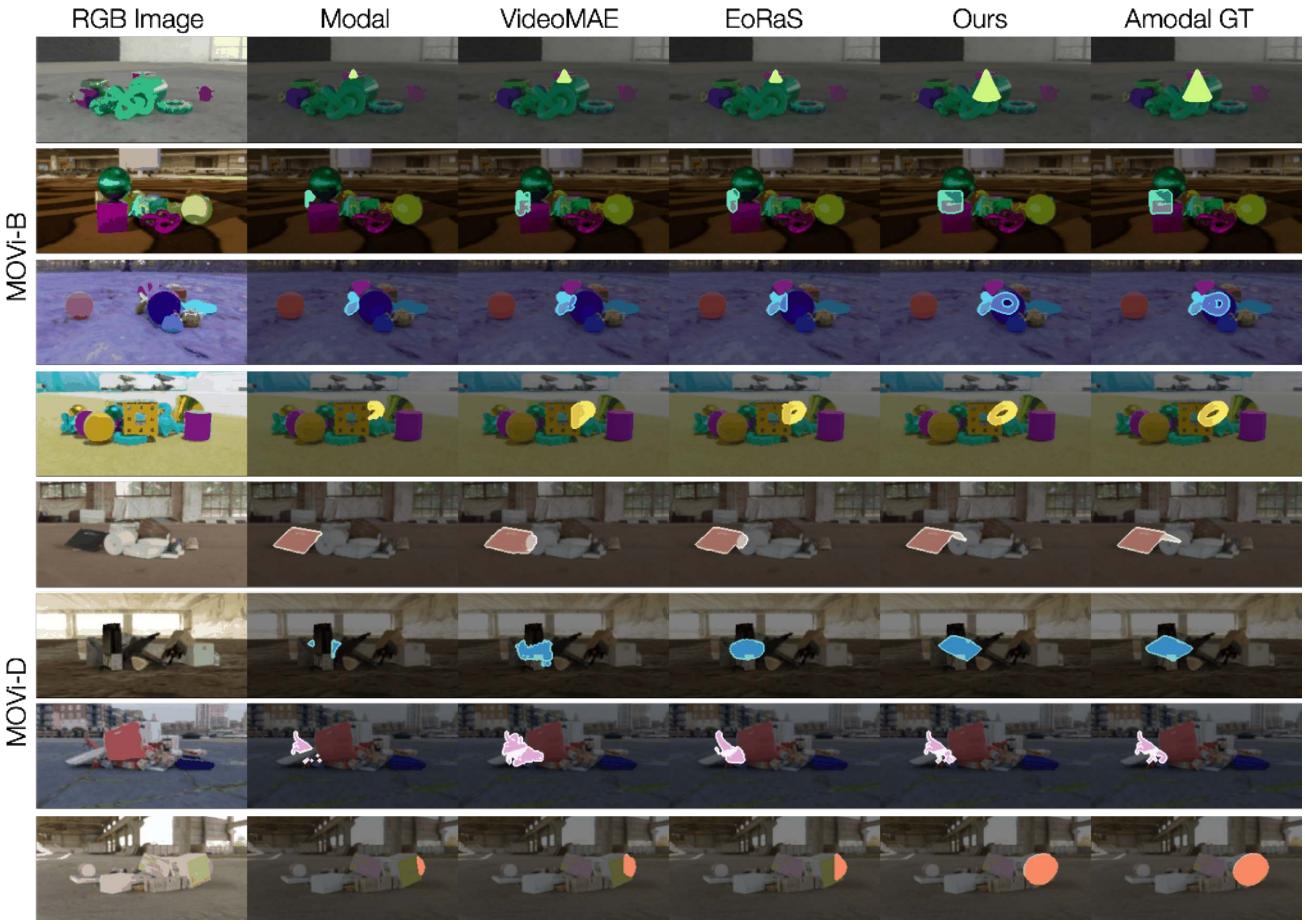


Figure 6.20. Qualitative results on MOVi-B/D.

Figure 6.21. Qualitative results for amodal content completion for in-the-wild scenarios.

Chapter 7

Generating 2.5D egocentric depth sequences of dynamic scenes

Technical report information

Khurana, T. and Ramanan, D., 2024. Predicting Long-horizon Futures by Conditioning on Geometry and Time. arXiv preprint arXiv:2404.11554.

7.1 Introduction

Recent innovations in generative visual modeling have paved the way for a variety of applications. In this work, we focus on the task of conditionally generating (or forecasting) the future from past observations. Our motivation is from an embodied perspective. Evidence from neuroscience suggests *predictive coding* to be a fundamental phenomena for biological processing of visual streams [239]; specifically, biological agents process the future by first predicting what may occur and then updating predictions based on actual observations (similar to classic dynamic models such as kalman filters [135, 150]). Predictive modeling is the backbone of autonomous systems such as self-driving vehicles that forecast environment motion for downstream applications like motion planning [35, 151].

Why is this hard? One of the challenges in operationalizing such a predictive task is that the future is inherently *multi-modal*; consider an outdoor scene of a busy intersection where cars may continue straight or turn. Encoding such uncertainty has been a notorious challenge, but

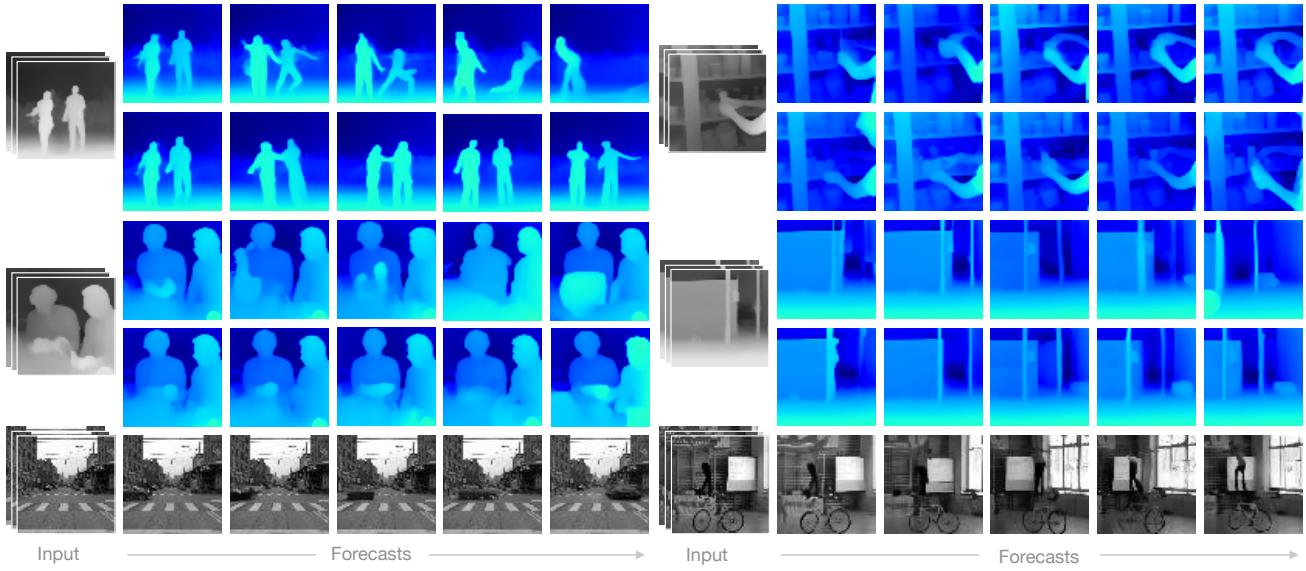


Figure 7.1. **Predicting long-horizon futures by conditioning on geometry and time.** In this work, we focus on the task of forecasting sensor observations given the past. Since the unobserved future can unfold in multiple ways, we capitalize on the recent explosion in large-scale pretraining of 2D diffusion networks, which are able to model the multi-modal distribution of natural images. By introducing invariances in data and additionally learning to condition on frame timestamps, we are able to equip 2D diffusion models with the ability to perform predictive video modeling using moderately-sized training data. Since we are able to query arbitrary timestamps, we find new sampling schedules that perform better than traditional autoregressive / hierarchical sampling strategies. Here, we show two pseudo-depth **futures** each, given the **past** pseudo-depth for four scenes, along with forecasts from training with luminance.

recent generative modeling techniques such as diffusion networks provide an attractive formalism for generating multiple *samples* from the multi-modal future. As such, our work follows a growing body of work on video-based diffusion models [21, 28, 109, 331]. But crucially, rather than generating video samples unconditionally or conditioned on textual prompts, we generate future frames conditioned on past observations. However, this introduces a significant practical challenge of satisfying compute demands that are required for learning from massive-scale video datasets.

Our approach relies on two key insights. First, we take the view that accurate video prediction can be achieved by using recent 2D image diffusion models [252] alone. This is because such models are trained on a massive scale of image data that (inevitably) contains multiple stages or instances of *temporal* events (c.f. Fig. 7.2). We add a control mechanism to image diffusion models in the form of *timestamps* that help build a temporal understanding, and are fairly easy to obtain. Moreover, by training on videos with differing framerates, our

timestamp-conditioned model can support a variety of video prediction tasks including short-horizon forecasting, autoregressive long-horizon forecasting, and even frame interpolation (by conditioning on fractional timestamps). This flexibility to sample an arbitrary timestamp in the future lets us probe newer (and stronger) sampling schedules, other than just autoregressive and heirarchical sampling that is most commonly used by prior work [99, 109, 293].

Modalites Our second key insight is motivated by embodied applications such as robotics / self-driving vehicles. Oftentimes, we are not concerned with the photometric properties of the future (e.g., "what will be the color of this car?") but rather geometric properties (e.g.,"*where* will this car be?") [152]. Geometric processing of depth sensors is commmon in point cloud processing [202, 314, 316] & occupancy forecasting [4, 148, 198] from 3D LiDAR sweeps, and legged locomotion using only egocentric depth [3, 47]. However, such depth data is not as widely as available as passive camera imagery. To leverage the latter, we show that one can use (pseudo) depth, which can readily be obtained at-scale for videos by running recent monocular depth estimators [20]. We show that simply choosing to forecast in grayscale rather than color already simplifies the forecasting problem to a great degree. More importantly, introducing invariances in data allows us to finetune image diffusion models with only 1000 videos in about 7 hours (11 hours for training them from scratch with same data)!

Contributions In summary, we present a video prediction diffusion network that can be efficiently fine-tuned from foundational image networks by additionally conditioning on frame timestamps. The flexibility in sampling an arbitrary future, allows us to propose stronger sampling schedules than prior work. We also demonstrate that our design choices allow our model to be trained on a modest but diverse set of ∼1000 videos from the TAO dataset [52], that encompasses a variety of indoor and outdoor scenes, spanning a large vocabulary of objects. We use a variety of baselines [53, 99, 293] (including nonlinear regression, constant and linear prediction) to illustrate the effectiveness of different modalities. To illustrate the effectiveness of multi-modal forecasting, we make use of probabilistic (top-K) metrics developed in the forecasting community [39].

7.2 Related work

Extracting priors from image diffusion models Denoising diffusion models [106, 271] have emerged as an expressive and powerful class of text-to-image generative models. Because of the massive scale of data used to train models like Stable Diffusion [252], Imagen [258] and DALL·E [235], numerous follow-up works have investigated and built upon their rich representations.

Specifically for novel-view synthesis, a few works [189, 261] aimed at extracting geometric, pose priors from Stable Diffusion [252] for object or scene-level novel-view-synthesis. Other sparse-view 3D reconstruction works [283, 373, 377] also draw motivation from the same concept for distilling the information from image diffusion into 3D models. A new paradigm of text/image-to-3D assets emerged, where many works [226, 265, 276] iteratively enforced 3D consistency from the outputs of image diffusion models, whereas others repurposed the image models for directly predicting tri-plane representations [110]. In fact, a dedicated study was conducted for understanding the 3D priors learnt by image diffusion models [359].

Similarly for the task of video or motion diffusion, some works [268, 293, 323] have attempted to "inflate" image diffusion models to suit video generation, with normalization tricks, a general phenomenon that has appreared before for designing convolutional video understanding architectures [34]. This also extends to the task of 3D motion generation, be it for humans [61] or object trajectories [5, 97]. In a similar spirit, we address the task of video forecasting, emphasizing the fact that in order to repurpose 2D diffusion models to suit the video-based task of forecasting given the past, it is important to extract and control the axis of time, by explicit conditioning on fractional timestamps.

Video diffusion models For video diffusion, algorithms have been built on top of recurrent or 3D architectures, including 3D convolutions [109], and RNNs [342], usually coupled with large-scale training datasets. Apart from these, there has been a meteoric rise in recent developments in dedicated text-to-video diffusion models, ranging from industrial-scale pretraining [28, 92, 107, 171], to multi-modality conditioning and generation networks [331]. Some of these methods are even designed for extremely-long autoregressive video generation [99, 293, 349]. We instead explore the setting where in addition to a moderately-sized data, only limited training resources are available for building a model that conditions on an input timestamp, instead of text (therefore, find the open-sourced Stable Video Diffusion [21] to be out of resource bounds). We also find better sampling schedules than autoregressive and hierarchical sampling.

Training with masked-autoencoder objectives The ground-breaking findings from learning self-supervisable representations with masked autoencoders [101], have recently been adopted by image and video transformer architectures [122, 284, 297, 352, 353], and diffusion models designed for a variety of tasks [308, 335]. Although we do not explicitly train in the fashion of masked autoencoders, we touch upon a similar finding when designing the timestamp conditioning mechanism for optimizing the forecasting performance at inference.

Forecasting for autonomous systems In robotics, an important precursor to motion planning is forecasting what the scene and its agents will look like in the future [39, 114, 318]. In self-driving, this spans the field of point cloud [314, 316], and recently, occupancy forecasting [4, 148, 198]. Forecasting videos of depth has a direct analogue to works that forecast range images of point clouds from LiDAR sensors [202]. For the task of legged locomotion in quadrupeds, egocentric-depth is increasingly becoming the sole modality that robots rely on [3, 47]. This is largely for the reason that depth acts a low-level actionable cue that helps generalization across a vast set of diverse environments for robot navigation. We are motivated by this, and explore forecasting future geometries for use in autonomous systems.

7.3 Method

We lean on recent image-to-image diffusion architectures, specifically Zero-1-to-3 [189], trained for changing the camera viewpoint of an object given its RGB image. We repurpose its image and camera pose conditioning for the task of timestamp-conditioned video forecasting given multiple past contexts.

Problem formulation

Given a set of context frames $\mathbf{c} \in \mathbb{R}^{K\times H\times W\times C}$ from a video of a (static or dynamic) scene, our goal is to generate a frame $\mathbf{x} \in \mathbb{R}^{1\times H\times W\times C}$ for the same scene but from a different point in time, t. Let all timestamps in consideration be $\mathbf{t} \in \mathbb{R}^{K+1}$. Then, we want to learn a function g that generates an estimate of the unobserved frame $\mathbf{x}$ given context frames $\mathbf{c}$ and timesteps $\mathbf{t}$,

$$\hat{\mathbf{x}} = g(\mathbf{c}, \mathbf{t}) \tag{7.1}$$

Since, $\hat{\mathbf{x}}$ is unobserved, it inherently follows a multi-modal distribution, making its prediction underconstrained. To this end, we exploit pretrained large diffusion models like Stable Diffusion [166, 252] that can model and sample from such multi-modal distributions of natural images. We can use single-frame 2D diffusion models for the task of video prediction, as their large-scale pretraining likely covers the space of temporal events and the different stages of their unfolding. In Fig. 7.2, we show different stages of two temporal events, prompted from Stable Diffusion v2. However, such architectures are not straight-forward to use, as time-conditioned video prediction demands for two new capabilities: first, the ability to generate a new frame that is consistent with the historical context frames $\mathbf{c}$, and second, the ability to listen to the continuous valued

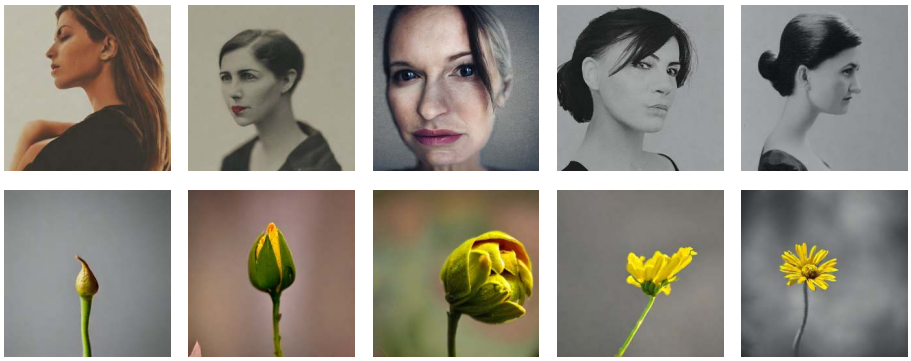


Figure 7.2. **Using 2D diffusion models for video prediction** As part of designing the *video* prediction architecture, we make the important design choice of using *image* diffusion models. Owing to the scale of data such models are trained on, we can expect them to understand indepedent stages of *temporal* events such as 'turning head from left to right', and 'flower bud opening up'. We show individual frames prompted from Stable Diffusion v2. We propose to add a control knob to image models in the form of timestamps that helps in temporal understanding.

timestamps, $\mathbf{t}$.

Given the above, we formalize the task of video prediction in the context of diffusion models as follows. Given a dataset of videos with a known FPS, we extract snippets of length $K+1$ and construct a training sample as $\{\mathbf{x}, \mathbf{c}, \mathbf{t}\}$. Using this training data, we start from the natural scene-level data distribution learnt by Stable Diffusion Image Variations [161, 166] and finetune it for controlling both the conditioning with the context frames, and timestamp scalars. Architecturally (ref. Fig. 7.3), we use a denoising UNet ϵ_θ [189], that looks at 64×64 images. For any timestep $i \sim [0, 1000]$, we train ϵ_θ with the well-adopted noise prediction objective for diffusion training,

$$\min_{\theta} \mathbb{E}_{z\sim x, i, \epsilon\sim\mathcal{N}(0,1)} ||\epsilon - \epsilon_\theta(z_i, i, f(\mathbf{c}, \mathbf{t}))||_2^2. \tag{7.2}$$

where $f(\mathbf{c}, \mathbf{t})$ is the conditioning embedding discussed in the following subsections. At inference, we start from pure gaussian noise, and iteratively denoise it, steering the denoised image in the direction of the conditioning embedding.

Conditioning on context views

We use a two-stream image conditioning protocol from prior work [189] but modify it to suit our multi-frame setting. For conditioning on low-level features of the input context frames (such as depth, texture, and motion patterns of scene actors), we concatenate the K frames with the noisy input image to the UNet. For conditioning on higher-level features of the input context frames

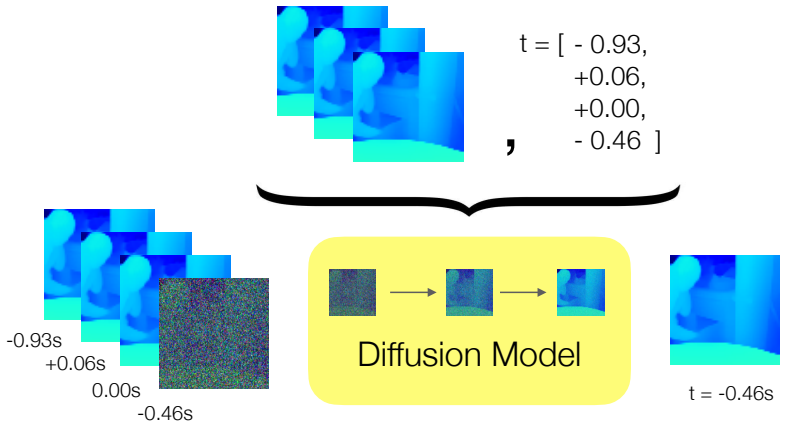


Figure 7.3. **High-level architecture** We use a diffusion model that conditions on three video frames, their corresponding timestamps and a query timestamp. It generates a single video frame for the query. We adopt the two-stream conditioning from image-to-image models [189], and (1) channel-concatenate the context frames with the noisy input to diffusion model, and (2) CLIP-encode the context frames for cross-attention across the UNet layers. Context and query timestamps are positionally encoded and concatenated with CLIP embeddings.

(such as the scene elements, contextual background, and observed camera trajectory), we pass the context frames through the CLIP image encoder [232] to get their image embeddings. We additionally construct a "residual" CLIP embedding for the target frame, by learning the weights on K embeddings and taking their weighted average. Intuitively, this "guides" the target image with a residual embedding that can be hooked onto, in order to generate the prediction.

Conditioning on timestamp scalars

In addition to building a conditioning mechanism for the context views, we also need to let the denoising UNet know, which timestamps the context frames belonged to, and which timestamp we are probing for. To accomplish this, we positionally encode the timestamp scalar with sinusoidal embeddings,

$$\gamma(t) = (\sin 2^0\pi t, \cos 2^0\pi t, \ldots, \sin 2^{L-1}\pi t, \cos 2^{L-1}\pi t) \tag{7.3}$$

This ensures that even if every timestamp value is not seen during training, any high-frequency variation of it can be approximated in the frequency domain at inference. We concatenate this with CLIP embeddings, and cross-attend them at every residual block in the UNet architecture.

Even though at *inference* this method addresses forecasting, we *train* it for a 'random

timestamp prediction' objective (*i.e.*, the order of K frames and their timestamps can be arbitrary), instead of the task of forecasting itself. We detail more results from this finding in Sec. 7.4.

Stitching together a video from individual frames

At inference, we generate long-horizon forecasts by predicting one frame at-a-time, which means that our model has to be queried more than once. Consider the case where we want to predict depth maps for T timesteps in the future. Prior work for long-horizon generation tends to make use of T sequential autoregressive next-frame predictions [99, 293], or $\log(T)$ hierarchical [99, 109, 268] predictions that first predict a low framerate future that is iteratively refined into $2\times$ higher framerate predictions (until T frames are generated). However, both sampling strategies have their drawbacks; autoregressive prediction may suffer from "drift" as the historical window of frames (to be conditioned on) will eventually contain only predicted frames rather than actual ground-truth histories. On the other hand, hierarchical sampling may not exhibit enough temporal coherence.

Interestingly, because our approach explicitly conditions on both input and output timestamps when making predictions, our trained model can support both such sampling strategies in addition to other more flexible approaches. We describe two such flexible approaches, which Sec. 7.4 shows perform better than the conventional sampling. First, given pairs of past frames and their timestamps, $\{\mathbf{c}_{-k:-1}, \mathbf{t}_{-k:-1}\}$, one can directly jump to all futures $t \in [1, T]$ independently. We term this ***D**irect* sampling. While this predicts more plausible futures because 'real' historical frames are used for conditioning, generated frames aren't temporally coherent (every frame might be sampled from a different future).

To improve temporal consistency, we propose *mixing* forecasts from direct sampling (which are accurate but temporally inconsistent) with forecasts from autoregressive sampling (which are temporally consistent but not as accurate as they are conditioned on the previously-predicted past, $\{\mathbf{c}_{t-k:t-1}, \mathbf{t}_{t-k:t-1}\}$). This means that for outputs x_D^T and x_A^T generated from direct and autoregressive sampling respectively, we can linearly combine these two inference pathways during the reverse diffusion process similar to classifier-free guidance [104],

$$x_D^t = g(\mathbf{c}_{-k:-1}, \mathbf{t}_{-k:-1}) \qquad x_A^t = g(\mathbf{c}_{t-k:t-1}, \mathbf{t}_{t-k:t-1})$$

$$x_M^t = x_A^t + w_m \cdot (x_D^t - x_A^t) \tag{7.4}$$

where w_m is the mixing guidance and g is a generative model. We term this sampling schedule, ***M**ixed* sampling. Intuitively, this makes samples from direct inference more coherent, and samples from autoregressive inference more plausible, as they now condition on a 'real' past. This also curbs the tendency of autoregressive inference to blow up at longer horizons as the output sample can now always fall back on predictions with direct inference.

Training details For all experiments in this work, $K = 3, L = 160, w_m = 2.0$. We train our architecture with classifier free guidance, *i.e.* we randomly remove the conditioning to generate unconditional frames (which can be used as a guidance signal during inference [104]). During training, the diffusion model predicts noise, and we set the probability of dropping the conditioning for classifier free guidance to 10%. During inference, we use a guidance of 2.0 for all experiments, with DDPM sampling for 40 iterative denoising steps. We do not perform diffusion in the latent space, but train and evaluate on images of size 64×64 using the Stable Diffusion Image Variations [161] UNet. To circumvent the use of VAE, we learn two new convolutional layers at the start and end of the UNet that help the input image to adjust to the weights of the latent space diffusion model, similar in spirit to prior works [203, 289] that also do not depend on the VAE. We learn all new layers $10\times$ faster than other layers, for training from scratch. We train the network with a batch size of 12 for 10k iterations (which takes $\sim$7 hours on 8 NVIDIA RTX A6000s), using AdamW with $\beta_1 = 0.95, \beta_2 = 0.999, \epsilon = 1e^{-8}$ and weight decay of $1e^{-6}$, with a learning rate of $1e^{-4}$.

