Parameter-Efficient Neuro-Symbolic Action Anticipation via Iterative Context Refinement

FNU Aryan

CMU-RI-TR-25-45

October 25, 2025

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

Thesis Committee:
Katia Sycara, *chair*
Deva Ramanan
Himangi Mittal

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

To my late grandfather

Abstract

As robots and intelligent systems increasingly interact with humans, the ability to understand users by anticipating their actions becomes significantly more important. Current approaches to action anticipation leverage the inference capabilities of large foundational models but are limited in their application by the complexity and resource requirement, as well as the difficulty of training. This work introduces a novel neuro-symbolic approach, SymAnt, that unifies the expressiveness of neural networks with symbolic knowledge in the form of knowledge and scene graphs as context to effectively anticipate actions from short videos. By leveraging symbolic context, our approach significantly reduces model size as well as the need for extensive contextual observations. We present a joint-graph search approach that utilizes scene and knowledge graphs to reason over objects' spatial relationships as well as their attributes, affordances, and actions, allowing for action predictions with less context. Inspired by diffusion models, we propose an iterative action denoising approach that uses symbolic knowledge as initialization and refines the predicted sequence of future actions to improve accuracy, coherence, and plausibility. Through our experiments, we demonstrate the effectiveness of our neuro-symbolic approach, outperforming current state-of-the-art methods by up to 5% on a set of four diverse datasets, including *Breakfast*, *50 Salads*, *EPIC Kitchens*, and *EGTEA Gaze+* while reducing the model size by over 98% compared to the best neural-only baseline.

Acknowledgments

The past two years at CMU have been a journey I'll always carry with me, full of learning, growth, and a fair share of ups and downs. I've been lucky to have incredible mentors, friends, and family by my side through it all. They stood by me when things got tough and kept pushing me to keep going, to keep improving. I honestly wouldn't have made it through without them.

First and foremost, I want to express my deepest gratitude to my advisor, Katia Sycara, for her unwavering support, insightful guidance, and steady encouragement throughout my time at CMU, both in research and in shaping my broader professional path. Katia is not only an exceptional researcher but also a generous, driven, and deeply inspiring mentor. Her sharp eye for impactful yet practical problems and her relentless dedication to the work have left a lasting impression on me. I'm especially thankful for the thoughtful conversations, moments of clarity, and the many long (and often never-ending) group meetings, which helped shape me not only as a researcher capable of designing and executing ideas, but also as someone who can communicate them with purpose and conviction.

I am immensely grateful to Simon Stepputtis for his constant support, thoughtful guidance, and collaborative spirit throughout my thesis projects. Over the past two years, Simon has been both a second mentor and a close friend. He's been honest when I needed a reality check, encouraging when things felt overwhelming, and always ready to dive deep into discussions, technical or otherwise. From brainstorming ideas and convincing Katia to take a chance on them, to debugging stubborn code and polishing paper drafts (including this one), Simon has been there at every step. His mentorship has meant a lot to me, not just as a junior student learning the ropes, but as someone genuinely lucky to have had such a sharp and generous mind to learn from.

I would also like to thank the postdocs in our lab, Joseph Campbell, Yaqi Xie, and Woojun Kim, for their steady support and thoughtful feedback during group meetings. I'm deeply grateful to all my labmates who made each day in the lab a truly rewarding experience. I've learned and grown alongside some of the most brilliant and generous people I've met. Special thanks to Dana Hughes, Ini Oguntola, Muhan Lin, Muyang Yan, Renos

Zabounidis, Samuel Li, Sarthak Bhagat, Sha Yi, Shreya Sharma, Silong Yong, Srujan Deolasee, Suyang Shi, Udit Arora, Venkata Nagarjun, and Zifu Wan for making this journey all the more memorable.

I'm deeply thankful to Prof. S. Indu from DTU, who first introduced me to research during my undergraduate years and supported me throughout my journey to graduate school. My path in robotics and at the Robotics Institute truly began with the Robotics Institute Summer Scholars (RISS) program. I'm incredibly grateful to Rachel Burcin and Prof. John Dolan for hosting what I believe is the best undergraduate research program in the world. Thank you to Prof. Chen Wang and Prof. Sebastian Scherer for believing in me and opening the door to RISS, it changed my life, and the lives of many others around me. And to Rachel, the RISS simply wouldn't be the same without your boundless energy, enthusiasm, and unwavering support for students like me.

I was fortunate to find my Pittsburgh family in the form of an incredible group of friends who made the city feel like home. I'm especially grateful to my amazing roommate, Vihaan, for being a constant source of support and laughter. I'd also like to thank Aadesh, Chehak, Dhruv, Khush, Mehal, Neham, Parth, Peya, Rupali, Sagar, Sagnik, Sanjali, Saksham, Sashank, Shashwat, Shivangi, Srujan, Stevan, Tirtha, Yash, and Yatharth for the countless unforgettable memories. I truly miss our weekend tradition of gathering together as it always helped lift the weight of the week. I'm also deeply thankful to my friends from undergrad and school, Bharat, Gaurav, Harsha, Kunal, Mahika, Mrigakshi, Naveen, Rajkhush, Rahul, Ravita, Siddharth, and Vishal, for supporting me from afar and staying close despite the distance.

I am extremely grateful to my mom and dad for their unwavering love, constant encouragement, and faith in my dreams. Coming from a small town in India, I never imagined I'd one day study at such a prestigious institution in the United States. None of this would have been possible without their sacrifices and belief in me. They always encouraged me to pursue my passions and carve my own path, and for that, I'll always be thankful. I'm also deeply grateful to my cousins, grandparents, and extended family for cheering me on from afar. Each of them, in their own way, has shaped the person I am today, and for that, I will be forever grateful.

Funding

This work was supported by Defense Advanced Research Projects Agency (DARPA) grant FA8750-23-2-1015 as part of the Assured Neuro Symbolic Learning and Reasoning (ANSR) program, and the Army Research Laboratory (ARL) grants W911QX24F0049 and W911NF2320007.

Contents

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

List of Figures

List of Tables

Chapter 1

Introduction

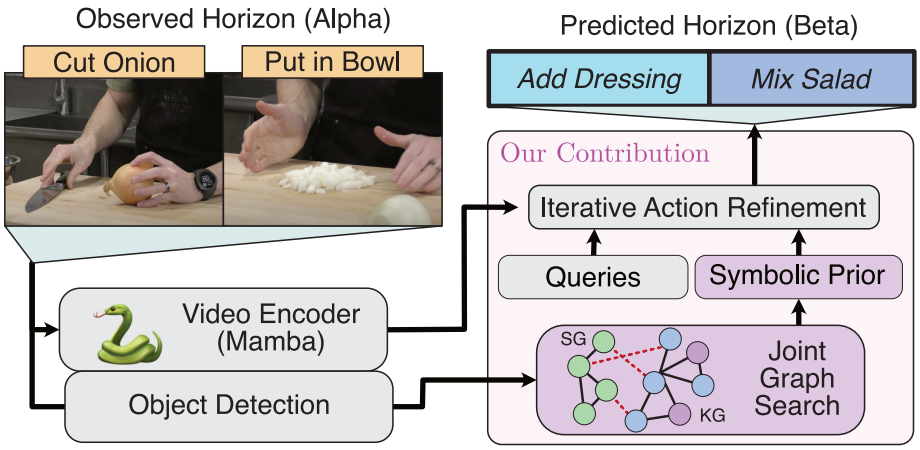

Figure 1.1: Overview of our neuro-symbolic approach (contributions in purple): We utilize a scene graph (SG, green nodes) and knowledge graph (KG, blue nodes for entities, purple for attributes/affordances/actions) to capture spatial and contextual information about the scene to initialize a novel diffusion-inspired decoder capable of predicting long-horizon actions from short video contexts.

Action anticipation refers to predicting a temporally ordered sequence of *future* actions after observing only a brief snippet of past video. Action anticipation is crucial for intelligent agents in tasks such as human–robot collaboration [10, 25], autonomous driving [36, 50], and reconnaissance [7, 47]. Accurately forecasting the behaviour of other agents allows systems to adjust their own behaviour proactively, providing valuable time for planning, resource allocation, and hazard avoidance. Early work

treated the problem as sequence modelling with recurrent neural networks (RNNs) and gated variants such as LSTMs, which captured short-term dynamics but struggled with long-range dependencies. CNN–RNN hybrids improved spatial encoding yet retained the sequential bottleneck. The introduction of self-attention enabled transformer-based architectures, which model long-range context in parallel and dramatically improved accuracy. Most recently, generative diffusion frameworks—including Future Transformer and DiffAnt [29, 82]—frame anticipation as an iterative denoising process that can represent multiple plausible futures. However, these approaches still require large datasets and extensive observations. To address this challenge, we propose the integration of symbolic knowledge as additional context for action anticipation. Figure 1.1 provides an overview of our method, which uses a Mamba [33] video encoder and object detector, a joint graph search approach, and an iterative action refinement.

We introduce a method that represents this context as a *scene graph* (SG)—a directed graph whose nodes are detected object instances and whose edges encode pairwise spatial relations such as *next to*, *on*, or *holding*. Scene graphs were popularised by the Visual Genome dataset and are now a standard intermediate representation for image retrieval, VQA, and robotic manipulation. Likewise, we employ a *knowledge graph* (KG), a semantic network that stores class-level facts, attributes, and affordances (e.g., *tomato* $\rightarrow$ *cuttable*), giving the model access to commonsense information that is not always visually observable. A key contribution of our work is the joint-graph search: For example, a *tomato* node in the KG is connected to an affordance node *cuttable*, while the SG represents a *knife* with the affordance *can cut* next to it; a likely future action could therefore be *cut tomato*. Combining spatial and contextual information is crucial for successfully making inferences. To this end, we propose to combine scene graphs that encode spatial relationships and account for multiple object instances with the knowledge of affordances and actions in the knowledge graph through a dynamic graph-merging and a subsequent search approach. This unified representation lets the model exploit both *where* objects are and *what* they can do.

To ultimately predict actions in dynamic environments with humans where behaviors may quickly change, in our second contribution, we take inspiration from diffusion models [17] that can iteratively adjust and refine the sequence of antic-

ipated actions. Given their ability to iteratively refine their predictions from an initialization conditioned on some context, diffusion models are well suited to not only generate images [39], videos [69], or motion control [72], but also to predict and continuously refine a sequence of future actions. By leveraging past video observations for conditioning, diffusion models can generate plausible future action sequences [82]. However, applying diffusion models to action anticipation introduces unique challenges. Traditional diffusion processes operate in either fully discrete [6] or continuous [52, 63] state spaces, with respect to their inputs and outputs. In contrast, action anticipation requires inputs comprising learned continuous representations of observed video sequences and symbolic context, while the outputs are discrete action sequences. Recent works [30, 82] address this issue by embedding discrete actions in a continuous latent space. Although effective, this approach requires two additional networks: one to embed discrete actions into the latent space for the forward diffusion process, and another to decode latent actions back into discrete ones. This reliance on extra networks adds complexity, increases computational overhead, and risks error propagation during both training and inference. Inspired by the diffusion process and in contrast to existing approaches, our action decoder is initialized with a symbolic context, instead of noise, iteratively refining it into a continuous action representation conditioned on the latent video encoding.

