Enhancing Robot Perception and Interaction Through Structured Domain Knowledge

Sarthak Bhagat

CMU-RI-TR-24-28

June 22

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

Thesis Committee:
Professor Katia Sycara, *chair*
Professor Katerina Fragkiadaki
Ini Oguntola

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

To my mother, father, and sister.

Abstract

Despite the advancements in deep learning driven by increased computational power and large datasets, significant challenges remain. These include difficulty in handling novel entities, limited mechanisms for human experts to update knowledge, and lack of interpretability, all of which are crucial for human-centric applications like assistive robotics. To address these issues, we propose leveraging structured information sources, such as knowledge graphs, to enhance the robustness and reliability of deep learning models by utilizing additional domain knowledge. By integrating these knowledge sources through neurosymbolic architectures, which combine neural networks and symbolic reasoning, we can improve model interpretability, generalization, and flexibility. This approach enables AI systems to understand complex scenes and human actions better, ultimately leading to more reliable and transparent performance in real-world scenarios. Our work highlights the potential of augmenting neural networks with additional domain knowledge. Particularly, we demonstrate the benefit of this approach in the task of learning novel objects in a sample-efficient manner and action anticipation from short-video contexts in a human-robot collaborative setting.

Acknowledgments

Is it the journey or the destination? Neither. It's the people.

I consider myself extremely fortunate to have had some of the most amazing mentors, friends, and family supporting me during one of the most challenging yet rewarding phases of my life. They not only stood by me during difficult times but also pushed me to become a better version of myself. As my two transformative years at CMU come to an end, I feel immense gratitude for the incredible people who supported me through all the ups and downs of this journey.

Firstly, I would like to extend my deepest and most heartfelt gratitude to my advisor, Katia Sycara, for her unwavering support, profound wisdom, and constant encouragement throughout my time at CMU. Her insights and attention to detail have inspired me to strive for excellence in everything I undertook during my Master's. Her remarkable judgment in selecting impactful yet practically feasible problems to work on is something from which I have learned a great deal. I am personally grateful for the moments of insight, enjoyment, and long discussions during our never-ending group meetings. These experiences significantly contributed to my growth, not only as a researcher meticulously designing and conducting experiments but also as an orator persuasively presenting the value of my proposed ideas.

I am immensely grateful to Simon Stepputtis for his support, guidance, and intellectual collaboration throughout my thesis projects. Over the past two years, he has served as a mentor in every sense of the word, assisting me from the inception of ideas, persuading Katia to pursue my concepts, to debugging intricate code, and refining paper drafts like this one. His mentorship has been invaluable, providing me with the guidance and resources that any junior student could hope for from a senior peer. Additionally, I extend my thanks to other postdocs in the lab, Joseph Campbell, Yaqi Xie, and Woojun Kim, for their continual support during group meetings.

I am immensely thankful to all my lab members who made each day in the lab incredibly rewarding. I gained so much knowledge from each one of them, sharing both joyful memories and profound research discussions. I express my gratitude (in alphabetical order) to Aryan Mangal, Dana Hughes, Ini Oguntola, Joseph Campbell, Renos Zabounidis, Samuel Li, Sha Yi, Shreya Sharma, Simon Stepputtis, Srujan Deolasee, Venkata Nagarjun, Wei-Hao (Zack) Zeng, Woojun Kim, Yaqi Xie, Yu Quan Chong, Yimin Tang, Yue (Sophie) Guo, and Yuzhe (Bryan) Lu. I will also miss the annual tradition of Thanksgiving dinner, which added warmth and camaraderie to our lab's bond. I deeply appreciate the chance to work alongside them and grow through their presence.

I'm deeply grateful to my dear friends, Anish Madan, Dvij Kalaria, Poorvi Hebbar, Pranay Gupta, Pushkal Katara, Shagun Uppal, Shreya Sharma, and Sriram

Narayanan, for their invaluable contributions to my personal and professional growth. Our table tennis sessions in Robolounge were not just a way to take my mind off work, but moments of pure joy. I'll forever treasure our shared adventures, from traveling to new cities to the unforgettable nights of dancing and singing. The 4 AM dining table discussions with Anish, Shreya, and Shagun, the gym sessions with Pranay where we pushed each other towards consistency, late-night coffee chats with Pranay, Dvij, Shagun, and Shreya, and the endless FIFA sessions with Pranay and Pushkal, where our shouts filled the room – these are memories that will stay with me forever. These past two years have been a whirlwind, and I'm thankful to have experienced it all with such incredible people.

I would also like to extend my gratitude to someone without whom this journey wouldn't have been possible: Shagun. You challenged me to become a better version of myself every single day. Your support during the stressful job search, your "never-say-never" attitude that instilled in me the belief that I could tackle any challenge, your cheerful nature that found joy in the smallest moments even when everything seemed bleak, and your resilience and fortitude — these are the qualities I deeply admire about you. They will remain etched in my memory forever, strengthening our bond in ways I couldn't have imagined, transforming my good into better, and my better into best.

Lastly, I want to express my profound gratitude to my parents, whose persistent courage propelled me across 8000 miles to pursue my studies. They not only supported me through challenging times but also instilled unwavering confidence in me, which has served as my anchor during the stormiest of days. I am forever indebted to the quality education they provided, which has not only shaped me as a researcher but also as a person. While I can never fully repay my parents for the countless sacrifices they've made since my childhood, I aspire to honor their love and support in every way I can. To my dear sister, Vidita, I owe an immeasurable debt of gratitude. She has been my guiding light, constantly challenging me to be a better human being. From the nights filled with laughter over the most trivial matters to the profound discussions about our careers and lives, our bond has only grown stronger with time. Words fail to capture the depth of our relationship, but I am endlessly thankful for her presence in my life. And to my beloved furry friend, Oreo, who has shown me the true meaning of unconditional love and has been my source of joy during the darkest of days. Your boundless affection has been my motivation to persevere, even when faced with adversity. I also want to extend my heartfelt appreciation to my late paternal and maternal grandparents, whose wisdom and guidance have always steered me toward the path of goodness, emphasizing that being a good human being should always precede any other accomplishment in life. Together, these cherished individuals have shaped me into the person I am today, and for that, I will be eternally grateful.

Funding

I gratefully acknowledge the financial support from multiple funding sources, which made this research possible. Specifically, I would like to acknowledge the support from the Defense Advanced Research Projects Agency (DARPA) under grants FA8750-23-2-1015 and HR001120C0036, the Air Force Office of Scientific Research (AFOSR) under grants FA9550-18-1-0251 and FA9550-18-1-0097, and the Army Research Laboratory (ARL) under grants W911NF-19-2-0146 and W911NF-2320007.

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

Despite the increased utility of deep learning models driven by advancements in hardware computation capacity and the availability of large datasets, these models still face several significant challenges that limit their effectiveness in various contexts. Key issues include a lack of interpretability, which makes it difficult to understand and trust model predictions; and a limited ability to effectively handle novel entities or classes or mechanisms for human experts to modify and update the system's knowledge. These limitations are particularly problematic in applications involving human interaction where reliability and trust are essential.

To mitigate these shortcomings, this work proposes leveraging structured information sources, such as knowledge graphs, to enhance the robustness and reliability of deep learning models. Knowledge graphs provide a rich source of structured domain knowledge that can ground model predictions in more reliable and interpretable information. Since KGs are inherently interpretable by humans and easy to edit, their augmentation with neural architectures can provide enhanced explainability and adaptability. By integrating these knowledge sources, we can improve the model's ability to generalize to new entities and classes and facilitate easier updates and modifications by human experts.

More specifically, this work advocates for the use of neurosymbolic architectures that combine the strengths of neural networks, such as their ability to handle high-dimensional data and generalize to unseen data—with the interpretability and flexibility provided by symbolic approaches. By leveraging these combined strengths, neurosymbolic architectures can enhance both the performance and reliability of deep learning models. This approach aims to bridge the gap between the advancement of recent deep learning approaches and the reasoning capabilities of structured neurosymbolic approaches, ultimately leading to enhanced performance and explainability of AI systems in complex, human-centric environments.

In the realm of robotics, where the integration of AI is becoming increasingly prevalent,

there exists a pressing need for advanced architectures capable of handling complex real-world scenarios. While prior approaches that utilize feature engineering have proven useful in domains of limited scope, data-driven neural network-based methods offer a superior solution for problems requiring an understanding of underlying concepts in the visual domain. Despite their efficacy in these complex areas, significant challenges remain. These deep learning models often struggle to recognize novel objects on which they were not trained, and they lack interpretability — an essential component for human-centric systems. This deficiency in comprehending various objects, their affordances, and attributes can be addressed by utilizing structured domain knowledge priors in the form of a knowledge graph. Such a knowledge source provides grounding for the efficient extraction of various entities in the scene, referred to in this work as concepts.

Inspired by the human ability to draw upon symbolic knowledge to interpret and interact with the environment, we propose a neurosymbolic architecture that combines deep learning with symbolic reasoning. By augmenting state-of-the-art recognition models with a symbolic knowledge graph, our approach enables these models to effectively understand novel concepts in a few-shot continual learning setting. Traditional neural network-based approaches excel only at recognizing concepts they have been explicitly trained on. However, by supplementing these models with a knowledge graph, we can extend their scope to include previously unseen concepts by leveraging the understanding of related concepts and inter-concept relationships. This integration allows the model to consider not only visual concepts, such as objects and scenes but also abstract concepts, such as affordances and attributes, thereby enhancing its overall understanding of the scene. Through an extensive set of experiments, we demonstrate the effectiveness of our approach, outperforming existing methods in tasks such as few-shot classification and extraction of novel non-visual concepts on benchmark datasets.