7.4 Experiments

Benchmarking Setup

Datasets To cover a wide range of dynamic environments from a number of domains like activity recognition and self-driving, we use the large-vocabulary diverse tracking dataset, TAO [52]. TAO is a collection of seven different datasets that is originally used for multi-object tracking. For its unconstrained *dynamic* nature of videos, we repurpose it for predictive modeling. For rigid scenes, we also include video sequences from Common Objects in 3D (CO3Dv2) [244]. CO3Dv2 is a collection of 19k video sequences spanning objects from 51 MS-COCO [181] categories, designed for use in object-level 3D reconstruction and new-view synthesis of *static* scenes. We experiment with three different modalities: RGB videos, their luminance channels and most importantly, sequences of *pseudo-depth*, where the pseudo-depth is obtained from a single-frame monocular depth estimator, ZoeDepth [20], that predicts metric depth for scenes. We randomly sample the

input and output frames in a window of 8s across the entire length of a video and shuffle the frame ordering for training. For dataset splits of TAO and using metric depth from CO3Dv2, please see supplement.

Evaluation settings For benchmarking, we consider two settings. First, we evaluate single-frame forecasting. Because this is a scalable evaluation, we benchmark all baselines discussed below and do all ablations for the setting where methods are asked to generate a single prediction for either the future +1s or +10s with input frames given at {-1.0, -0.5, 0}s. Note the forecasting windows are motivated by and reminiscent of motion planning benchmarking [31, 152].

Second, we evaluate multi-frame forecasting for up to +10s long horizon. This setting allows us to empirically evaluate the proposed direct and mixed sampling schedules. The input is still provided at {-1.0, -0.5, 0}s and samplers generate predictions for future {+1, +2, +3, ..., +10s}.

Metrics For evaluating depth prediction across both TAO and CO3Dv2 datasets, we adopt the scale and shift invariant L1 error on relative depth maps from monocular depth estimation literature [164], where scale and shift are computed as a minimization of the following least squares objective:

$$(s,t) = \arg\min_{(s,t)} \sum_{i=1}^{M} (s\mathbf{d}_i + t - \mathbf{d}_i^*)^2 \tag{7.5}$$

Here, $\mathbf{d}_i$ is the set of per-pixel predicted depths, and $\mathbf{d}_i^*$ are the corresponding groundtruth values. Using Eq. 4, the L1 error is computed as $e = \frac{1}{M}\sum_{i=1}^{M} |s\mathbf{d}_i + t - \mathbf{d}_i^*|$. For evaluating both grayscale and RGB modalities, we follow prior work in novel-view synthesis [206, 373] and compute the peak-signal-to-noise ratio (PSNR), which measures mean color difference. We take motivation from the forecasting literature in the autonomous driving domain [39] and use Top-k versions of both L1 and PSNR metrics: we take k samples from the model and report the best L1 / PSNR of k. When benchmarking multi-frame depth forecasting, we compute an average trajectory error (ATE, *i.e.* L1 error across the entire predicted sequence), and compute the Top-k errors across a set of k trajectories.

Baselines We compare to state-of-the-art video prediction architectures MCVD [293], FDM [99] and RIVER [53] and construct three simple baselines for video prediction: (1) constant past which predicts the current frame as the future, (2) linear extrapolation from the two temporally closest context frames, and (3) non-linear regression, which is trained for the task of forecasting the next +1.0s using our architecture but without cross-attention layers (therefore, no conditioning)

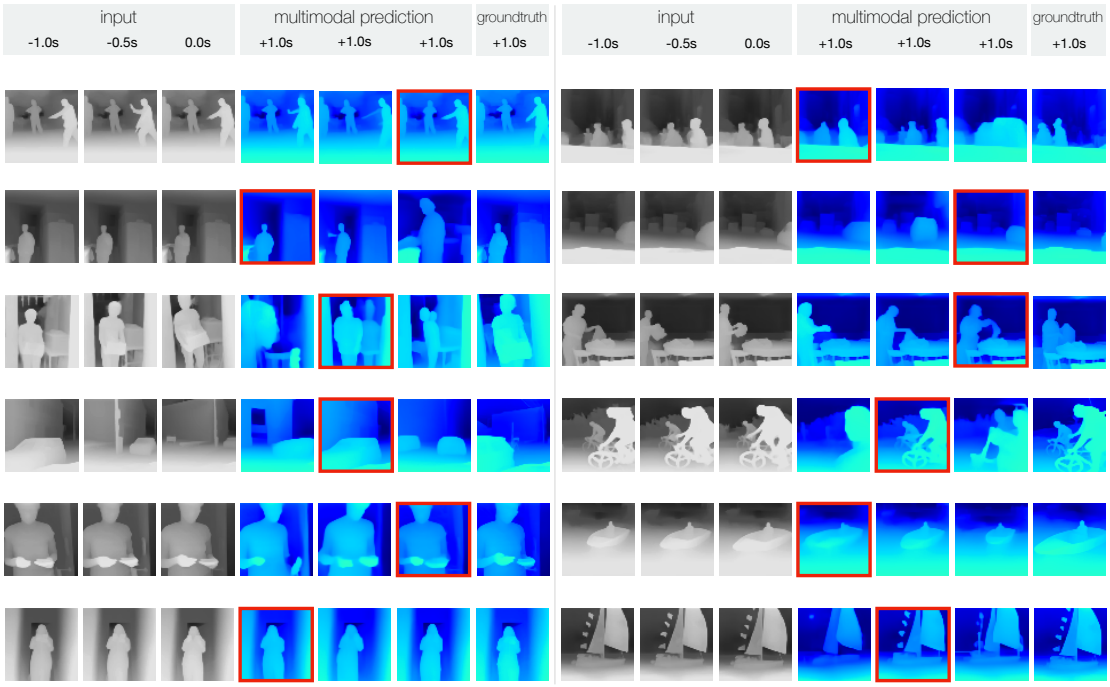


Figure 7.4. **Qualitative analysis of single-frame short horizon forecasting** We show examples of **input**-**output**-groundtruth triplets. Given 3 past frames as input, we show 3 different samples of the future from our diffusion network, and the corresponding groundtruth. Prediction highlighted in red is the closest to groundtruth. Despite learning from only 1000 videos and training for only 7 hours, our method learns to generate multiple realistic futures and listens to low-level details in the historical context frames (e.g., scene structure, actors performing events, and overall camera motion). For reference, the events across examples in row major form could be described as, 'playing in field', 'crossing road', 'doing laundry', 'driving (front view)', 'exiting room while holding a box', 'picking up from table', 'driving (side view)', 'biking', 'fidgeting', 'boating with camera zooming in', 'standing in hallway', 'sailing'.

with an L2 loss on the predicted depth from diffusion model. We retrain MCVD [293], FDM [99] and RIVER [53] on our TAO pseudo-depth dataset and use them at inference for single-frame forecasting given three past frames. For MCVD, we use the 'concat' variant as it has lower memory requirements.

Finally, in the setting where the scene is rigid but camera has a non-zero motion, like in CO3Dv2, we compare to a state-of-the-art method for sparse (3-) view reconstruction, SparseFusion [373], on the task of novel-view depth synthesis. Here, we evaluate on a randomly sampled set of test sequences from the core subset proposed in a prior work [244]. This subset consists of

10 object categories from CO3Dv2. All experiments, including qualitative analysis, on CO3Dv2 against SparseFusion can be found in the supplement.

Comparison to state-of-the-art

We begin the quantitative analysis by comparing our method to MCVD [293], FDM [99] and RIVER [53] for future timestamp prediction in dynamic videos.

Short horizon forecasting We evaluate our method and all baselines for single-frame +1s forecasting in Tab. 7.1. We find that our method outperforms state-of-the-art video prediction methods, MCVD [293], FDM [99] and RIVER [53]. We posit that against MCVD, our randomized frame prediction objective during training and additional conditioning on timestamps, helps in learning better temporal coherence across frames. FDM, specifically, is not designed for scenes that have dynamic actors, so may perform suboptimally when learning to handle dynamics. RIVER's bottleneck is video prediction in a significantly low dimensional latent space which results in imprecise reconstructions at inference.

When comparing our method to simple baselines such as the (non-learned) constant past and linear extrapolation, and the unimodal non-linear regression, it becomes readily apparent that, (1) both constant past and linear extrapolation are *strong* baselines for scenes that are static and have been captured by a stationary camera, and (2) regression, expectedly, stands out as an even stronger baseline (often used by pioneering work in occupancy forecasting [152]) but regresses to the mean of multi-modal distribution of possible futures. This mean-seeking behaviour still suffices for most scenes and metrics (such as our *mean* Top-1 L1 error), but our method provides the increased capability of sampling multiple futures which reduces the probabilistic Top-5 L1 further. An in-depth qualitative analysis of all baselines along with our method, and a training / inference runtime analysis, can be found in the supplement.

Long horizon forecasting First, we evaluate the single-frame forecasting for +10s using three different sampling schedules as discussed in Sec. 7.3. Note that in the single-frame case, direct and hierarchical sampling are equivalent as the first lowest framerate layer of hierarchical sampling generates the +10s frame *directly* from the given inputs. Compared to the baselines in Tab. 7.2, we find that our proposed mixed sampling strategy performs the best at the probabilistic L1, while surprisingly constant prediction suffices for the Top-1 metric.

Second, in Tab. 7.3, we benchmark different sampling schedules discussed for the multi-frame forecasting case with Top-k ATE, where samplers predict a 1fps sequence up to 10s in the future. First, we find that directly jumping to a future frame, performs better than the conventional

Method	Top-1 L1	Top-3 L1	Top-5 L1
Linear extrapolation	21.25	21.25	21.25
Non-linear regression	7.96	7.96	7.96
Constant past	**7.15**	7.15	7.15
RIVER [53]	10.82	10.32	10.17
MCVD [293]	10.54	7.83	7.12
FDM [99]	9.99	7.78	7.24
Ours	8.40	**6.93**	**6.59**

Table 7.1. **Comparison to state-of-the-art** We evaluate future depth prediction for +1s against state-of-the-art video prediction methods by retraining them for pseudo-depth prediction, and against other simple or non-learned baselines. We find that our method beats prior work with a substantial margin.

Method	Top-1 L1	Top-3 L1	Top-5 L1
Linear extrapolation	21.80	21.80	21.80
Non-linear regression	14.76	14.76	14.76
Constant past	**11.61**	11.61	11.61
Ours (autoreg.)	12.93	11.24	10.77
Ours (direct)	12.65	11.13	10.65
Ours (mixed)	12.39	**10.97**	**10.51**

Table 7.2. **Single-frame long horizon forecasting** We evaluate future depth prediction for +10s against the discussed baselines. Given our timestamp conditioning, we are able to explore more flexible sampling schedules like direct and mixed, which perform better than the widely used autoregressive sampling.

autoregressive and hierarchical sampling schedules. Specifically, for autoregressive sampling, the error in prediction starts adding up as the diffusion models starts conditioning on predicted frames rather than the groundtruth past. For hierarchical sampling, the future is coarsely decided by the first set of predictions. After this, intermediate frames can only be interpolated and the future cannot be refined. Finally, for mixed sampling, we find that it produces more accurate and coherent futures as it benefits from the advantages of both direct & autoregressive sampling (Fig. 7.5).

Comparison between different modalities

We also explore luminance and RGB modalities for single-frame +1s video prediction. Specifically, instead of pseudo-depth, we train our model for luminance and RGB prediction under the short-

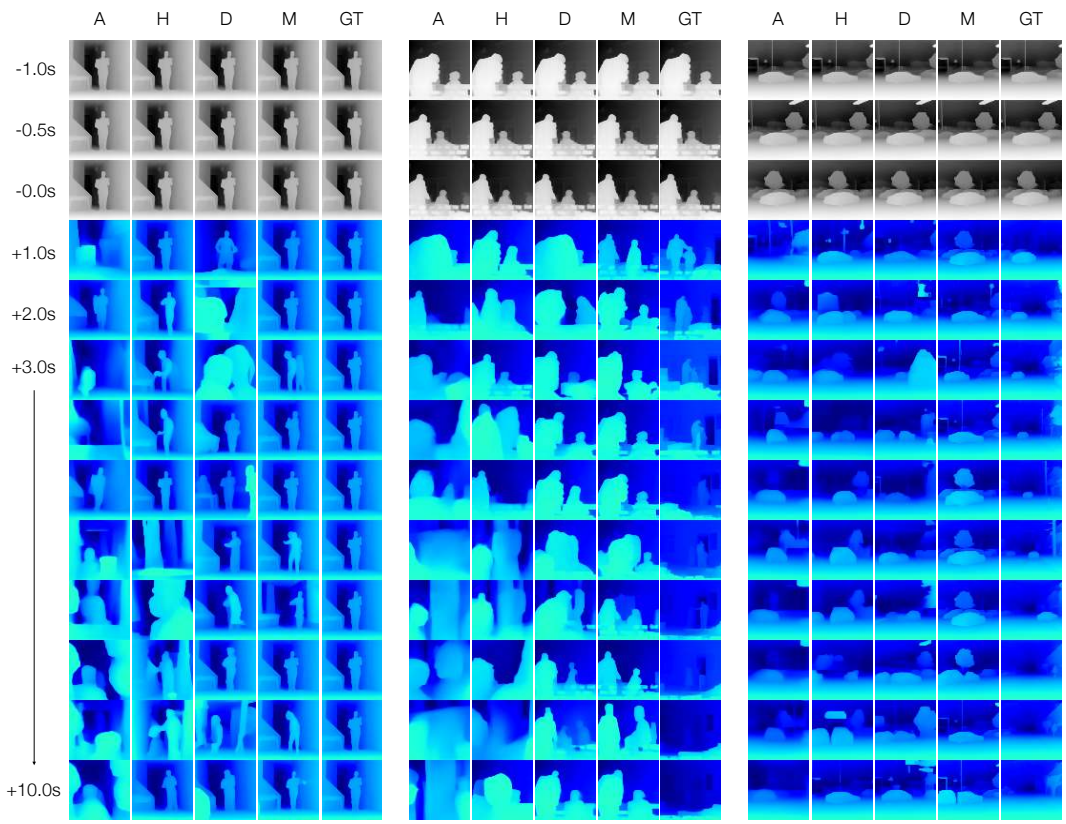


Figure 7.5. **Comparison between sampling strategies** We qualitatively analyse the **predictions** from the four discussed sampling strategies given same **past** alongside the **G**round**T**ruth: **A**utoregressive and **H**ierarchical [99, 109, 293], and **D**irect and **M**ixed, which are enabled by our timestamp conditioning. As detailed in Sec. 7.3, we find that autoregressive sampling suffers from "drift", and the performance of hierarchical sampling is governed by its first layer of forecasts (*i.e.* lacks flexibility). While direct sampling does better, it cannot produce coherent futures. Concretely, we propose mixed sampling, which mixes both the coherence of autoregressive and the accuracy of direct samples. For reference, the samples from left to right could be described as, 'standing in hallway', 'interaction between two people', 'side-view from a driving car'.

horizon forecasting setting. When evaluating RGB, we factor out the luminance channel from the prediction and use that for benchmarking against our luminance prediction model. In Tab. 7.4, we see that introducing invariances in the input data (such as learning from luminance rather than a combination of color and texture), helps in making forecasting easier. Quantitatively, the Top-5 PSNR increases by a large margin of ∼2.1 points.

We also compare our depth and RGB prediction models by running Stable Diffusion Depth2Img on the predicted depth. We find that, (1) our depth is readily usable for downstream tasks, and

Method	Top-1 ATE	Top-3 ATE	Top-5 ATE
Ours (autoreg.)	15.20	13.56	13.06
Ours (hierar.)	15.15	13.77	13.32
Ours (direct)	13.54	12.73	12.43
Ours (mixed)	**12.16**	**11.73**	**11.58**

Table 7.3. **Multi-frame long horizon forecasting** We evaluate multiple sampling strategies for generating a sequence of future depths upto +10s. We evaluate with Top-k ATE and find that our proposed mixed sampling, which is able to generate accurate and coherent futures, performs the best of all.

Method	Top-1	Top-3	Top-5	Evaluation
Ours-L	**16.32**	**17.07**	**17.33**	Luminance
Ours-RGB	12.16	14.47	15.24	
Ours-D	**16.28**	**16.44**	**16.50**	Color
Ours-RGB	14.10	15.40	15.80	
Ours-D	**8.40**	**6.93**	**6.59**	Pseudo-depth
Ours-L	22.68	19.17	17.61	
Ours-RGB	27.05	20.88	19.33	

Table 7.4. **Comparison between different modalities.** We quantitatively enable a fair comparison between modalities by evaluating them for either pseudo-depth, luminance, or RGB forecasting. We consistently find that invariant modalities like depth and luminance perform drastically better than RGB at video prediction. **L**uminance and **C**olor models are evaluated with PSNR and **D**epth with L1.

(2) it is infact easier to do RGB prediction by learning to forecast scene depth first! For details on the depth2img parameters and text prompts used, please see supplement.

Finally, we compare all three modalities for the task of pseudo-depth forecasting. This requires running ZoeDepth [20] on our predictions from the luminance and RGB models. We once again find that it is easier to directly learn to forecast depth, without depending on color or scene texture.

Architecture ablations

We ablate our design decisions in Tab. 7.5 for +1s forecasting. For a fair comparison with MCVD, FDM and RIVER and to see how much performance boost we get from the Stable Diffusion Image Variations weights, we attempt to train from scratch. Surprisingly, this training does not take much longer than finetuning (11 hours for training from scratch vs. 7 hours for finetuning), and performs remarkably well (still better than the state-of-the-art). We further attempt to

reduce the number of parameters in our network by removing 1 convolution block each from the UNet encoder and decoder. This brings number of parameters closer to the state-of-the-art video prediction models, and training the smaller model from scratch still beats all baselines. For exact parameter counts, see supplement.

Next, we find that the CLIP embedding is essential to conditioning on the past context frames and results in a drop of ∼1.4 points if ablated. Finally, we ablate the design decisions for the timestamp conditioning.

Anchoring timestamps When designing the timestamp conditioning, we find that it helps to condition on relative rather than absolute timestamps. This includes, "anchoring" timestamps to a constant frame in the input such that that frame always occurs at t=0s. For our experiments, we choose the third context frame as anchor, and this frame at timestamp +0s becomes the 'current' frame for the diffusion network. This practice has recently been adopted by methods [295] that use diffusion models for conditioning on 3D cues such as camera pose.

Timestamp randomization One of our key insights is that training directly for the task of forecasting is sub-optimal to training for a random frame prediction objective. Specifically, the drop in performance is rather significant (∼1.8 Top-1 L1 points). This aligns with the insights from masked autoencoder literature [101, 284, 297] where randomization in masking results in better representations. Analogously, destroying structure in the data and making the final task harder for the diffusion models, helps in building robust temporal understanding.

Applications

In Fig. 7.6, we show qualitative examples of different applications our approach can be used for: (1) generating videos at varying framerates for different horizons given the same context frames, (2) frame interpolation at fractional timestamps between the given context frames, and (3) looking back in the past with negative timestamps given the future frames as context.

7.5 Discussion

We focus on the problem of predicting the future from past sensor observations, and take motivation from the neuroscience literature on predictive coding. Since the future is multi-modal and can therefore unfold in multiple ways, we lean on the explosive advancements in large-scale pretraining of diffusion models, that can internally represent such multi-modal distributions. With two key modifications to *image* diffusion networks, we come up with a method for predictive

Method	Top-1	Top-3	Top-5
Ours	8.40	**6.93**	**6.59**
- pretrained weights	**7.89**	7.04	6.78
- 2 × conv blocks	8.50	7.27	6.89
- CLIP embedding	9.19	8.25	7.95
- timestamp anchoring	9.00	7.08	6.62
- random timestamps	10.24	7.89	7.31

Table 7.5. **Architecture ablations.** We ablate our method under the single-frame +1s forecasting setting with L1 error. We assess the benefits from using pretrained weights [161], a large model, and CLIP embeddings for context frames. We additionally investigate the design choices in creating the timestamp conditioning, by using relative timestamps and randomizing their order. Ablations indicate that all design choices play a crucial role.

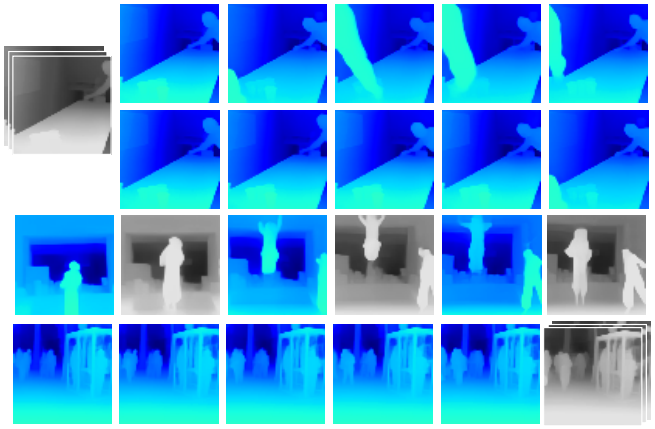


Figure 7.6. **Video applications.** We show examples of how the formulation of our method unlocks multiple video applications: variable framerate forecasting (top row at 1FPS, second row at 5FPS), (third row) frame interpolation given the frames in gray, and (last row) backcasting at 5FPS given the future. For reference, events from top to bottom could be described as, 'playing pool', 'jumping', 'walking on a busy street'.

video modeling. We find that for training with moderately-sized datasets, it helps to introduce invariances in the data – such as forecasting only pseudo-depth or luminance of real-world images. Physical quantities like pseudo-depth are readily usable by downstream tasks in robot autonomy (locomotion and planning) as they represent the time-to-contact. We introduce a mechanism for diffusion models to condition on a frame's timestamp. This allows models to perform better at the task of forecasting (especially when they are *not trained* for forecasting). Timestamp conditioning also lets us come up with flexible sampling schedules for long-horizon forecasting. We find that these new sampling schemes perform better than conventional autoregressive or

hierarchical sampling strategies.

7.6 Appendix

In this appendix, we extend our discussion of the performance of our method on predicting diverse future geometries, both qualitatively and quantitatively.

Dataset splits and evaluation

We use the diverse TAO [52] dataset for learning dynamism in unconstrained scenes. Since TAO is a tracking *benchmark*, its training set is smaller than the validation or test sets. For this reason, we train on the validation set of TAO (∼1000 videos) and report all results on one randomly sampled subsequence each, from the train set (containing about 500 videos). The randomly sampled set is fixed across all experiments for fair comparison.

In the case of evaluation on rigid scenes, we use CO3Dv2 [244]. Although CO3Dv2 has groundtruth depth that is obtained from COLMAP [262], it is not dense. For this reason, we still run ZoeDepth on CO3Dv2 and use those pseudo-depth maps for training our method, but use the valid depths from groundtruth for computing the probabilistic L1 metric on CO3Dv2 for both our method and the baseline, Sparsefusion [373]. In the following section, we analyse depth forecasting on CO3Dv2 qualitatively and quantitatively.

Novel-view synthesis

We consider the case where a moving camera captures a static scene. In literature, this has been studied under the umbrella of novel-view synthesis from dense [206] or sparse views [244, 373]. The setting we evaluate (context from {-1s, -0.5s, 0s} and prediction at +1s) falls under sparse view reconstruction/synthesis. We use a variant of our model trained on CO3Dv2 alongside TAO. Note that the state-of-the-art method, SparseFusion [373], which we use as a baseline has access to future camera pose for rendering the novel-view from its reconstruction, whereas for our method, the camera pose is *unknown*. Along with the scene, it is sampled from the timestamp-conditioned diffusion model during inference. Despite this disadvantage, we find that our method predicts plausible depths for the objects, *in addition* to the depth predictions for the object backgrounds, which is ignored by SparseFusion. We cover some qualitative analysis in Fig. 7.7.