In summary, we propose the following contributions:

- Our novel approach integrating scene and knowledge graphs as supplementary context, achieving fast action anticipation from minimal observation while allowing for smaller models compared to other approaches.
- We introduce a diffusion-inspired action decoder that iteratively refines action sequences from symbolic context.
- We demonstrate state-of-the-art performance of our approach on four datasets (*Breakfast*, *50 Salads*, *EPIC Kitchens*, and *EGTEA Gaze+*) across short- and long-context prediction benchmarks while significantly reducing model size.

Chapter 2

Related Work

This section reviews related work relevant to our neuro-symbolic action anticipation framework. We begin by discussing symbolic knowledge and its role in computer vision, followed by an exploration of diffusion models, including their application to action anticipation and related tasks such as trajectory prediction. Finally, we review prior work specifically focused on action anticipation.

2.0.1 Symbolic Knowledge for Computer Vision

The integration of structured domain knowledge, particularly through knowledge graphs [43], into vision models is significantly enhancing grounding, interpretability, and performance [51, 56, 57, 84]. Neuro-symbolic vision pipelines [44] leverage the structure and hierarchy of knowledge graphs to enhance a wide range of vision tasks, including object detection [22, 48], scene graph generation [76], transfer learning [15, 58], vision-language pre-training [4], classification [71, 78], and image captioning [79, 83]. For example, [57] introduces a graph search mechanism over knowledge graphs, and subsequent work [8] extended this to include novel concepts. NeSCA [9] builds upon this insight to identify object affordances and apply this additional information to action anticipation. In parallel, scene graphs have also been extensively utilized in computer vision for tasks such as object detection, image captioning, and visual question answering [13, 16, 38, 53, 59]. In the realm of action anticipation, scene graphs have been employed to predict future actions by modeling object interactions

and their temporal evolution [65]. While previous methods either leveraged knowledge graphs or scene graphs, our approach uniquely integrates both to create a richer contextual framework. This allows our approach to not only reason over general object knowledge but also over the number of object instances and their spatial relationships. To our knowledge, our approach is the first to leverage a combination of scene and common-sense knowledge information for action anticipation.

2.0.2 Diffusion Models

Denoising diffusion models [39] have recently emerged as a powerful class of generative models that learn complex data distributions through the iterative denoising process. These models have achieved notable success across diverse domains, including image generation [20, 66, 74], natural language generation [31, 52, 75], text-to-image synthesis [34], and audio generation [49], object detection [14], video forecasting, infilling [40], and action segmentation [55]. Additionally, diffusion models have also been investigated for discrete state-spaces [6, 67]. Some recent action anticipation works [30, 82] have used diffusion models to generate latent embeddings of discrete actions. However, unlike these works, we initialize the denoising process with a symbolic context, improving accuracy while reducing parameters and the need for extensive observations.

2.0.3 Action Anticipation

Action anticipation addresses the problem of predicting future actions based on previously observed behavior, generally utilizing video clips of humans or agents performing various tasks. Early work focused on action anticipation from third-person video perspectives [2, 26, 29, 45]. However, recently, with the development of multiple challenge benchmarks [18, 19, 32], action anticipation has increasingly been considered from the perspective of first-person (egocentric) vision. Early methods for long-term action anticipation used RNNs [2, 24] and TCNs [3] to model action progressions over time. More recently, Transformer-based models [27, 29, 61, 81] have demonstrated superior performance in learning long-range temporal relationships, making them the preferred choice for long-term action anticipation. NeSCA [9] focuses on short-context anticipation where the observed time horizon is relatively smaller, and additional

context can be very beneficial. Our approach achieves state-of-the-art performance across both third-person and egocentric video datasets, excelling in both short-context anticipation and general long-term action anticipation, demonstrating its versatility and effectiveness.

Chapter 3

Methodology

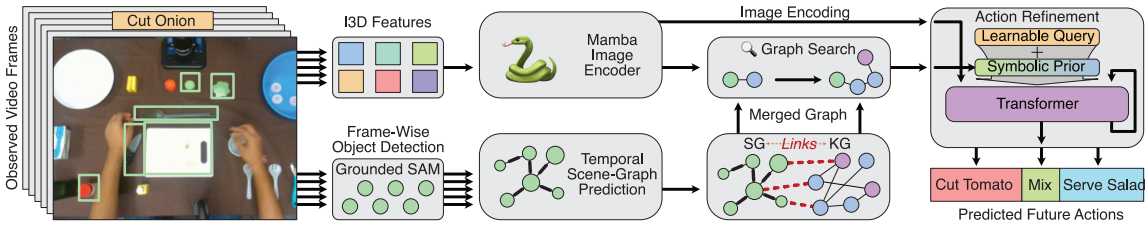

Figure 3.1: The overall architecture of our novel approach for long-term action anticipation: Utilizing a short video of prior observations, we extract a relevant scene graph and utilize a knowledge graph as additional context to predict a sequence of future actions. We propose a joint graph search to capture additional context, utilizing it as a prior for a diffusion-inspired iterative action refinement.

In this section, we introduce our method, *SymAnt*, a novel approach for action anticipation that utilizes symbolic domain knowledge as additional context to improve prediction through a diffusion-inspired action decoder. We first explain the problem of action anticipation and our overall pipeline before highlighting the components of our method as shown in overview fig. 3.1, specifically our symbolic context aggregation framework, followed by the video encoder and action decoder architectures.

3.0.1 Problem Statement

Given a set of features from an observed video (V_o), $\mathbf{F} \in \mathbb{R}^{L \times G}$ with G dimensions for L frames, the objective of the anticipation task is to predict the subsequent future

actions Z after the end of the shown video. These actions consist of action classes $\mathbf{A} \in \mathbb{R}^{Z \times C}$ and their corresponding durations $\mathbf{d} \in \mathbb{R}^{Z \times 1}$, where C represents the total number of action classes. On a high level, as shown in fig. 3.1, SymAnt utilizes a Mamba-based encoder ($\mathbf{E} = f_{enc}(\mathbf{F})$) to encode the video observations and a novel denoising action decoder $f_{dec}(\dots)$, initialized with learned prior $f_{SN}(\dots)$ from the temporal scene graph and symbolic knowledge.

3.0.2 Symbolic Context Aggregation

The key to our approach is utilizing symbolic knowledge to derive additional context information from the objects that can be seen in a short-context video observation. We achieve this by creating a *joint-graph* from a scene graph created using the video observations V_o and a fixed, external, knowledge graph $\mathbf{K} = \{\mathbf{K}_n, \mathbf{K}_A\}$ providing additional information for the detected entities, where $\mathbf{K}_n, \mathbf{K}_A$ are nodes and adjacency matrix of $\mathbf{K}$, respectively. Given an observed video sequence and a knowledge graph, which contains various concepts such as objects, their attributes, affordances, and actions, we utilize a context aggregation module that effectively integrates this information. Our context aggregation module operates in three key steps to achieve this integration, which are explained as follows:

1. **Temporal Scene Graph Generation**: Scene graphs excel at capturing spatial relationships between entities in the scene; however, for the purpose of action anticipation, we modify them to capture temporal changes as well. We start by utilizing GroundedSAM [64] on each of the observed frames V_o^l to detect initial concepts $\mathbf{c}^l$ in the scene, where l is the frame number. This forms the basis of our frame-level scene graphs $\mathbf{S}^l$. Spatial relationship between two scene entities is predicted with a neural network $f_{\text{edge}}()$, classifying whether or not two nodes have a spatial relationship, and if so, estimates the relationship type with a respective weight, indicating its importance. The set of spatial relations are: '*on*', '*next to*', '*behind*', '*in front of*', '*above*', '*across*', '*below*', '*inside*', '*under*', '*left*', '*right*', '*in*'. Concepts in the SG are represented by a representation $f_{ViT}(box)$ [21] derived as a latent representation of the entity represented in SAM's bounding box. To aggregate scene graphs $\mathbf{S}^l$ across multiple time frames and construct a temporal scene graph $\mathbf{S}$, we adopt a dynamic update mechanism. For each new frame l, any object not

Algorithm 1 Graph Merging Process

Input: Scene Graph $\mathbf{S} = \{\mathbf{S}_n, \mathbf{S}_A\}$; Knowledge Graph $\mathbf{K} = \{\mathbf{K}_n, \mathbf{K}_A\}$
Initialize: Merged Graph $\mathbf{M}\{\mathbf{M}_n, \mathbf{M}_A\} = (\mathbf{S}, \mathbf{K})$; Active Nodes $\mathbf{M}_{active} = \{\}$
for each $\mathbf{S}_n^i \in \mathbf{S}_n$ **do**
  Connect $\mathbf{S}_n^i$ and $\mathbf{K}_n^j$ in $\mathbf{M}$, where $\mathbf{K}_n^j$ is the node corresponding to $\mathbf{S}_n^i$ in $\mathbf{K}$
  $\mathbf{M}_{active} \leftarrow \mathbf{M}_{active} + \{\mathbf{S}_n^i, \mathbf{K}_n^j\}$
end for

already present in the temporal scene graph is added, along with its relationships to the existing nodes. When a relationship between two pre-existing nodes is updated, the new relationship is added to the graph while retaining the previous one. To prevent the accumulation of stale information, the importance of older relationships is decayed over time. This decay mechanism naturally diminishes the significance of outdated relationships as new observations are incorporated.

2. **Graph Merging Module**: The goal of this module is to combine scene-specific information from the temporal scene graph $\mathbf{S}$ with domain knowledge from the provided knowledge graph $\mathbf{K}$ based on the common entities present in the environment. This merging process creates a comprehensive joint-graph $\mathbf{M}$ that integrates spatial and domain-specific information, as explained in algorithm 1.

Given a temporal SG $\mathbf{S} = \{\mathbf{S}_n, \mathbf{S}_A\}$, where $\mathbf{S}_n$ and $\mathbf{S}_A$ are nodes and adjacency matrix of $\mathbf{S}$ respectively, we form connections between the graphs by linking the mutual nodes between the scene and knowledge graph to create the joint-graph $\mathbf{M} = \{\mathbf{M}_n, \mathbf{M}_A\}$, which includes the entire scene and knowledge graph. In scenarios where the scene graph contains multiple instances of the same node, all such instances are connected to the corresponding single node in the knowledge graph. The nodes that link the scene and knowledge graph are referred to as active nodes M_{active}. These nodes serve as the starting points for our joint-graph search (see the next section for details about graph search).