Extending this idea to the context of videos (– long-horizon interactions), we also demonstrate the utility of our approach in assistive robotics, focusing on tasks like collaborative cooking where accurate action anticipation is crucial for effective human-robot interaction. By imbuing the system with domain knowledge about scene objects and their respective affordances, we enable it to anticipate human intentions from short observation contexts — an area where previous work on action anticipation has fallen short.

Moreover, our work showcases the efficacy of our approach by integrating it with a transformer-based action anticipation architecture. This integration enhances the model's ability to analyze human interactions with scene objects by dynamically adjusting attention mechanisms based on the encoded knowledge of object affordances. Through empirical evaluations, we show that our approach not only outperforms current action anticipation baselines on standard benchmarks but also proves effective in the context of a real-world

dummy kitchen setup.

In essence, our work underscores the necessity of neurosymbolic architectures augmented with structured domain knowledge priors in robotics, particularly in tasks requiring a nuanced understanding of the environment and seamless interaction with humans. The approach aims to enhance the visual scene understanding capabilities of deep learning models via the augmentation of reliable knowledge sources to perform few-shot continual learning of concepts and action anticipation from short video contexts for effective human-robot collaboration. Our approach paves the way for more versatile and interpretable AI systems in real-world applications by bridging the gap between deep learning and symbolic reasoning.

Chapter 2

Sample-Efficient Learning of Novel Visual Concepts

2.1 Introduction

The ability to recognize objects from visual inputs [109] is crucial for the success of agents that interact in real or simulated environments [13, 84, 88]. Beyond applications in agent development, object recognition is also vital for image captioning [86], scene understanding [104], vision-language understanding [92], and many other domains. Recent contributions to foundational vision models [28, 43] and wider availability of computational resources has enabled many of these applications. One benefit of such models is their ability to drastically reduce the amount of training data needed when utilizing them as priors to train new visual tasks, e.g. in the domain of object recognition. However, while very capable, these pre-trained models often fail to perform well in few-shot learning settings that require them to recognize novel objects from a small set of sample images [102]. Beyond object recognition, assigning abstract concepts and affordances is an even more challenging task as concepts such as *wearable* are only indirectly related to a visual representation for several tasks including visual question answering [26], visual question generation [72, 91], and other scene understanding tasks [74]. Inspired by how humans learn to utilize few-shot learning by connecting novel concepts to their prior domain knowledge and experience, neuro-symbolic architectures [42] can address some of these shortcomings by imbuing neural networks with symbolic knowledge graphs (KG) [4]. Utilizing the interconnected domain knowledge of the graph, novel concepts can be added in a few-shot manner by augmenting the graph with the new nodes and thus, also limiting the need to re-train large parts of the neural architecture. Depending on the representation of novel nodes, such approaches are largely invariant to the topology of the

graph, only requiring the final neural outputs to be expanded and trained while intermediate components may be able to only require little fine-tuning. The availability of interconnected domain knowledge also allows easy integration of non-visual abstract concepts and affordances as relationships can be formed between such concepts and existing entities.

In the spirit of [67], our approach utilizes a neural network approach in conjunction with an optimized KG constructed from the Visual Genome Multi-Label (VGML) [67] dataset. In this work, we improve and extend this setup to few-shot multi-label classification (FS-MLC) by proposing a pipeline that adds new information to existing domain knowledge via `RelaTe` : a multimodal relationship prediction transformer that, given a small set of images and a latent representation of the linguistic concept, automatically connects novel objects, abstract concepts, and affordances to existing domain knowledge. In particular, `RelaTe` will evaluate the information propagated through the KG that is relevant to these images in the context of a latent concept representation from GloVe [76] and determine which nodes are applicable to be connected to the novel target concepts. Subsequently, we propose to extend the capacity of the final multi-label classifier by adding an output neuron associated with the new concept. The related weights of this extra neuron as well as the graph neural network are then trained and fine-tuned, respectively, to learn how to incorporate the new information. Thus, our approach utilizes a dynamically changing neuro-symbolic architecture that efficiently incorporates additional concepts in a sample-efficient manner.

This chapter is adapted from our work [17] published in the 2^{nd} Conference on Lifelong Learning Agents (CoLLAs), 2023.

2.2 Related Works

Few-shot multi-label classification (FS-MLC) remains a challenging problem despite some recent advances [11, 23, 106]. On its own, few-shot classification is difficult due to various factors like catastrophic forgetting [39] and limited data sets; however, these problems are amplified in the multi-label case when novel target classes occur in conjunction with already existing concepts, making their identification and training even more challenging. One avenue of addressing this issue is the utilization of domain knowledge, which can reduce the complexity of this problem by reducing the reliance on labeled data [21, 101] and instead, drawing from the encoded knowledge. Such domain knowledge can be acquired in multiple ways, either by explicitly formulating and utilizing a data structure or by utilizing a foundational neural network that is "large enough" to encode the general knowledge. Examples of such large models are GPT [71], particularly MiniGPT-4 [111], CLIP [78], and Flamingo [9]. However, in this work, we focus on imbuing neural networks with symbolic knowledge in the form of a

Knowledge Graph as such data structure is amenable to human interpretation [41] and quick augmentation in order to address the FS-MLC problem. Nevertheless, we will compare our approach to publicly available large-language models (LLMs).

Few-Shot Multi-Label Classification Utilizing concepts has shown to be an efficient approach to learning interpretable policies [107]. One approach to learning the FS-MLC task is to define novel objects as the sum of their parts, allowing such approaches to learn how to recombine known, simpler concepts that represent the target class [46, 54]. However, the addition of novel fundamental concepts remains an active field of research. Several approaches have addressed the problem of adding new concepts from a small number of samples by utilizing additional modalities [65, 68], structured primitives [77], generative modeling [15, 80], and meta-learning methods [19]. However, these approaches are usually limited to simulated ([70, 77]) or less demanding datasets ([19, 64, 68, 80]) that do not reflect the richness and intricacy of real-world concepts that we encounter in our daily lives. One of the first papers addressing the problem of FS-MLC in great detail is Alfassy et al. [11], which tackled the problem of limited data by representing sample images and their labels in a latent space and defining various set operations over these representations to synthesize additional samples through the combination of latent image features. Similarly, the work presented in Yan et al. [106] proposed a multimodal approach that utilized word embeddings to align verbal and visual representations in a latent feature space, allowing the creation of a mechanism that obtains visual prototypes for unseen labels by sampling an image from the latent space pinpointed by a language description of the novel entity. In our work, we propose a framework for extracting abstract concepts from complex real-world images and demonstrate enhanced performance over current few-shot learners by utilizing the connection between linguistic and visual concept representations.

Neuro-Symbolic Few-Shot Learning In addition to the techniques discussed above, incorporating domain-specific knowledge shows great potential as it can assist in recognizing and adding new concepts in a more sample-efficient manner, especially in scenarios with limited data. A commonly used approach to utilize symbolic KGs in deep learning is graph neural networks [55, 97, 105], providing a multitude of benefits from interpretability [90] to the utilization of interconnected information. The hierarchy and structure present in KGs have resulted in their use as priors for neuro-symbolic vision systems [47] for a number of applications ranging from transfer learning [8] to vision-language pre-training [10]. Chen et al. [23] introduced a static knowledge-guided graph routing framework consisting of two graph propagation frameworks to transfer both visual and semantic features, enabling information

transfer between correlated features to train a better classifier with limited samples. Marino et al. [67] and Fang et al. [30] utilized this structured knowledge to identify the underlying concepts present in the image. With a comprehensive graph, the structured knowledge embedded in it can even be used to extract critical information about previously unseen classes in either a few- [22, 75] or zero-shot [45, 48, 56, 99, 100, 103] manner. However, a limitation of these approaches is the use of a static KG. The work presented in Wang et al. [98] and Kim et al. [50] partially addressed this problem by dynamically changing edge weights and re-computing latent node representations respectively, but the graph's structure and encoded knowledge fundamentally remain the same. In this work, we propose a mechanism to update both aspects of the neuro-symbolic architecture by dynamically extending the KG with novel nodes, computing representations that are conditioned on the target images, and updating the neural components of our classification approach both structurally and in regards to its trained weights. This also allows our approach to incorporate novel objects that go beyond the visual domain, including abstract concepts and affordances while alleviating the assumption, as in prior work [12, 24, 40, 89, 113] that an exhaustive KG has to exist.

2.3 Few-Shot Object Recognition with Neuro-Symbolic Architectures

In this work, we propose a twofold approach. Firstly, we employ a neuro-symbolic object recognition approach called Graph Search Neural Networks (GSNN), as originally introduced by Marino et al. [67]. To enhance the performance of this pipeline, we propose multiple modifications, namely adding image conditioning and incorporating node types (see Section 2.3.1). Secondly, we introduce a novel approach called `RelaTe` , which automatically extends the KG to integrate new concepts while, simultaneously, extending the neural components of the system to incorporate them (see Section 2.3.2). In the following sections, we provide detailed explanations of each component.