In Tab. 7.6, we formally evaluate the task of novel-view synthesis. Since CO3Dv2 has multi-view object data captured in the form of videos, we structure this problem as, given frames at -1.0s,

Method	Donut	Apple	Hydrant	Vase	Cake	Ball	Bench	Suitcase	Teddybear	Plant	Overall
SparseFusion	11.54	28.94	**19.04**	**14.29**	26.28	27.64	**75.89**	34.32	40.04	**71.53**	**34.95**
Ours	**7.22**	**19.23**	30.39	21.82	**19.57**	**20.65**	91.83	**33.56**	**38.44**	75.23	35.79

Table 7.6. **Novel-view synthesis results on Co3Dv2 core subset.** We evaluate our method for the task of novel-view depth synthesis with Top-1 L1 error on normalized depth, against a recent approach for 3-view reconstruction. Over the set of categories in the core subset of CO3Dv2, we see that SparseFusion performs better overall. Unlike SparseFusion, our method does not have the access to future camera pose or object mask. Despite this, it is able to generate plausible depth maps for object turn-table sequences in Co3Dv2. We only compute the metric on valid groundtruth depths inside the given object mask in CO3Dv2, without penalizing the background forecasts.

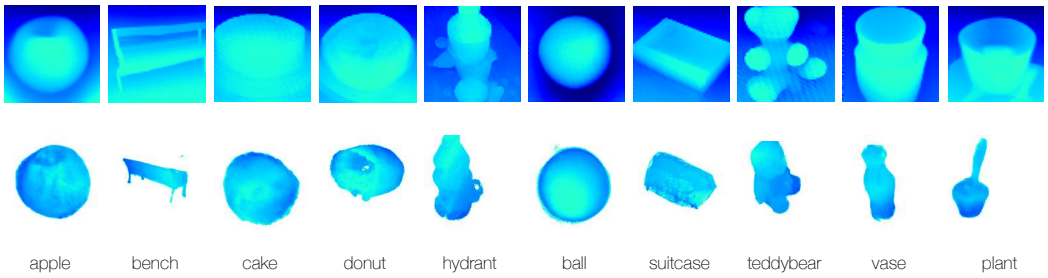


Figure 7.7. **Qualitative comparison to novel-view synthesis** We train and qualitatively evaluate our method on CO3Dv2. From the core subset [244] of 10 categories in CO3Dv2, we show novel-view synthesis from both our method (**top**) and SparseFusion [373] (**bottom**). While SparseFusion has access to the parameters of both the input and new (or future) view, these are implicitly estimated by our method from the camera trajectory encoded in the past frames. Therefore, our method does not rely on known camera poses! Qualitatively, our method performs favourably on the task of new-view synthesis from 3 input views, while handling dynamics and backgrounds in general for a wide variety of scenes.

-0.5s, 0.0s, we want to predict the frame at +1.0s. For our method, only the future timestamp is available. For SparseFusion, instead of future timestamp, future camera pose information is available. Quantitatively, we find that our method performs better than SparseFusion on a few categories (`donut`, `apple`, `ball`, `suitcase`, etc.) because of more smooth depth forecasts (ref. Fig. 7.7). Other than that, for categories where camera viewpoint matters more (`hydrant`, `bench`, `plant` etc.) for rendering the geometry, SparseFusion does better.

More importantly, the extension of our method for the task of novel-view synthesis coupled with its performance on forecasting for dynamic scenes, we show that we can handle object backgrounds, and dynamic video settings such as in TAO [52], unlike methods for sparse-view static object/scene reconstruction like SparseFusion.

Figure 7.8. With same training data/architecture/duration, a depth or luminance model learns better temporal coherence than RGB.

Qualitative comparison with baselines

In Fig. 7.9-7.12, we qualitatively compare to all baselines discussed in Tab. 1. We see that predicting the most recent past frame as the future serves as a strong baseline. Non-linear regression, regresses to the mean of the future distribution. FDM [99], RIVER [53], MCVD [293] and our method instead, sample *modes* of the future distribution. Linear extrapolation is not shown but it serves as a strong baseline when the scene is static. Overall, we see that our method produces more realistic and diverse outputs, as compared to MCVD [293] which usually does not diverge much from the input views. RIVER [53] struggles to learn temporal coherence because of its processing in the low-dimensional latent space, and FDM [99] is not able to learn precise object boundaries likely because it is not designed to handle dynamic scenes.

Mean vs. mode-seeking behavior Fig. 7.9, row 1 shows how the non-linear regression baseline hallucinates multiple possible futures, thereby introducing artifacts because of this phenomenon (e.g., multiple people are visible in the output). In contrast to this, our method and other state-of-the-art approaches are able to sample multiple futures separately, commonly referred to as the mode-sampling or mode-seeking behavior.

Depth vs. luminance vs. RGB Fig. 7.8 shows a qualitative comparison between forecasts from different modalities; we see that RGB forecasting tends to be noisy. Temporal coherence is better learned with invariant modalities such as pseudo-depth or luminance. While many recent works do show successful RGB video generation [114, 171], they typically train on far more data than us (days of compute on 10 million videos vs 7 hours of compute on 1000 videos).

Comparison to state-of-the-art on long-horizon forecasting

In Tab. 7.7, we compare the performance of our method shown in the main paper, by retraining FDM [99] and RIVER [53] for +10s forecasting. Note that this is not an apples-to-apples

Method	L1			ATE		
	Top-1	Top-3	Top-5	Top-1	Top-3	Top-5
FDM [99]	16.05	13.15	12.29	16.57	14.38	13.79
RIVER [53]	13.21	11.71	11.26	13.34	12.28	11.93
Ours	**12.39**	**10.97**	**10.51**	**12.16**	**11.73**	**11.58**

Table 7.7. **Long horizon forecasting** We evaluate future depth prediction for +10s against FDM and RIVER, two state-of-the-art methods for video generation in the single-frame (with L1) and multi-frame (with ATE) settings. Given our timestamp conditioning, we are able to explore more flexible sampling schedules like mixed sampling, which performs better than the widely used autoregressive sampling strategies for FDM and RIVER.

comparison to our method, as even for the case where we want to predict just the +10s frame with the baselines, they are forced to predict every intermediate (0s to 10s) frame because this is the only way to reach the future +10s. On the other hand, when we evaluate our method for single-frame +10s forecasting, we directly jump to that timestamp.

Quantitatively, (1) errors are higher when methods are used for predicting sequences of future frames, rather than when evaluated for a single timestamp in future, and (2) across the discussed settings, our method performs the best of all with the proposed mixed sampling.

Memory requirements and speed In Tab. 7.8, we detail the memory and speed requirements of our method and its variants along with the state-of-the-art for the task of +1s single-frame forecasting. First, we find that at inference, our method samples the fastest from the diffusion model. Second, FDM [99] uses the least amount of memory as it has the smallest model. RIVER [53] also uses lesser memory for a lighter architecture since it learns video generation in significantly low dimensional latent space. While these methods allow for a smaller memory footprint, as seen qualitatively and quantitatively, none of them is able to learn persistence and temporal coherence of objects and scenes. For a fair comparison to baselines, we see that a variant of our model that is not initialized with the Stable Diffusion Image Variations weights finishes training in 11 hours, still better than all baselines. Another variant of our model that has lesser parameters and is more comparable to baselines, is much faster to both train and sample from.

Table 7.8. Resource requirements of baselines for single-frame +1s forecasting.

Method	Params. (M)	Mem. (GB)	Train (hrs.)	Test (s)
RIVER [53]	236	12	32	6.90
MCVD [293]	565	19	66	12.50
FDM [99]	**80**	**8**	72	24.41
Ours	860	21	**7**	4.09
Ours (scratch)	860	21	11	4.09
Ours (small, scratch)	399	16	8	**3.78**

All numbers are provided for batch size = 1. For RIVER, a VQ-GAN needs to be trained

whose number of parameters (68M) are added to the RIVER parameters (168M). Note that all our variants quantitatively perform better than the state-of-the-art as shown in the main paper, and these differences in the training and inference resources are even more pronounced when the state-of-the-art methods are used for multi-frame long-horizon future generation.

Stable Diffusion Depth2Img In the main paper, we show that given the same amount of training resources, it is better to train a depth video prediction diffusion model and use this 'temporally-aware' depth in conjunction with a single-frame depth-to-image model (such as Stable Diffusion Depth2Img [252]) than an RGB video prediction model. To get this RGB image for every predicted *future* depth frame, we input the RGB image at timestamp `t = 0s` (which is the last input timestamp), alongside every predicted depth frame from the future, into the Stable Diffusion Depth2Img model one-at-a-time. We use the LLaVA [185] model to caption the RGB image at `t = 0s` which is input as the text prompt for the depth-to-image generation. We use `'ugly looking, bad quality, cartoonish'` as the negative text prompt. The guidance scale is set to 5.0 and the conditioning strength is set to 0.3.

LLaVA prompting To get the text prompt from LLaVA, we use the HuggingFace `llava-hf/llava-1.5-7b-hf` weights, and use the input prompt for LLaVA as, `"<image>\nUSER: Caption the image in one long sentence.\nASSISTANT:"`. All text returned after this prompt is used as input to Stable Diffusion Depth2Img. The RGB images are resized at a 512 × 512 resolution, and a max output length of 500 characters is used.

Limitations Our method suffers from two important limitations. First, our method is biased towards hallucinating people and cars. For the other categories, the future is rather difficult to forecast. This limitation arises from the fact that TAO, which is used for training, has ∼52% of the objects as people, with the second most common category being cars. However, when even a small finetuning set of varied objects from CO3Dv2 [244] are used, our method does perform well on forecasting the future for those categories.

Second, we find the pseudo-depth produced by our method (and others) is low-fidelity, lacking details that otherwise appear in inputs. We posit that this is because although neural networks are universal function approximators, they struggle with modeling high-frequencies.

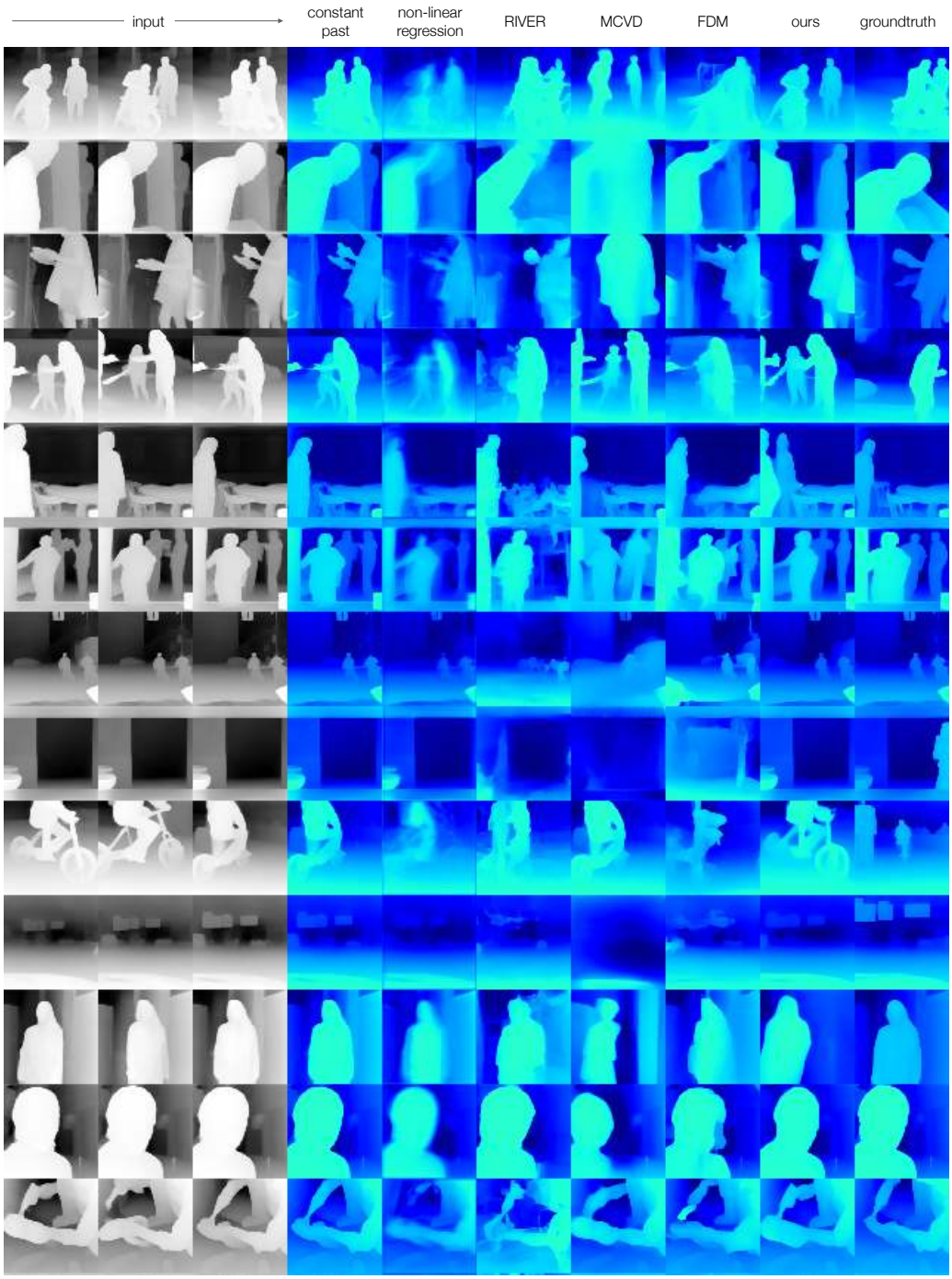


Figure 7.9. **Qualitative comparison to baselines (1 of 4).** We compare to all baselines for the task of short-horizon forecasting on TAO-val set. Given inputs at -1.0, -0.5, 0.0s in gray, methods predict future pseudo-depth at +1.0s. Lighter color is closer depth.

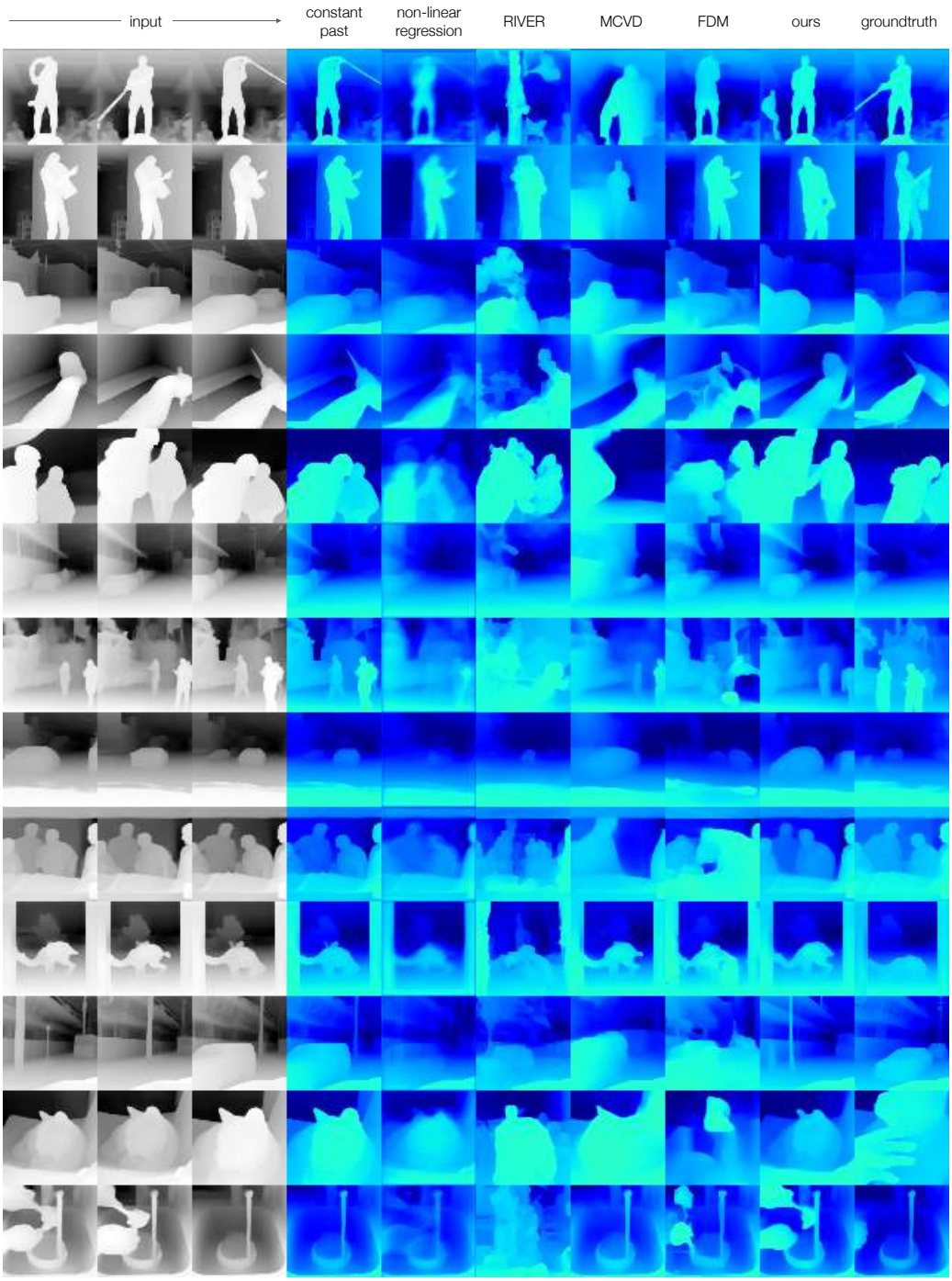


Figure 7.10. **Qualitative comparison to baselines (2 of 4).** We compare to all baselines for the task of short-horizon forecasting on TAO-val set. Given inputs at -1.0, -0.5, 0.0s in gray, methods predict future pseudo-depth at +1.0s. Lighter color is closer depth.

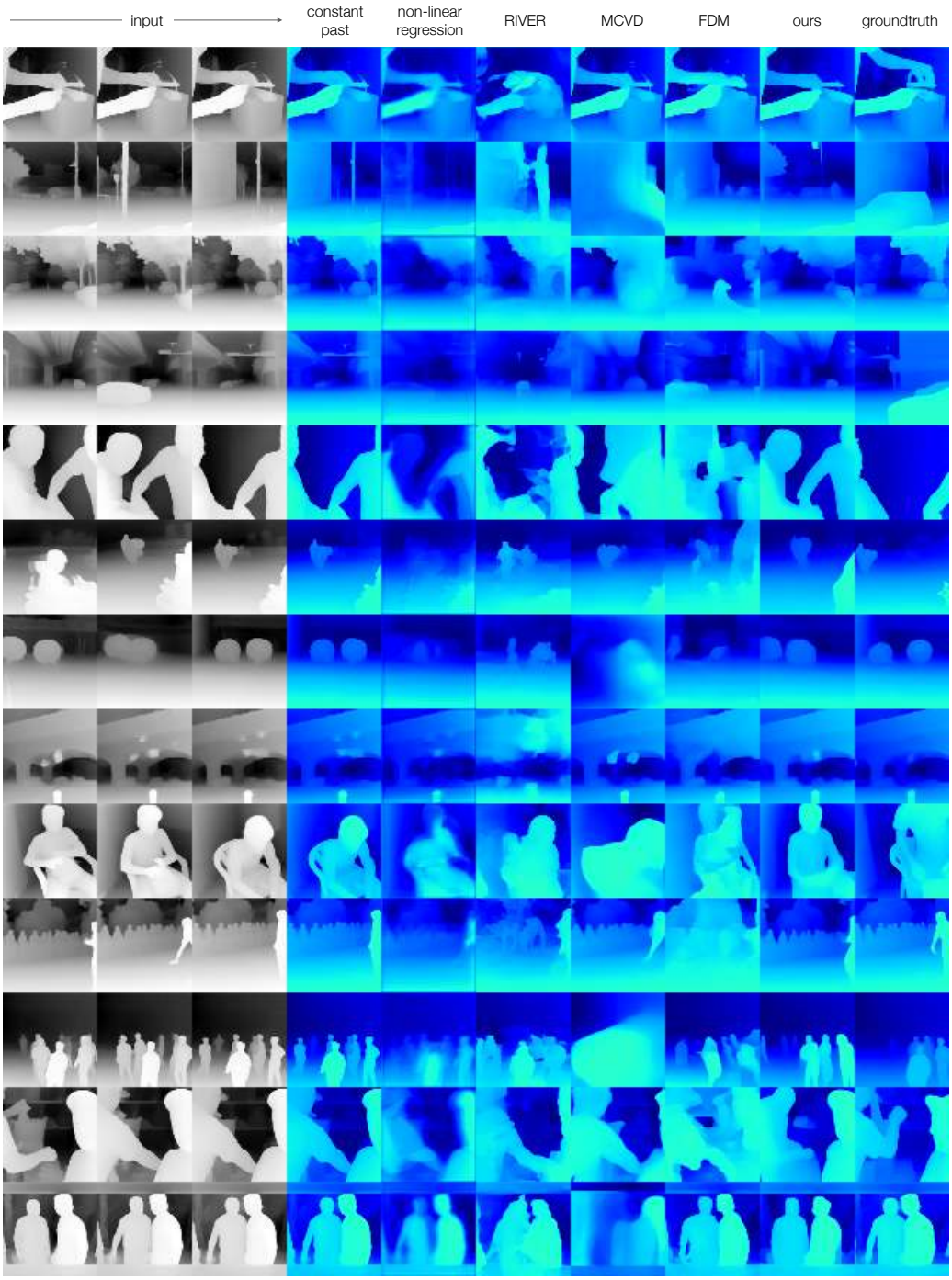


Figure 7.11. **Qualitative comparison to baselines (3 of 4).** We compare to all baselines for the task of short-horizon forecasting on TAO-val set. Given inputs at -1.0, -0.5, 0.0s in gray, methods predict future pseudo-depth at +1.0s. Lighter color is closer depth.

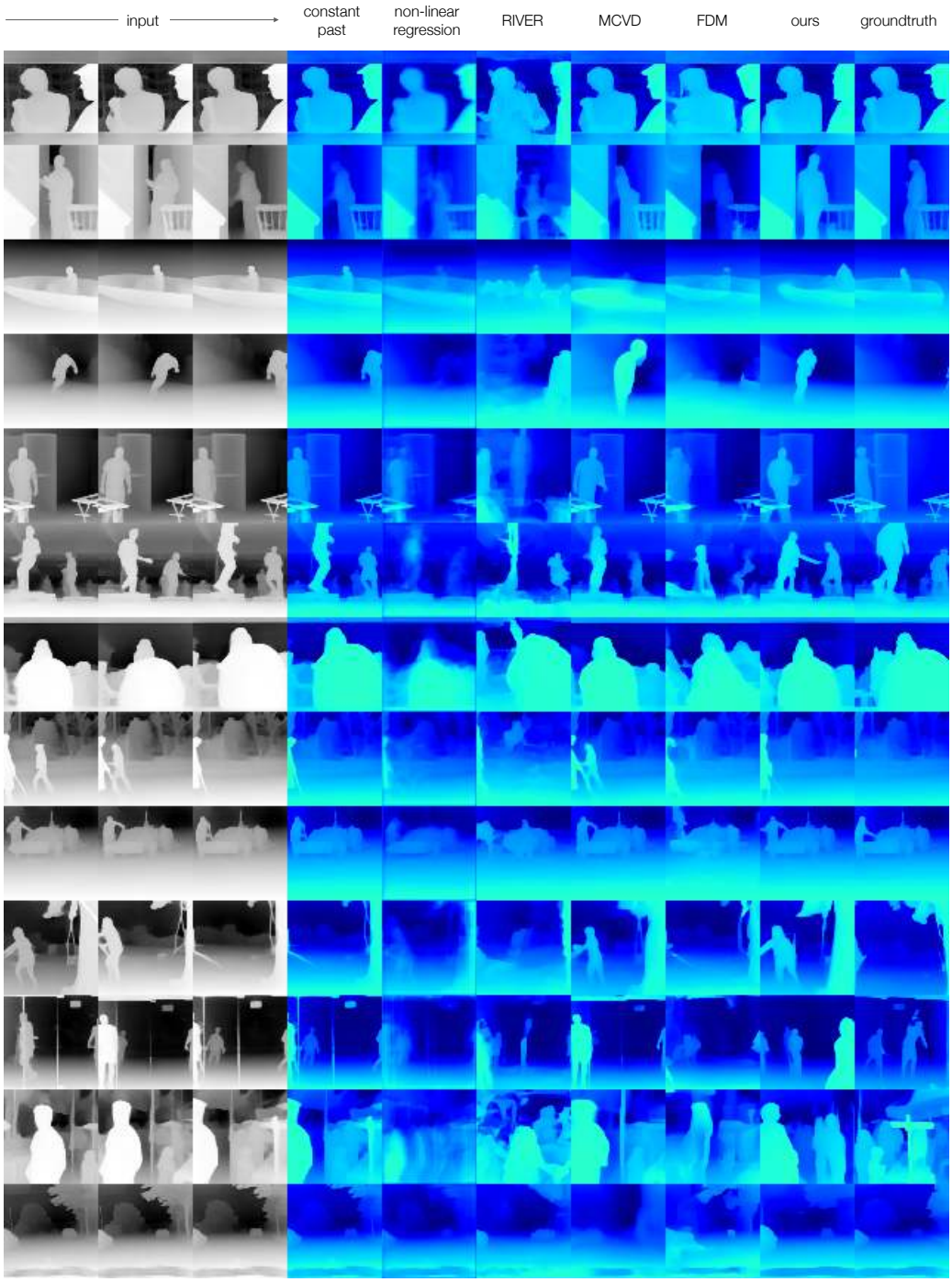


Figure 7.12. **Qualitative comparison to baselines (4 of 4).** We compare to all baselines for the task of short-horizon forecasting on TAO-val set. Given inputs at -1.0, -0.5, 0.0s in gray, methods predict future pseudo-depth at +1.0s. Lighter color is closer depth.