3. **Joint-Graph Search**: The goal of joint-graph search is to iteratively expand nodes around the currently active nodes, using both scene data and knowledge graph information related to these active nodes, providing comprehensive contextual information about the scene. The expansion continues until all relevant domain information is obtained, i.e., when no additional nodes contribute meaningful context to the current observation. We conduct the joint-graph search using a three-stage

Algorithm 2 Joint-Graph Search Network

Input: Observed Video Features: $\mathbf{F}$; Active Nodes : $\mathbf{M}_{active}$
while True **do**
  Step 1: Get all inactive neighbors of $\mathbf{M}_{active}$
  $\mathbf{M}_{nei} \leftarrow$ get_neighbors($\mathbf{M}_{active}$, $\mathbf{M}$)
  Step 2: Propagate active nodes and neighbors
  $\mathbf{M}_{nei}$, $\mathbf{M}_{active} \leftarrow f_{prop}(\mathbf{M}_{nei}, \mathbf{M}_{active}, \mathbf{M})$
  Step 3: Identify Important Neighbors
  for each $\mathbf{M}^i_{nei} \in \mathbf{M}_{nei}$ **do**
    $\mathbf{I}^i \leftarrow f_{imp}(\mathbf{M}^i_{nei}, \mathbf{F})$
    if $\mathbf{I}^i > \gamma$ **then**
      $\mathbf{M}_{active} \leftarrow \mathbf{M}_{active} + \mathbf{M}^i_{nei}$
    end if
  end for
  Step 5: Dynamic Propagation for Expansion
  $\mathbf{I}^{\max} \leftarrow \max(\mathbf{I})$
  if $\mathbf{I}^{\max} < \eta$ **then**
    Break the loop: Expansion threshold did not meet
  end if
end while
Step 6: Get the context from fully expanded graph
$\mathbf{z}_T = f_{cont}(\mathbf{M}_{active}, \mathbf{F})$

approach inspired by [5, 8, 57] (also see algorithm 2). The individual stages include 1) a propagation network, which calculates the node representations of the neighbors of currently active nodes $\mathbf{M}_{active}$ in the graph, 2) an importance network, which decides which nodes should be expanded and iteratively prunes the graph during the search, and 3) a context network which generates a context embedding from the fully expanded graph. Inspired by prior work [5], we utilize a dynamically determined number of iterations of the propagation and importance network until no additional relevant knowledge is added in a single iteration. The functions of each individual network are as follows:

- **Propagation Network**: The propagation network, $f_{prop}(\dots)$ updates the states of active nodes $\mathbf{M}_{active}$, as well as the states of neighboring nodes that are connected to these active nodes given the context of the video $\mathbf{F}$.
- **Importance Network**: The Importance Network, $f_{imp}(\dots)$ operates in al-

ternation with the Propagation Network, deciding whether adjacent nodes to currently active ones should be activated, thereby preventing exponential growth of the contextual graph.

- **Context Network**: The context network, $f_{cont}(\dots)$ generates a final embedding $\mathbf{z}_T$ using the active nodes in the fully expanded graph, which is subsequently used as context for the decoder initialization.

3.0.3 Video Encoder

In addition to the symbolic knowledge derived from the observed video, we extract features directly from the video frames using I3D [33], which employs 3D convolutions to learn rich motion and appearance representations. These input features $\mathbf{F}$ are further transformed using the Mamba [33] model as it improves training and inference speed without compromising accuracy compared to transformer-based encoders utilized in FUTR [29] and DiffAnt [82] (see ablations for details). We combine the image features with a sinusoidal positional encoding $\mathbf{P} \in \mathbb{R}^{L \times D}$, enable the Mamba encoder to generate refined representations $\mathbf{E} \in \mathbb{R}^{L \times D}$. These representations are then processed through a linear classifier to map them to the discrete actions, enabling recognition of observed actions in videos, denoted as $\tilde{\mathbf{A}}^{\text{obs}}$. Training these representations for action recognition ensures that they capture meaningful temporal and spatial structures. This representation is used as conditioning for the iterative action denoising process, as explained in the next section.

3.0.4 Iterative Symbolic Action Diffusion

We combine and utilize the complementary strengths of high-level abstract information encoded in the fully expanded joint-graph and the fine-grained visual details present in the video stream in our denoising decoder. Inspired by the reverse process of diffusion models, our approach is built on the hypothesis that the context embedding $\mathbf{z}_T$ derived from the fully expanded graph inherently contains a noisy representation of future events. Consequently, the decoder, utilizing a transformer backbone, iteratively refines this noisy representation through multiple denoising steps—from step T to step 0 as detailed in algorithm 3. This iterative refinement gradually transforms the

Algorithm 3 Iterative Symbolic Action Decoding

Input: Symbolic Context: $\mathbf{z}_T$, Encoded Observations: $\mathbf{E}$, Action Queries: $\mathbf{Q}$
Initialize: $\mathcal{L}^{\text{AR}} \leftarrow 0$; $\mathbf{z}_T \leftarrow \mathbf{z}_T + \mathbf{Q}$
for $t = T$ to 1 **do**
  Step 1: Temporal Encoding
  $\tilde{\mathbf{z}_t} \leftarrow \mathbf{z}_t + \mathbf{T}_E(t)$
  Step 2: Refinement Step
  $\mathbf{z}_{t-1} = \text{refinement\_step}(\tilde{\mathbf{z}_t}, \mathbf{E})$
  Step 3: Projection
  $\tilde{\mathbf{A}_{t-1}}, \tilde{\mathbf{d}_{t-1}} \leftarrow \text{projection\_net}(\mathbf{z}_{t-1})$
  Step 4: Calculate Intermediate anticipation Loss
  $\mathcal{L}^{\text{AR}} \leftarrow \mathcal{L}^{\text{AR}} + \lambda \mathcal{L}^{\text{ant}}(\tilde{\mathbf{A}_{t-1}}, \tilde{\mathbf{d}_{t-1}})$
end for
Step 5: Calculate Final Anticipation Loss
$\mathcal{L}^{\text{AR}} \leftarrow \mathcal{L}^{\text{AR}} + \mathcal{L}^{\text{ant}}(\tilde{\mathbf{A}}, \tilde{\mathbf{d}})$

noisy representation into coherent predictions of upcoming actions. Importantly, each refinement is conditioned on the video features, ensuring that the predictions are semantically meaningful and closely aligned with the observed visual content.

To facilitate focused reasoning for each action while maintaining their interdependencies, we employ a query-based approach [11, 29]. The action queries, denoted as $\mathbf{Q} \in \mathbb{R}^{N \times D}$, consist of N learnable tokens temporally aligned with future action sequences, with each query corresponding to a specific action [29]. Importantly, these queries are not fixed representations; rather, they are learned dynamically, allowing the model to refine their embeddings throughout training. These action queries are combined with the context embedding $\mathbf{z}_T$ from the joint graph-search to form the input for the action decoder at the initial denoising step T.

To ensure effective learning of the denoising process, we apply supervision at each refinement step. At each refinement step t, the model transforms the action embeddings $\mathbf{z}_t$ into intermediate action outputs $\tilde{\mathbf{A}}_t \in \mathbb{R}^{N \times (C)}$ and timestamp outputs $\tilde{\mathbf{d}}_t \in \mathbb{R}^{N \times 1}$ using an output projection module. This module comprises two fully connected multi-layer perceptrons, each followed by a softmax activation that generates probabilities for action classes and durations, respectively. By aligning these intermediate outputs with the ground truth, the model progressively refines its predictions rather than depending solely on the final output. This iterative supervision

is enforced through an anticipation loss as explained in the section 3.0.5. After T refinement steps, the final prediction $\mathbf{z}_0$ is processed by the same output projection module to generate the final action labels $\tilde{\mathbf{A}}$ and durations $\tilde{\mathbf{d}}$, which serve as the model's outputs.

3.0.5 Training Losses

This section details the loss functions used in end-to-end training SymAnt. We begin with the action recognition loss for observed actions, followed by the importance loss for joint-graph search. Finally, we describe the iterative refinement loss and initialization loss for the decoder.

Recognition Loss

We apply an action recognition loss to align the predicted and ground truth action sequences for the observed portion of the video. This serves a dual purpose: first, it ensures that the model learns to recognize past actions accurately; second, it enhances the quality of past video embeddings, making them more informative for future action anticipation. The recognition loss $\mathcal{L}^{\text{rec}}$ is formulated as a cross-entropy loss between the predicted action logits $\mathbf{A}^{\text{obs}}$ and ground truth action $\tilde{\mathbf{A}}^{\text{obs}}$ as:

$$\mathcal{L}^{\text{rec}} = -\sum_{i=1}^{L}\sum_{j=1}^{C} \mathbf{A}^{\text{obs}}_{i,j} \log \tilde{\mathbf{A}}^{\text{obs}}_{i,j} \tag{3.1}$$

where L is the observed video frames and C is the number of actions.

Importance Loss

The importance loss ensures that the joint-graph search network correctly identifies and activates the relevant actions and affordances. To achieve this, we compute $L2$ loss between the predicted importance scores $\tilde{\mathbf{I}}$ and the ground truth importance scores $\mathbf{I}$:

$$\mathcal{L}^{\text{imp}} = \sum_{w=1}^{W} \left(\mathbf{I}^{w} - \tilde{\mathbf{I}}^{w}\right)^2 \tag{3.2}$$

where W represents the set of active nodes along with their directly connected inactive neighbors. The ground truth importance scores are derived from the future action labels. Specifically, nodes corresponding to actions, affordances, and entities associated with the ground-truth future actions are assigned an importance score of 1, while all other nodes are assigned a score of 0. This loss guides the model to focus on the most relevant parts of the joint-graph.

Action Refinement Loss

The action refinement loss $\mathcal{L}^{\text{AR}}$ ensures both the proper functioning of the decoder's iterative refinement steps and the accuracy of the final predicted actions. It encourages the decoder to progressively refine its predictions while ensuring that the final output actions remain reliable. This loss is computed as a weighted sum of the anticipation loss (defined later) at the intermediate refinement steps and at the final output. The action refinement loss is given by:

$$\mathcal{L}^{\text{AR}} = \lambda \sum_{t=T-1}^{1} \mathcal{L}^{\text{ant}}(\tilde{\mathbf{A}}_t, \tilde{\mathbf{d}}_t) + \mathcal{L}^{\text{ant}}(\tilde{\mathbf{A}}, \tilde{\mathbf{d}}) \tag{3.3}$$

where T is the number of steps in the denoising process, and λ is the weight for the intermediate losses, taken as $1/T$.