2.3.1 Neuro-Symbolic Object Recognition

At its core, our work considers the problem of detecting a set $\mathbb{C}$ of concepts in a given image $\boldsymbol{I}$, while affording the ability for a human Subject Matter Expert (SME) to extend the system's capability by detecting additional, novel concepts in a sample efficient manner from a small set of images. In this section, we first describe our inference pipeline, inspired by Marino et al. [67] before discussing novel concept addition in Section 2.3.2. Figure 3.4 describes the three main steps of the inference pipeline: First, we extract a set of candidate objects $\mathbb{F}_{\boldsymbol{I}}$ that

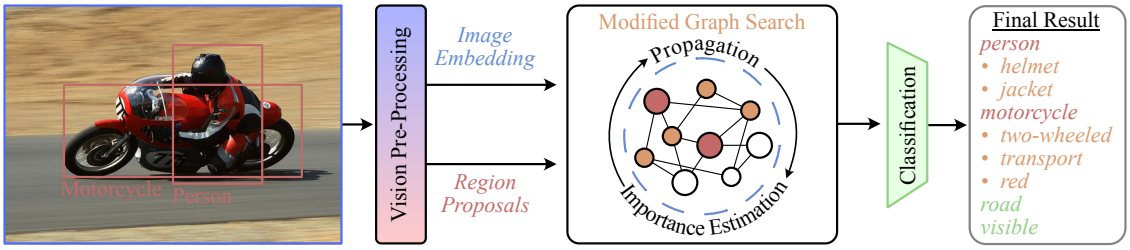


Figure 2.1: Example inference explaining our general inference pipeline, inspired by Marino et al. [67]. Given a novel image, we utilize ViT and Faster R-CNN to extract an image embedding (blue) and a set of initial object proposals (red). The initial proposals initialize our knowledge graph (red nodes) while the Modified Graph-Search identifies additional nodes (orange) conditioned on the overall image embedding (blue). Finally, our classifier evaluates the active nodes and produces a list of detected objects (green) in addition to the already detected nodes from the graph.

initialize our KG $\mathcal{G}$ and extract a global image embedding $\boldsymbol{e_I}$ (see Section 2.3.1); Second, we utilize the GSNN to propagate information through the graph while extending the prior work to also condition on the global image embedding $\boldsymbol{e_I}$, alleviating the need for edge types, and utilizing semantic node types; Third, the final classifier evaluates all active nodes of graph $\mathcal{G}$ (where $\mathcal{P}_{\boldsymbol{I}}$ is the sub-graph of $\mathcal{G}$ containing the active nodes for a particular image $\boldsymbol{I}$) in order to provide a holistic view of the input image and predict the final set of concepts $\mathbb{C}_{\boldsymbol{I}}$.

Extracting Candidate Objects

In the first step, we employ a pre-trained object detection pipeline, namely Faster R-CNN [79] to extract the initial set $\mathbb{F}_{\boldsymbol{I}}$ of candidate objects from image $\boldsymbol{I}$. Faster R-CNN is pre-trained on the COCO [60] dataset to predict the 80 concepts of COCO, but omit the 16 classes designated for our FS-MLC experiments as defined in Alfassy et al. [11] for a total of 64 trained concepts $\mathbb{C}_{\text{COCO}}$. For this approach, we did not conduct any further fine-tuning on other datasets, nor did we change the outputs of Faster R-CNN. The initial set of objects $\mathbb{F} \subset \mathbb{C}_{\text{COCO}}$ is then utilized to activate the initial set of nodes $\mathbb{N}_{\mathbb{F}}$ in graph $\mathcal{G}$. In contrast to the prior work that uses VGG [85], we use a pre-trained ViT [28] model to calculate an overall image embedding $\boldsymbol{e_I} \in \mathbb{R}^v$ with feature size v that is utilized to provide a global context for our modified graph-search approach as well as the final classifier. ViT is pre-trained on the ImageNet-21k [27] dataset and then fine-tuned on the ImageNet-10k dataset without any further modifications.

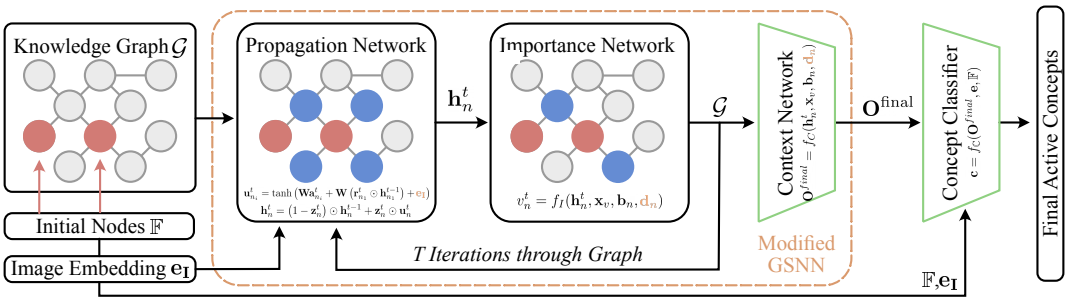


Figure 2.2: Overview of the Modified Graph Search Neural Network (GSNN). In contrast to prior work, we condition the propagation network on the global image embedding $\boldsymbol{e_I}$ and introduce a node type d_n while dropping edge types.

Modified Graph Search Neural Network

In this section, we provide a detailed explanation of the different components of the GSNN inspired by Marino et al. [67], as well as the proposed modification of conditioning its components on the input image, removing edge types, and introducing node types. Figure 2.2 shows the modified GSNN over graph $\mathcal{G}$ which contains three core components: a) the propagation network which computes an embedding for each node given its neighbors in the context of the current image $\boldsymbol{I}$, b) the importance network which decides which nodes are relevant and should be kept for potential future expansion given the current image $\boldsymbol{I}$, and c) the context network which generates final node embeddings. The context network is dependent on both the current image and the associations derived from the KG via multiple rounds of applying the propagation and importance network. The goal of the GSNN is to, in an iterative manner, propagate and prepare the information encoded in the KG that is relevant to image $\boldsymbol{I}$ by alternating the propagation and importance network over T steps. After T rounds, the context network provides the representation needed for the final concept classifier. The selection of T is a crucial hyper-parameter of our method and Section A.6 evaluates this choice in detail. Each of these components and the final classifier are described in the following sections along with our proposed modifications.

Propagation Network: Given the initial set of nodes $\mathbb{N}_{\mathbb{F}}$, the propagation network is designed to produce an output $\boldsymbol{O} \in \mathbb{R}^{N \times F}$, where N is the number of nodes and F the feature size for the latent node embedding, encoding the information of each node's neighborhood. Each row of $\boldsymbol{O}$ represents a feature vector $\boldsymbol{h}$ for the respective node which is initialized with all zeros outside of the first element, which contains the node ID, x_v. We utilize the graph structure, encoded in an adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ to retrieve the hidden states $\boldsymbol{h}$ of active nodes based on their neighborhood in the graph. In contrast to prior work, we also provide the propagation network with the global image encoding $\boldsymbol{e_I}$ in order to ensure

that information is propagated according to the image context (Ablations are provided in Section A.7)

Initially, we calculate a vector $\boldsymbol{a}_n$ representing the neighbourhood of each active node at iteration t given that $\boldsymbol{A}_{n_i}$ is the relative adjacency matrix for node n_i:

$$\boldsymbol{a}_{n_i}^t = \boldsymbol{A}_{n_i}^\mathsf{T} \left[\boldsymbol{h}_1^{t-1}, \boldsymbol{h}_2^{t-1}, \ldots, \boldsymbol{h}_N^{t-1}\right]^\mathsf{T} + b \tag{2.1}$$

Given this neighbourhood vector $\boldsymbol{a}_{n_i}^t$ for each node, we calculate $\boldsymbol{z}_{n_i}^t$ and $\boldsymbol{r}_{n_i}^t = \sigma\left(\boldsymbol{W}\boldsymbol{a}_{n_i}^t + \boldsymbol{W}\boldsymbol{h}_{n_i}^{t-1}\right)$, where all $\boldsymbol{W}$ are different and trainable weights of the neural network.

Subsequently, we calculate the update $\boldsymbol{u}_{n_i}^t$ for each node's hidden state as follows, where each $\boldsymbol{W}$ is a separately trainable weight matrix:

$$\boldsymbol{u}_{n_i}^t = \tanh\left(\boldsymbol{W}\boldsymbol{a}_{n_i}^t + \boldsymbol{W}\left(\boldsymbol{r}_{n_1}^t \odot \boldsymbol{h}_{n_1}^{t-1}\right) + \boldsymbol{e_I}\right) \tag{2.2}$$

In contrast to the work presented in Marino et al. [67], we calculate this update conditioned on the global image context $\boldsymbol{e_I}$, allowing the modified GSNN to incorporate image-specific information. Section 2.4.2 evaluates this benefit in further detail. The final hidden state $\boldsymbol{h}_{n_i}^t$ for each node in $\mathcal{P}_{\boldsymbol{I}}$ is subsequently calculated as a weighted sum of the previous hidden state $\boldsymbol{h}_{n_i}^{t-1}$ and the previously computed update vector $\boldsymbol{u}_{n_i}^t$:

$$\boldsymbol{h}_{n_i}^t = \left(1 - \boldsymbol{z}_{n_i}^t\right) \odot \boldsymbol{h}_{n_i}^{t-1} + \boldsymbol{z}_{n_i}^t \odot \boldsymbol{u}_{n_i}^t \tag{2.3}$$

Together with the importance network detailed in the next section, the propagation throughout the graph is done over T iterations, thus, allowing the utilization of the interconnected knowledge provided in the knowledge graph $\mathcal{G}$. Learning to utilize the symbolic knowledge of the graph efficiently is of utmost importance for our few-shot learning goal described in Section 2.3.2.