Chapter 8

Learning foundational 3D priors via dynamic novel view synthesis

Publication information

Chen, K., Khurana, T., and Ramanan, D., 2025. Reconstruct, Inpaint, Finetune: Dynamic Novel-view Synthesis from Monocular Videos. In Proceedings of Neural Information Processing Systems (NeurIPS).

8.1 Introduction

Rapid advances in static 3D scene representations [144, 206] have paved the way for spacetime understanding of the dynamic world. This has enabled photorealistic content creation and immersive virtual reality applications. In this work, we focus on the problem of novel-view synthesis from casually-captured monocular videos of dynamic scenes.

Why is this hard? Prior work on dynamic view synthesis addresses this task from two extremes. The first class of methods "test-time" optimize a new 4D representation from scratch for every new test video. While this ensures physically-plausible scene geometry, careful choices in modeling scene motion – in the form of an independent deformation field, or learnable temporal offsets – have to be made [176, 298, 322]. More importantly, it can take on the order of hours to optimize and render a novel-view video. An attractive alternative is to train large feed-forward video models *directly* for view synthesis [111, 364]. While inference on such models is dramatically faster (on the order of milliseconds), the resulting renderings often are not as accurate as their

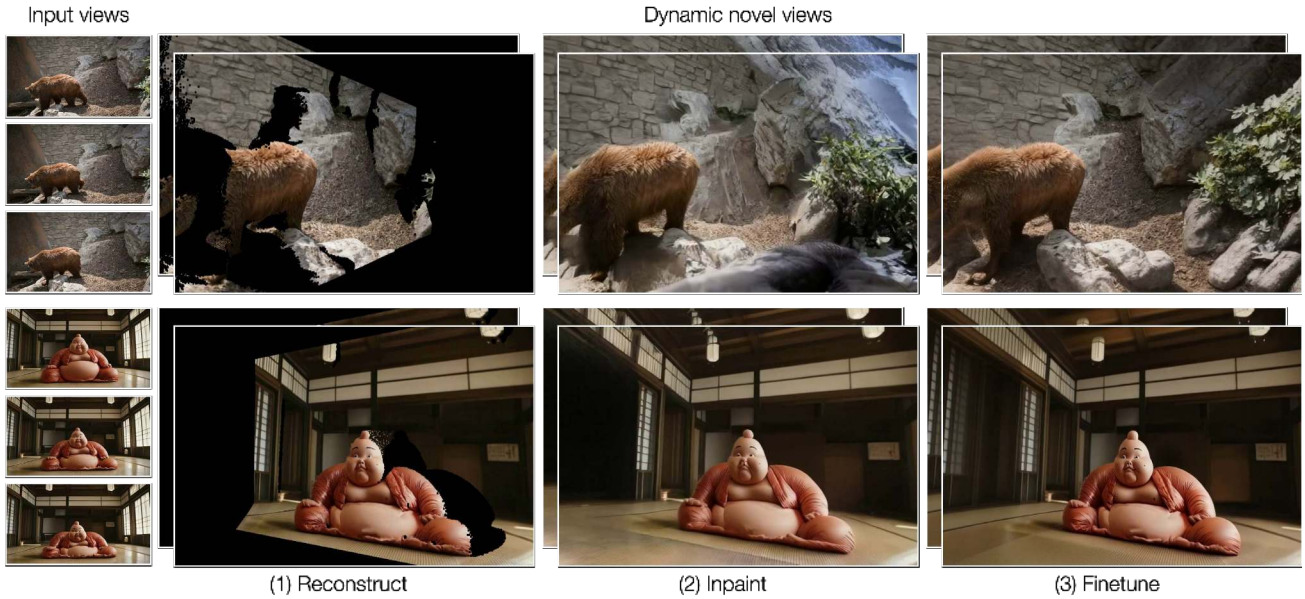


Figure 8.1. We present CogNVS, a video diffusion model that enables novel-view synthesis of dynamic scenes. Given an in-the-wild monocular video of a dynamic scene, we first reconstruct the scene, render it from the target novel-view and inpaint any unobserved regions. Because CogNVS can be pre-trained via self-supervision, it can also be test-time-finetuned on a given target video, enabling it to zero-shot generalize to novel domains. Our simple pipeline outperforms almost all prior state-of-the-art for dynamic novel-view synthesis. We show outputs from CogNVS from two unseen videos; a generated video above, and a real-world video below.

test-time optimized counterparts. From a pragmatic perspective, such models need to be trained on mega-scale multi-view training data, which is difficult to obtain for dynamic scenes.

Our method addresses the above challenges by decomposing the problem of dynamic view-synthesis into three distinct stages. First, we lean on the success of non-rigid structure from motion [178, 192, 363] approaches that produce reconstructions of visible scene regions, sometimes known as "2.5D" reconstructions (since occluded regions are not reconstructed). We point out that such reconstructions can be trivially produced for casual mobile videos captured with depth sensors and egomotion [84]. When such reconstructions are rendered from a target novel view, previously-hidden regions will not not be rendered. To "inpaint" these regions, we train a 2D video-inpainter – CogNVS – by fine-tuning a video diffusion model (CogVideoX [344]) to condition on the partially-observable novel-view pixels. Importantly, we allow CogNVS to *also* update the appearance of previously-visible pixels, allowing our pipeline to model view-dependent (dynamic)

scene effects.

The **key insight** of our work is that CogNVS can be trained on any 2D videos via self-supervision. However, rather than training our inpainter with random 2D masks, we make use of 3D multi-view supervision that better captures 3D scene visiblity, similar to prior art [307]. Specifically, given a 2D training video (ref. Fig. 8.1), we first reconstruct it (with an off-the-shelf method such as MegaSAM) and then render the reconstruction from a random camera trajectory. This rendering is used to identify co-visible pixels from the source video that remain visible in the novel views. This original source video and its co-visible-only masked variant can now form a training pair for 3D-consistent video inpainting. Importantly, because such a training pair does *not* require ground-truth 3D supervision, CogNVS can be trained on diverse in-the-wild 2D videos. We use dynamic scenes from TAO [52], SA-V [241], Youtube-VOS [333], and DAVIS [225]. Equally as important, we use the same paradigm to *test-time finetune* CogNVS on the test video-of-interest. We show that this allows our pipeline to "zero-shot" generalize to test videos that were never seen during training. We argue that our test-time finetuning of 2D diffusion models can be seen as the "best-of-both-worlds", by leveraging large-scale training data (for data-driven robustness) and test-time optimization (for accuracy).

In summary, our contributions are as follows: (1) We decompose dynamic view synthesis into three stages of reconstruction, inpainting and test-time finetuning, (2) we use a large corpus of *only 2D videos* for training CogNVS, and (3) we do extensive *zero-shot* benchmarking on three evaluation datasets against state-of-the-art methods and show improvements on dynamic view synthesis.

8.2 Related Work

Novel-view synthesis has seen recent advancements with the rise of implicit scene representations like NeRFs [206] and Gaussian primitives [46, 144, 145]. We have seen widespread efforts in scaling these representations to model larger scenes [280, 288, 326], making them faster to fit [32, 40, 57, 69, 89, 144, 167, 209], anti-aliased [12, 13, 14, 118, 180], and extend to representing dynamic scenes [86, 167, 227, 273]. The most popular paradigms have been the adoption of dynamic NeRFs [206, 218, 227] and deformable Gaussian primitives [144, 195, 322, 343] for modeling scene dynamics, apart from using voxel grids [151, 152] or learnable tokenzation [365]. Most approaches need multi-view posed videos as input, and only recently monocular view synthesis has gained traction [79, 168, 175, 298]. However, each of the aforementioned approaches have to

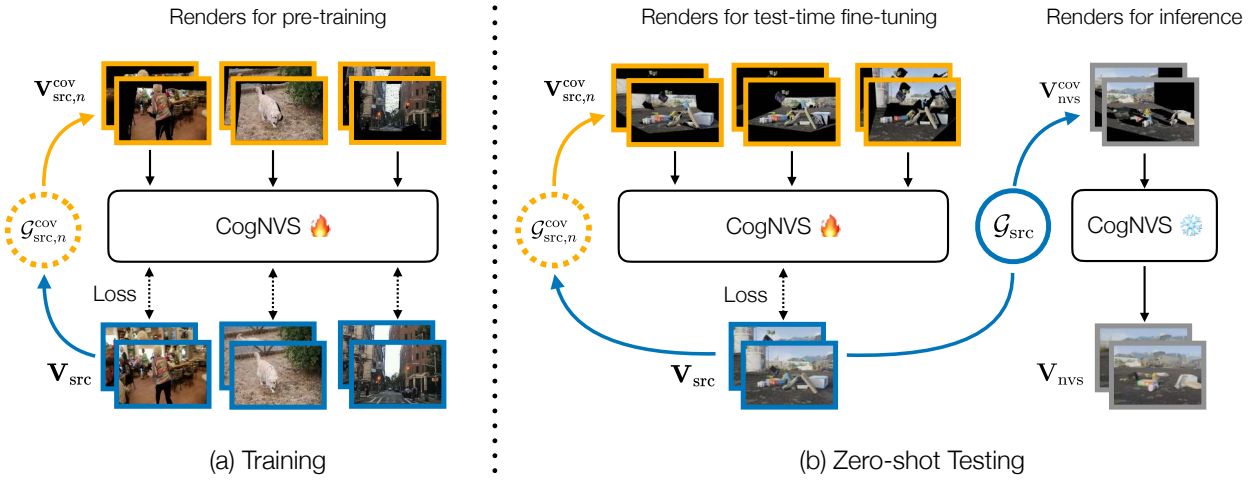


Figure 8.2. **CogNVS overview.** During training (**left**), given a 2D source video (in blue) of a dynamic scene, we first reconstruct the scene using off-the-shelf monocular reconstruction algorithms like MegaSAM [178] to obtain the 3D scene geometry, $\mathcal{G}_{\text{src}}$ and camera odometry, $\mathbf{c}_{\text{src}}$. We then sample a set of arbitrary camera trajectories $\{\mathbf{c}_1, \cdots, \mathbf{c}_N\}$ to simulate plausible occluded geometries, $\{\mathcal{G}^{\text{cov}}_{\text{src},1}, \cdots, \mathcal{G}^{\text{cov}}_{\text{src},N}\}$ which when rendered from original camera trajectory, $\mathbf{c}_{\text{src}}$ produces a mask of source pixels that are co-visible in the sampled trajectory (in orange). The source video and its masked variant produce a self-supervised training pair for learning CogNVS, our video inpainting diffusion model (visualized in Fig. 8.3). At inference (**right**), we finetune CogNVS on the given input sequence by similarly constructing self-supervised training pairs. The final novel-view is then generated using the finetuned CogNVS in a feed-forward manner.

be test-time optimized separately for every new test video, are slow to optimize and yet fail to recover highly-detailed dynamic scene content [168]. Moreover, there is no focus on predicting the unobservable scene content, which is exacerbated by benchmarking metrics that only evaluate co-visible pixels [84] in training and inference views and therefore encourage benchmarking on novel views that are not too far apart from the training views. Our approach instead reformulates dynamic view synthesis as an inpainting task, which specifically focuses on generating parts of the scene that were occluded from the training views, thereby facilitating extreme novel view synthesis for dynamic scenes. Our large-scale pretraining for feed-forward novel-view inpainting enables data-driven robustness.

Data-driven novel-view synthesis approaches have emerged [189, 247, 292, 324, 340, 354, 357] which train for view synthesis in a feed-forward manner with large-scale data. One class of methods is based on transformer architectures, more often than not trained with multi-view supervision and

rendering in the loop [111, 131, 245, 310, 329, 364]. Another class of methods reformulate novel-view synthesis as a conditional generation task and use diffusion-based generative architectures [189, 264] for the same. Initially, the focus was on developing data-driven pipelines for static novel-view synthesis [37, 186, 187, 189, 306, 310, 357, 364], or exploiting data-driven priors [43, 149, 226, 233, 276, 281, 304, 350], using multi-view posed image inputs. However, the focus is now shifting to dynamic view synthesis of casually captured videos in an unconstrained setting [130, 172, 247, 292, 340, 354]. These approaches allow a greater level of hallucination of unseen scene components which instills the capability of view synthesis for camera poses that are far apart from the training views. We fit into this setting. While the data-driven learning provides faster inference times and a broad generalization, it compromises on 3D geometric accuracy and physical-plausibility which introduces unrealistic artifacts in the synthesized outputs (*e.g.*, objects suddenly exist or cease to exist) [354]. In this work, we highlight that test-time finetuning is crucial to preserving the 3D geometry of the scene and reducing implausible artifacts.

Test-time finetuning is a long-standing paradigm to curb distribution shifts in machine learning algorithms and improve their generalization. It's origin lies in early-age algorithms for optical character recognition [24] and text classification [132], where the algorithm adjusts itself *after* observing the test data. A decade back, it popularly resurfaced for super-resolution [266] where learning to super-resolve an image was achieved by downsampling and super-resolving the test image. Domain generalization approaches for vision [44, 45, 80, 127, 277, 369] soon took inspiration from this breakthrough and recently, chain-of-thought prompting [309] and general LLM reasoning [6] in natural language processing adapted this paradigm. The most recent adoption was seen in 4D reconstruction and tracking [72], and we similarly explore this paradigm further in our work.

8.3 Method

Given a monocular video of a dynamic scene, $\mathbf{V}_{\text{src}} = \{\mathbf{V}^t_{\text{src}}\}_{t=1}^{T}$, we want to generate a novel view of the observed scene, $\mathbf{V}_{\text{nvs}} = \{\mathbf{V}^t_{\text{nvs}}\}_{t=1}^{T}$ from a target camera pose. As discussed (c.f. Fig. 8.2), we achieve this by decomposing the task into three distinct stages – (1) obtain an off-the-shelf reconstruction of the observed scene over time, (2) render the scene from the novel views and inpaint the non-co-visible regions, and (3) curb the train-test distribution shift with test-time finetuning. We now descibe each of the stages in detail.

Dynamic view synthesis as structured inpainting

We use off-the-shelf SLAM frameworks, like MegaSAM [178], to obtain a reconstruction of the given scene. Formally, let the underlying 3D structure of the world as observed by $\mathbf{V}_{\text{src}}$ be represented by, $\mathcal{G}_{\text{src}} = \{\mathbf{X}^t_{\text{src}}\}_{t=1}^T$, where $\mathbf{X}^t_{\text{src}}$ are the evolving 3D primitives (points, Gaussians, etc.) across time, t. Any physical properties of the primitives are omitted from this discussion for simplicity. Let the recovered camera poses from which $\mathbf{V}_{\text{src}}$ was observed be, $\mathbf{c}_{\text{src}} = \{\mathbf{c}^t_{\text{src}}\}_{t=1}^T$, where $\mathbf{c}$ denotes a camera pose and is formulated as, $\mathbf{c} = (\mathbf{R}, \mathbf{t}) \in \text{SE}(3)$ lie group. The source video $\mathbf{V}_{\text{src}}$ can be obtained by using a rendering function $\mathcal{R}$ as,

$$\mathbf{V}_{\text{src}} = \mathcal{R}\big(\mathcal{G}_{\text{src}}, \mathbf{c}_{\text{src}}\big)$$

Learning to inpaint novel views For obtaining $\mathbf{V}_{\text{nvs}}$, we note that a subset of 3D primitives that must be visible from $\mathbf{c}_{\text{src}}$, are already available in the reconstructed scene geometry, $\mathcal{G}_{\text{src}}$. Therefore, a partial observation of the world in the form of *co-visible* pixels [84] from novel views, $\mathbf{c}_{\text{nvs}} = \{c^t_{\text{nvs}}\}_{t=1}^T$, can be rendered as follows,

$$\mathbf{V}^{\text{cov}}_{\text{nvs}} = \mathcal{R}\big(\mathcal{G}_{\text{src}}, \mathbf{c}_{\text{nvs}}\big)$$

At this point, the novel view synthesis is incomplete, and all missing regions have to be generated. To this end, we train a conditional video diffusion model, CogNVS (denoted by ϵ_θ) built on top of a recently proposed transformer-based video diffusion model [344]. CogNVS takes in the partially observed novel view video and generates an inpainted novel-view of the scene. The overall CogNVS pipeline first employs a 3D causal VAE to compress the conditioning $\mathbf{V}^{\text{cov}}_{\text{nvs}}$ and target novel-view $\mathbf{V}_{\text{src}}$ into latent representations $\mathbf{z}_{\text{cond}} = \mathcal{E}(\mathbf{V}^{\text{cov}}_{\text{nvs}})$ and $\mathbf{z}_0 = \mathcal{E}(\mathbf{V}_{\text{src}})$ respectively, enabling efficient training while preserving temporal coherence and photometric fidelity. Here, $\mathcal{E}$ is the VAE encoder. Gaussian noise is then added to the target latent $\mathbf{z}_0$, and the resulting noisy latent is concatenated with the conditional latent $\mathbf{z}_{\text{cond}}$. This joint representation is passed through a self-attention transformer equipped with 3D rotary positional embeddings (3D-RoPE) [275] and adaptive layer normalization, which predicts the added noise. The training objective follows a score matching formulation:

$$\min_\theta \; \mathbb{E}_{\substack{\mathbf{z}_0=\mathcal{E}(\mathbf{V}_{\text{src}}),\, \mathbf{z}_{\text{cond}}=\mathcal{E}(\mathbf{V}^{\text{cov}}_{\text{nvs}}),\\ k\sim\mathcal{U}\{1,\ldots,K\},\, \epsilon\sim\mathcal{N}(0,I)}} \big\| \epsilon_\theta(\mathbf{z}_k, k, \mathbf{z}_{\text{cond}}) \; - \; \epsilon \big\|_2^2$$

Here, $\mathbf{z}_k = \sqrt{\bar{\alpha}_k}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_k}\epsilon$ denotes the noisy latent at a uniformly sampled timestep k, where $\bar{\alpha}_k$ is the cumulative signal preserving factor. While CogVideoX was originally designed as

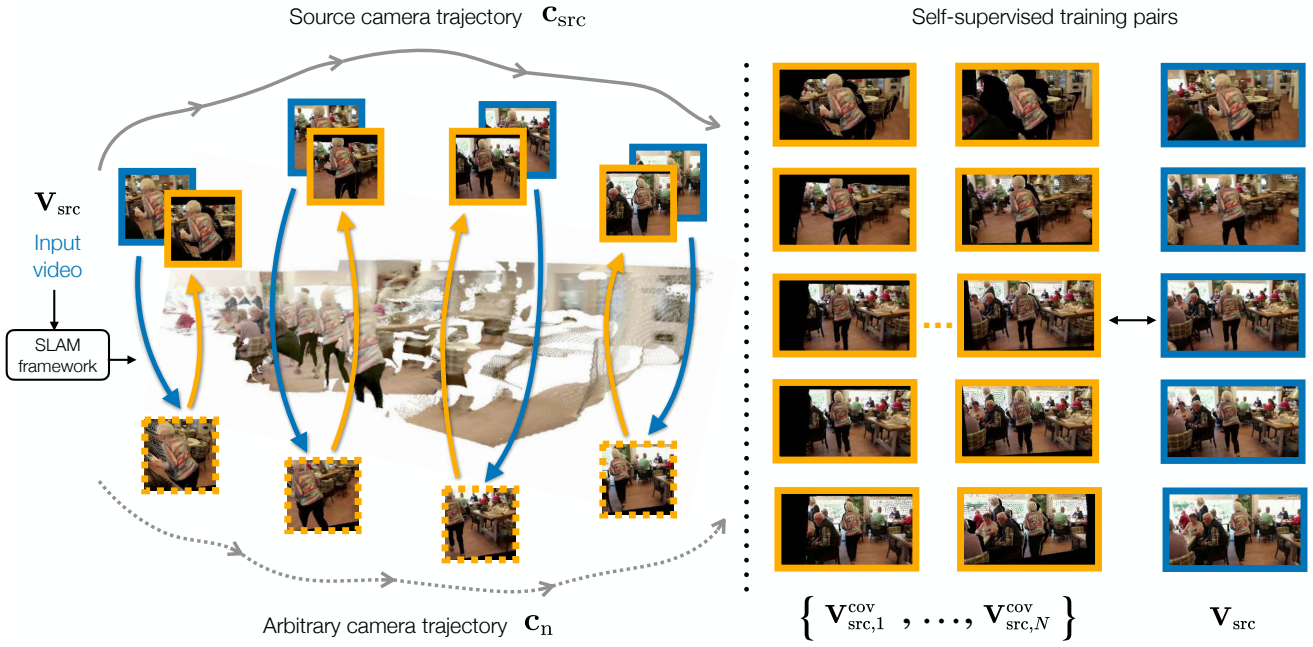


Figure 8.3. **Self-supervised training data generation.** To curate a large training set for video inpainting, we first reconstruct an input source 2D video (in blue) with an off-the-shelf monocular SLAM system. After reconstruction, we randomly sample N pairs of 'start' and 'end' camera poses around a spherical region, $\mathcal{S}$ of the estimated camera pose in the given 2D video. $\mathcal{S}$ is bounded by a predefined deviation in the spherical coordinate axes, similar to a prior work [357]. We sample a SE(3) camera trajectory that interpolates the start and end poses while looking at the center of the scene. We render the reconstruction from this novel trajectory (in dotted-orange), and use the rendering to identify co-visible pixels in the original source view (in orange). The source video and its masked variant are used to produce a self-supervised training pair for training CogNVS, our "3D-aware" video inpainting diffusion model.

an image-to-video diffusion model that zero-pads conditional image patches to match the length of the target video, we adapt it for a video-to-video setting, where the shapes of the conditional and target inputs are inherently aligned and no padding is needed. In practice, CogNVS is trained with datasets of 2D videos which are used to generate self-supervised training pairs. We discuss this below.

Data generation for self-supervised training

We propose to train CogNVS in a self-supervised manner. This allows us to use a large corpus of 2D videos. For each casually captured monocular video $\mathbf{V}_{\text{src}}$, we obtain its 3D reconstruction $\mathcal{G}_{\text{src}}$ and odometry $\mathbf{c}_{\text{src}}$ from off-the-shelf SLAM frameworks [178]. As demonstrated in Fig. 8.3, we

sample N arbitrary camera trajectories in order to create training pairs from 2D videos, described as follows.

We first obtain the "center" of the scene by considering the pixel at the optical center in the first frame of the given video, similar to a prior work [354]. We then construct a bounded region $\mathcal{S}$ in spherical coordinates, around the camera center of c^1_{src}. Within this region, we uniformly sample start and end spherical coordinates of each new camera trajectory, and then again sample two intermediate camera locations between the start and end spherical coordinates to ensure smoothness during interpolation. Camera poses are obtained by converting the spherical coordinates into euclidean space to get translations, and camera rotations are obtained such that the look-at vector always points to the center of the scene. Using the four sampled camera poses, we do bicubic interpolation on the SE(3) manifold. This results in a set of smooth camera trajectories, $\{\mathbf{c}_{\text{n}}\}_{n=1}^{N}$ which are then used to construct the training pairs. With N trajectories, we can obtain "partial" novel-view renderings as,

$$\mathbf{V}^{\text{cov}}_n = \mathcal{R}\big(\mathcal{G}_{\text{src}}, \mathbf{c}_n\big)$$

Between $\mathbf{V}^{\text{cov}}_n$ and $\mathbf{V}_{\text{src}}$, only a subset of primitives from $\mathcal{G}_{\text{src}}$ are *co-visible*. Let this subset be denoted by $\mathcal{G}^{\text{cov}}_{\text{src},n}$ for the n^{th} trajectory. Then, partial renderings of the source video are given by,

$$\mathbf{V}^{\text{cov}}_{\text{src},n} = \mathcal{R}\big(\mathcal{G}^{\text{cov}}_{\text{src},n}, \mathbf{c}_{\text{src}}\big) \qquad \text{s.t.} \qquad \mathcal{D} = \{(\mathbf{V}^{\text{cov}}_{\text{src},n}, \mathbf{V}_{\text{src}})\} \forall n \in [1, N]$$

is the set of training pairs created by one monocular video. We repeat this for all 2D videos considered.