The anticipation loss is a combination of action loss and duration loss. The action loss $\mathcal{L}^{\text{ant}}_{\text{action}}$ is defined as the cross-entropy between ground-truth actions $\mathbf{A}$, which are provided by the dataset and predicted actions $\tilde{\mathbf{A}}$, while the duration regression loss $\mathcal{L}^{\text{ant}}_{\text{duration}}$ is defined using the $L2$ loss between target durations $\mathbf{d}$ and predicted durations $\tilde{\mathbf{d}}$.

$$\mathcal{L}^{\text{ant}}_{\text{action}} = -\sum_{i=1}^{N}\sum_{j=1}^{C} \mathbf{A}_{i,j} \log(\tilde{\mathbf{A}}_{i,j}) \tag{3.4}$$

$$\mathcal{L}^{\text{ant}}_{\text{duration}} = \sum_{i=1}^{N} (\mathbf{d}_i - \tilde{\mathbf{d}}_i)^2 \tag{3.5}$$

Initialization Loss:

The initialization loss ensures that the symbolic initialization is correctly interpreted by the neural decoder and that this information is preserved throughout the iterative denoising process. Specifically, it enforces that action nodes that are active in the fully expanded joint-graph remain present in the final predictions. The initialization loss is formulated as cross-entropy between active action nodes in the expanded joint-graph and the predicted action logits $\tilde{\mathbf{A}}$:

$$\mathcal{L}^{\text{init}} = -\sum_{i=1}^{N}\sum_{j=1}^{C} \mathbb{1}_{j \in C_K} \log(\tilde{\mathbf{A}}_{i,j}) \tag{3.6}$$

where $\mathbb{1}_{j \in \mathbf{C}_K}$ is an indicator function that is 1 if $j \in \mathbf{C}_K$ and $\mathbf{C}_K$ is a subset actions that are active in the joint-graph after expansion.

Total Loss

The total training loss $\mathcal{L}_{\text{total}}$ for SymAnt is a weighted summation of all individual loss components described above. These include the recognition loss, importance loss, action refinement loss (which itself includes anticipation and duration losses), and the initialization loss. Formally, it is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}^{\text{rec}} + \gamma\mathcal{L}^{\text{imp}} + \mathcal{L}^{\text{AR}} + \delta\mathcal{L}^{\text{init}} \tag{3.7}$$

where γ, and δ are scalar weights that control the contribution of importance and initialization loss terms during training which are typically taken 0.1 and 0.01 respectively.

Chapter 4

Experiments

In this section, we evaluate our approach on four widely used action anticipation datasets: 50 Salads [70], Breakfast [46], EpicKitchens [19], and EGTEA Gaze+ [54], and compare the results with neuro-symbolic and neural-only baselines, including large language model baselines. Additionally, we present qualitative results on a custom real-world kitchen setup and conduct ablation studies to analyze the contributions of different components of our neuro-symbolic approach, demonstrating its efficiency and performance for action anticipation from short contexts over long prediction horizons.

4.0.1 Datasets

Breakfast [46] and **50 Salads** [70] are fixed camera datasets. The Breakfast [46] dataset consists of 1,712 videos featuring 52 different individuals making breakfast in 18 different kitchens, totaling 77 hours of footage. Each video is categorized into one of 10 activities related to breakfast preparation and annotated with 48 fine-grained actions. The 50 Salads [70] dataset includes 50 top-view videos of 25 people preparing a salad, containing over 4 hours of RGB-D video data annotated with 17 fine-grained action labels and 3 high-level activities. **EpicKitchens** [19] and **EGTEA Gaze+** [54] are egocentric datasets. EpicKitchens comprises 39,596 segments labeled with 125 verbs, 352 nouns, and 2,513 verb-noun combinations (actions), totaling 55 hours of video. EGTEA Gaze+ contains 28 hours of videos with

10.3K action annotations, including 19 verbs, 51 nouns, and 106 unique actions. We also utilize a **real-world mini-kitchen** setup from [9] for fine-tuning our model, which contains 20 videos recorded in a controlled kitchen environment.

4.0.2 Training and Architecture Details

We configure the hidden dimension D to 128 for the Breakfast dataset and 512 for all other datasets. The number of action queries N is set to 8 for Breakfast and 20 for 50 Salads. For EpicKitchens and EGTEA Gaze+, even though the task involves multi-label action classification [60], we still predict sequential actions, setting $N = 50$. To ensure a fair comparison with state-of-the-art long-term action anticipation methods, we use pre-extracted I3D features [12] as input visual features F for all datasets, provided by [23] and [60]. We sample the I3D features with a stride of 3 for the Breakfast and 50 Salads datasets, and with a stride of 1 for EpicKitchens and EGTEA Gaze+. During training, the observation rate α is set to $\{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$ for Breakfast and 50 Salads, and additionally to $\{0.6, 0.7, 0.8\}$ for EpicKitchens and EGTEA. We used the Adam optimizer and trained for 70 epochs for 50 Salads and 50 epochs for the other datasets. The total number of denoising steps is set to $T = 10$, and these configurations were optimized via hyperparameter tuning. The average training time across all four datasets is approximately 5 hours on an Nvidia RTX 6000 GPU, with an inference time of around 16.5 ms per video (using a pre-computed scene graph). Despite incorporating a graph-based symbolic network alongside traditional neural networks, training and inference times remain largely unchanged, as we generate only a single joint graph for each input video. During training, GPU utilization is approximately 18 GB.

4.0.3 Evaluation Metrics

Mean Accuracy over Classes (MoC): We evaluate performance on the Breakfast and 50 Salads datasets using the mean accuracy over classes (MoC). For a given anticipation duration, MoC is computed as the average accuracy per class over all future timestamps. Building on prior work [2, 29, 82], we first observe an initial portion of the video, denoted by α (typically 0.2 or 0.3). To evaluate short-context action anticipation, we also include $\alpha = 0.05$ and 0.1, following [9]. Anticipation

Table 4.1: Comparison of long-term anticipation performance with short observation horizons ($\alpha = 0.05, 0.1$) on the Breakfast dataset. The table shows results for different values of β.

	Breakfast β ($\alpha = 0.05$)				**Breakfast** β ($\alpha = 0.1$)			
Method	0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
KG[8]	5.44	4.95	4.22	3.98	6.02	5.15	4.86	4.51
CNN[2]	5.76	5.52	5.45	4.80	7.84	6.62	6.02	5.17
RNN[2]	6.16	5.60	5.53	4.96	7.67	6.73	6.15	5.22
Deepseek-r1[35]	5.34	8.25	5.00	6.23	7.41	10.20	7.06	8.09
FUTR[29]	9.54	7.24	6.42	5.58	14.70	12.55	12.10	11.71
ChatGPT o1[62]	7.61	6.67	7.89	5.15	-	-	-	-
NeSCA[9]	9.91	7.95	6.86	5.88	15.53	13.52	13.07	11.94
Ours	**12.19**	**9.80**	**8.91**	**8.15**	**17.11**	**14.32**	**14.06**	**12.92**

begins immediately after the observed frames, but the predicted segment length is determined by $\beta \in \{0.1, 0.2, 0.3, 0.5\}$, which is defined as a fraction of the entire video—not just the unobserved portion. We evaluate performance across 4 splits for the Breakfast dataset and 5 splits for the 50 Salads dataset.

Mean Average Precision (mAP): For the EpicKitchens and EGTEA Gaze+ datasets, we evaluate performance using mean average precision (mAP) following [60], which is a multi-label classification metric that quantifies the accuracy of predicting specific action classes. In our approach, an initial portion of each untrimmed video, α is used as input to forecast all subsequent action classes, which occur over the remaining $(1 - \alpha)$ duration of the video. Consistent with [60], we set $\alpha = \{0.25, 0.50, 0.75\}$ during evaluation and report mAP scores separately for low-shot (rare) and many-shot (frequent) scenarios.

4.0.4 Comparison to the State-of-the-art

We present our experimental results for long-term action anticipation on the 50 Salads and Breakfast datasets in tables 4.1 to 4.4. In tables 4.1 and 4.2, we evaluate the model's anticipation performance using short observation contexts ($\alpha = 0.05, 0.1$). We compare SymAnt with neuro-symbolic methods (e.g., NeSCA [9]), neural-only approaches [2, 29, 82], a purely symbolic KG-Baseline [9] and state-of-the-art large language models (LLMs), including OpenAI's ChatGPT o1 [62] and Deepseek-r1 [35].

Table 4.2: Comparison of long-term anticipation performance with short observation horizons ($\alpha = 0.05, 0.1$) on the 50 Salads dataset. The table shows results for different values of β.

	50 Salads β ($\alpha = 0.05$)				**50 Salads** β ($\alpha = 0.1$)			
Method	0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
KG[8]	6.92	6.21	6.01	5.58	7.13	6.48	6.07	5.78
CNN[2]	7.42	6.97	6.67	6.40	8.50	7.80	7.45	6.92
RNN[2]	7.98	6.90	6.48	6.42	8.78	7.92	7.57	7.26
Deepseek-r1[35]	6.67	5.36	4.04	2.80	10.20	6.92	4.95	3.30
FUTR[29]	8.90	7.46	7.29	8.63	15.17	11.34	11.31	11.36
ChatGPT o1[62]	15.64	13.12	13.38	11.18	15.18	15.07	13.47	12.29
NeSCA[9]	<u>17.86</u>	<u>16.25</u>	<u>10.84</u>	<u>9.38</u>	<u>23.15</u>	<u>17.28</u>	<u>16.62</u>	<u>13.96</u>
Ours	**21.98**	**19.41**	**17.75**	**14.33**	**27.92**	**23.82**	**20.01**	**18.83**

Table 4.3: Long-term anticipation results on the Breakfast dataset for observation horizons $\alpha = 0.2, 0.3$. SymAnt consistently outperforms both non-symbolic and symbolic baselines.

	Breakfast β ($\alpha = 0.2$)				**Breakfast** β ($\alpha = 0.3$)			
Method	0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
Deepseek-r1[35]	15.77	11.94	10.61	6.60	16.40	13.13	9.87	6.08
CNN[2]	17.90	16.35	15.37	14.54	22.44	20.12	19.69	18.76
RNN[2]	18.11	17.20	15.94	15.81	21.64	20.02	19.73	19.21
UAAA[1]	16.71	15.40	14.47	14.20	20.73	18.27	18.42	16.86
TempAgg[68]	24.20	21.10	20.00	18.10	30.40	26.40	23.80	21.20
Timecond.[45]	18.41	17.21	16.42	15.84	22.75	20.44	19.64	19.75
Cycle Cons[3]	25.88	23.42	22.42	21.54	29.66	27.33	25.58	25.20
A-ACT[37]	26.70	24.30	23.20	21.70	30.80	28.30	26.10	25.80
FUTR[29]	27.70	24.55	22.83	22.04	32.27	29.88	27.49	25.87
ActFusion[30]	**28.25**	<u>25.52</u>	<u>24.66</u>	<u>23.25</u>	<u>35.79</u>	31.76	29.64	28.78
Diffant[82]	25.33	24.59	24.39	22.74	32.13	<u>31.83</u>	<u>31.18</u>	**30.77**
Ours	<u>28.22</u>	**27.55**	**27.02**	**25.67**	**34.22**	**32.06**	**31.19**	<u>30.11</u>

Table 4.4: Long-term anticipation results on the 50 Salads dataset for observation horizons $\alpha = 0.2, 0.3$. SymAnt achieves strong performance improvements across all prediction lengths.