Importance Network. The importance network alternates with the propagation network over T cycles and decides whether or not an adjacent node to a currently active node should be made active. This is an important step as purely expanding nodes at every step has the potential to become computationally impractical if $\mathcal{G}$ is large. The importance v_n^t of each node at timestep t is calculated as follows:

$$v_n^t = f_I(\boldsymbol{h}_n^t, \boldsymbol{x}_v, \boldsymbol{b}_n, \boldsymbol{d}_n) \tag{2.4}$$

where, in contrast to the original GSNN, we propose the addition of $\boldsymbol{d}_n$ which represents a one-hot vector describing the node type ("object", "affordance", or "attribute") instead of

using an edge type and $f_I(\dots)$ is a multi-layer perceptron (MLP). Nodes above a certain threshold γ are maintained for the next propagation cycle. Additionally, we also learn a node bias term $\boldsymbol{b}_n$ for each node in the knowledge graph that intuitively captures a global meaning of the respective node. Note that this bias does not depend on a particular image $\boldsymbol{I}$.

Context Network. After T iterations, the final node embeddings are created via the context network [14]. Similar to the importance network, it is formulated as:

$$\boldsymbol{O}^{\text{final}} = f_C(\boldsymbol{h}_n^t, \boldsymbol{x}_v, \boldsymbol{b}_n, \boldsymbol{d}_n) \tag{2.5}$$

However, instead of predicting a scalar value indicating a node's importance, it generates the final state representation of the expanded nodes in $\mathcal{G}$, where $f_C(\dots)$ is another MLP.

Final Concept Classifier

The third and final step is the classification of the active concepts $\mathbb{C}_{\boldsymbol{I}} \subset \mathbb{C}$ in the input image $\boldsymbol{I}$ where $\mathbb{C}$ is a set of all possible concepts. These concepts are computed from the state representations $\boldsymbol{O}^{\text{final}}$ of all the expanded nodes in the active graph $\mathcal{P}$ along with the global image embedding $\boldsymbol{e}_{\boldsymbol{I}}$ and the originally detected classes $\mathbb{F}_{\boldsymbol{I}}$ from Faster R-CNN. Utilizing a single fully connected layer, a probability distribution over all the concepts is predicted $\boldsymbol{c} = f_{\mathbb{C}}(\boldsymbol{O}^{final}, \boldsymbol{e}_{\boldsymbol{I}}, \mathbb{F}_{\boldsymbol{I}})$. In order to make the result amenable to interpretation by a human user, we also provide the graph of active nodes $\mathcal{P}$, thus providing insights into why certain classifications may have been made.

2.3.2 Novel Concept Learning in a Dynamic Neuro-Symbolic Architecture

In addition to improving the neuro-symbolic architecture of GSNN our remaining two main contributions are as follows: a) a multi-modal Relation Prediction Transformer – `RelaTe` , that aids a human SME when adding novel concepts to the symbolic knowledge graph (Section 2.3.2) and b) introducing a framework to also dynamically updating neural parts of the inference pipeline described in Section 2.3.1.

Extending the Knowledge Graph with Relation Prediction Transformer

Figure 2.3 introduces our proposed approach – `RelaTe` . Given a small set of SME-provided images $\mathbb{I}_{\text{SME}}$ showing a novel concept as well as the partial graphs $\mathcal{P}$ for each image, `RelaTe` predicts how this novel concept can be incorporated into the existing knowledge graph. Thereby,

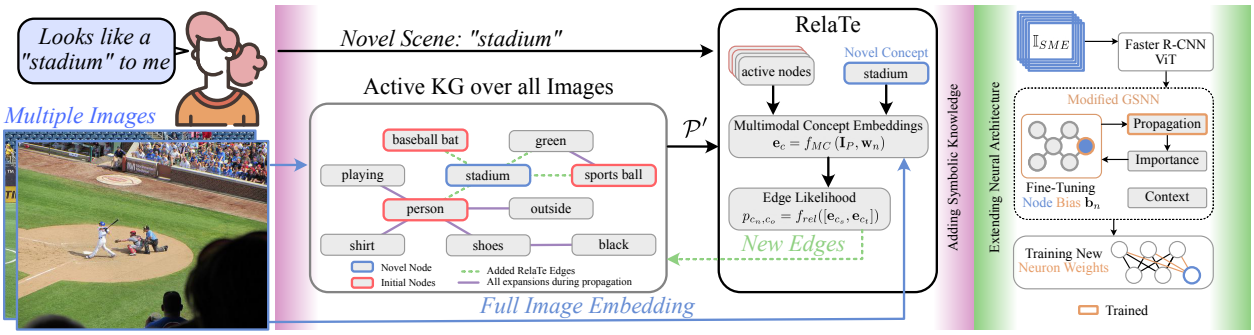


Figure 2.3: Given a novel concept of a *stadium* by a human expert along with one or more images for it, `RelaTe` estimates the optimal connectivity between the novel concept and existing nodes in the KG (i.e. *person*, *baseball bat*, *sports ball*, *green*). Subsequent inferences on similar images will yield the novel concept and allow for generalization through the domain knowledge encoded in the KG.

`RelaTe` provides an efficient and intuitive way of quickly adding new symbolic knowledge to the graph.

Multi-modal Cross-Attention Framework. While we use the image processing pipeline of Dosovitskiy et al. [28], we introduce a multi-modal approach to relating linguistic concept representations to images that contain a novel concept. In particular, we utilize GloVe [76] word embeddings in order to retrieve a context-invariant representation $\boldsymbol{w}_c = f_{GloVe}(c) \in \mathbb{R}^{F_w}$ for any given concept $c \in \mathbb{C}$. These representations are particularly useful when generalizing to novel concepts due to the potential similarity of new concept embeddings to a semantically similar word that may be known to the KG already. In order to combine the linguistic and visual representations, we utilize a cross-attention framework in which the image is represented as a sequence of patches $\boldsymbol{I}_P$ [28, 35]. We then combine both modalities as follows:

$$\boldsymbol{e}_{c_e} = f_{MC}\left(\boldsymbol{I}_P, \boldsymbol{w}_n\right) \tag{2.6}$$

where, $\boldsymbol{w}_c$ is the word embedding of any given concept c, and f_{MC} is explained in detail in Section A.1.

Post-Attention Fusion. Given the embeddings $\boldsymbol{e}_{c_e}$ for each node in each $\mathcal{P}_{\boldsymbol{I}}$ across all images in $\mathbb{I}_{\text{SME}}$ and the novel concept $\boldsymbol{e}_{c_n}$, we create pairs between the novel concept embedding $\boldsymbol{e}_{c_n}$ and the existing nodes' embeddings $\boldsymbol{e}_{c_e}$. We calculate the likelihood of an edge being present between a source and target nodes by concatenating the embeddings of each node pair as follows:

$$p_{c_s,c_t} = f_{rel}([\boldsymbol{e}_{c_s}, \boldsymbol{e}_{c_t}]) \tag{2.7}$$

Here, $f_{rel}(\dots)$ is an MLP predicting a scalar likelihood that an edge is present between the

pair of nodes in the direction from source to target. $\boldsymbol{e}_{c_s}$ and $\boldsymbol{e}_{c_t}$ are populated by every combination of the novel concept node and existing graph nodes for novel objects; however, for abstract concepts and affordances, we only calculate incoming connections (see Section 2.4.1 for details). Finally, at most k nodes that are above a specified threshold $p_{c_n,c_o} > \gamma$ are added to the KG. We empirically choose a suitable k depending on the concept type while γ is a global hyper-parameter.

Updating the Neural Architecture

Adding the novel concept to the KG alone does not directly yield improved classification performance as the new node does not have a trained node bias $\boldsymbol{b}_n$ yet, nor does the propagation network know how to generate node embeddings $\boldsymbol{h}_n$ for the novel concept. Additionally, the final classifier needs to be extended to enable the prediction of the novel class. In this section, we detail the process of training the node bias, fine-tuning the propagation module, and extending the classifier in further detail (also see Figure 2.3, describing how we extend the neural architecture).

Fine-tuning the Node Bias and Graph Propagation. To fine-tune the propagation network and train the node bias $\boldsymbol{b}_n$ of the novel concept, we utilize a small dataset $\mathcal{D}_{\text{SME}}$ generated from the images $\mathbb{I}_{\text{SME}}$ given by the human SME that demonstrate the novel concept. $\mathcal{D}_{\text{SME}}$ is subsequently expanded by applying transformations to all images in $\mathbb{I}_{\text{SME}}$. Further, we define a small curated dataset $\mathcal{D}_C$ with the intention of preventing catastrophic forgetting that contains $\sim 2\%$ of the original VGML training data. The dataset $\mathcal{D}_C$ is selected through Maximally Diverse Expansion Sampling (MDES) which selects a representative set of inputs from the original VGML dataset that activates a diverse set of nodes in the graph $\mathcal{G}$ (see Section A.9 for details). Prior to training the propagation network and node bias, the novel node's bias $\boldsymbol{b}_n$ is initialized as the average of its adjacent nodes' bias, and the corresponding novel node is forcefully activated in each training sample in $\mathcal{D}_{\text{SME}} \cup \mathcal{D}_C$ that contains an image from $\mathcal{D}_{\text{SME}}$. Forcing the activation of the novel node includes it in the downstream classification task and thus enables the fine-tuning of the propagation network and node bias.