Test-time finetuning for target domain adaptation

At test time, to reduce domain gap arising due to different scene properties (lighting, appearance, motion) we use the source test video $\mathbf{V}_{\text{src}}$ to adjust the priors of CogNVS and create self-supervised finetuning pairs, $\mathcal{D}$ as described above. We therefore adapt the model weights θ on-the-fly with M gradient steps with η step size as follows,

$$\theta \;\leftarrow\; \theta \;-\; \eta\, \nabla_\theta \big\| \epsilon_\theta(\mathbf{z}_k,\, k,\, \mathbf{z}^n_{\text{cond}}) - \epsilon \big\|^2_2,$$

where $\mathbf{z}^n_{\text{cond}}$ is the latent of the n^{th} self-supervised training pair input. At the end of finetuning, we obtain the desired novel view $\mathbf{V}_{\text{nvs}}$ from CogNVS by using the partially observed novel-view, $\mathcal{R}(\mathcal{G}_{\text{src}}, \mathbf{c}_{\text{nvs}})$ as the input conditioning, and running a reverse diffusion process.

8.4 Experiments

Experimental setup

Datasets We train CogNVS on four in-the-wild video datasets, SA-V [241], TAO [52], Youtube-VOS [333], and DAVIS [220]. We sample 3000, 3000, 4000 and 100 videos respectively from each of the datasets, giving us a total training video pool of $\approx$ 10,000 videos. For pretraining, we randomly select a new subsequence of 49-frames in every epoch and construct its training pairs. For benchmarking, we follow prior work [168, 292] and use a combination of Kubric-4D, ParallelDomain-4D [292] and Dycheck [84]. These have a held-out test set of 20, 20 and 5 videos each. Note that our evaluation on Kubric-4D, ParallelDomain-4D and Dycheck is zero-shot as the datasets are not seen during training. Since the Kubric-4D and ParallelDomain-4D are synthetic, we use their groundtruth point clouds and odometry for a fair comparison to baselines. For Dycheck, we use MegaSAM for reconstruction and align the estimated point cloud with the groundtruth to solve for scale ambiguity.

Baselines For Kubric-4D, we consider GCD [292] and Gen3C [247], alongside a concurrent work, TrajectoryCrafter [354]. For ParallelDomain-4D, we consider the same baselines except Gen3C, which only evaluates on Kubric-4D, as there is no open-source implementation available yet. For Dycheck, we consider recent work like Shape-of-Motion [298], MoSca [168], CAT4D [324]. Note that we do not benchmark test-time optimization approaches on Kubric-4D and ParallelDomain-4D, because their performance degrades catastrophically on novel views that are far apart from training views. For more quantitative analysis of CAT4D, see appendix.

Metrics For pixel-wise photometric evaluation, we adopt the widely used PSNR, SSIM, and LPIPS family of metrics for evaluating reconstruction quality via novel-view synthesis. We additionally benchmark the generation quality with FID and KID. This is in line with the benchmarking proposed in several diffusion-based view synthesis works [189, 247, 287, 292].

During pretraining, we load the official CogVideoX-5B-I2V checkpoint and fully finetune all 42 transformer blocks. We use the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and $\beta_3 = 0.98$, a learning rate of $2 \times 10e - 5$, and a batch size of 8 for 12,000 steps. To fit within 48GB VRAM, we employ DeepSpeed ZeRO-2 [234] to partition model states across 8 A6000 Ada GPUs in a distributed setting. Pretraining completes in approximately 3 days.

During test-time finetuning, we maintain the same optimizer and learning rate but reduce

the number of steps to 200 for shorter sequences (e.g., Kubric-4D) and 400 for longer ones (e.g., DyCheck). For all experiments, we use an input resolution of $\mathbb{R}^{49\times480\times720}$, set the classifier-free guidance scale to 6, and run 50 inference steps. A single novel-view sequence generates in $\sim$5 mins on an A6000 Ada.

Training pair details To generate self-supervised training pairs, we randomly perturb the source camera trajectory to create diverse camera paths. In the spherical coordinate system, we sample random elevations from [-15°, 15°], azimuths from [-30°, 30°], and radius deviations from [-0.15, 0.15], followed by bicubic interpolation. This procedure enables flexible generation of training pairs across arbitrary camera trajectories. For pretraining, we use $N = 2$ camera views per training videos. For test-time finetuning, we set $N = 5$ for DyCheck and $N = 9$ for both Kubric-4D and ParallelDomain-4D, due to their wider novel-view gaps. When a video sequence exceeds CogVideoX's default input length of 49 frames, we randomly sample a 49-frame subsequence in each epoch. On DyCheck, we additionally apply a noise injection strategy to simulate real-world degradation on training pairs, as analyzed in Section B.

Evaluation protocol For Kubric-4D and ParallelDomain-4D, we follow the official GCD evaluation protocol. On DyCheck, we consistently report evaluation metrics at an image resolution of 360 (width) $\times$ 480 (height). We render dynamic Gaussian representations from Mosca and Shape-of-Motion with a black background for fair comparison. Both methods optimize camera poses using the ground-truth novel views to improve photometric metrics; we retain this step to stay consistent with their original implementation. CAT4D, although diffusion-based, fits a 4D-GS representation (with minor extensions) after synthesizing multi-view videos. When evaluating CogNVS on MegaSAM renders, we append a static background extracted from the full input video to better capture long-term context. The effectiveness of background stacking is validated in Section B. Also note that Shape-of-Motion and MoSca optimize for evaluation camera poses during evaluation using ground-truth novel view videos. Whether CAT4D adopts this step is unknown. We do not do this camera pose optimization at test-time. Since the DyCheck evaluation sequences are more than 49-frames in length, we isolate the static scene regions and stack them in 3D across time. This accumulated background is then rendered onto each frame which helps, to a large extent, "pre-inpaint" the static background regions using fused information from multiple 49-frame length sequences.

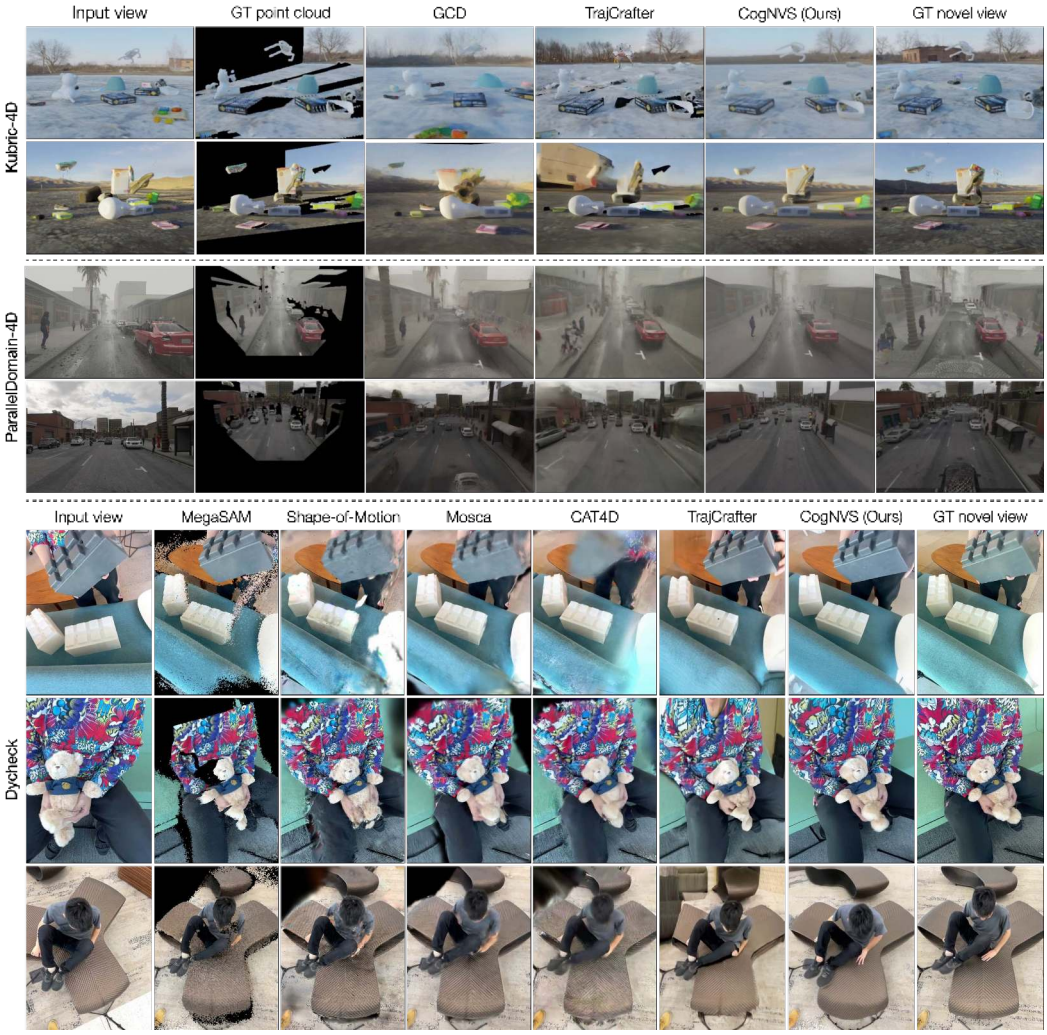


Figure 8.4. We show a qualitative comparison with state-of-the-art approaches for dynamic novel-view synthesis on Kubric-4D (**top**), ParallelDomain-4D (**middle**) and DyCheck (**bottom**). Note how reconstruction alone, either by groundtruth depth, MegaSAM [178], Shape of Motion [298], or MoSca [168] cannot synthesize a complete novel view. Optimization based approaches like Shape of Motion, and MoSca, blur the dynamic regions when fitting 4D representations. CAT4D [324], whose visuals are taken from its project page due to unavailable code, struggles to generalize. TrajectoryCrafter [354] over-hallucinates the occluded regions and does not preserve geometry. GCD [292] performs well because it was trained on Kubric-4D and ParallelDomain-4D. Our method can instead produce photorealistic and 3D-consistent novel-views for the given scenes in a *zero-shot* manner with test-time finetuning, even starting from point cloud renders that are incomplete and noisy (*e.g.*, from MegaSAM for DyCheck). It is consistently able to synthesize sharp dynamic objects, which the other baselines struggle with. Please see the video in the appendix.

Table 8.1. Comparison to state-of-the-art for dynamic view synthesis on Kubric-4D and ParallelDomain-4D. We find that our method, that operates zero-shot unlike Gen3C and GCD, achieves state-of-the-art performance across a majority of metrics. † Note that Gen3C only evaluates on Kubric-4D and there is no open-source code that would allow us to benchmark it on ParallelDomain-4D.

Method	Kubric-4D					ParallelDomain-4D				
	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	KID ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	KID ↓
GT	15.12	0.671	0.328	175.01	0.063	18.79	0.499	0.409	197.99	0.129
GCD [292]	18.59	0.555	0.383	121.57	0.020	21.77	0.665	0.400	90.58	0.022
Gen3C† [247]	19.41	0.630	0.290	98.58	n/a	n/a	n/a	n/a	n/a	n/a
TrajCrafter [354]	20.93	0.730	0.257	130.20	0.024	21.46	0.719	0.342	95.38	0.026
CogNVS	22.63	0.760	0.232	102.47	0.008	24.34	0.797	0.302	102.43	0.033

Table 8.2. Comparison to state-of-the-art for dynamic novel-view synthesis on Dycheck. First, we note that our method can be run on top of any reconstruction approach and the better the reconstruction (*e.g.*, replacing MegaSAM with MoSca), the better the view synthesis. Second, we see that our method can achieve state-of-the-art FID / KID scores because test-time optimization approaches [168, 178, 298] result in blurry dynamic regions and cannot hallucinate new scene content, and completely feed-forward approaches [354] cannot return precise geometry. Our method instead gets the "best of both worlds".

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	KID ↓
MegaSAM [178]	12.16	0.299	0.698	239.57	0.148
Shape-of-Motion [298]	15.30	0.476	0.494	164.29	0.073
MoSca [168]	16.22	0.472	0.586	148.18	0.063
TrajCrafter [354]	12.74	0.337	0.749	140.35	0.059
CogNVS (MegaSAM)	15.19	0.382	0.622	94.48	0.030
CogNVS (MoSca)	16.94	0.449	0.598	92.83	0.031

Comparison to state-of-the-art

Kubric-4D and ParallelDomain-4D We first do zero-shot benchmarking of CogNVS on two synthetic datasets that come with high-fidelity dense depth and accurate camera odometry annotations. For a fair comparison to all baselines, we use the depth and poses to backproject the given scene into a canonical coordinate frame. Given this scene, we generate self-supervised pairs for test-time finetuning. Upon inference (see Tab. 8.1), we find that CogNVS beats prior work on photometric evaluation with PSNR, SSIM, LPIPS, even when baselines are not evaluated zero-shot (GCD is trained on Kubric-4D and ParallelDomain-4D and Gen3C is trained on Kubric-4D). In Fig. 8.4, we show the plausible and realistic novel-views predicted by our method on both datasets as compared to the baselines. This is quantitatively demonstrated by better FID and KID scores. A concurrent work, TrajectoryCrafter [328] performs competitively. We also evaluate

Table 8.3. We ablate our design choices of large-scale pretraining and test-time finetuning on three randomly chosen sequences from Kubric-4D test set. We find that no pretraining is detrimental to the performance of CogNVS, so much so that the PSNR drops by 5 points, thereby devoiding CogNVS of data-driven robustness. Test-time finetuning is also essential as without the adaptation of CogNVS to the test video, the performance in terms of PSNR drops by ~ 3 points.

Pretrain	Finetune	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	KID ↓
✗	✓	18.62	0.691	0.318	201.29	0.051
✓	✗	20.06	0.662	0.284	185.48	0.038
✓	✓	**23.29**	**0.779**	**0.240**	**158.98**	**0.036**

the rendered visible scene structure from groundtruth depth, for establishing a lower bound on dynamic-view synthesis.

DyCheck We evaluate the performance of our method on a real-world dataset of casually captured iPhone videos in Tab. 8.7. First, note that since CogNVS can be applied on top of reconstructions from any method, we show two variants. Better initial reconstruction (in this case, with MoSca rather than MegaSAM) allows for better dynamic view synthesis. Second, of all approaches, our method produces the most visually plausible novel views, as captured by drastically better FID and KID. Third, note how TrajectoryCrafter [354], also based on video diffusion which was the second-best method on Kubric-4D, is unable to handle the distribution shift in Dycheck (shallow field-of-view, close-up videos of moving objects) and fails to generalize. Whereas our method benefits from test-time finetuning and is able to adjust to any new data-distribution at test-time. Other test-time optimization approaches (Shape-of-Motion, MoSca) do better as long as evaluation views are close to training views, because there is only one distribution they need to fit to.

Ablation studies

Effect of test-time finetuning We study the effectiveness of the test-time finetuning stage of our method. Row 2 vs. 3 in Tab. 8.3 show that proposed self-supervised finetuning is crucial for adaptation of CogNVS to a target video's distribution at test-time. Once the self-supervised test-time finetuning stage is completed, our method yields outputs with high fidelity, showcasing improved precision, and more contextually and geometrically consistent 3D appearances, as shown in Fig. 8.5.

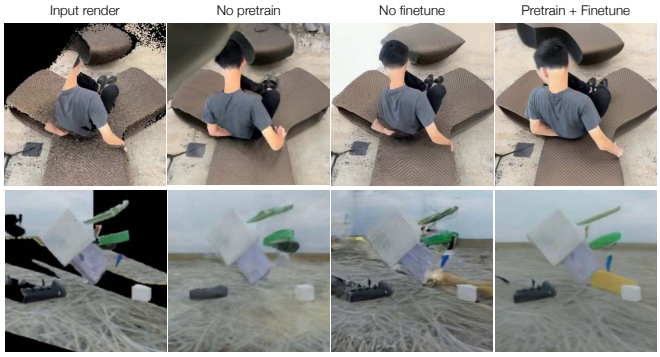


Figure 8.5. We qualitatively analyze the effect of pretraining and test-time finetuning. We note that without the data-driven robustness and generalization of pretraining (**second column**), CogNVS cannot hallucinate missing regions properly (*e.g.*, inpainted region in first row is still black in top left corner). Finally, without test-time finetuning (**third column**), 3D consistency and adherence to scene lighting and appearance properties cannot be ensured (*e.g.*, overall darker scene in second row, and output off by a few pixels at the bottom and right side of the image in first row, thereby inhibiting geometric consistency).

Effect of large-scale pretraining We also study the usefulness of the large-scale pretraining stage with 2D videos from 4 training datasets. In this case, test-time finetuning alone with a self-supervised objective, cannot pull CogNVS out of the local minima it reaches without a good initialization. This is a common failure mode of many test-time optimization approaches that overfit to the training views but default to rendering artifacts such as blurry dynamic regions [168]. We show in Tab. 8.3 (Row 1 vs. 3) and Fig. 8.5 that pretraining is essential for data-driven robustness.

Effect of reconstruction quality Although we touch upon how the initial reconstruction affects the quality of dynamic view synthesis, we describe in detail here. We create a pertubed version of the Kubric-4D dataset, by obtaining reconstruction and odometry from MegaSAM and aligning the reconstruction to groundtruth to solve for scale ambiguity. Quantitative results show a ~ 3 points drop on PSNR and a consistently worse performance on all metrics with sub-optimal reconstructions and cameras. This also addresses the gap in the photometric performance (with PSNR, SSIM, LPIPS) of MegaSAM-based CogNVS on DyCheck. For the quantitative and qualitative analysis of this ablation, please see the appendix.

Table 8.4. Effect of reconstruction quality on Kubric-4D. We quantitatively evaluate CogNVS's performance with the use of two different reconstructions for Kubric-4D. Groundtruth depth gives an upperbound on view synthesis performance by CogNVS. Our first observation, perhaps unsurprisingly, is that the quality of MegaSAM reconstruction is subpar to that of the groundtruth. This difference is quality is also translated to the novel-view synthesis task with CogNVS, where CogNVS used with groundtruth depth does 3 and 45 points better at PSNR and FID respectively as compared to CogNVS used on top of MegaSAM.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	KID ↓
MegaSAM	12.73	0.299	0.644	280.62	0.164
GT	15.12	0.671	0.328	175.01	0.063
CogNVS (MegaSAM)	19.62	0.621	0.313	147.83	0.033
CogNVS (GT)	**22.63**	**0.760**	**0.232**	**102.47**	**0.008**

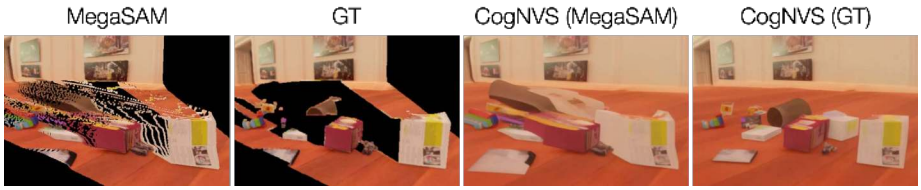


Figure 8.6. We show the effect of using different qualities of reconstruction. Note that the groundtruth depth of the scene is perfect because it is derived synthetically. This re-rendered depth results in more realistic object placements in the scene as compared to the predictions using the depth from MegaSAM. This is because the MegaSAM depth is noisy at the object edges and therefore results in smeared objects in the novel view predictions.

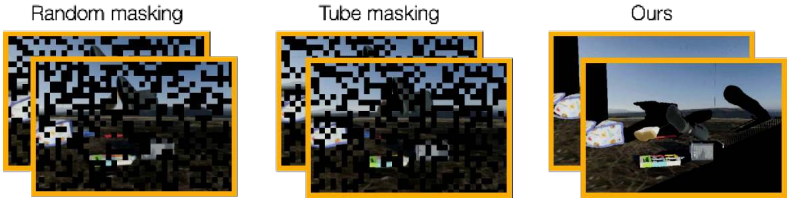


Figure 8.7. We illustrate the different masking strategies considered, as proposed by a prior work [284]. For random masking (**left**), the masked out patches are different in each frame of the input video. For tube masking (**center**), a random set of patches is masked but this set is constant across multiple frames of the video. For our structured masking (**right**), we derive the mask by rendering visible scene reconstruction from the novel views.

Effect of reconstruction quality In Tab. 8.4, we show the effect of using different sources of reconstructions on the entire Kubric-4D evaluation set. Specifically, we compare the structure and odometry from MegaSAM [178] and the synthetic depth groundtruth from Kubric-4D [292]. We find that both quantitatively and qualitatively (c.f. Fig. 8.6), our pipeline benefits more

Table 8.5. Effect of masking strategy on Kubric-4D. We study the effect of building CogNVS as an inpainting model using other masking strategies, specifically, random and tube masking [284]. We find that random masking is the least optimal as it does not mimic the test-time scenario, tube masking does better, but our structured masking strategy is the best for the proposed structured inpainting task.

Mask	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	KID ↓
Random	20.62	0.755	0.310	187.79	0.059
Tube	21.75	0.778	**0.236**	173.55	0.041
Ours	**23.29**	**0.779**	0.240	**158.98**	**0.036**

from better reconstructions. This is because the quality of reconstruction directly affects the input to CogNVS, and if the input point cloud is noisy (*e.g.*, smearing at the object borders), the prediction of the novel view also becomes inaccurate.

Ablation on masking strategy Since CogNVS is an inpainting model, we ablate different masking strategies to train CogNVS on three sequences from Kubric-4D, instead of the proposed structured masking. In Tab. 8.5 and Fig. 8.7, we use random and tube masking from VideoMAE [284] and apply them with a 50% masking ratio on the input video sequences divided into 16 × 16 patches. We find that random masking is the least optimal as it does not resemble the structured inpainting task at test-time. Tube masking is more amenable to the test-time inpainting pattern, which reflects as better photometric and generative metrics. Of all, our structured masking obtained by rendering scene reconstructions into the novel views performs the best.

Ablation on test-time finetuning epochs Following the same data setup as above, we assess how the length of test-time finetuning affects the final prediction from CogNVS. In Fig. 8.8 (left), as expected we observe that the performance improvement in the first few epochs is high (both in terms of PSNR and FID going from 0 to 50 epochs) and saturates as the number of epochs are increased further (up to 200).

Top-K evaluation Following the same data setup as above, we compute probabilistic PSNR and FID metrics for CogNVS's performance on Kubric-4D in the form of Top-k metrics (where the best of k number is reported) in Fig. 8.8 (right). As we sample multiple modes from CogNVS's learnt distribution, the Top-k metrics for PSNR and FID become better and start to saturate near k=8.

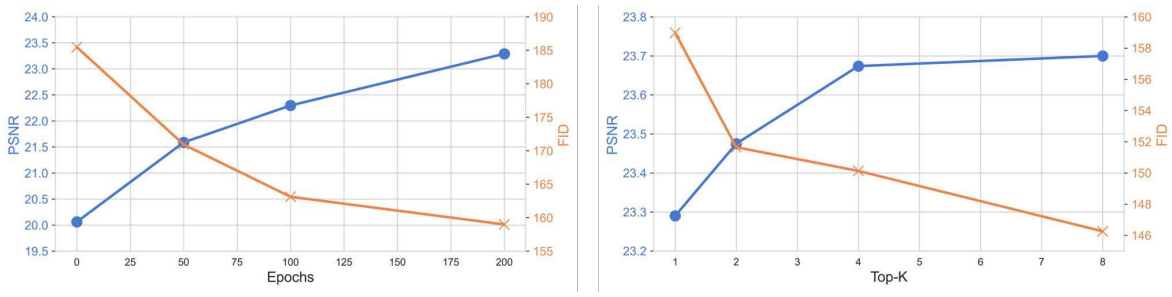


Figure 8.8. We conduct ablations on the number of epochs used for test-time finetuning (**left**) and number of samples drawn from CogNVS for a probabilistic evaluation (right). Both experiments suggest similar trends; performance improves with an increase in the number of finetuning epochs and increase in the number of samples drawn from our diffusion model. Performance saturates once a threshold is reached.