	50 Salads β ($\alpha = 0.2$)				**50 Salads** β ($\alpha = 0.3$)			
Method	0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
Deepseek-r1[35]	7.83	5.87	3.76	2.30	6.00	4.99	3.56	2.39
CNN[2]	21.24	19.03	15.98	9.87	29.14	20.14	17.46	10.86
RNN[2]	30.06	25.43	18.74	13.49	30.77	17.19	14.79	9.77
ChatGPT o1[62]	21.60	21.75	17.86	12.93	28.38	25.08	20.44	15.99
UAAA[1]	24.86	22.37	19.88	12.82	29.10	20.50	15.28	12.31
TempAgg[68]	25.50	19.90	18.20	15.10	30.60	22.50	19.10	11.20
Timecond.[45]	32.51	27.61	21.26	15.99	35.12	27.05	22.05	15.59
Cycle Cons[3]	34.76	28.41	21.82	15.25	34.39	23.70	18.95	15.89
A-ACT[37]	35.40	29.60	22.50	16.10	35.70	25.30	20.10	16.30
FUTR[29]	<u>39.55</u>	27.54	23.31	17.77	35.15	24.86	24.22	15.26
Obj.Prompt[77]	37.40	28.90	24.20	18.10	28.00	24.00	24.30	19.30
ActFusion[30]	<u>39.55</u>	28.60	23.61	19.90	**42.80**	27.11	23.48	22.07
Diffant[82]	36.13	<u>34.00</u>	<u>30.46</u>	<u>25.29</u>	34.09	<u>30.14</u>	<u>26.34</u>	<u>20.23</u>
Ours	**40.78**	**35.08**	**31.19**	**28.19**	<u>35.23</u>	**30.94**	**29.34**	**25.50**

Specifically, for ChatGPT-o1—which can ingest any number of images before making predictions—we sampled images from observed videos (based on the observation ratio α) and prompted the model to predict subsequent future actions along with their relative durations. We then compared the predicted actions with the ground truth using a prediction ratio β, similar to our model's evaluation. For Deepseek-r1, since its API does not accept images, we prompted it with the ground truth observed actions while keeping the remaining prompt and evaluation the same. Due to budget constraints, we evaluated ChatGPT-o1 only using the 0.05 observation split of the Breakfast; this split was chosen because it yields results closest to those on the 50 Salads dataset when compared to our model. SymAnt outperforms all baselines, surpassing the closest one by 2–3% and LLM baselines by at least 5% MoC accuracy.

In tables 4.3 and 4.4, we report anticipation performance with increased observation context ($\alpha = 0.2, 0.3$), which is commonly used in prior work [29, 30, 82]. Our comparison includes methods that utilize action labels from observed video [1, 2, 45] as well as approaches that rely solely on latent features such as I3D [3, 29, 37, 68, 77]. We also compare against the LLM baselines [35, 62] as well as recent diffusion-based action anticipation methods [30, 82], which are most similar to our approach but

Method	EPIC Kitchens			EGTEA GAZE+		
	All	Freq.	Rare	All	Freq.	Rare
I3d[12]	37.7	53.5	23.0	72.1	79.3	53.3
ActionVLAD[28]	29.2	53.5	18.6	73.3	79.0	58.6
TimeCeption[41]	35.6	55.9	26.1	74.1	79.7	59.7
VideoGraph[42]	22.5	49.4	14.0	67.7	77.1	47.2
EGO-TOPO[60]	38.0	56.9	29.2	73.5	80.7	54.7
ANTICIPATR[61]	39.1	58.1	29.1	76.8	83.3	55.1
Diffant[82]	38.1	55.0	31.0	77.3	83.5	61.4
Ours	**42.47**	**59.32**	**32.27**	**78.66**	**84.96**	**65.51**

Table 4.5: Performance comparison of our method on the EPIC Kitchens and EGTEA GAZE+ datasets. We use the mAP metric to evaluate performance across all dataset splits, including rare and frequent action types.

do not incorporate symbolic knowledge. Note that we exclude ANTICIPATR [61] from these experiments due to its different evaluation protocol [80]. The results indicate that SymAnt outperforms LLM baselines [35, 62] by an average of 15%, non-symbolic and non-diffusion-based baselines [29, 37] by an average of 4–5% and achieves a 2–3% MoC accuracy improvement over recent diffusion-based non-symbolic baselines [30, 82]. These results, particularly the higher margin of improvement for longer prediction horizons, show the effectiveness of our joint-graph approach in encoding long-term contextual information and ultimately validate the efficacy of our method.

We further evaluate our approach on the EpicKitchens and EGTEA Gaze+ datasets, as summarized in table 4.5. We compare our method against conventional methods [12, 28, 41, 42, 60], transformer-based techniques such as ANTICIPATR [61], and recent diffusion-based approach Diffant [82]. Consistent with prior work [60, 82], we report results for all, rare, and frequent splits. Our method surpasses all baselines, achieving an average improvement of 3 mAP score on EPIC Kitchens and 2 mAP score on EGTEA GAZE+. This demonstrates its superior ability to capture and anticipate even the rarest action categories.

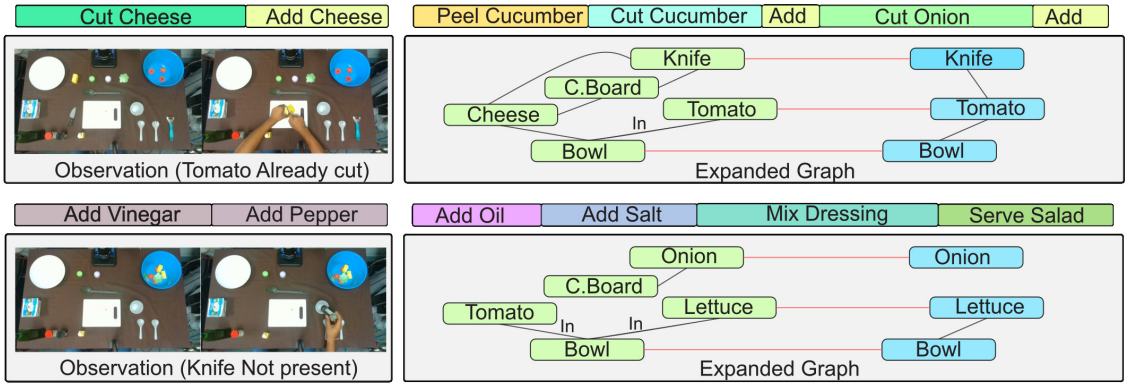

Figure 4.1: Qualitative evaluation of scene and knowledge graph integration in real-world kitchen scenario. Observed video and actions on the left, with predicted actions on the right, demonstrating the expanded graphs (SG in green, KG in blue). Only the relevant portions of the joint graph are shown.

4.0.5 Qualitative Analysis on Real-world Videos

We conducted a qualitative analysis to demonstrate the effectiveness of incorporating symbolic context from scene and knowledge graphs, particularly in handling challenging real-world corner cases. We captured real-world overhead videos in our dummy kitchen setup, focusing on scenarios where small but crucial details, such as the presence or absence of a knife, were altered. To adapt our model to this real-world mini-kitchen setup, we first fine-tuned it on the general salad-making activities dataset from [9], recorded in a similar kitchen environment. We then evaluated our model on these challenging cases by observing 10% of each video and successfully predicting the subsequent 50% of the content. Figure 4.1 illustrates two such cases, along with their corresponding expanded joint-graphs (showing only the relevant parts) and the predicted actions. In the first example, a tomato is already cut and placed in a bowl when the observation is made. The scene graph explicitly captures the "in" relation between the tomato and the bowl, allowing our model to recognize that no further tomato-related actions are necessary – a crucial detail that non-symbolic, neural models, which rely solely on visual features, might overlook. In the second example, the scene graph correctly identifies the absence of a knife, and the knowledge graph further reasons about it, ensuring that no cutting action is predicted. Unlike neural models that might overlook such minor details, our method explicitly reasons about

			Breakfast				**50 Salads**			
			$\beta(\alpha = 0.2)$		$\beta(\alpha = 0.3)$		$\beta(\alpha = 0.2)$		$\beta(\alpha = 0.3)$	
Enc.	Dec.	KG	0.1	0.5	0.1	0.5	0.1	0.5	0.1	0.5
T	T	-	27.03	20.82	32.77	24.89	36.77	20.12	31.43	18.61
M	T	-	26.70	20.25	31.50	23.76	36.20	21.60	32.81	17.84
M	R	-	27.20	23.60	32.25	28.58	37.20	27.70	34.76	20.13
-	R	I	17.81	11.86	22.55	18.31	25.20	12.41	23.65	16.54
M	D	C	27.50	24.08	32.58	29.92	38.70	27.44	34.82	23.67
M	R	C	27.10	25.41	33.70	29.42	39.58	27.61	34.57	23.93
M	R	I	**28.22**	**25.67**	**34.22**	**30.11**	**40.78**	**28.19**	**35.23**	**25.50**

Table 4.6: Comparison of anticipation accuracy across different model configurations for Breakfast and 50 Salads dataset/. Enc. (Encoder): M (Mamba), T (Transformer), – (None). Dec. (Decoder): R (Iterative Refinement), T (Transformer), D (Diffusion model with explicit forward process [30, 82]). KG (Knowledge Graph Integration): I (Initialization), C (Conditioning), – (None).

individual concepts in the scene separately. Overall, these examples demonstrate how our explicit symbolic reasoning significantly enhances the model's ability to capture and act upon subtle yet crucial scene details, improving its overall robustness in real-world scenarios.

4.0.6 Ablation Studies

In the following section, we ablate various components of our method to assess their impact. Specifically, we evaluate the effectiveness of the Mamba encoder, the contributions of symbolic domain knowledge, and the role of our iterative refinement architecture for the decoder. Additionally, we present a quantitative analysis of symbolic initialization on neural predictions, examine the effect of different loss terms on anticipation accuracy, and analyze the effect of refinement steps.