Extending the Classification Module After training the propagation network and node bias for a limited number of epochs, the classification module with the novel neuron for concept c_n is unfrozen and added to the fine-tuning process. Furthermore, we reduce the learning rate of the propagation module and freeze the node bias $\boldsymbol{b}_n$ in this step of fine-tuning. As the classifier is depending on a valid node bias and the propagation network produces meaningful node embeddings for c_n, we delay the training of the classifier; however, we allow for continuous training of the propagation network to better capture the image-conditioned representation learning of the novel node. When training the classifier, its training objective

is reduced to a binary classification problem that predicts whether or not the novel concept is active in $\mathcal{D}_{\text{SME}} \cup \mathcal{D}_C$ while only calculating gradients for the added neuron so as to not alter the prediction capabilities of the existing concepts. This approach drastically reduces the number of parameters that need to be optimized, allowing the dataset to be comparatively small.

2.4 Evaluation

We evaluate the effectiveness of our proposed approach for novel concept recognition in two separate settings. First, we compare it against current state-of-the-art FS-MLC baselines on the COCO [60] dataset, and second, we perform a qualitative analysis of adding abstract concepts, affordances, and scene summaries to the underlying neuro-symbolic architecture. Particularly, we demonstrate that our method, when trained on Visual Genome, outperforms FS-MLC baselines on COCO and even improves performance further when trained on the COCO training data. Further, we conduct an extensive ablation study analyzing the impact of the various components of the neuro-symbolic architecture as well as our proposed `RelaTe` approach. Finally, we further investigate the implications of different node addition strategies, while additional experiments regarding the number of iteration step T, curated dataset $\mathcal{D}_C$, the addition of node types, and further qualitative analysis including deeper analysis in regards to failure cases and large language models are available in the Appendix. The source code can be found at: `https://github.com/sarthak268/sample-efficient-visual-concept-learning`.

2.4.1 Data and Metrics

As our method depends on the existence of an initial knowledge graph, we initialize the graph $\mathcal{G}$ from the Visual Genome Multi-Label (VGML) [67] dataset. While based on Visual Genome [52], VGML improves VG by drastically simplifying the graph, only using the 200 most common objects and 100 most common attributes, plus an additional 16 nodes to completely cover all COCO classes. We subsequently modify the graph for our FS-MLC task by removing 16 nodes that are defined as test nodes in Alfassy et al. [11], resulting in a total of 300 nodes while also removing all images related to these 16 FS-MLC target nodes from the training dataset of VGML. Furthermore, we impose the requirement that all nodes representing affordances and attributes must be leaf nodes in the graph to simplify graph structure further. Additionally, we remove edge labels and introduce a one-hot vector indicating whether a node is an object, attribute, or affordance, and discovered that the edge types did not impact the performance of the approach. We evaluate these changes in

comparison to the original knowledge graph from Marino et al. [67] in Section A.8.

The coverage of all COCO classes is allowing us to compare novel object recognition performance between multiple COCO baselines and our approach. Particularly, the 16 FS-MLC test classes include *bicycle*, *boat*, *stop sign*, *bird*, *backpack*, *frisbee*, *snowboard*, *surfboard*, *cup*, *fork*, *spoon*, *broccoli*, *chair*, *keyboard*, *microwave*, and *vase*. In our additional evaluation of novel abstract concepts, affordance, and scenes, we utilize the full knowledge graph with all 316 nodes. We utilize our modified VGML dataset to train the GSNN, classification head of ViT, and final concept classifier in an end-to-end fashion. Further, `RelaTe` is trained on the entire Visual Genome dataset after removing the 16 test classes from it. The training for `RelaTe` includes concepts that are not present in VGML.

Evaluation Metric. In order to compare the efficacy of our approach in the FC-MLC task, we utilize mean average precision (mAP), macro average precision (Macro AP), and the top-K score. mAP is computed by taking the mean of the AP scores computed for each label, where AP is the area under the precision-recall curve plotted for each label. Similarly, Macro AP is computed by averaging the AP scores for each label across all instances and then averaging the results across all classes. To compute the top-K score, we compute the percentage of K most confident predictions of our model that are correctly predicted, i.e. precision of K most confident predictions.

2.4.2 Novel Concept Recognition

In this section, we detail our comparison of utilizing `RelaTe` with our updated neuro-symbolic architecture to add novel concepts. Particularly, we compare against multiple state-of-the-art baselines on FS-MLC tasks over the COCO dataset while training our model on VGML and later fine-tuning on COCO. Further, we demonstrate the utility of adding scene summaries, e.g. *kitchen* from kitchen appliances, abstract concepts, and affordances in a comprehensive ablation study on the VGML dataset.

COCO Novel Multi-Object Recognition

We evaluate the efficacy of adding novel visual objects in a sample-efficient manner using our approach by comparing it against current state-of-the-art baselines in FC-MLC applications. As defined in Alfassy et al. [11], we use a set $\mathbb{I}_{SME}$ of five images per novel class and train all 16 novel classes one by one. Table 2.1 shows the results comparing our method to three state-of-the-art baselines as well as four additional ablations. For each method, "Source" indicates the training dataset for the respective model while $\mathcal{D}_{SME}$ and $\mathcal{D}_C$ indicate which dataset was used for the few-shot learning. If both datasets are used, they are randomly

interleaved. Lines 1 to 3 in Table 2.1 demonstrate the performance of our three baselines. Using a naive approach to adding novel concepts to Marino et al. [67], line 4 trains only the final classifier (by adding a novel neuron) on the same training dataset as used in lines 4 to 7 without adding novel information to the knowledge graph altogether. In addition to training the classifier, adding the novel node to the knowledge graph, but not training the propagation network and node bias is shown in line 5, indicating that our modified GSNN is mostly invariant of the knowledge graph despite the lack of fine-tuning, underlining the strength of having domain knowledge (compare line 4 and 5), yielding a 17% performance increase. However, further improvements can be made when fine-tuning the propagation network and node bias. Compared to Yan et al. [106] in line 3, which achieves 68.12% (the best baseline), utilizing the neuro-symbolic architecture and our proposed `RelaTe` architecture in line 6, we achieve a Macro AP score of 70.26, despite training on VGML, which is statistically significant with a standard deviation of $\sigma = 0.45$ at p-value $1.252e^{-3}$ trained over four seeds. Further, we also fine-tuned our method from line 6 on the training set of COCO and reported the results of 70.30% with $\sigma = 0.19$, with a p-value of $9.1e^{-5}$ over four seeds in line 7 of Table 2.1. In each case, we parameterize `RelaTe` with an unlimited k value to add as many relations as possible. Given that our results in line 6 are resulting from a model trained on an entirely different dataset, i.e. VGML, yet performs very similarly to being trained on COCO allows the conclusion that our approach has the ability to transfer knowledge between datasets through the utilization of a knowledge graph.

Recognizing Affordances, Attributes, and Scenes

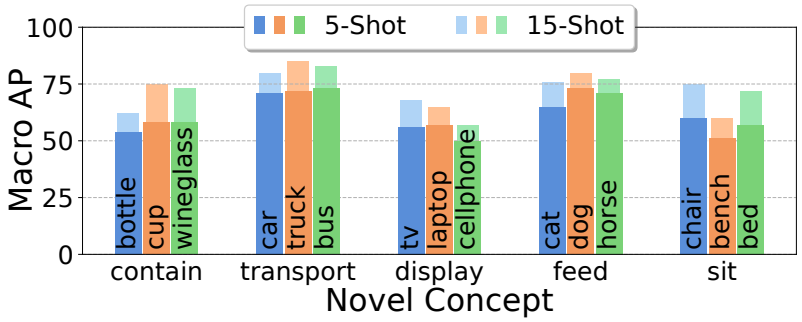
Unlike other approaches to few-shot novel concept detection that rely on novel objects being visible in the input image, our approach can go beyond such limitations through the utilization of interconnected information in the knowledge graph. In addition to adding visual concepts as shown in Section 2.4.2, we demonstrate how non-visible concepts like abstract concepts, attributes, and scene summaries can be added. While the borders between what is visual and what is not are sometimes blurry, particularly in the case of scenes, utilizing the knowledge graph highlights the ability to draw conclusions from a set of partial observations. E.g., given that *refrigerator*, *oven*, and *microwave* were detected, we can conclude that the input image likely shows the *kitchen*

	Method	Source	$\mathcal{D}_{SME}$	$\mathcal{D}_C$	Macro AP
1	[11]	COCO	✓	—	58.10
2	[23]	COCO	✓	—	63.50
3	[106]	COCO	✓	—	68.12
4	Fine-tuning (classifier)	VGML	✓	✓	52.22
5	Fine-tuning + `RelaTe`	VGML	✓	✓	69.26
6	Ours	VGML	✓	✓	70.26
7	Ours	COCO	✓	✓	**70.30**

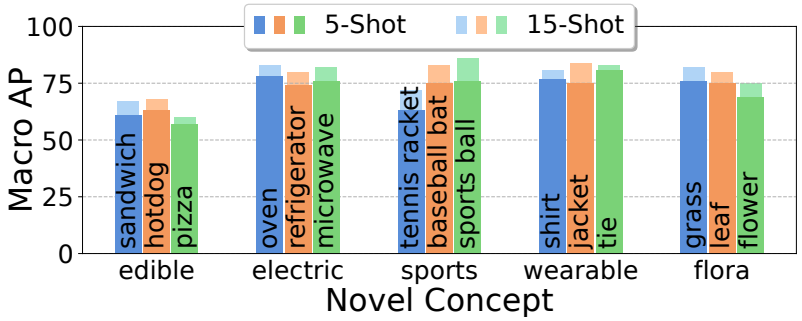
Table 2.1: Experimental results on COCO dataset for five-shot multi-label classification of previously unseen concepts.

concept, which can subsequently be added to the knowledge graph as a novel concept. In the following two sections, we discuss the addition of abstract concepts and scene summaries.