Ablation on background stacking and noise addition We conduct an ablation on the MegaSAM reconstructions of DyCheck for the effect of static background stacking described in the previous section. In Tab. 8.6 and Fig. 8.9, we see that stacking the background on DyCheck provides a large in photometric performance. Secondly, we propose to add noise to dynamic object depths during training, especially for out-of-distribution data. This is essential as our creation of self-supervised training pairs only masks out certain pixels from the source video which leaves no room for CogNVS to be able to see real-world noise. To simulate real noise, at say object edges, we estimate the noise between (pseudo) groundtruth depth (coming from iPhone LiDARs or a state-of-the-art depth estimator, say, MoGe) and the predicted depth (coming from a SLAM framework like MegaSAM). This estimated noise for the source pixels, is added to the visible scene reconstruction but in the ray direction of the pixels visible in the arbitrary cameras. This results in noisy visuals that make CogNVS training more robust, especially to out-of-distribution cases. In Tab. 8.6 and Fig. 8.9, we demonstrate the improvements in performance by training CogNVS to inpaint in the presence of distracting noise artifacts.

Evaluation with masked metrics on DyCheck In addition to the metrics reported in the main paper, we also report masked photometric errors as proposed by a prior work [84]. While this metric only evaluates the visible scene content and how any view-dependent changes were handled during novel-view synthesis, it does not encourage the generation of unseen scene regions. On this metric, CogNVS performs competitively as compared to baselines.

Table 8.6. We quantitatively evaluate the effect of static background stacking and noise addition on DyCheck. Note that background stacking helps DyCheck because the video sequences are longer than 49-frames that CogNVS can handle. This gives us 3 points performance boost in PSNR. Adding real-world noise to dynamic objects helps make CogNVS robust to noise and therefore it reduces artifacts like smeared object edges, reflected in a much lower FID metric.

Background stacking	Noise addition	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	KID ↓
✗	✗	11.55	0.304	0.848	197.93	0.204
✓	✗	14.27	0.352	**0.737**	180.67	0.171
✓	✓	**14.30**	**0.354**	0.740	**156.83**	**0.141**

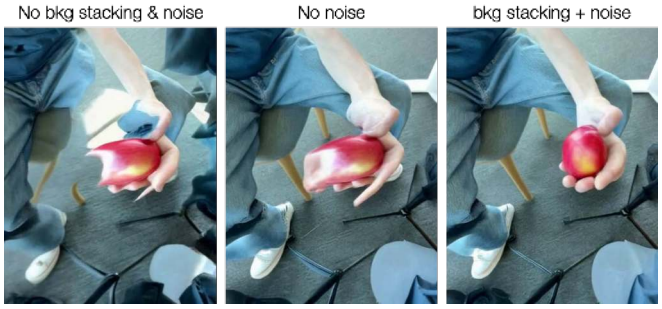


Figure 8.9. We visualize the 'apple' evaluation sequence from DyCheck for analysis of the effect of background stacking over time and noise addition strategy during training to simulate realistic in-the-wild scenarios. First (column 1 vs. 2), we see that for longer videos, stacking the static background from the entire input video helps accumulate multi-view cues about the static background. Second (column 2 vs. 3), we see that due to the noise addition strategy during training, CogNVS is more robust to real-world noise patterns like smearing across object (in this case, apple) edges.

8.5 Discussion

In this work, we focus on the problem of dynamic novel-view synthesis from monocular videos. Contrary to prior state-of-the-art that approaches this task from two extremes (either test-time optimization for every new video from scratch, or large-scale feed-forward novel view synthesis) – we propose a simple setup that is the "best-of-both-worlds". We reformulate dynamic view synthesis as an inpainting task and lean on the success of reconstruction algorithms like MegaSAM that can estimate the structure and geometry of in-the-wild videos. We first train a video inpainter, CogNVS, on pairs of co-visible novel-view pixels and target novel-views via self-supervision on only 2D videos. At test-time, we propose to finetune CogNVS, again via self-supervision, to adjust to the target video distribution. The proposed setup provides data-driven robustness with the large-scale pretraining of a video inpainting model, and enhances 3D accuracy of the

Table 8.7. We report masked perceptual quality metrics as proposed by prior work [84]. This metric only evaluates the visible regions of the scene and so does not encourage generation of unseen scene components. Note that our method performs competitively as compared to the baselines which only focus on modeling the visible scene content.

Method	mPSNR ↑	mSSIM ↑	mLPIPS ↓
MegaSAM [178]	14.60	0.517	0.609
Shape-of-Motion [298]	16.47	0.639	0.409
MoSca [168]	17.82	0.635	0.507
CAT4D [324]	17.39	0.607	0.341
TrajCrafter [354]	13.60	0.518	0.663
CogNVS (MegaSAM)	15.35	0.549	0.557
CogNVS (MoSca)	17.33	0.607	0.530

predictions with test-time finetuning.

Limitations CogNVS does not currently take advantage of open-source 3D and 4D video datasets and trains on a relatively small set of 2D videos. While the zero-shot evaluation can achieve better photorealistic performance than prior state-of-the-art even with this unprivileged training data, the model and its geometric inpainting capabilities can be enhanced by adding more training data from all three – 2D, 3D and 4D data sources. Additionally, the performance of CogNVS is dependent on the quality of dynamic scene reconstruction obtained from off-the-shelf structure from motion algorithms. When groundtruth structure and odometry is available, such as from ubiquitous depth sensors, CogNVS's performance can be increased. A limitation of the data generation pipeline is that the sampled arbitrary camera trajectories are not able to mimic the diversity of camera trajectories that are encountered in real-life, which is a bottleneck to the performance of CogNVS. A better strategy would be to create a "data-driven" trajectory sampler that samples from a set of real-world trajectories observed in the training set.

We now cover more qualitative visuals from evaluation datasets and in-the-wild videos.

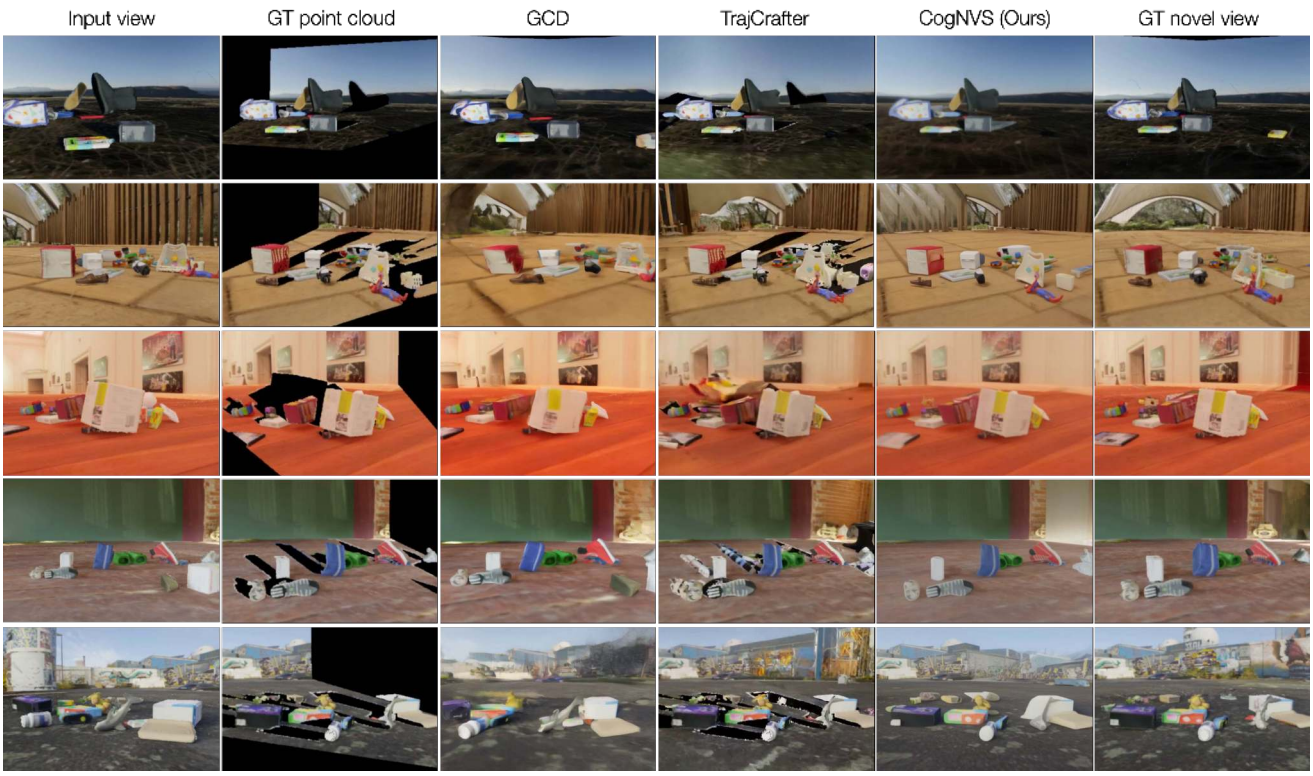


Figure 8.10. We show supplementary qualitative comparison on Kubric-4D. Note that TrajectoryCrafter is able to generate a reasonable background for the unseen scene regions, but is not able to inpaint the shadows / masks created by foreground objects. GCD is trained on Kubric-4D so performs reasonably well but struggles to preserve the precise geometry. CogNVS achieves better performance as compared to baselines and is the closest is geometric consistency to the groundtruth novel view.

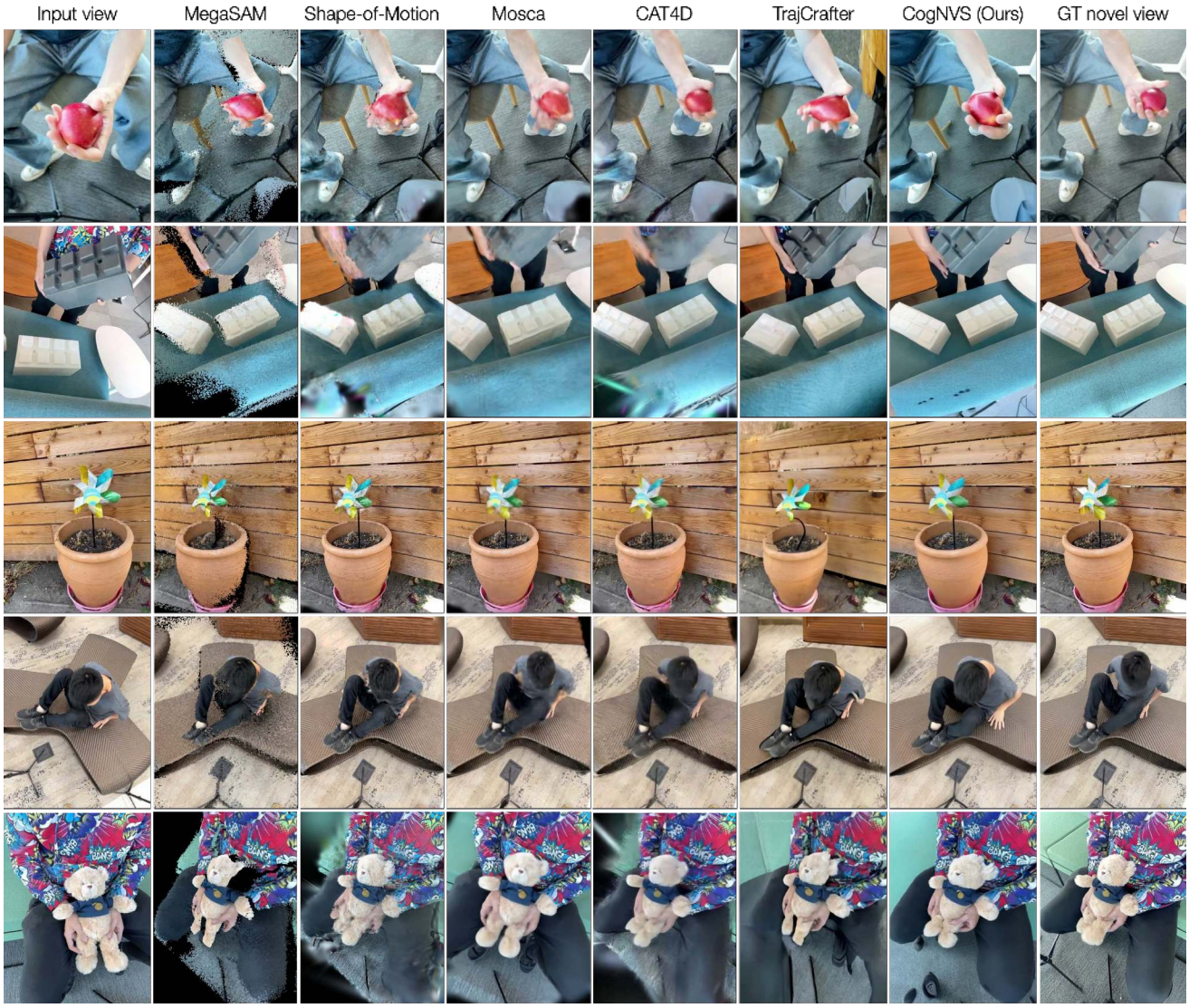


Figure 8.11. We show supplementary qualitative comparison on DyCheck with CogNVS which surpasses the performance of all prior state-of-the-art. Note that baselines either do not hallucinate the unseen regions in the novel-view (Shape-of-Motion, MegaSAM), show blurry dynamic regions (MoSca, CAT4D), or are not able to preserve the underlying geometry of the scene (TrajectoryCrafter).

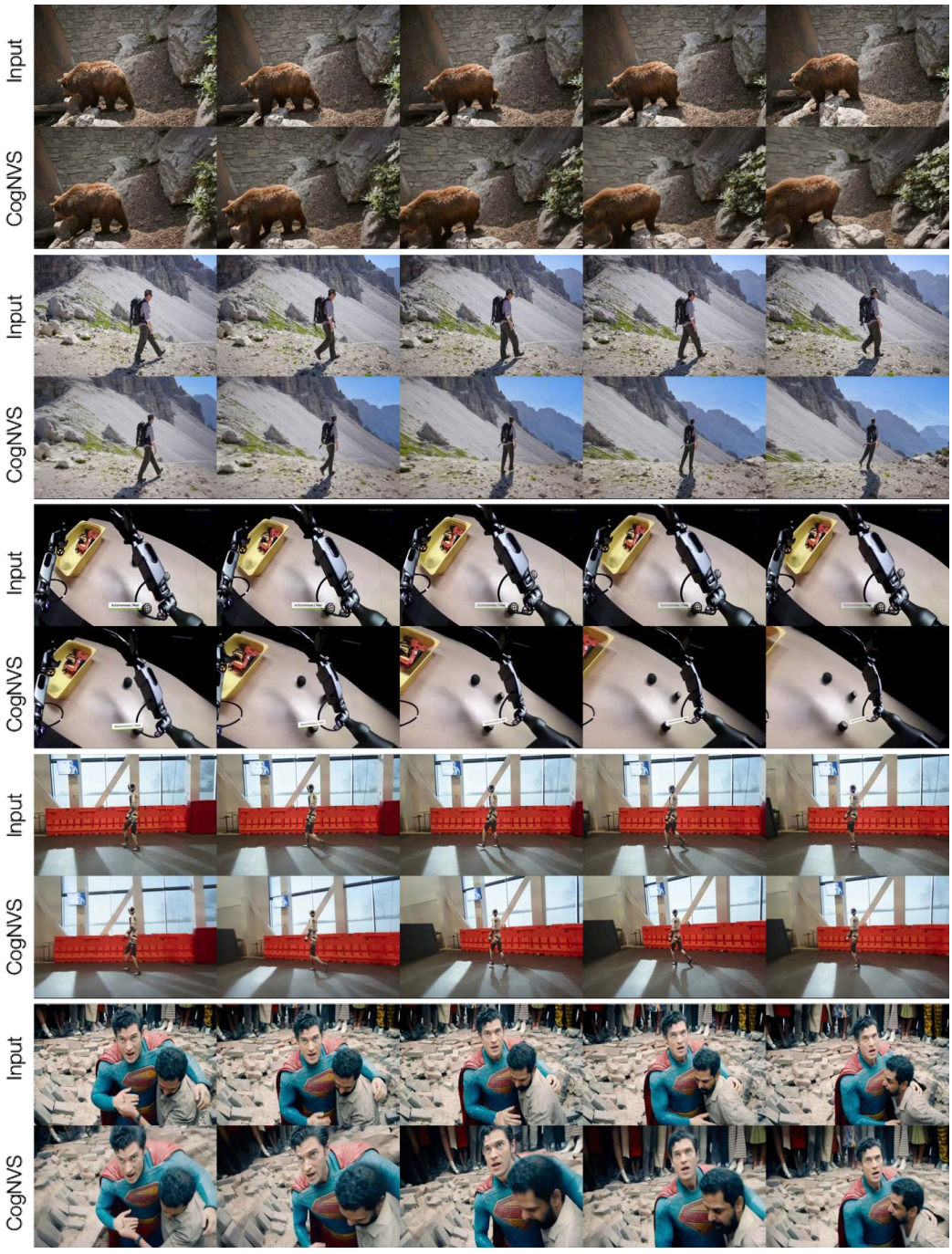


Figure 8.12. Qualitative results on in-the-wild examples. Part 1 of 2.

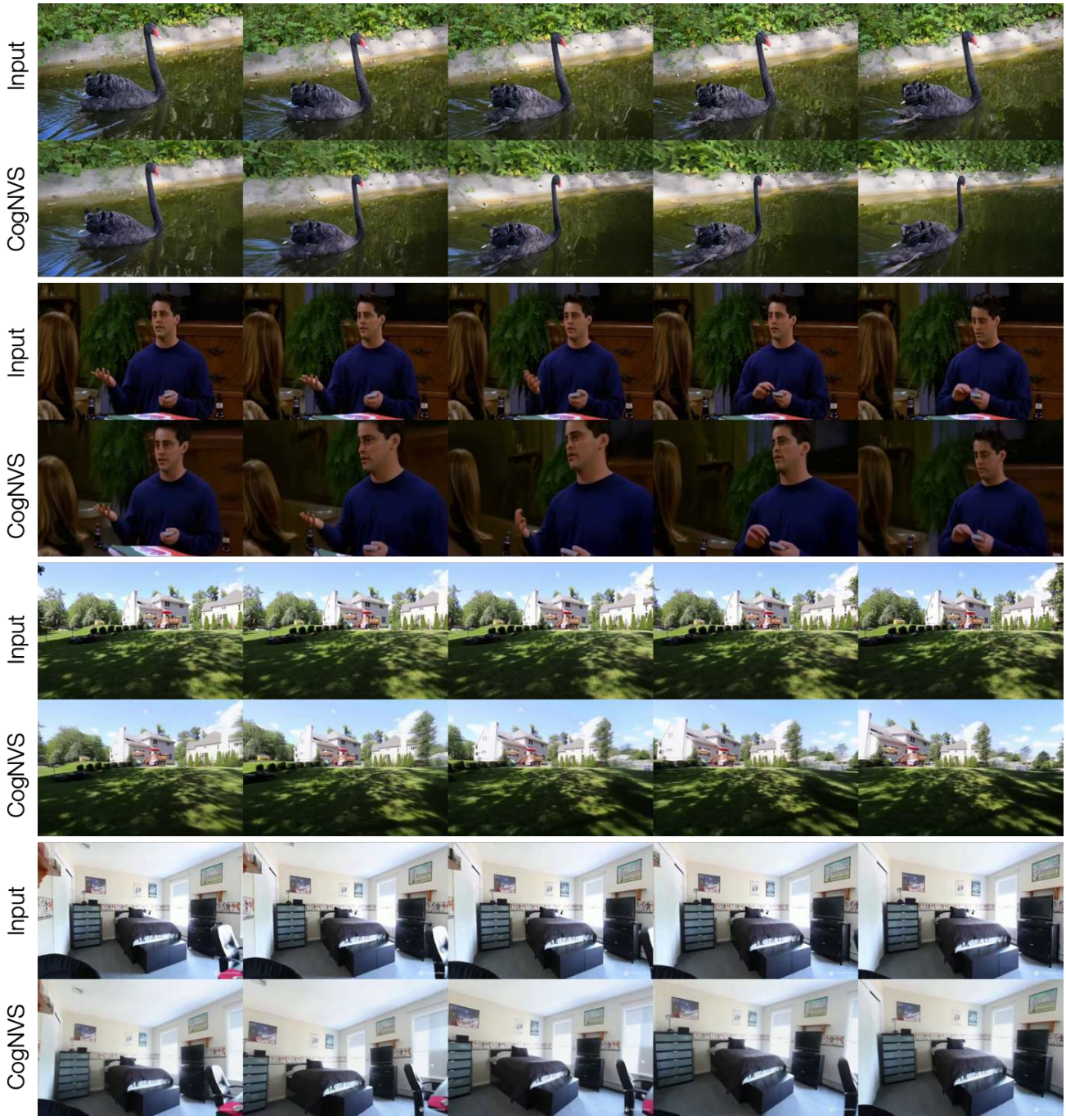


Figure 8.13. Qualitative results on in-the-wild examples (static scenes included in last two rows). Part 2 of 2.

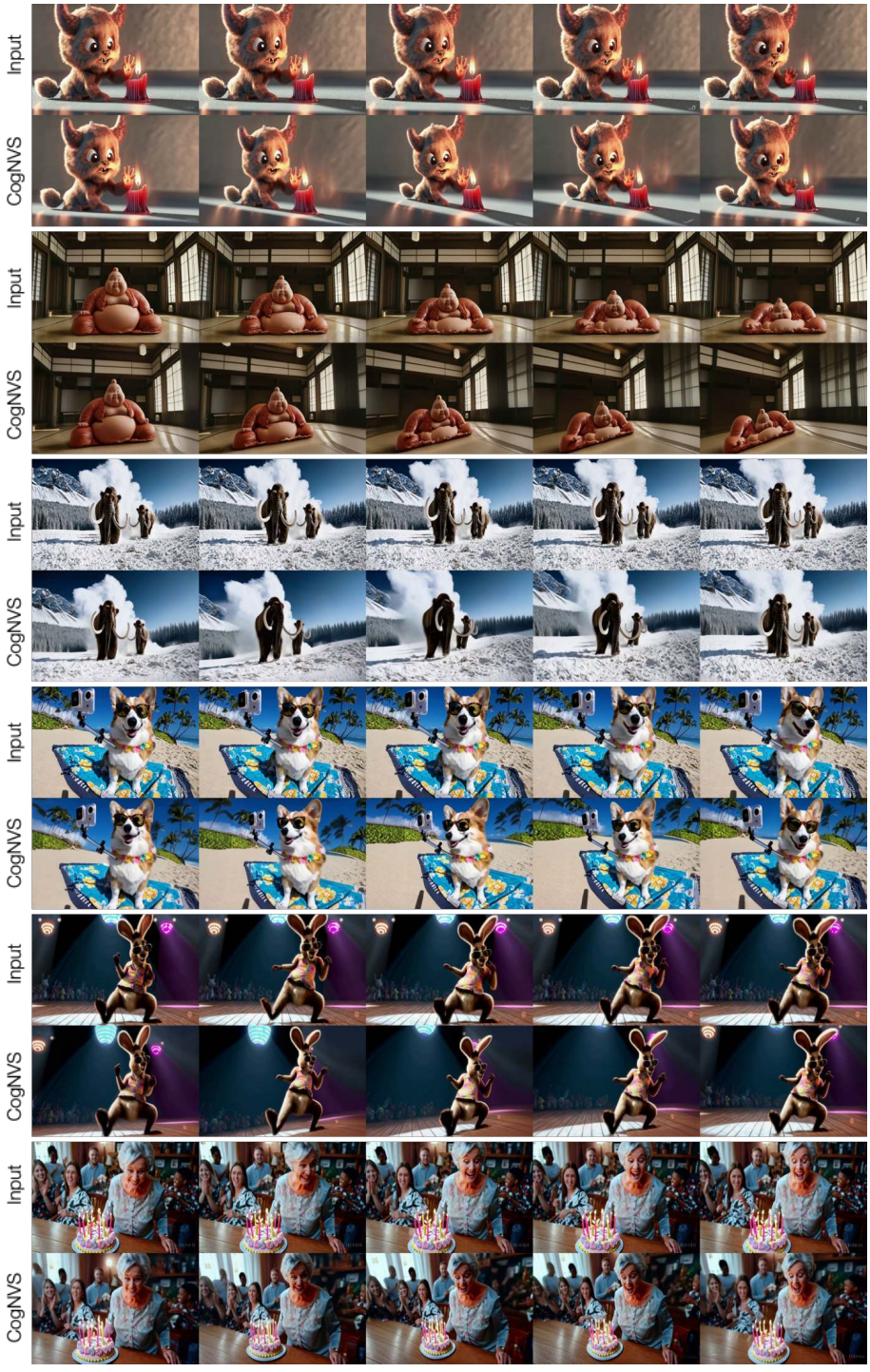


Figure 8.14. Qualitative results on synthetic videos from SORA.