Encoder, Decoder and Knowledge Integration

We conduct an ablation study to compare different choices for the encoder, decoder, and symbolic knowledge integration, and assess their impact on action anticipation performance. Table 4.6 presents results on the Breakfast and 50 Salads datasets, omitting results for ($\beta = 0.2, 0.3$) as they follow a uniform trend across different

			EPIC Kitchens			EGTEA Gaze+		
Enc.	Dec.	KG	All	Frequent	Rare	All	Frequent	Rare
T	T	-	37.9	55.3	29.3	72.1	75.4	59.2
M	T	-	38.6	56.1	29.7	72.1	76.8	59.4
M	D	C	38.5	55.3	29.8	73.6	78.4	61.6
M	R	-	<u>39.7</u>	<u>56.4</u>	<u>31.8</u>	<u>74.4</u>	<u>79.7</u>	<u>63.8</u>
M	R	I	**42.4**	**59.3**	**32.2**	**78.7**	**84.9**	**65.5**

Table 4.7: Comparison of anticipation accuracy across different model configurations for EPIC Kitchens and EGTEA Gaze+ dataset. Enc. (Encoder): M (Mamba), T (Transformer). Dec. (Decoder): R (Iterative Refinement), T (Transformer), D (Diffusion model with explicit forward process [30, 82]). KG (Knowledge Graph Integration): I (Initialization), C (Conditioning), – (None).

values of β. Rows 1 and 2 indicate no statistically significant difference between the Mamba (M) and Transformer (T) encoders, suggesting that Mamba can serve as a drop-in replacement for Transformer-based video encoding. Comparing rows 2 and 3 demonstrates that incorporating an iterative denoising decoder (Refinement, R) significantly enhances long-horizon action anticipation, particularly for larger β values. Additionally, rows 3 and 7 highlight the impact of knowledge-guided initialization (I), showing a 2–3% performance improvement on average across all observation and prediction settings. Rows 4 and 7 compare the impact of symbolic initialization without video feature conditioning. The lower performance in row 4 underscores the importance of low-level visual features for accurate predictions. Furthermore, rows 5 and 7 reveal that a traditional diffusion-based decoder Diffusion(D), which follows a conventional forward-reverse denoising process, significantly underperforms compared to our iterative refinement approach. Finally, rows 6 and 7 show that conditioning the decoder on symbolic knowledge (C) instead of using it for initialization results in a significant accuracy drop, emphasizing the effectiveness of our design choice. Similar trends are observed for the EPIC Kitchens and EGTEA Gaze+ datasets and are shown in table 4.7. Rows 1 and 2 indicate no statistically significant difference between the Mamba (M) and Transformer (T) encoders. However, comparing rows 2 and 4 reveals that incorporating an iterative denoising decoder (Refinement, R) substantially improves mAP. Additionally, rows 4 and 5 highlight the impact of knowledge-guided initialization (I), yielding an average mAP increase of 4 across

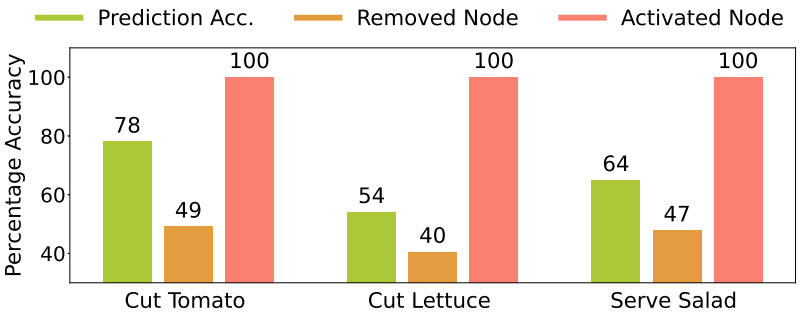

Figure 4.2: Quantitative analysis of anticipation accuracy when we manually alter the expanded graph. The results show a drop in accuracy when nodes are explicitly deactivated and a 100% accuracy when nodes are explicitly activated, highlighting the impact of symbolic context on action anticipation.

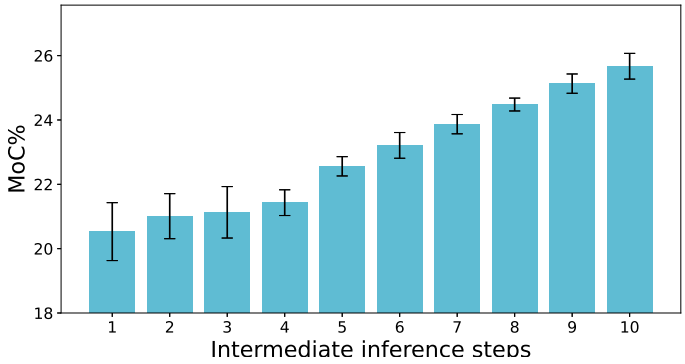

Figure 4.3: Analysis of intermediate denoising steps. We show mean and standard deviation accuracy over intermediate inference steps for a model trained for 10 denoising steps on Breakfast dataset.

all splits. Furthermore, rows 3 and 5 demonstrate that a conventional diffusion-based decoder (Diffusion, D), which follows a standard forward-reverse denoising process, significantly underperforms compared to our iterative refinement approach. This demonstrates that symbolic knowledge enhances performance over neural-only approaches, and further underscores the benefits of our iterative denoising process.

Quantitative Impact of Symbolic Knowledge

To quantitatively assess the impact of symbolic initialization on our model's final neural predictions, we designed an experiment to evaluate how each activated action

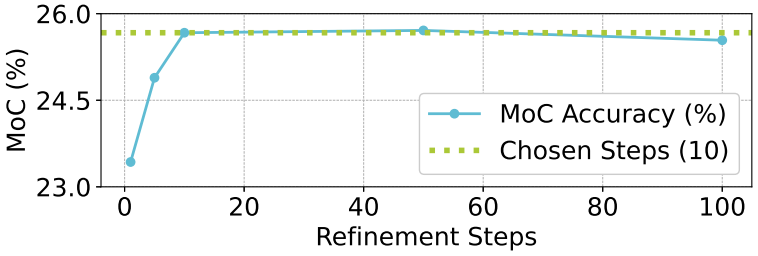

Figure 4.4: MoC accuracy vs. number of steps in the refinement process on the Breakfast dataset, showing that accuracy at 50 and 100 steps does not significantly improve compared to 10 steps.

node in the expanded graph influences the model's output. Specifically, we manually altered (activated or deactivated) selected action nodes in the graph and examined the resulting changes in neural predictions. This experiment was performed on the 50 Salads dataset, focusing on the actions *Cut Tomato*, *Cut Lettuce*, and *Serve Salads*. For each target action, we generated a filtered subset of the dataset in which the action is guaranteed to appear in the ground-truth anticipation. Anticipation accuracy for a target action on its respective subset is defined as the percentage of samples in which the action appears in any of the model's N predicted actions. First, we measured the baseline accuracy—i.e., the model's standard behavior without any manual graph modifications. Next, we conducted two separate experiments: one in which we manually deactivated the target node in the expanded graph (regardless of its initial status) and another in which we manually activated the target node irrespective of its original state. By comparing the anticipation accuracy under these conditions, we evaluated the influence of symbolic initialization on the final predictions. The results, shown in fig. 4.2, reveal significant accuracy variations corresponding to different graph modifications. Notably, removing the target node leads to a substantial drop in accuracy, whereas enforcing its activation increases the accuracy to 100%. These findings underscore the critical role of graph context and initialization in shaping the model's predictions, thereby highlighting the importance of symbolic context during inference.

Analysis of Refinement Steps

To analyze the impact of the number of refinement steps on model accuracy, we evaluated performance using different numbers of steps in the denoising process, as shown in fig. 4.4. In this experiment, we used Breakfast dataset with an observation rate of 0.2% and a prediction rate of 0.5%. Since the accuracy achieved at 50 and 100 steps was not statistically better than the mean accuracy obtained at 10 steps, we selected 10 as the ideal number of steps to balance runtime and accuracy. Furthermore, fig. 4.3 presents the accuracy at various intermediate steps for a model trained with 10 refinement steps on the Breakfast dataset. For each split, we trained ten models and reported the overall mean and standard deviation of the accuracy at each inference step. Our observations reveal that the accuracy starts low, increases to a peak as more refinement steps are applied, and is accompanied by a decreasing standard deviation. This behavior indicates that our model is effectively capable of iteratively refining its anticipated actions.

Impact of Different Loss Terms

To evaluate the impact of different loss terms on the model's performance, we conducted an ablation study on various loss components using the Breakfast and 50 Salads datasets, as shown in table 4.8. The results reveal that recognition loss significantly enhances accuracy, highlighting its importance in the model's performance. While initialization loss does not lead to substantial improvements in accuracy, both qualitative and quantitative analyses of symbolic knowledge's influence on the model's predictions suggest that it plays a critical role in maintaining proper alignment between the neural and symbolic components. This alignment, as discussed in the thesis, ultimately improves the overall observability and coherence of the model's predictions.

4.0.7 Parameter Efficiency

In this section, we discuss the efficiency of our model with respect to its parameter size, which is an important metric when deploying models on various, potentially limited hardware. Table 4.9 summarizes the results on the Breakfast dataset for

			Breakfast				**50 Salads**			
			$\beta(\alpha = 0.2)$		$\beta(\alpha = 0.3)$		$\beta(\alpha = 0.2)$		$\beta(\alpha = 0.3)$	
AR	Rec.	Init.	0.1	0.5	0.1	0.5	0.1	0.5	0.1	0.5
✓			23.75	19.21	26.64	23.28	32.19	22.05	30.61	17.52
✓		✓	24.02	19.44	26.21	23.75	32.78	22.70	31.74	17.26
✓	✓		28.07	**25.81**	34.12	29.51	40.69	27.98	**35.51**	25.31
✓	✓	✓	**28.22**	25.67	**34.22**	**30.11**	**40.78**	**28.19**	35.23	**25.50**

Table 4.8: Results for different loss terms on the Breakfast and 50 Salads datasets, highlighting the significant impact of the recognition loss on model performance.

		$\beta(\alpha = 0.2)$				$\beta(\alpha = 0.3)$			
Method	Param	0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
Ours	400k	25.27	23.88	23.49	22.87	30.41	28.60	26.54	25.00
FUTR	1M	27.70	24.55	22.83	22.04	32.27	29.88	27.49	25.87
Ours	800k	26.41	24.54	23.32	21.58	31.10	30.61	29.58	27.63
Diffant	80M	25.33	24.59	24.39	22.74	32.13	31.83	31.18	30.70
Ours	1.2M	27.83	27.54	25.91	25.05	32.54	31.56	31.06	29.78
Ours	3M	**28.22**	**27.55**	**27.02**	**25.67**	**34.22**	**32.06**	**31.21**	**30.11**

Table 4.9: Performance comparison of different variants of our model with varying parameters, alongside closest baselines on the Breakfast dataset. Our model demonstrates comparable performance to larger models, even with fewer parameters, highlighting its efficiency.

different values of α and β. Our models with 3M and 1.2M parameters consistently outperform all baselines with a 3% and 2% average performance improvement over the best baseline (with 80M parameters). The 800k model demonstrates competitive results, on par with state-of-the-art methods, only demonstrating a 1% performance reduction. Notably, even our 400k model, despite having 200 times fewer parameters than the baseline, only drops by about 1.5% in performance. These results underline the efficiency of neuro-symbolic approaches by leveraging symbolic knowledge as part of their inference pipeline.