Adding Non-Visual Concepts. We conduct further experiments to assess the ability of `RelaTe` to incorporate non-visual concepts into the knowledge graph by relating it to relevant existing domain knowledge. In contrast to novel object recognition, we parameterize `RelaTe` with a threshold $k = 3$ in order to enforce a sparser connection of affordances and attributes to existing nodes.



(a) Novel affordances learning given a set of five images.



(b) Novel attribute learning, given a set of five images.

Figure 2.4: Analysis of the performance when adding novel affordances and attributes to the knowledge graph. We evaluate the performance on five-shot (dark colors) and fifteen-shot (bright colors) learning.

Figure 2.4 shows our experiments on adding novel affordances (Figure 2.4a) and attributes (Figure 2.4b). In each case, we selected a set $\mathbb{I}_{SME}$ with five and fifteen sample images containing three separate concepts that should be assigned to the novel concept. Subsequently, we evaluate the performance of the resulting model on 50 test images from within the same concept classes as well as 50 test images that do not show any of the trained targets. Our results show that added non-visual concepts have an average Macro AP of 66.7% given five sample images and 75.1% given fifteen images. Recall that for novel object detection as shown in Table 2.1, the Macro AP score is 70.26%. We hypothesize that the slightly lower performance on abstract

concepts roots in the difficulty of not having a clear visual representation for such concepts. However, when increasing the training samples to fifteen, we outperform the object detection reported in Table 2.1, which, in the context of deep learning, is still a relatively small sample size.

Novel Scene Recognition: In addition to adding novel objects and abstract concepts, `RelaTe` can also assist in the addition of compound concepts. For example, the existence of a *oven*, *microwave* and *refrigerator* implies a scene that can be defined as a *kitchen* that is the sum of the underlying parts. Encoding such knowledge poses a slightly different problem as compound concepts require reasoning over multiple adjacent concepts. This ability can be imbued by our KG, but can also be found in large foundational neural networks, particularly LLMs. Table 2.2 demonstrates the ability of three LLM baselines, MiniGPT-4 [111], CLIP [78], and Flamingo [9], to draw higher-level conclusions about the general scene shown in an image to our method, attempting the same task. We evaluate each scene on 25 test images of previously unseen samples and report the existence of the compound concept within the estimated concepts. For the LLMs, particularly the free-form response models, we queried the models to see if the image shows any of the five target scenes. With an average accuracy of 84.4%, this experiment underlines the utility of having interconnected knowledge that augments our few-shot detection pipeline, allowing the GSNN to successfully draw high-level conclusions from a set of basic concepts. While LLMs demonstrate partial success in identifying the high-level scenes, explicitly modeling the symbolic knowledge provides significant improvements despite the LLM's large general knowledge encoded within their trained model architecture. In additional experiments for the *kitchen* example, we showed that the likelihood of classifying the *kitchen* scene from a *refrigerator* or *microwave* alone is 24% and 28% respectively while the likelihood to identify it from an image containing both base concepts is 88%, showing that our model accurately learned that a *kitchen* is the sum of its parts.

		Scene Concept					
	Model	*stadium*	*kitchen*	*zoo*	*school*	*bedroom*	Avg.
8	CLIP (0-shot)	16	100	100	56	72	68.8
9	Flamingo (0-shot)	20	4	40	24	28	23.2
10	Mini-GPT (0-shot)	24	96	64	24	96	60.8
11	Flamingo (5-shot)	68	36	40	72	80	59.2
12	Ours (5-shot)	90	84	84	72	92	**84.4**

Table 2.2: Novel scene prediction in comparison to free-form text generation models.

Ablations

In this section, we ablate the different components of our FS-MLC pipeline on the VGML dataset, recognizing the novel objects defined in Alfassy et al. [11]. Table 2.3 summarizes

	Components			Fine-tuned		KG Configuration		All Classes			Novel Classes		
	ViT	FRCNN	KG	GSNN	CLF	RelaTe	MDES	T-1	T-5	mAP	T-1	T-5	mAP
1	✓	-	-	-	-	-	-	84.2	63.4	31.8	85.7	65.5	34.6
2	✓	✓	-	-	✓	-	-	84.7	64.2	33.0	86.1	67.2	37.3
3	✓	✓	✓	-	✓	-	-	87.4	68.8	36.5	91.8	72.8	68.0
4	✓	✓	✓	-	✓	✓	-	89.8	69.8	38.5	91.6	72.8	68.2
5	✓	✓	✓	✓	✓	✓	-	**90.4**	72.4	41.7	92.2	73.6	69.3
6	✓	✓	✓	-	✓	✓	✓	90.0	70.1	39.5	91.8	73.0	68.8
7	✓	✓	✓	✓	✓	✓	✓	90.3	**72.9**	**42.0**	**92.4**	**73.9**	**69.5**

Table 2.3: Experimental results on Visual Genome dataset, ablating the components of our method

these results where lines 1 and 2 show the performance on the test set across all classes and our 16 novel few-shot classes given their Top-1 (T-1) and Top-5 (T-5) performance when using pure neural end-to-end architectures. In each case, novel classes are trained with five demonstration images and evaluated on the test set of VGML. Line 3 adds a KG with the GSNN approach proposed in Marino et al. [67] and fine-tunes the final classifier (CLF-column) on the novel classes with an $\sim 3\%$ improvement in Top-K score. From this, we conclude that novel classes may also need to be added to the knowledge graph. Line 4 uses our proposed `RelaTe` approach to add the novel classes to the graph; however, it does not tune the GSNN with respect to the propagation network and node bias (GSNN-column). Adding nodes to the graph yields another $3-5\%$ improvement over line 3. Line 5 fine-tunes the propagation network and node bias with our methodology described in Section 2.3.2, improving performance by another $\sim 2\%$. Finally, lines 6 and 7 show the impact of our curated fine-tuning dataset $\mathcal{D}_C$ as compared to an equally sized random dataset over the original VGML dataset. This demonstrates the importance of MDES to prevent catastrophic forgetting. In summary, Table 2.3 highlights our approach's ability to effectively expand its understanding of novel concepts with limited samples by effectively utilizing the knowledge graph. Further experiments on the original KG are available in Section A.3 while a qualitative comparison of the ground-truth graph connections in comparison to the ones `RelaTe` adds is available in Section A.4.

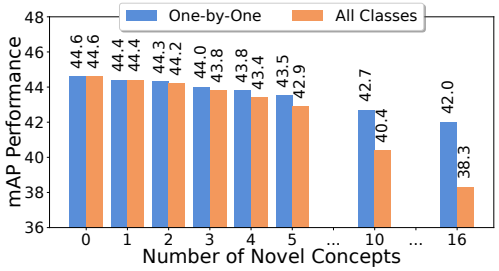


Figure 2.5: Strategy for adding the 16 novel concepts: one-by-one vs. all

Evaluating Node Addition Procedure

While `RelaTe` allows the addition of novel concepts individually, or as a group, we hypothesize that the node addition strategy has an impact on the overall performance of the model. Figure 2.5 shows the performance on the VGML dataset when adding the 16 nodes one-by-one (blue) or all at once (orange), omitting intermediate nodes 6-9 and 11-15 for simplicity. The trends show that adding one concept at a time and fine-tuning the classifier as well as the GSNN for each of them before adding the next concept yields a higher-performing model. This not only facilitates the extraction and comprehension of new concepts but also prevents the model from getting overwhelmed with multiple concepts simultaneously, thus, minimizing the risk of forgetting previously acquired knowledge.

Interpretability of Results

Our approach provides interpretability through the explicit propagation of the initially detected concepts $\mathbb{F}_{\boldsymbol{I}}$ through the graph $\mathcal{G}$, providing insights as to why certain final concepts have been classified. However, while these propagations are not a direct output of the model, they provide an auxiliary insight into the internal workings of the FS-MLC pipeline. Figure 3.4 shows how these propagations can be useful in interpreting the concept classifier's result.

Chapter 3

Neuro-Symbolic Short-Context Action Anticipation

3.1 Introduction

Action anticipation is a crucial step in the development of intelligent agents [29] for the task of human-robot collaboration (HRC). For example, accurately anticipating a human's future action allows a robot to assist them in their task proactively without needing to be instructed at every step, reducing the cognitive load on human operators, thus, allowing them to focus more on their work [73]. Prior work in action anticipation has mainly focused on short-term or next-action anticipation [32, 33, 34, 37, 69, 81, 82, 83]; however, to enable proactive agent behavior, multiple future actions must be predicted for long horizons as the immediate next action may not always be the most appropriate assistive action. For example, taking over a task that the human is already doing or about to start may interfere with the user's immediate actions, necessitating the prediction of multiple future actions. Selecting one of multiple future actions depends on various factors, including if such a task can be executed in parallel by the robot and if its likelihood of occurring is sufficiently high given the current observation. In an assistive task, predicting action sequences requires a quick understanding of the user's current behavior, necessitating making decisions given short observation contexts of task-relevant behavior. However, making accurate predictions of future actions given only a short horizon of relevant observations is challenging due to its inherent lack of context. To this end, we propose `NeSCA`, **Ne**uro-**S**ymbolic Short-**C**ontext **A**ction Anticipation, which imbues a neural action anticipation pipeline with additional symbolic domain knowledge in the form of a Knowledge Graph (KG).