Part IV
Conclusion and Future Work

In this thesis, we focus on building, leveraging, and adapting foundational priors to achieve scalable spatiotemporal (4D) perception. Our overarching goal is to bridge the gap between abundant 2D visual data and the limited availability of 4D supervision, in order to build perception algorithms for understanding the dynamic 3D structure and motion of everyday scenes.

In the first part of the thesis, by constructing foundational priors from scratch via next-timestep prediction on LiDAR sequences, we show that explicit 4D bottlenecks are key to robust dynamic scene forecasting and can directly benefit downstream applications such as motion planning. In the second part, we show that learned depth priors from large reconstruction models can be used to track objects even across occlusions, and reconstruct dynamic scenes from sparse views, thereby significantly surpassing prior state-of-the-art. By the last chapter, we demonstrate that meaningful 4D understanding can emerge with minimal finetuning (with less data and low compute), by carefully reformulating existing perception tasks with self-supervised objectives.

Altogether, we highlight the importance of leveraging 2D priors from large-scale foundation models to circumvent the need for fully supervised approaches for 4D perception tasks. A key takeaway is that to unlock the full potential of these models, downstream objectives should be aligned as closely as possible with their original pretext tasks. Doing so enables effective adaptation with minimal curated data and low computational overhead.

The next step for a 4D perception stack is to be useful for downstream applications, for instance, deployment in autonomous systems. Two important questions to ask in this regard are (1) How can we make better foundation models? (2) Where should we use these foundational priors? We discuss these future directions below.

1. **How can we make better foundation models?** In this thesis, we build upon existing foundation model architectures trained at scale on 2D images and videos. Specifically, we look at depth estimation and video generation models, where the heavylifting is done primarily by pretrained DINO backbones or transformer architectures respectively. This undoubtedly ensures *scalability* which is the first benchmark for building "better" foundation models. More importantly, while being scalable, the next push must be toward building *efficient* foundation models. Efficiency could mean building models that can train faster, or can infer in real-time. One claim is that designing training objectives with soft inductive biases will be key to enabling efficient training and inference. Take for instance, Genie 3 [65], which is speculated to be an autoregressive image diffusion model that conditions on 8 discrete controls (4 rotation, 4 translation), and Wan2.1-First Last Frame Video [294] model that is learning to find correspondences between two input images by generating a

video linking them. For both models, one could argue that the intermediate features are 3D-aware. Better features have been shown to speed up diffusion training [356]. Similarly, 3D-aware features could be used for 3D tasks *outside* the foundational framework during inference, by learning lightweight modules on top of them.

2. **Where should we use these foundational priors?** In this thesis, we demonstrated the use of foundational priors for various tasks such as amodal tracking and segmentation, and novel-view synthesis from sparse or monocular inputs. Focusing on the generative formulation of foundation models, their two key properties are compositionality (generating scenes that never existed before by combining separate concepts) and multimodality (generating multiple plausible scenes for, say, a single text prompt). One claim is that newer tasks must be explored which can exploit these properties. For instance, to build a closed-loop simulator, an effective reformulation is to do novel-view synthesis in a *reactive* manner —where the world reacts to changes in the observer's camera pose. Traditionally, it is difficult to collect training data for this task, as it would require multiple observations of a dynamic world starting from the same initial state—something only feasible in simulation. However, leveraging the multimodal and compositional nature of diffusion models, it may be possible to circumvent this need.

Bibliography

[1] nuScenes LiDARSeg. https://www.nuscenes.org/nuscenes#lidarseg.

[2] Vitaly Ablavsky and Stan Sclaroff. Layered graphical models for tracking partially occluded objects. *TPAMI*, 33(9):1758–1775, 2011.

[3] Ananye Agarwal, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Legged locomotion in challenging terrains using egocentric vision. In *Conference on Robot Learning*, pages 403–415. PMLR, 2023.

[4] Ben Agro, Quinlan Sykora, Sergio Casas, and Raquel Urtasun. Implicit occupancy flow fields for perception and prediction in self-driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1379–1388, 2023.

[5] Hyemin Ahn, Esteve Valls Mascaro, and Dongheui Lee. Can we use diffusion probabilistic models for 3d motion prediction? *arXiv preprint arXiv:2302.14503*, 2023.

[6] Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. The surprising effectiveness of test-time training for few-shot learning. *arXiv preprint arXiv:2411.07279*, 2024.

[7] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *CVPR*, 2016.

[8] John Amanatides and Andrew Woo. A Fast Voxel Traversal Algorithm for Ray Tracing. In *EG 1987-Technical Papers*. Eurographics Association, 1987. doi: 10.2312/egtp.19871000.

[9] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2021.

[10] Ali Athar, Jonathon Luiten, Paul Voigtlaender, Tarasha Khurana, Achal Dave, Bastian Leibe, and Deva Ramanan. Burst: A benchmark for unifying object recognition, segmentation and tracking in video. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1674–1683, 2023.

[11] Renée Baillargeon and Julie DeVos. Object permanence in young infants: Further evidence. *Child development*, 62(6):1227–1246, 1991.

[12] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-

Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Int. Conf. Comput. Vis.*, 2021.

[13] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[14] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Int. Conf. Comput. Vis.*, 2023.

[15] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019.

[16] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1806–1819, 2011.

[17] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *ICCV*, 2019.

[18] 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.

[19] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, 2016. doi: 10.1109/ICIP.2016.7533003.

[20] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

[21] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[22] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[23] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 22563–22575, 2023.

[24] Léon Bottou and Vladimir Vapnik. Local learning algorithms. *Neural computation*, 4(6): 888–900, 1992.

[25] Breakthrough. PySceneDetect: Video scene cut detection tool, 2024. URL https://github.com/Breakthrough/PySceneDetect. Accessed: 2024-11-12.

[26] Ted J Broida, S Chandrashekhar, and Rama Chellappa. Recursive 3-d motion estimation from a monocular image sequence. *IEEE Transactions on Aerospace and Electronic Systems*, 26(4):639–656, 1990.

[27] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.

[28] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

[29] Harry Edwin Burton. The optics of euclid. *Journal of the Optical Society of America*, 35 (5):357–372, 1945.

[30] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

[31] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021.

[32] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.

[33] Joel Carranza, Christian Theobalt, Marcus Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. *ACM Trans. Graph.*, 22:569–577, 07 2003. doi: 10.1145/882262.882309.

[34] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[35] Sergio Casas, Abbas Sadat, and Raquel Urtasun. Mp3: A unified model to map, perceive, predict and plan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14403–14412, 2021.

[36] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019.

[37] Eric R Chan, Koki Nagano, Matthew A Chan, Alexander W Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3d-aware diffusion models. In *Int. Conf. Comput. Vis.*, 2023.

[38] Michael Chan, Dimitri Metaxas, and Sven Dickinson. Physics-based tracking of 3d objects in 2d image sequences. In *Proceedings of 12th International Conference on Pattern Recognition*,

volume 1, pages 432–436. IEEE, 1994.

[39] 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 Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019.

[40] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Int. Conf. Comput. Vis.*, 2021.

[41] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Conference on Robot Learning*, pages 66–75. PMLR, 2020.

[42] Guikun Chen and Wenguan Wang. A survey on 3d gaussian splatting. *arXiv preprint*, 2024.

[43] Kaihua Chen, Deva Ramanan, and Tarasha Khurana. Using diffusion priors for video amodal segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[44] Liang Chen, Yong Zhang, Yibing Song, Jue Wang, and Lingqiao Liu. Ost: Improving generalization of deepfake detection via one-shot test-time training. *Advances in Neural Information Processing Systems*, 35:24597–24610, 2022.

[45] Liang Chen, Yong Zhang, Yibing Song, Ying Shan, and Lingqiao Liu. Improved test-time adaptation for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24172–24182, 2023.

[46] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *Eur. Conf. Comput. Vis.*, 2024.

[47] Xuxin Cheng, Kexin Shi, Ananye Agarwal, and Deepak Pathak. Extreme parkour with legged robots. *arXiv preprint arXiv:2309.14341*, 2023.

[48] Javier Civera, Andrew J Davison, and JM Martinez Montiel. Inverse depth parametrization for monocular slam. *IEEE transactions on robotics*, 24(5):932–945, 2008.

[49] Felipe Codevilla, Matthias Miiller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation*, pages 1–9. IEEE, 2018.

[50] Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9329–9338, 2019.

[51] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers, 2023.

[52] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. Tao: A large-scale benchmark for tracking any object. In *Computer Vision–ECCV 2020:*

16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16, pages 436–454. Springer, 2020.

[53] Aram Davtyan, Sepehr Sameni, and Paolo Favaro. Efficient video prediction via sparsely conditioned flow matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23263–23274, 2023.

[54] Edilson de Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. *SIGGRAPH*, 2008.

[55] 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.

[56] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. *arXiv preprint arXiv:2107.02791*, 2021.

[57] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[58] Michel Denis. *Space and spatial cognition: A multidisciplinary perspective*. Routledge, 2017.

[59] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*, 2017.

[60] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, Pushmeet Kohli, Vladimir Tankovich, and Shahram Izadi. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2016*, 35, July 2016. URL https://www.microsoft.com/en-us/research/publication/fusion4d-real-time-performance-capture-challenging-scenes-2/.

[61] Yuming Du, Robin Kips, Albert Pumarola, Sebastian Starke, Ali Thabet, and Artsiom Sanakoyeu. Avatars grow legs: Generating smooth human motion from sparse tracking inputs with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 481–490, 2023.

[62] Kiana Ehsani, Roozbeh Mottaghi, and Ali Farhadi. Segan: Segmenting and generating the invisible. In *CVPR*, 2018.

[63] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *NeurIPS*, pages 2366–2374, 2014.

[64] Andreas Ess, Konrad Schindler, Bastian Leibe, and Luc Van Gool. Improved multi-person tracking with active occlusion handling. In *ICRA Workshop on People Detection and Tracking*, 2009.

[65] Philip J. Ball et al. Genie 3: A new frontier for world models. 2025.

[66] Georgios D Evangelidis and Emmanouil Z Psarakis. Parametric image alignment using

enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1858–1865, 2008.

[67] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[68] Ke Fan, Jingshi Lei, Xuelin Qian, Miaopeng Yu, Tianjun Xiao, Tong He, Zheng Zhang, and Yanwei Fu. Rethinking amodal video segmentation from learning supervised signals with object-centric representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1272–1281, 2023.

[69] Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds. *arXiv:2403.20309*, 2024.

[70] 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.

[71] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.

[72] Haiwen Feng, Junyi Zhang, Qianqian Wang, Yufei Ye, Pengcheng Yu, Michael J. Black, Trevor Darrell, and Angjoo Kanazawa. St4rtrack: Simultaneous 4d reconstruction and tracking in the world. *arXiv preprint arxiv:2504.13152*, 2025.

[73] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. Tracking by prediction: A deep generative model for mutli-person localisation and tracking. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1122–1132. IEEE, 2018.

[74] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):267–282, 2007.

[75] Patrick Follmann, Rebecca König, Philipp Härtinger, Michael Klostermann, and Tobias Böttger. Learning to see the invisible: End-to-end trainable amodal instance segmentation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1328–1336. IEEE, 2019.

[76] David A Forsyth and Jean Ponce. A modern approach. *Computer vision: a modern approach*, 17:21–48, 2003.

[77] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. *CVPR*, pages 2002–2011,

2018.

[78] Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. In *ECCV*, 2024.

[79] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[80] Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei Efros. Test-time training with masked autoencoders. *Advances in Neural Information Processing Systems*, 35:29374–29385, 2022.

[81] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Int. Conf. Comput. Vis.*, 2021.

[82] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5712–5721, 2021.

[83] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. 2022.

[84] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. In *NeurIPS*, 2022.

[85] Jianxiong Gao, Xuelin Qian, Yikai Wang, Tianjun Xiao, Tong He, Zheng Zhang, and Yanwei Fu. Coarse-to-fine amodal segmentation with shape prior. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1262–1271, 2023.

[86] Quankai Gao, Qiangeng Xu, Zhe Cao, Ben Mildenhall, Wenchao Ma, Le Chen, Danhang Tang, and Ulrich Neumann. Gaussianflow: Splatting gaussian dynamics for 4d content creation. *arXiv preprint arXiv:2403.12365*, 2024.

[87] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. CAT3D: Create Anything in 3D with Multi-View Diffusion Models. *NeurIPS*, 2024.

[88] Shan Gao, Zhenjun Han, Ce Li, Qixiang Ye, and Jianbin Jiao. Real-time multipedestrian tracking in traffic scenes via an rgb-d-based layered graph model. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2814–2825, 2015.

[89] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Int. Conf. Comput. Vis.*, 2021.

[90] 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.

[91] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 32(11):1231–1237, 2013.

[92] Rohit Girdhar et al. Emu video: Factorizing text-to-video generation by explicit image conditioning. 2023.

[93] Helmut Grabner, Jiri Matas, Luc Van Gool, and Philippe Cattin. Tracking the invisible: Learning where the object might be. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1285–1292. IEEE, 2010.

[94] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, Eugene Byrne, Zach Chavis, Joya Chen, Feng Cheng, Fu-Jen Chu, Sean Crane, Avijit Dasgupta, Jing Dong, Maria Escobar, Cristhian Forigua, Abrham Gebreselasie, Sanjay Haresh, Jing Huang, Md Mohaiminul Islam, Suyog Jain, Rawal Khirodkar, Devansh Kukreja, Kevin J Liang, Jia-Wei Liu, Sagnik Majumder, Yongsen Mao, Miguel Martin, Effrosyni Mavroudi, Tushar Nagarajan, Francesco Ragusa, Santhosh Kumar Ramakrishnan, Luigi Seminara, Arjun Somayazulu, Yale Song, Shan Su, Zihui Xue, Edward Zhang, Jinxu Zhang, Angela Castillo, Changan Chen, Xinzhu Fu, Ryosuke Furuta, Cristina Gonzalez, Prince Gupta, Jiabo Hu, Yifei Huang, Yiming Huang, Weslie Khoo, Anush Kumar, Robert Kuo, Sach Lakhavani, Miao Liu, Mi Luo, Zhengyi Luo, Brighid Meredith, Austin Miller, Oluwatumininu Oguntola, Xiaqing Pan, Penny Peng, Shraman Pramanick, Merey Ramazanova, Fiona Ryan, Wei Shan, Kiran Somasundaram, Chenan Song, Audrey Southerland, Masatoshi Tateno, Huiyu Wang, Yuchen Wang, Takuma Yagi, Mingfei Yan, Xitong Yang, Zecheng Yu, Shengxin Cindy Zha, Chen Zhao, Ziwei Zhao, Zhifan Zhu, Jeff Zhuo, Pablo Arbelaez, Gedas Bertasius, David Crandall, Dima Damen, Jakob Engel, Giovanni Maria Farinella, Antonino Furnari, Bernard Ghanem, Judy Hoffman, C. V. Jawahar, Richard Newcombe, Hyun Soo Park, James M. Rehg, Yoichi Sato, Manolis Savva, Jianbo Shi, Mike Zheng Shou, and Michael Wray. Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives, 2023.

[95] Alex Graves. *Long short-term memory*, pages 37–45. 2012.

[96] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[97] 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.

[98] 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.

[99] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. *Advances in Neural Information Processing Systems*, 35:27953–27965, 2022.

[100] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

[101] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[102] Michael Himmelsbach, Felix V Hundelshausen, and H-J Wuensche. Fast segmentation of 3d point clouds for ground vehicles. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 560–565. IEEE, 2010.

[103] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[104] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[105] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Adv. Neural Inform. Process. Syst.*, volume 33, pages 6840–6851, 2020.

[106] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[107] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[108] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[109] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022.

[110] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023.

[111] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. In *Int. Conf. Learn. Represent.*, 2024.

[112] Cheng-Yen Hsieh, Tarasha Khurana, Achal Dave, and Deva Ramanan. Tao-amodal: A benchmark for tracking any object amodally. *arXiv preprint arXiv:2312.12433*, 2023.

[113] Cheng-Yen Hsieh, Tarasha Khurana, Achal Dave, and Deva Ramanan. Tracking any object amodally. *arXiv preprint arXiv:2312.12433*, 2023.

[114] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving.

arXiv preprint arXiv:2309.17080, 2023.

[115] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *arXiv preprint arXiv:2404.15506*, 2024.

[116] Peiyun Hu, Jason Ziglar, David Held, and Deva Ramanan. What you see is what you get: Exploiting visibility for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11001–11009, 2020.

[117] Peiyun Hu, Aaron Huang, John Dolan, David Held, and Deva Ramanan. Safe local motion planning with self-supervised freespace forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12732–12741, 2021.

[118] Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. In *Int. Conf. Comput. Vis.*, 2023.

[119] Yuan-Ting Hu, Hong-Shuo Chen, Kexin Hui, Jia-Bin Huang, and Alexander G Schwing. Sail-vos: Semantic amodal instance level video object segmentation-a synthetic dataset and baselines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3105–3115, 2019.

[120] Yuan-Ting Hu, Hong-Shuo Chen, Kexin Hui, Jia-Bin Huang, and Alexander G. Schwing. Sail-vos: Semantic amodal instance level video object segmentation-a synthetic dataset and baselines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3105–3115, 2019.

[121] Yuan-Ting Hu, Jiahong Wang, Raymond A Yeh, and Alexander G Schwing. Sail-vos 3d: A synthetic dataset and baselines for object detection and 3d mesh reconstruction from video data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1418–1428, 2021.

[122] Bingkun Huang, Zhiyu Zhao, Guozhen Zhang, Yu Qiao, and Limin Wang. Mgmae: Motion guided masking for video masked autoencoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13493–13504, 2023.

[123] Piao Huang, Shoudong Han, Jun Zhao, Donghaisheng Liu, Hongwei Wang, En Yu, and Alex ChiChung Kot. Refinements in motion and appearance for online multi-object tracking. *arXiv preprint arXiv:2003.07177*, 2020.

[124] Yan Huang and Irfan Essa. Tracking multiple objects through occlusions. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 1051–1058. IEEE, 2005.

[125] 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*

International Conference on Computer Vision, pages 6272–6281, 2019.

[126] Michael Isard and John MacCormick. Bramble: A bayesian multiple-blob tracker. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 34–41. IEEE, 2001.

[127] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34: 2427–2440, 2021.

[128] Omid Hosseini Jafari, Dennis Mitzel, and Bastian Leibe. Real-time rgb-d based people detection and tracking for mobile robots and head-worn cameras. In *ICRA*, 2014.

[129] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014.

[130] Hyeonho Jeong, Jinho Chang, Geon Yeong Park, and Jong Chul Ye. Dreammotion: Space-time self-similarity score distillation for zero-shot video editing. *arXiv e-prints*, pages arXiv–2403, 2024.

[131] Hanwen Jiang, Qixing Huang, and Georgios Pavlakos. Real3d: Scaling up large reconstruction models with real-world images. *arXiv preprint arXiv:2406.08479*, 2024.

[132] Thorsten Joachims et al. Transductive inference for text classification using support vector machines. In *Icml*, volume 99, pages 200–209, 1999.

[133] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE international conference on computer vision*, pages 3334–3342, 2015.

[134] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Godisart, Bart Nabbe, Iain Matthews, et al. Panoptic studio: A massively multiview system for social interaction capture. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(1):190–204, 2017.

[135] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[136] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Adv. Neural Inform. Process. Syst.*, volume 35, pages 26565–26577, 2022.

[137] Shyamgopal Karthik, Ameya Prabhu, and Vineet Gandhi. Simple unsupervised multi-object tracking. *arXiv preprint arXiv:2006.02609*, 2020.

[138] Michael Kavsek. The influence of context on amodal completion in 5-and 7-month-old infants. *Journal of Cognition and Development*, 5(2):159–184, 2004.

[139] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth

estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[140] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9492–9502, 2024.

[141] Lei Ke, Yu-Wing Tai, and Chi-Keung Tang. Deep occlusion-aware instance segmentation with overlapping bilayers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4019–4028, 2021.

[142] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat, track & map 3d gaussians for dense rgb-d slam, 2024. URL https://arxiv.org/abs/2312.02126.

[143] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42 (4):1–14, 2023.

[144] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 2023.

[145] Leonid Keselman and Martial Hebert. Approximate differentiable rendering with algebraic surfaces. In *European Conference on Computer Vision*, pages 596–614. Springer, 2022.

[146] Saad M Khan and Mubarak Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *European Conference on Computer Vision*, pages 133–146. Springer, 2006.

[147] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[148] Tarasha Khurana. Argoverse 2 challenge on 4d occupancy forecasting at the workshop on autonomous driving, cvpr. In *https://eval.ai/web/challenges/challenge-page/1977/overview*, 2023.

[149] Tarasha Khurana and Deva Ramanan. Predicting long-horizon futures by conditioning on geometry and time. *arXiv preprint arXiv:2404.11554*, 2024.

[150] Tarasha Khurana, Achal Dave, and Deva Ramanan. Detecting invisible people. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3174–3184, 2021.

[151] Tarasha Khurana, Peiyun Hu, Achal Dave, Jason Ziglar, David Held, and Deva Ramanan. Differentiable raycasting for self-supervised occupancy forecasting. In *European Conference on Computer Vision*, pages 353–369. Springer, 2022.

[152] Tarasha Khurana, Peiyun Hu, David Held, and Deva Ramanan. Point cloud forecasting as a proxy for 4d occupancy forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1116–1124, 2023.

[153] KangSan Antonio Kim and Solomon Appekeyh. 4d lidar and sensor fusion for autonomous rover missions oussema jounia*, alex moicab, gavin furtadoc, eshana mariam johnd, varsha santhoshe, daniel asantef. 2023.

[154] Kyungnam Kim and Larry S Davis. Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. In *European Conference on Computer Vision*, pages 98–109. Springer, 2006.

[155] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2014.

[156] Diederik P. Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[157] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[158] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *ECCV*. Springer, 2012.

[159] Kurt Koffka. *Principles of Gestalt Psychology*. Routledge, 2013.

[160] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10138–10148, 2021.

[161] Lambda Labs. Stable diffusion image variations. In *https://huggingface.co/lambdalabs/sd-image-variations-diffusers*, 2022.

[162] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019.

[163] Ferdinand Langer, Andres Milioto, Alexandre Haag, Jens Behley, and Cyrill Stachniss. Domain transfer for semantic segmentation of lidar data using deep neural networks. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8263–8270. IEEE, 2020.

[164] Katrin Lasinger, René Ranftl, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv preprint arXiv:1907.01341*, 2019.

[165] Laura Leal-Taixé, Gerard Pons-Moll, and Bodo Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 120–127. IEEE, 2011.

[166] Tai Sing Lee and David Mumford. Hierarchical bayesian inference in the visual cortex. *JOSA A*, 20(7):1434–1448, 2003.

[167] Yao-Chih Lee, Zhoutong Zhang, Kevin Blackburn-Matzen, Simon Niklaus, Jianming Zhang, Jia-Bin Huang, and Feng Liu. Fast view synthesis of casual videos with soup-of-planes. In *Eur. Conf. Comput. Vis.*, 2025.

[168] Jiahui Lei, Yijia Weng, Adam Harley, Leonidas Guibas, and Kostas Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. *arXiv preprint arXiv:2405.17421*, 2024.

[169] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, volume 26, pages 655–664. Wiley Online Library, 2007.

[170] Ke Li and Jitendra Malik. Amodal instance segmentation. In *ECCV*. Springer, 2016.

[171] Xin Li, Wenqing Chu, Ye Wu, Weihang Yuan, Fanglong Liu, Qi Zhang, Fu Li, Haocheng Feng, Errui Ding, and Jingdong Wang. Videogen: A reference-guided latent diffusion approach for high definition text-to-video generation. *arXiv preprint arXiv:2309.00398*, 2023.

[172] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

[173] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018.

[174] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4521–4530, 2019.

[175] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[176] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.

[177] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.

[178] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megasam: Accurate, fast, and robust structure and motion from casual dynamic videos. *arXiv preprint arXiv:2412.04463*, 2024.

[179] Yiqing Liang, Numair Khan, Zhengqin Li, Thu Nguyen-Phuoc, Douglas Lanman, James Tompkin, and Lei Xiao. Gaufre: Gaussian deformation fields for real-time dynamic novel view synthesis. *arXiv preprint*, 2023.

[180] Zhihao Liang, Qi Zhang, Wenbo Hu, Lei Zhu, Ying Feng, and Kui Jia. Analytic-splatting: Anti-aliased 3d gaussian splatting via analytic integration. In *Eur. Conf. Comput. Vis.*, 2024.