4.0.8 Relaxed Evaluation and Improved Training

In standard Mean over Class (MoC) accuracy evaluations, we perform a one-to-one matching between the ground truth future actions and the predicted future

	Breakfast $\beta(\alpha = 0.2)$				**50 Salads** $\beta(\alpha = 0.3)$			
Model	0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
FUTR	28.19	24.29	23.90	24.10	45.11	46.78	42.88	45.43
Ours (G)	**35.63**	34.47	**30.88**	**34.38**	52.11	51.35	46.67	45.16
Ours (I)	32.87	**34.76**	28.97	32.01	**75.57**	**72.81**	**73.31**	**70.56**

Table 4.10: Performance comparison under a relaxed evaluation criterion, highlighting the impact of our improved training strategy for modeling action interdependencies. G : General Training, I : Improved Training.

actions, training the model to replicate the exact future sequence. However, this strict approach may not be ideal for real-world tasks, where the precise order of actions is not critical. Instead, the model should learn a generalized distribution allowing flexibility in the order agnostic tasks while ensuring that crucial dependencies are maintained. For instance, in a salad-making task, while serving the salad must always occur after all preparation steps, the model can predict cutting actions in any sequence. To accommodate this, we introduce a relaxed evaluation criterion based on the concept of temporal constraints. For each action, we define a set of subsequent actions that are expected to occur only after it. For instance, the action "Cut Tomato" is typically followed by "Put Tomato in Bowl," "Mix Ingredients," and "Serve Salad." While evaluation, a predicted action is considered correct if it satisfies two conditions: first, it appears before all its associated temporal constraints actions in the predicted sequence, and second, it is present in the ground truth sequence. This evaluation ensures the logical correctness of the temporal order of actions while allowing flexibility when the precise sequence is not essential.

Following our relaxed evaluation criterion, we designed a training paradigm that better captures the interdependencies between actions (accounting for both order-sensitive and order-agnostic relationships) instead of simply predicting a fixed action sequence. Although our approach already excels at modeling these dependencies, we further refine our training methodology, particularly in how training labels are constructed, to enhance the model's understanding of these relationships. Specifically, we modify the original ground truth video labels to reflect the principle that while individual tasks within a subtask may require strict temporal order, the overall sequence of subtasks can be order-agnostic. To achieve this, we first categorize actions

into three groups based on their temporal constraints: initial actions (those with a large number temporal constraints), terminal actions (those with minimal temporal constraints), and intermediate actions (all remaining tasks). Next, we compute transition probabilities for each action using training data, representing the likelihood of one action following another. Finally, using the ground truth sequences, transition probabilities, and the defined action hierarchy, we generate multiple anticipation labels for each video. In these labels, the order of actions within a subtask—after the initial action—is determined probabilistically based on the transition probabilities, while the order of initial actions across subtasks is randomized. This training paradigm enables the model to learn both the structured dependencies within subtasks and the flexibility across subtasks, ultimately leading to more accurate and context-aware action anticipation.

Table 4.10 presents the results with our relaxed evaluation on the 50 Salads and Breakfast datasets. Rows 1 and 2 compare the performance of our model without the improved training against FUTR, demonstrating that incorporating symbolic knowledge, along with iterative refinement, enhances the prediction of long-term action possibilities and better captures the interdependencies between actions, resulting in an average improvement of 6%. Rows 2 and 3 compare the performance of our method with and without the improved training. For the Breakfast dataset, no significant improvement is observed with the improved training, while for the 50 Salads dataset, there is an average performance boost of approximately 20%. This improvement is primarily attributable to the presence of multiple interchangeable subtasks in the 50 Salads dataset, an aspect not observed in the Breakfast dataset. Overall, these results show that our improved training strategy is particularly effective in tasks with interchangeable subtasks within higher-level tasks.

Chapter 5

Limitations and Future Work

While our method shows promising results for action anticipation, particularly due to the use of symbolic context, our method requires the existence of a sufficiently rich knowledge graph. While extensive knowledge graphs are available online [73] or can be generated from recent advances in large language models, we created our graph specifically for the cooking sub-task across our four major datasets. In particular, we want to avoid activating irrelevant nodes, such as those related to driving, when operating in kitchen-based environments without having to learn that such nodes would not be helpful overall. However, it's worth noting that since we use the same knowledge graph across all datasets and benchmarks, the effort required to generate it is relatively small. Furthermore, our dynamic propagation method is capable of explicitly learning which nodes are useful, making it robust to non-perfect knowledge graphs. In the future, we plan to explore the automatic knowledge graph generation pipeline either through lifelong learning or extraction from large foundational models. Another limitation is the potential leakage of information from the neural pipeline. As demonstrated in fig. 4.2, even when certain action nodes are removed from the graph, the model still predicts the corresponding actions, indicating some level of information leakage bypassing the symbolic components. While we demonstrate that changing the graph still has a significant influence on the predicted action sequence, future work will focus on disentangling the information encoded in the symbolic knowledge with the information flowing freely through the graph.

Chapter 6

Conclusion

In this work, we introduce SymAnt, a neuro-symbolic framework that merges scene graphs with an external knowledge graph through a joint-graph search and refines this symbolic prior using a diffusion-inspired iterative decoder. By allowing the decoder to start from a semantically meaningful prior instead of random noise, SymAnt achieves unified symbolic-neural reasoning—the first approach of its kind for action anticipation. Across the Breakfast, 50 Salads, EPIC-Kitchens, and EGTEA Gaze+ datasets, it improves accuracy by up to five percentage points over the best purely neural baselines while reducing model parameters by approximately 98%. Ablation studies confirm that both symbolic initialization and iterative refinement are essential to these gains, and models as small as 400 k parameters remain competitive. Even when only 5–10% of a video is observed, the symbolic prior steers the model toward reliable predictions, outperforming purely neural and large-language-model baselines and enabling practical on-device inference for mobile robots and drones. SymAnt still depends on a well-curated domain knowledge graph, making graph construction a non-trivial upfront cost, and a small amount of semantic information can leak through the neural pathway when corresponding graph nodes are suppressed, suggesting the need for tighter disentanglement. Future work will explore automatic, continual knowledge-graph construction and stronger gating mechanisms to keep neural and symbolic signals aligned, broadening the applicability of neuro-symbolic anticipation across diverse environments. In summary, this thesis demonstrates that coupling structured knowledge with compact neural encoders yields an action-anticipation system that

is accurate, interpretable, and resource-efficient, offering a practical blueprint and a motivation for re-introducing symbolic reasoning into modern, data-driven perception pipelines.

Bibliography

[1] Yazan Abu Farha and Juergen Gall. Uncertainty-aware anticipation of activities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 4.3, 4.4, 4.0.4

[2] Yazan Abu Farha, Alexander Richard, and Juergen Gall. When will you do what?-anticipating temporal occurrences of activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5343–5352, 2018. 2.0.3, 4.0.3, 4.1, 4.0.4, 4.2, 4.3, 4.4, 4.0.4

[3] Yazan Abu Farha, Qiuhong Ke, Bernt Schiele, and Juergen Gall. Long-term anticipation of activities with cycle consistency. In *Pattern Recognition: 42nd DAGM German Conference, DAGM GCPR 2020, Tübingen, Germany, September 28–October 1, 2020, Proceedings 42*, pages 159–173. Springer, 2021. 2.0.3, 4.3, 4.4, 4.0.4

[4] Houda Alberts, Teresa Huang, Yash Deshpande, Yibo Liu, Kyunghyun Cho, Clara Vania, and Iacer Calixto. Visualsem: a high-quality knowledge graph for vision and language. *arXiv preprint arXiv:2008.09150*, 2020. 2.0.1

[5] FNU Aryan, Simon Stepputtis, Sarthak Bhagat, Joseph Campbell, Kwonjoon Lee, Hossein Nourkhiz Mahjoub, and Katia Sycara. Symbolic graph inference for compound scene understanding. In *International Workshop on Ontologies and Standards for Robotics and Automation (WOSRA 2024)*, 2024. 3.0.2

[6] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021. 1, 2.0.2

[7] Ershad Banijamali, Mohsen Rohani, Elmira Amirloo, Jun Luo, and Pascal Poupart. Prediction by anticipation: An action-conditional prediction method based on interaction learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15601–15610, 2021. 1

[8] Sarthak Bhagat, Simon Stepputtis, Joseph Campbell, and Katia Sycara. Sample-efficient learning of novel visual concepts. In *Conference on Lifelong Learning*

Agents, pages 637–657. PMLR, 2023. 2.0.1, 3.0.2, 4.1, 4.2

[9] Sarthak Bhagat, Samuel Li, Joseph Campbell, Yaqi Xie, Katia Sycara, and Simon Stepputtis. Let me help you! neuro-symbolic short-context action anticipation. *IEEE Robotics and Automation Letters*, 2024. 2.0.1, 2.0.3, 4.0.1, 4.0.3, 4.1, 4.0.4, 4.2, 4.0.5

[10] Achim Buerkle, William Eaton, Niels Lohse, Thomas Bamber, and Pedro Ferreira. EEG based arm movement intention recognition towards enhanced safety in symbiotic human-robot collaboration. *Robotics and Computer-Integrated Manufacturing*, 70:102137, 2021. doi: 10.1016/j.rcim.2021.102137. URL https://doi.org/10.1016/j.rcim.2021.102137. 1

[11] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 3.0.4

[12] 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. 4.0.2, **??**, 4.0.4

[13] Xiaojun Chang, Pengzhen Ren, Pengfei Xu, Zhihui Li, Xiaojiang Chen, and Alex Hauptmann. A comprehensive survey of scene graphs: Generation and application. *arXiv preprint arXiv:2104.01111*, 2021. 2.0.1

[14] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19830–19843, 2023. 2.0.2

[15] Tianshui Chen, Liang Lin, Riquan Chen, Xiaolu Hui, and Hefeng Wu. Knowledge-guided multi-label few-shot learning for general image recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1371–1384, 2020. 2.0.1

[16] Weilin Cong, William Wang, and Wang-Chien Lee. Scene graph generation via conditional random fields. *arXiv preprint arXiv:1811.08075*, 2018. 2.0.1

[17] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):10850–10869, 2023. doi: 10.1109/TPAMI.2023.3261988. 1

[18] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European conference on computer vision (ECCV)*, pages 720–736, 2018. 2.0.3

[19] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4125–4141, 2020. 2.0.3, 4, 4.0.1

[20] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2.0.2

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3.0.2

[22] Yuan Fang, Kingsley Kuan, Jie Lin, Cheston Tan, and Vijay Chandrasekhar. Object detection meets knowledge graphs. In *International Joint Conferences on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence, 2017. 2.0.1

[23] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3575–3584, 2019. 4.0.2

[24] Antonino Furnari and Giovanni Maria Farinella. What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 6252–6261, 2019. 2.0.3