NeSCA utilizes domain knowledge to connect scene objects to their relevant affordances [36, 112] through a structured prior. For example, with the knowledge that a *tomato* has the affordance being *cuttable* and knowing about the presence of a *knife* that can be used for cutting, NeSCA can boost the attention between these concepts to increase the likelihood of the human's intent of *cutting tomatoes* in the future while simultaneously attenuating the attention between other unrelated features (see Fig. 3.4). Imbuing a neural network that can effectively comprehend complex inputs like videos with symbolic knowledge can greatly enhance the performance of downstream tasks [17], i.e., subsequent action anticipation and user assistance. Empirically, we find that utilizing the knowledge graph that connects objects to their affordances reduces the required task-relevant observation by $\approx 50\%$ when predicting future actions as compared to current state-of-the-art baselines.

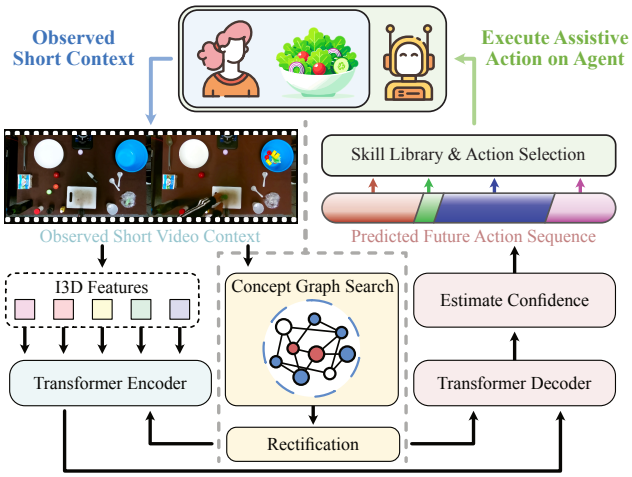


Figure 3.1: NeSCA: Given a short video segment (blue), our system anticipates future actions and their respective confidences (color gradients) utilizing our proposed neuro-symbolic attention approach to refocus attention between visual features. Finally, if sufficiently confident about the prediction, a robot executes assistive actions (green).

To process high-dimensional inputs like videos, transformers [94] have proven to be efficient at comprehending sequential data and lend themselves well to action anticipation from videos [38], but remain largely black-box end-to-end approaches. On the other hand, structured domain knowledge remains interpretable and has previously been investigated in the image domain [66], demonstrating improved performance for image classification [17]. In this work, we seek to integrate neural video comprehension with external symbolic domain knowledge pertaining to the objects in the scene, linking them to their respective affordances. Given a previously unseen video sequence, we extract the relevant scene objects via a neural object detector and employ graph search through our KG to assign relevant affordances to them. To achieve the integration of the video understanding and extracted domain knowledge, we propose to imbue the attention mechanism of the transformer with an addition rectification matrix that influences how queries and keys interact with each other. Intuitively speaking, the learned knowledge-conditioned rectification matrix boosts or attenuates the attention between various video features, thus, aiding the prediction of future actions. A particular benefit of this approach is that our proposed method significantly improves performance when

only short-horizon contexts are given – a key aspect for effective human-robot collaboration that prior works in action anticipation [2, 3, 31, 38, 49, 83] only addressed to a limited extent. Before utilizing our action anticipation approach for human-robot collaboration, we demonstrate its efficacy on two common long-term action-anticipation benchmarks, namely the *50Salads* [87] and *Breakfast* [53] datasets, and show superior performance as compared to current state-of-the-art methods.

Having demonstrated the efficacy of `NeSCA`, we showcase a joint salad creation task in a real-world tabletop scenario that leverages the sequence of predicted future human actions. Given a set of predicted actions, the system selects an appropriate action for the robot to execute while the user keeps working on their current task given a set of selection criteria (see Figure 3.4). Among others, these criteria mainly include checking whether the anticipated action's pre-conditions are already satisfied and if the action is predicted with sufficient confidence. When such an action is identified, the robot executes the action to support the user. With our approach, we achieve a 50.1% accuracy in selecting and executing an appropriate assistive action while also reducing the required length of context to half compared to the current state-of-the-art to achieve a similar success rate.

In summary, our contributions are as follows:

- We propose a novel approach utilizing knowledge graphs to augment the attention mechanism for transformer-based action anticipation, which we refer to as `NeSCA`.
- Through extensive experiments, we demonstrate that our proposed method outperforms current state-of-the-art methods for action anticipation on two challenging benchmarks, *50Salads* and *Breakfast*.
- We show how our proposed method can be utilized for effective HRC that anticipates tasks and subsequently supports human users in the creation of a salad in a real-world tabletop manipulation setting.

This chapter is adapted from our work [16] published in the ICCV 2023 Workshop on AI for Creative Video Editing and Understanding.

3.2 Related Work

Action anticipation is a field of research that is currently gaining a lot of attention due to its usefulness in areas such as autonomous driving and human-robot interaction [110]. In this study, we introduce a new approach that makes use of structured domain knowledge to predict long-term action sequences based solely on short video contexts.

Knowledge Graphs for Computer Vision. The emergence of the utilization of struc-

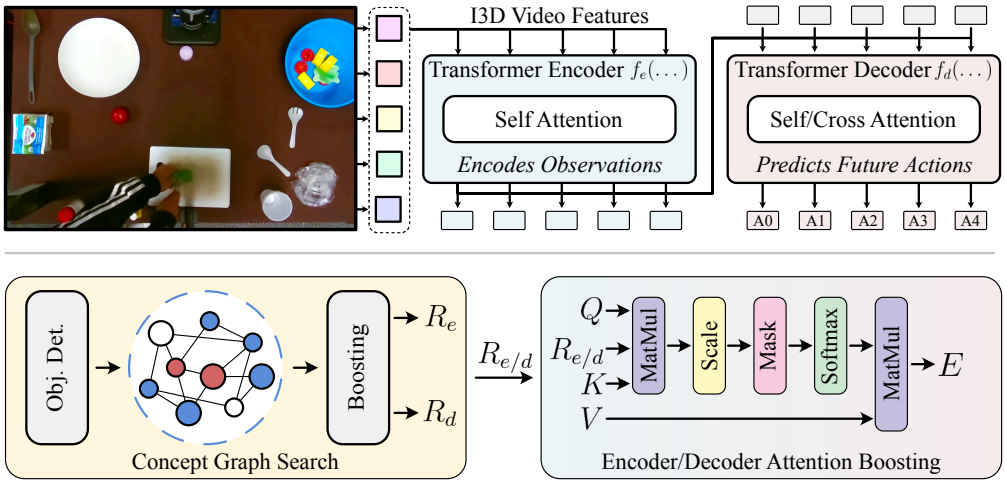


Figure 3.2: `NeSCA` utilizes a transformer architecture for action anticipation (top); however, in parallel, Concept Graph Search (bottom) is utilized to obtain the set of active concepts, including related affordances, in the scene. These concepts are further used to refocus the attention in the transformer toward the relevant visual features.

tured domain knowledge, in the form of knowledge graphs, in vision models is gaining traction as it grounds their predictions by establishing a comprehensive understanding of entities and their interconnected relationships, thereby, enhancing overall model interpretability and performance. [66] introduced a knowledge graph as a structured prior for image classification and proposed the Graph Search Neural Network, demonstrating its performance improvement by integrating knowledge graphs into the vision classification pipeline. Further, [17] extended it to include the augmentation of novel concepts, encompassing visual objects and compound concepts such as affordances, attributes, and scenes. In this work, we extend the idea by refining the propagation framework from [17] to identify relevant object affordances along with the tools that can be used to afford it in the desired manner. To the best of our knowledge, our approach is the first one to utilize the information about the affordances of the objects in the scene to perform action anticipation. Our methodology represents a novel approach to leveraging information regarding the affordances of objects within a scene for action anticipation.

Action Anticipation. The task of action anticipation from videos [44] revolves around predicting future actions based on a specific segment of the video. With recent advancements in foundational vision models and the availability of large-scale human-centric datasets [25], this domain has gained significant attention. Many recent approaches have been developed to predict a single future action within a short time frame, typically spanning a few seconds [15, 32, 33, 34, 37, 69, 81, 82, 83]. However, a notable emerging trend is long-term action anticipation, which emphasizes predicting a sequence of future actions occurring in the distant

future from a lengthy video [2, 3, 31, 38, 49, 83]. While much attention has been paid to predicting long-term actions with ample video context, limited research has addressed using short video contexts to predict long-term future action sequences. Our work addresses this particularly challenging task: Action anticipation for long-horizon predictions given only a short observation context.

Human-Robot Collaboration. Several approaches have performed human-robot collaboration by anticipating what actions might be useful in the current setup [96], by either utilizing gaze information from the user [5] or performing action prediction [7, 51]. Approaches utilizing action anticipation for human-robot collaboration circumvent the need for explicit task specification, accommodating situations where human intent is ambiguous or multiplicitous, while also streamlining everyday tasks by eliminating the time-consuming process of articulating actions. While these works can infer current actions, they fall short in capturing the temporal aspect of visual inputs to make predictions not only about ongoing actions but also anticipate future actions. Other methods have been proposed utilizing a human-in-the-loop approach to improve the learned policy in an online manner [61, 93]; however, enabling these interactions can be expensive, and therefore, offline finetuning approaches have been identified as an effective solution to deploy robots in real-world scenarios [59, 95]. This work integrates the advantages of predicting future actions and offline fine-tuning with a finite curated dataset in a novel environment to enhance the prediction of useful actions considering the subject's actions.