[181] 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*, pages 740–755. Springer, 2014.

[182] Zhiqiu Lin, Siyuan Cen, Daniel Jiang, Jay Karhade, Hewei Wang, Chancharik Mitra, Tiffany Ling, Yuhan Huang, Sifan Liu, Mingyu Chen, Rushikesh Zawar, Xue Bai, Yilun Du, Chuang Gan, and Deva Ramanan. Towards understanding camera motions in any video. *arXiv preprint arXiv:2504.15376*, 2025.

[183] Huan Ling, David Acuna, Karsten Kreis, Seung Wook Kim, and Sanja Fidler. Variational amodal object completion. *Advances in Neural Information Processing Systems*, 33:16246–16257, 2020.

[184] Fangfu Liu, Wenqiang Sun, Hanyang Wang, Yikai Wang, Haowen Sun, Junliang Ye, Jun Zhang, and Yueqi Duan. Reconx: Reconstruct any scene from sparse views with video diffusion model. *arXiv preprint arXiv:2408.16767*, 2024.

[185] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[186] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems*, 36:22226–22246, 2023.

[187] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10072–10083, 2024.

[188] Qiankun Liu, Qi Chu, Bin Liu, and Nenghai Yu. Gsm: Graph similarity model for multi-object tracking. International Joint Conferences on Artificial Intelligence Organization, 2020.

[189] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9298–9309, 2023.

[190] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. 2019.

[191] Hao Lu, Tianshuo Xu, Wenzhao Zheng, Yunpeng Zhang, Wei Zhan, Dalong Du, Masayoshi Tomizuka, Kurt Keutzer, and Yingcong Chen. Drivingrecon: Large 4d gaussian reconstruction model for autonomous driving. *arXiv preprint arXiv:2412.09043*, 2024.

[192] Jiahao Lu, Tianyu Huang, Peng Li, Zhiyang Dou, Cheng Lin, Zhiming Cui, Zhen Dong, Sai-Kit Yeung, Wenping Wang, and Yuan Liu. Align3r: Aligned monocular depth estimation for dynamic videos. *arXiv preprint arXiv:2412.03079*, 2024.

[193] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022.

[194] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *3DV*, 2024.

[195] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pages 800–809. IEEE, 2024.

[196] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *CVPR*, pages 3569–3577, 2018.

[197] Wei-Chiu Ma, De-An Huang, Namhoon Lee, and Kris M Kitani. Forecasting interactive dynamics of pedestrians with fictitious play. In *CVPR*, 2017.

[198] Reza Mahjourian, Jinkyu Kim, Yuning Chai, Mingxing Tan, Ben Sapp, and Dragomir Anguelov. Occupancy flow fields for motion forecasting in autonomous driving. *IEEE Robotics and Automation Letters*, 7(2):5639–5646, 2022.

[199] Jiageng Mao, Minzhe Niu, Chenhan Jiang, Hanxue Liang, Xiaodan Liang, Yamin Li, Chaoqiang Ye, Wei Zhang, Zhenguo Li, Jie Yu, et al. One million scenes for autonomous driving: Once dataset. *arXiv preprint arXiv:2106.11037*, 2021.

[200] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the Eighth IEEE International Conference on Computer Vision (ICCV 2001)*, volume 2, pages 416–423. IEEE, 2001.

[201] Willi Menapace, Aliaksandr Siarohin, Ivan Skorokhodov, Ekaterina Deyneka, Tsai-Shien Chen, Anil Kag, Yuwei Fang, et al. Snap video: Scaled spatiotemporal transformers for text-to-video synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7038–7048, 2024.

[202] Benedikt Mersch, Xieyuanli Chen, Jens Behley, and Cyrill Stachniss. Self-supervised point cloud prediction using 3d spatio-temporal convolutional networks. In *Conference on Robot Learning*, pages 1444–1454. PMLR, 2022.

[203] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12663–12673, 2023.

[204] 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.

[205] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019.

[206] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020.

[207] Himangi Mittal, Brian Okorn, and David Held. Just go with the flow: Self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[208] Michael Montemerlo, Jan Becker, Suhrid Bhat, Hendrik Dahlkamp, Dmitri Dolgov, Scott Ettinger, Dirk Haehnel, Tim Hilden, Gabe Hoffmann, Burkhard Huhnke, et al. Junior: The stanford entry in the urban challenge. *Journal of field Robotics*, 25(9):569–597, 2008.

[209] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022.

[210] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015.

[211] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020.

[212] Masatoshi Okutomi and Takeo Kanade. A multiple-baseline stereo. In *CVPR*, volume 93, pages 63–69, 1991.

[213] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[214] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *arXiv preprint arXiv:1811.06711*, 2018.

[215] Yumiko Otsuka, So Kanazawa, and Masami K. Yamaguchi. Development of modal and amodal completion in infants. *Perception*, 35(9):1251–1264, 2006.

[216] Ege Ozguroglu, Ruoshi Liu, Dídac Surís, Dian Chen, Achal Dave, Pavel Tokmakov, and

Carl Vondrick. pix2gestalt: Amodal segmentation by synthesizing wholes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3931–3940, 2024.

[217] Ioannis Papakis, Abhijit Sarkar, and Anuj Karpatne. Gcnnmatch: Graph convolutional neural networks for multi-object tracking via sinkhorn normalization. *arXiv preprint arXiv:2010.00067*, 2020.

[218] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021.

[219] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*. IEEE, 2009.

[220] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016.

[221] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. UniDepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[222] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011.

[223] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[224] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in neural information processing systems*, pages 305–313, 1989.

[225] 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.

[226] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[227] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[228] Charles R Qi, Yin Zhou, Mahyar Najibi, Pei Sun, Khoa Vo, Boyang Deng, and Dragomir Anguelov. Offboard 3d object detection from point cloud sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6134–6144,

2021.

[229] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with KINS dataset. In *CVPR*, 2019.

[230] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with kins dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2019.

[231] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. 2021.

[232] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[233] Amit Raj, Srinivas Kaza, Ben Poole, Michael Niemeyer, Ben Mildenhall, Nataniel Ruiz, Shiran Zada, Kfir Aberman, Michael Rubenstein, Jonathan Barron, Yuanzhen Li, and Varun Jampani. Dreambooth3d: Subject-driven text-to-3d generation. In *Int. Conf. Comput. Vis.*, 2023.

[234] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pages 1–16. IEEE, 2020.

[235] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[236] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[237] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[238] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021.

[239] Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87, 1999.

[240] Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. Maximum margin planning.

In *Proceedings of the 23rd international conference on Machine learning*, pages 729–736, 2006.

[241] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[242] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[243] N. Dinesh Reddy, Robert Tamburo, and Srinivasa G. Narasimhan. Walt: Watch and learn 2d amodal representation from time-lapse imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9356–9366, 2022.

[244] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10901–10911, 2021.

[245] Jiawei Ren, Cheng Xie, Ashkan Mirzaei, Karsten Kreis, Ziwei Liu, Antonio Torralba, Sanja Fidler, Seung Wook Kim, Huan Ling, et al. L4gm: Large 4d gaussian reconstruction model. *Advances in Neural Information Processing Systems*, 37:56828–56858, 2024.

[246] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[247] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas MÃ¼ller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[248] Nicholas Rhinehart, Rowan McAllister, and Sergey Levine. Deep imitative models for flexible inference, planning, and control. *arXiv preprint arXiv:1810.06544*, 2018.

[249] John W Roach and JK Aggarwal. Determining the movement of objects from a sequence of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):554–562, 1980.

[250] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *ECCV*. Springer, 2016.

[251] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[252] 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.

[253] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10684–10695, 2022.

[254] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III*, pages 234–241. Springer International Publishing, 2015.

[255] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015.

[256] Abbas Sadat, Mengye Ren, Andrei Pokrovsky, Yen-Chen Lin, Ersin Yumer, and Raquel Urtasun. Jointly learnable behavior and trajectory planning for self-driving vehicles. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3949–3956. IEEE, 2019.

[257] Abbas Sadat, Sergio Casas, Mengye Ren, Xinyu Wu, Pranaab Dhawan, and Raquel Urtasun. Perceive, predict, and plan: Safe motion planning through interpretable semantic representations. *European Conference on Computer Vision*, 2020.

[258] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[259] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Kamyar Ghasemipour, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[260] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[261] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, et al. Zeronvs: Zero-shot 360-degree view synthesis from a single real image. *arXiv preprint arXiv:2310.17994*, 2023.

[262] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[263] Paul Scovanner and Marshall F Tappen. Learning pedestrian dynamics from the real world. In *ICCV*. IEEE, 2009.

[264] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023.

[265] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023.

[266] Assaf Shocher, Nadav Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3118–3126, 2018.

[267] A Siegel. The development of spatial representations of large-scale environments. in. *Advances in Child Development and Behavior/Academic Press*, 1975.

[268] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

[269] Chonghyuk Song, Gengshan Yang, Kangle Deng, Jun-Yan Zhu, and Deva Ramanan. Total-recon: Deformable scene reconstruction for embodied view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17671–17682, 2023.

[270] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[271] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR*, October 2021. URL https://arxiv.org/abs/2010.02502.

[272] Davide Spinello and Daniel J Stilwell. Nonlinear estimation with state-dependent gaussian observation noise. *IEEE Transactions on Automatic Control*, 55(6):1358–1366, 2010.

[273] Colton Stearns, Adam Harley, Mikaela Uy, Florian Dubost, Federico Tombari, Gordon Wetzstein, and Leonidas Guibas. Dynamic gaussian marbles for novel view synthesis of casual monocular videos. In *SIGGRAPH Asia 2024 Conference Papers*, 2024.

[274] Colton Stearns, Adam W. Harley, Mikaela Uy, Florian Dubost, Federico Tombari, Gordon Wetzstein, and Leonidas Guibas. Dynamic gaussian marbles for novel view synthesis of casual monocular videos. *arXiv preprint arXiv:2406.18717*, 2024.

[275] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[276] Jingxiang Sun, Bo Zhang, Ruizhi Shao, Lizhen Wang, Wen Liu, Zhenda Xie, and Yebin Liu. Dreamcraft3d: Hierarchical 3d generation with bootstrapped diffusion prior. *arXiv preprint arXiv:2310.16818*, 2023.

[277] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248. PMLR, 2020.

[278] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-

fast single-view 3d reconstruction. 2024.

[279] Jeff Tan, Donglai Xiang, Shubham Tulsiani, Deva Ramanan, and Gengshan Yang. Dressrecon: Freeform 4d human reconstruction from monocular video. *arXiv preprint arXiv:2409.20563*, 2024.

[280] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[281] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. In *Int. Conf. Comput. Vis.*, 2023.

[282] Matthias Tangemann, Steffen Schneider, Julius Von Kügelgen, Francesco Locatello, Peter Gehler, Thomas Brox, Matthias Kümmerer, Matthias Bethge, and Bernhard Schölkopf. Unsupervised object learning via common fate. *arXiv preprint arXiv:2110.06562*, 2021.

[283] Ayush Tewari, Tianwei Yin, George Cazenavette, Semon Rezchikov, Joshua B Tenenbaum, Frédo Durand, William T Freeman, and Vincent Sitzmann. Diffusion with forward models: Solving stochastic inverse problems without direct supervision. *arXiv preprint arXiv:2306.11719*, 2023.

[284] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022.

[285] Minh Tran, Khoa Vo, Kashu Yamazaki, Arthur Fernandes, Michael Kidd, and Ngan Le. Aisformer: Amodal instance segmentation with transformer. *arXiv preprint arXiv:2210.06323*, 2022.

[286] Minh Tran, Khoa Vo, Vuong Ho, Tri Nguyen, and Ngan Hoang Le. Amodal instance segmentation with diffusion shape prior estimation. In *The First Workshop on Populating Empty Cities–Virtual Humans for Robotics and Autonomous Driving at CVPR 2024*, 2024.

[287] Joseph Tung, Gene Chou, Ruojin Cai, Guandao Yang, Kai Zhang, Gordon Wetzstein, Bharath Hariharan, and Noah Snavely. Megascenes: Scene-level view synthesis at scale. In *European Conference on Computer Vision*, pages 197–214. Springer, 2024.

[288] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[289] Kevin Turner. Decoding latents to rgb without upscaling. In *Huggingface*.

[290] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017.

[291] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):

425–466, 2008.

[292] Basile Van Hoorick, Rundi Wu, Ege Ozguroglu, Kyle Sargent, Ruoshi Liu, Pavel Tokmakov, Achal Dave, Changxi Zheng, and Carl Vondrick. Generative camera dolly: Extreme monocular dynamic novel view synthesis. In *Eur. Conf. Comput. Vis.*, 2024.

[293] Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *Advances in Neural Information Processing Systems*, 35:23371–23385, 2022.

[294] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[295] Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9773–9783, 2023.

[296] Jikai Wang, Qifan Zhang, Yu-Wei Chao, Bowen Wen, Xiaohu Guo, and Yu Xiang. Ho-cap: A capture system and dataset for 3d reconstruction and pose tracking of hand-object interaction. *arXiv preprint arXiv:2406.06843*, 2024.

[297] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14549–14560, 2023.

[298] Qianqian Wang, Vickie Ye, Hang Gao, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. *arXiv preprint arXiv:2407.13764*, 2024.

[299] Qianqian Wang, Vickie Ye, Hang Gao, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. *arXiv preprint arXiv:2407.13764*, 2024.

[300] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. *arXiv preprint arXiv:2410.19115*, 2024.

[301] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.

[302] Xueyang Wang, Xiya Zhang, Yinheng Zhu, Yuchen Guo, Xiaoyun Yuan, Liuyu Xiang, Zerun Wang, Guiguang Ding, David Brady, Qionghai Dai, et al. Panda: A gigapixel-level human-centric video dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3268–3278, 2020.

[303] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and

Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *CVPR*, 2019.

[304] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36:8406–8441, 2023.

[305] Zihan Wang, Bowen Li, Chen Wang, and Sebastian Scherer. Airshot: Efficient few-shot detection for autonomous exploration. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11654–11661, 2024. doi: 10.1109/IROS58592.2024.10801738.

[306] Daniel Watson, William Chan, Ricardo Martin Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. In *Int. Conf. Learn. Represent.*, 2023.

[307] Ethan Weber, Aleksander Holynski, Varun Jampani, Saurabh Saxena, Noah Snavely, Abhishek Kar, and Angjoo Kanazawa. Nerfiller: Completing scenes via generative 3d inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20731–20741, 2024.

[308] Chen Wei, Karttikeya Mangalam, Po-Yao Huang, Yanghao Li, Haoqi Fan, Hu Xu, Huiyu Wang, Cihang Xie, Alan Yuille, and Christoph Feichtenhofer. Diffusion models as masked autoencoders. *arXiv preprint arXiv:2304.03283*, 2023.

[309] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[310] Xinyue Wei, Kai Zhang, Sai Bi, Hao Tan, Fujun Luan, Valentin Deschaintre, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. Meshlrm: Large reconstruction model for high-quality meshes. *arXiv preprint arXiv:2404.12385*, 2024.

[311] Xinshuo Weng and Kris Kitani. A baseline for 3d multi-object tracking. *arXiv preprint arXiv:1907.03961*, 1(2):6, 2019.

[312] Xinshuo Weng, Jianren Wang, Sergey Levine, Kris Kitani, and Nicholas Rhinehart. 4d forecasting: Sequential forecasting of 100,000 points. 2020.

[313] Xinshuo Weng, Jianren Wang, Sergey Levine, Kris Kitani, and Nicholas Rhinehart. Inverting the pose forecasting pipeline with spf2: Sequential pointcloud forecasting for sequential pose forecasting. *arXiv preprint arXiv:2003.08376*, 2020.

[314] Xinshuo Weng, Jianren Wang, Sergey Levine, Kris Kitani, and Nick Rhinehart. 4d forecasting: Sequantial forecasting of 100,000 points. In *Euro. Conf. Comput. Vis. Worksh*, volume 3, 2020.

[315] Xinshuo Weng, Jianren Wang, Sergey Levine, Kris Kitani, and Nicholas Rhinehart. Inverting the pose forecasting pipeline with spf2: Sequential pointcloud forecasting for sequential

pose forecasting. In *Proceedings of the 2020 Conference on Robot Learning*, pages 11–20, 2021.

[316] Xinshuo Weng, Junyu Nan, Kuan-Hui Lee, Rowan McAllister, Adrien Gaidon, Nicholas Rhinehart, and Kris M Kitani. S2net: Stochastic sequential pointcloud forecasting. In *European Conference on Computer Vision*, pages 549–564. Springer, 2022.

[317] Max Wertheimer. Gestalt theory. 1938.

[318] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2.0: Next generation datasets for self-driving perception and forecasting. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[319] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *WACV*. IEEE, 2018. doi: 10.1109/WACV.2018.00087.

[320] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, 2017.

[321] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Wang Xinggang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint*, 2023.

[322] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

[323] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7623–7633, 2023.

[324] Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T Barron, and Aleksander Holynski. Cat4d: Create anything in 4d with multi-view video diffusion models. *arXiv preprint arXiv:2411.18613*, 2024.

[325] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

[326] Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In *Eur. Conf. Comput. Vis.*, 2022.

[327] Yuting Xiao, Yanyu Xu, Ziming Zhong, Weixin Luo, Jiawei Li, and Shenghua Gao. Amodal segmentation based on visible region segmentation and shape prior. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2995–3003, 2021.

[328] Zeqi Xiao, Wenqi Ouyang, Yifan Zhou, Shuai Yang, Lei Yang, Jianlou Si, and Xingang Pan. Trajectory attention for fine-grained video motion control. In *Int. Conf. Learn. Represent.*, 2025.

[329] Desai Xie, Sai Bi, Zhixin Shu, Kai Zhang, Zexiang Xu, Yi Zhou, Sören Pirk, Arie Kaufman, Xin Sun, and Hao Tan. Lrm-zero: Training large reconstruction models with synthesized data. *arXiv preprint arXiv:2406.09371*, 2024.

[330] Yiming Xie, Chun-Han Yao, Vikram Voleti, Huaizu Jiang, and Varun Jampani. Sv4d: Dynamic 3d content generation with multi-frame and multi-view consistency. *arXiv preprint arXiv:2407.17470*, 2024.

[331] Jinbo Xing, Menghan Xia, Yuxin Liu, Yuechen Zhang, Yong Zhang, Yingqing He, Hanyuan Liu, Haoxin Chen, Xiaodong Cun, Xintao Wang, et al. Make-your-video: Customized video generation using textual and structural guidance. *arXiv preprint arXiv:2306.00943*, 2023.

[332] Katherine Xu, Lingzhi Zhang, and Jianbo Shi. Amodal completion via progressive mixed context diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9099–9109, 2024.

[333] 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.

[334] Kota Yamaguchi, Alexander C Berg, Luis E Ortiz, and Tamara L Berg. Who are you with and where are you going? In *CVPR 2011*. IEEE, 2011.

[335] Hong Yan, Yang Liu, Yushen Wei, Zhen Li, Guanbin Li, and Liang Lin. Skeletonmae: graph-based masked autoencoder for skeleton sequence pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5606–5618, 2023.

[336] Xiaosheng Yan, Feigege Wang, Wenxi Liu, Yuanlong Yu, Shengfeng He, and Jia Pan. Visualizing the invisible: Occluded vehicle segmentation and recovery. In *ICCV*, 2019.

[337] Chen Yang, Sikuang Li, Jiemin Fang, Ruofan Liang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Gaussianobject: High-quality 3d object reconstruction from four views with gaussian splatting. *SIGGRAPH Asia*, 2024.

[338] Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2129–2137, 2016.

[339] Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. Banmo: Building animatable 3d neural models from many casual videos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[340] Jiawei Yang, Jiahui Huang, Yuxiao Chen, Yan Wang, Boyi Li, Yurong You, Apoorva Sharma, Maximilian Igl, Peter Karkus, Danfei Xu, et al. Storm: Spatio-temporal reconstruction

model for large-scale outdoor scenes. *arXiv preprint arXiv:2501.00602*, 2024.

[341] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024.

[342] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation. *Entropy*, 25(10):1469, 2023.

[343] Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. In *Int. Conf. Learn. Represent.*, 2024.

[344] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[345] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint*, 2023.

[346] Jian Yao, Yuxin Hong, Chiyu Wang, Tianjun Xiao, Tong He, Francesco Locatello, David P. Wipf, Yanwei Fu, and Zheng Zhang. Self-supervised amodal video object segmentation. *Advances in Neural Information Processing Systems*, 35:6278–6291, 2022.

[347] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. *arXiv preprint arXiv:2012.05877*, 2020.

[348] Li Yi, Boqing Gong, and Thomas Funkhouser. Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15363–15373, 2021.

[349] Shengming Yin, Chenfei Wu, Huan Yang, Jianfeng Wang, Xiaodong Wang, Minheng Ni, Zhengyuan Yang, Linjie Li, Shuguang Liu, Fan Yang, et al. Nuwa-xl: Diffusion over diffusion for extremely long video generation. *arXiv preprint arXiv:2303.12346*, 2023.

[350] Meng You, Zhiyu Zhu, Hui Liu, and Junhui Hou. Nvs-solver: Video diffusion model as zero-shot novel view synthesizer. In *Int. Conf. Learn. Represent.*, 2025.

[351] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021.

[352] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10459–10469, 2023.

[353] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model

beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.

[354] Mark YU, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. *arXiv preprint arXiv:2503.05638*, 2025.

[355] Qian Yu, Gérard Medioni, and Isaac Cohen. Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *ICCV*, 2007.

[356] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

[357] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024.

[358] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8660–8669, 2019.

[359] Guanqi Zhan, Chuanxia Zheng, Weidi Xie, and Andrew Zisserman. What does stable diffusion know about the 3d scene? *arXiv preprint arXiv:2310.06836*, 2023.

[360] Guanqi Zhan, Chuanxia Zheng, Weidi Xie, and Andrew Zisserman. A general protocol to probe large vision models for 3d physical understanding. *arXiv preprint arXiv:2310.06836*, 2023.

[361] Guanqi Zhan, Chuanxia Zheng, Weidi Xie, and Andrew Zisserman. Amodal ground truth and completion in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28003–28013, 2024.

[362] Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised scene de-occlusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3784–3792, 2020.

[363] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024.

[364] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision*, pages 1–19. Springer, 2024.

[365] Lunjun Zhang, Yuwen Xiong, Ze Yang, Sergio Casas, Rui Hu, and Raquel Urtasun. Copilot4d: Learning unsupervised world models for autonomous driving via discrete diffusion. *arXiv preprint arXiv:2311.01017*, 2023.

[366] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on*

Computer Vision, pages 3836–3847, 2023.

[367] Ziheng Zhang, Anpei Chen, Ling Xie, Jingyi Yu, and Shenghua Gao. Learning semantics-aware distance map with semantics layering network for amodal instance segmentation. In *ACM Multimedia*, 2019.

[368] Qitao Zhao, Amy Lin, Jeff Tan, Jason Y. Zhang, Deva Ramanan, and Shubham Tulsiani. Diffusionsfm: Predicting structure and motion via ray origin and endpoint diffusion, 2025. URL https://arxiv.org/abs/2505.05473.

[369] Yizhou Zhao, Hengwei Bian, Kaihua Chen, Pengliang Ji, Liao Qu, Shao-yu Lin, Weichen Yu, et al. Metric from human: Zero-shot monocular metric depth estimation via test-time adaptation. In *NeurIPS*, 2024.

[370] Changshi Zhou, Rong Jiang, Feng Luan, Shaoqiang Meng, Zhipeng Wang, Yanchao Dong, Yanmin Zhou, and Bin He. Dual-arm robotic fabric manipulation with quasi-static and dynamic primitives for rapid garment flattening. *IEEE/ASME Transactions on Mechatronics*, pages 1–11, 2025. doi: 10.1109/TMECH.2025.3556283.

[371] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. *CVPR*, 2024.

[372] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. *arXiv:2004.01177*, 2020.

[373] Zhizhuo Zhou and Shubham Tulsiani. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12588–12597, June 2023.

[374] Yan Zhu, Yuandong Tian, Dimitris Mexatas, and Piotr Dollár. Semantic amodal segmentation. *arXiv preprint arXiv:1509.01329*, 2015.

[375] Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *CVPR*, 2017.

[376] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. FSGS: real-time few-shot view synthesis using gaussian splatting. *CoRR*, abs/2312.00451, 2023.

[377] Zi-Xin Zou, Weihao Cheng, Yan-Pei Cao, Shi-Sheng Huang, Ying Shan, and Song-Hai Zhang. Sparse3d: Distilling multiview-consistent diffusion for object reconstruction from sparse views. *arXiv preprint arXiv:2308.14078*, 2023.