[25] Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Predicting the future: A jointly learnt model for action anticipation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5562–5571, 2019. 1

[26] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Red: Reinforced encoder-decoder networks for action anticipation. *arXiv preprint arXiv:1707.04818*, 2017. 2.0.3

[27] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13505–13515, 2021. 2.0.3

[28] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 971–980, 2017. **??**, 4.0.4

[29] Dayoung Gong, Joonseok Lee, Manjin Kim, Seong Jong Ha, and Minsu Cho. Future transformer for long-term action anticipation. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3052–3061, 2022. 1, 2.0.3, 3.0.3, 3.0.4, 4.0.3, 4.1, 4.0.4, 4.2, 4.3, 4.4, 4.0.4

[30] Dayoung Gong, Suha Kwak, and Minsu Cho. Actfusion: a unified diffusion model for action segmentation and anticipation. *arXiv preprint arXiv:2412.04353*, 2024. (document), 1, 2.0.2, 4.3, 4.4, 4.0.4, 4.6, 4.7

[31] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*, 2022. 2.0.2

[32] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 2.0.3

[33] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 1, 3.0.3

[34] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10696–10706, 2022. 2.0.2

[35] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 4.1, 4.0.4, 4.2, 4.3, 4.4, 4.0.4

[36] Hongji Guo, Nakul Agarwal, Shao-Yuan Lo, Kwonjoon Lee, and Qiang Ji. Uncertainty-aware action decoupling transformer for action anticipation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18644–18654, 2024. 1

[37] Akash Gupta, Jingen Liu, Liefeng Bo, Amit K Roy-Chowdhury, and Tao Mei. A-act: Action anticipation through cycle transformations. *arXiv preprint arXiv:2204.00942*, 2022. 4.3, 4.4, 4.0.4

[38] Marcel Hildebrandt, Hang Li, Rajat Koner, Volker Tresp, and Stephan Günnemann. Scene graph reasoning for visual question answering. *arXiv preprint arXiv:2007.01072*, 2020. 2.0.1

[39] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2.0.2

[40] Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea

Dittadi. Diffusion models for video prediction and infilling. *arXiv preprint arXiv:2206.07696*, 2022. 2.0.2

[41] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Timeception for complex action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 254–263, 2019. **??**, 4.0.4

[42] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Videograph: Recognizing minutes-long human activities in videos. *arXiv preprint arXiv:1905.05143*, 2019. **??**, 4.0.4

[43] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514, 2021. 2.0.1

[44] Licheng Jiao, Jie Chen, Fang Liu, Shuyuan Yang, Chao You, Xu Liu, Lingling Li, and Biao Hou. Graph representation learning meets computer vision: A survey. *IEEE Transactions on Artificial Intelligence*, 4(1):2–22, 2022. 2.0.1

[45] Qiuhong Ke, Mario Fritz, and Bernt Schiele. Time-conditioned action anticipation in one shot. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9925–9934, 2019. 2.0.3, 4.3, 4.4, 4.0.4

[46] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 780–787, 2014. 4, 4.0.1

[47] Bolin Lai, Sam Toyer, Tushar Nagarajan, Rohit Girdhar, Shengxin Zha, James M. Rehg, Kris Kitani, Kristen Grauman, Ruta Desai, and Miao Liu. Human action anticipation: A survey. *arXiv preprint arXiv:2410.14045*, 2024. URL https://arxiv.org/abs/2410.14045. 1

[48] Christopher Lang, Alexander Braun, and Abhinav Valada. Robust object detection using knowledge graph embeddings. In *DAGM German Conference on Pattern Recognition*, pages 445–461. Springer, 2022. 2.0.1

[49] Yichong Leng, Zehua Chen, Junliang Guo, Haohe Liu, Jiawei Chen, Xu Tan, Danilo Mandic, Lei He, Xiangyang Li, Tao Qin, et al. Binauralgrad: A two-stage conditional diffusion probabilistic model for binaural audio synthesis. *Advances in Neural Information Processing Systems*, 35:23689–23700, 2022. 2.0.2

[50] Karen Leung, Edward Schmerling, Mo Chen, S. Shankar Sastry, and Marco Pavone. Trustworthy interaction-aware decision making and planning for autonomous vehicles. *The International Journal of Robotics Research*, 39(14): 1724–1747, 2020. doi: 10.1177/0278364920941832. 1

[51] Bowen Li, Sebastian Scherer, Yun-Jou Lin, Chen Wang, et al. Airloc: Object-based indoor relocalization. *arXiv preprint arXiv:2304.00954*, 2023. 2.0.1

[52] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022. 1, 2.0.2

[53] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. *Proceedings of the IEEE International Conference on Computer Vision*, pages 1261–1269, 2017. 2.0.1

[54] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European conference on computer vision (ECCV)*, pages 619–635, 2018. 4, 4.0.1

[55] Daochang Liu, Qiyue Li, Anh-Dung Dinh, Tingting Jiang, Mubarak Shah, and Chang Xu. Diffusion action segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10139–10149, 2023. 2.0.2

[56] Tomasz Malisiewicz and Alyosha Efros. Beyond categories: The visual memex model for reasoning about object relationships. *Advances in neural information processing systems*, 22, 2009. 2.0.1

[57] Kenneth Marino, Ruslan Salakhutdinov, and Abhinav Gupta. The more you know: Using knowledge graphs for image classification. *arXiv preprint arXiv:1612.04844*, 2016. 2.0.1, 3.0.2

[58] Sebastian Monka, Lavdim Halilaj, and Achim Rettinger. A survey on visual transfer learning using knowledge graphs. *Semantic Web*, 13(3):477–510, 2022. 2.0.1

[59] Maximilian Mozes, Martin Schmitt, Vladimir Golkov, Hinrich Schütze, and Daniel Cremers. Scene graph generation for better image captioning? *arXiv preprint arXiv:2109.11398*, 2021. 2.0.1

[60] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. Ego-topo: Environment affordances from egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 163–172, 2020. 4.0.2, 4.0.3, **??**, 4.0.4

[61] Megha Nawhal, Akash Abdu Jyothi, and Greg Mori. Rethinking learning approaches for long-term action anticipation. In *European Conference on Computer Vision*, pages 558–576. Springer, 2022. 2.0.3, **??**, 4.0.4

[62] OpenAI. Chatgpt o1. `https://openai.com/index/introducing-openai-o1-preview/`, 2024. Large language model. 4.1, 4.0.4, 4.2, 4.4, 4.0.4

[63] Ishan Prakash, S Indu, Mini Sreejeth, et al. Image flare removal using deep convolutional generative adversarial networks. In *2023 10th International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 834–839. IEEE, 2023. 1

[64] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 3.0.2

[65] Alexey Rodin et al. Action scene graphs for long-form understanding of egocentric videos. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1234–1243, 2024. 2.0.1

[66] 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. 2.0.2

[67] Ari Seff, Wenda Zhou, Farhan Damani, Abigail Doyle, and Ryan P Adams. Discrete object generation with reversible inductive construction. *Advances in neural information processing systems*, 32, 2019. 2.0.2

[68] Fadime Sener, Dipika Singhania, and Angela Yao. Temporal aggregate representations for long-range video understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 154–171. Springer, 2020. 4.3, 4.4, 4.0.4

[69] Uriel Singer, Adam Polyak, Tamir Hayes, Nupur Kumari Ravi, Aviv Navon, Gal Chechik, Amir Keller, Ankit Gupta, Oron Anschel, Alexander Schwing, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 1

[70] Sebastian Stein and Stephen J McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 729–738, 2013. 4, 4.0.1

[71] Wenqiang Tang, Zhouwang Yang, and Yanzhi Song. Selective interactive networks with knowledge graphs for image classification. *Knowledge-Based Systems*, 278: 110889, 2023. 2.0.1

[72] Guy Tevet, Yoav Gordon, Amir Hertz, Shai Shapiro, Daniel Friedman, Amit H Bermano, and Daniel Cohen-Or. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022. 1

[73] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE transactions on knowledge*

and data engineering, 29(12):2724–2743, 2017. 5

[74] Yunke Wang, Xiyu Wang, Anh-Dung Dinh, Bo Du, and Charles Xu. Learning to schedule in diffusion probabilistic models. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2478–2488, 2023. 2.0.2

[75] Peiyu Yu, Sirui Xie, Xiaojian Ma, Baoxiong Jia, Bo Pang, Ruiqi Gao, Yixin Zhu, Song-Chun Zhu, and Ying Nian Wu. Latent diffusion energy-based model for interpretable text modeling. *arXiv preprint arXiv:2206.05895*, 2022. 2.0.2

[76] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Bridging knowledge graphs to generate scene graphs. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 606–623. Springer, 2020. 2.0.1

[77] Ce Zhang, Changcheng Fu, Shijie Wang, Nakul Agarwal, Kwonjoon Lee, Chiho Choi, and Chen Sun. Object-centric video representation for long-term action anticipation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6751–6761, 2024. 4.4, 4.0.4

[78] Dehai Zhang, Menglong Cui, Yun Yang, Po Yang, Cheng Xie, Di Liu, Beibei Yu, and Zhibo Chen. Knowledge graph-based image classification refinement. *IEEE Access*, 7:57678–57690, 2019. 2.0.1

[79] Wentian Zhao and Xinxiao Wu. Boosting entity-aware image captioning with multi-modal knowledge graph. *IEEE Transactions on Multimedia*, 2023. 2.0.1

[80] Zeyun Zhong, Manuel Martin, Michael Voit, Juergen Gall, and Jürgen Beyerer. A survey on deep learning techniques for action anticipation. *arXiv preprint arXiv:2309.17257*, 2023. 4.0.4

[81] Zeyun Zhong, David Schneider, Michael Voit, Rainer Stiefelhagen, and Jürgen Beyerer. Anticipative feature fusion transformer for multi-modal action anticipation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6068–6077, 2023. 2.0.3

[82] Zeyun Zhong, Chengzhi Wu, Manuel Martin, Michael Voit, Juergen Gall, and Jürgen Beyerer. Diffant: Diffusion models for action anticipation. *arXiv preprint arXiv:2311.15991*, 2023. (document), 1, 2.0.2, 3.0.3, 4.0.3, 4.0.4, 4.3, 4.4, 4.0.4, **??**, 4.6, 4.7

[83] Yimin Zhou, Yiwei Sun, and Vasant Honavar. Improving image captioning by leveraging knowledge graphs. In *2019 IEEE winter conference on applications of computer vision (WACV)*, pages 283–293. IEEE, 2019. 2.0.1

[84] Xiangxin Zhu, Dragomir Anguelov, and Deva Ramanan. Capturing long-tail distributions of object subcategories. In *Proceedings of the IEEE Conference on*

Computer Vision and Pattern Recognition, pages 915–922, 2014. 2.0.1