3.3 Knowledge-Guided Action Anticipation

This section introduces our proposed method, `NeSCA`, as well as its application to Human-Robot Collaboration (HRC). At its core, `NeSCA`, consists of two core components: (1) a neuro-symbolic graph-search approach that extracts relevant scene concepts (i.e., objects and their affordances, see Sec. 3.3.1); and (2) a modified attention mechanism informed by our extracted concepts, allowing us to anticipate future actions from short observation context (see Sec. 3.3.2 and 3.3.3). With this set of predicted actions, we demonstrate the utility of `NeSCA` in an HRC task, utilizing our fast action anticipation from short observed contexts, which allows us to effectively assist a user (see Sec. 3.3.4).

Problem Statement. `NeSCA` (see Fig. 3.2) addresses the problem of predicting a sequence of future actions $\mathbf{a}$ from a *short* video observation $\mathbf{F}$ that can subsequently be used to provide effective assistance to a human user. In our setting, we observe α-percent of a video and predict actions in the next β-percent of the video, with respect to the average total video length obtained from the training set. Given an observed video sequence $\mathbf{F}$, we learn a

Figure 3.3: Example of our assistive HRC system: Shortly after the user starts to prepare the dressing, the robot identifies the intention and correctly assists the user in creating the dressing by adding further ingredients.

function $\mathbf{a} = f_\theta(\mathbf{F})$ that predicts a sequence of actions $\mathbf{a}$ happening after the end of $\mathbf{F}$. The video sequence $\mathbf{F} \in \mathbb{R}^{H \times W \times C \times N}$ is represented as a four-dimensional matrix describing the height H, width W, and channels C of each video frame $\mathbf{F_n}$, and the number of observed frames N. The action sequence $\mathbf{a} = [(a_0, d_0, c_0), ..., (a_M, d_M, c_M)]$ contains a list of M tuples describing the action sequence a, its duration d, and confidence c.

To actuate the robot, we propose a policy $a_r = \pi(f_\theta(\mathbf{F}), \mathbb{S})$ utilizing the predicted set of future actions. Given a skill library $\mathbb{S}$ and a list of future actions $f_\theta(\mathbf{F})$, π identifies a suitable action a_r for a robot to execute in our HRC task.

Training Procedure. We train our model from a dataset $\mathbb{D}$ where each sample $\mathbf{s}_i = [\mathbf{F}^i, \mathbf{a}^i]$ contains the video $\mathbf{F^i}$ and action-sequence $\mathbf{a}^i$, where i is the index of the video, along with a single agent performing a task. After training, we provide the trained action anticipation model $f_\theta(\dots)$ with a new, previously unseen video sequence, showing α percent (with respect to time) of the full video, tasking the policy with predicting the most likely action for each frame in the following β percent of the remaining video.

3.3.1 Extracting Domain Knowledge

We provide a hand-crafted KG as the source of domain knowledge, establishing connections between various concepts. In the following, we refer to objects and affordances as concepts. Each node in our KG is initialized by utilizing Grounded-DINO [62]. To utilize this knowledge during inference, given a short sequence of video frames, we extend our prior work [17] to the domain of videos. Intuitively, this approach utilizes a neural object detector to extract a set of initial concepts and subsequently utilizes them as a starting point for a graph search through the knowledge graph $\mathbb{K}$. We create the graph $\mathbb{K}$ consisting of two types of nodes: object nodes (e.g., *salt*, *knife*, *bowl*) and affordance nodes (e.g., *graspable*, *pourable*, *cuttable*). For example, a *tomato* has a connection to *cuttable*, which, in turn, connects to *knife*.

In the first step, we extract a set of relevant concepts from the video frames $\mathbf{F}$, using open-vocabulary object detection as proposed in [62]. These initial concepts $\mathbb{C}_O$ are then utilized as a starting point for our iterative Concept Graph Search (CGS), forming the

initial set of active concepts in our KG. CGS has two main components: a) the Propagation Network, which generates frame-conditioned representations (based on $\mathbf{F}$) for all candidate concepts directly connected to the active ones using Graph Attention Network v2 [18], and b) the Importance Network, responsible for computing a scalar importance value for each candidate node, given $\mathbf{F}$. At the end of each importance estimation, concepts above a predefined importance threshold are incorporated into the set of active concepts. This process is repeated for T iterations, alternating the Propagation and Importance Networks. Together the role of these networks is to perform message passing via nodes corresponding to concepts prevalent in the video. After expanding all relevant concepts $\mathbb{C}_F$ through T iterations, we generate a latent representation $\mathbf{c}_{KG}$ that encapsulates the information about the relevant objects in the scene along with their associated affordances. Intuitively, CGS allows us to extract the relevant concepts concerning the observed video and utilize them as additional domain knowledge during the action anticipation (see Sec. 3.3.2).

3.3.2 Action Anticipation with Domain Knowledge

This section introduces our main contribution, detailing how domain knowledge $\mathbf{c}_{KG}$ can be utilized for action anticipation. In particular, our architecture is motivated by [38]; however, we alter the attention mechanism of the encoder and decoder to allow for the integration of additional domain knowledge, thus, improving the contextual reasoning capabilities of the action-anticipation pipeline.

However, before we detail the novel attention mechanism in Section 3.3.3, we briefly outline the standard transformer-based part of our pipeline, consisting of an encoder $f_e(\dots)$ and decoder $f_d(\dots)$ (see Fig 3.2). The encoder $\mathbf{e}_{enc} = f_e(\mathbf{F})$ utilizes I3D [20] features $\mathbf{x}_n^{I3D}$ of the observed video $\mathbf{F}$ and produces a set of embeddings $\mathbf{e}_{enc} \in \mathbb{R}^{n \times D}$ for each video frame n and encoding dimension D. The encoder processes visual features extracted from the observed segment of a video $\mathbf{F}$ by employing multi-head self-attention. The resulting output is then provided to a classifier, $\mathbf{a}_O = f_{obs}(\mathbf{e}_{enc})$, determining the actions corresponding to the observed part of the video segment.

The decoder employs the embeddings of the observed sequence $\mathbf{e}_{enc}$ generated by $f_e(\dots)$ along with learnable tokens referred to as actions queries, initialized with zero vectors. Similarly to the encoder, the decoder $\mathbf{e}_{dec} = f_d(\mathbf{e}_{enc}, \chi)$ produces a set of embeddings $\mathbf{e}_{dec} \in \mathbb{R}^{p \times D}$ where p is the upper bound of the future actions that can be predicted and $\chi \in \mathbb{R}^{p \times D}$ are the p action queries. Subsequently, we utilize two separate, fully connected networks for predicting the future actions $\mathbf{a}_{pred}$ and their durations $\mathbf{d}_{pred}$ respectively.

$$\mathbf{d}_{pred} = f_{dur}(\mathbf{q}_n^{L_d}) \quad \text{and} \quad \mathbf{a}_{pred} = f_{act}(\mathbf{q}_n^{L_d}) \tag{3.1}$$

Model	Frame-wise (↑ larger is better) / Action Sequence (↓ smaller is better)								Next Action (↑)	
	5-10	5-20	5-30	5-50	10-10	10-20	10-30	10-50	5	10
50Salads										
KG Baseline [17]	6.92 / 4.88	6.21 / 6.20	6.01 / 7.44	5.58 / 7.92	7.13 / 4.50	6.48 / 5.98	6.07 / 7.37	5.78 / 7.88	8.0	9.0
Video-Llama [108]	- / 6.44	- / 7.20	- / 7.90	- / 9.12	- / 6.12	- / 6.80	- / 7.86	- / 9.02	6.0	7.0
GPT4-V [1]	- / 3.52	- / 5.94	- / 7.04	- / 7.12	- / 3.86	- / 4.67	- / 5.16	- / 7.05	12.0	32.0
CNN [31]	7.42 / 3.22	6.97 / 5.07	6.67 / 5.86	6.40 / 6.11	8.50 / 3.33	7.80 / 4.87	7.45 / 5.20	6.92 / 6.60	10.0	28.0
RNN [31]	7.98 / 3.00	6.90 / 5.46	6.48 / 6.30	6.42 / 6.16	8.78 / 2.94	7.92 / 4.83	7.57 / 5.20	7.26 / 6.52	12.0	30.0
FUTR [38]	8.90 / 2.98	7.46 / 4.52	7.29 / 5.40	8.63 / 6.80	15.17 / 2.74	11.34 / 4.04	11.31 / 4.98	11.36 / 6.04	12.0	36.0
NeSCA ($T = 0$)	7.95 / 3.08	7.86 / 4.42	6.15 / 5.20	7.10 / 6.58	24.0 / 2.60	16.90 / 3.72	11.17 / 4.98	11.30 / 6.80	10.0	34.0
NeSCA ($T = 1$)	**17.86** / **2.84**	**16.25** / **4.22**	**10.84** / **5.14**	**9.38** / **6.70**	**23.15** / **2.54**	**17.28** / 3.78	**16.62** / 4.76	**13.61** / **5.74**	**14.0**	**42.0**
NeSCA ($T = 2$)	13.67 / 2.90	9.60 / 4.40	8.62 / 5.32	8.51 / 6.60	22.86 / 2.56	16.86 / **3.71**	14.70 / **4.52**	12.66 / 5.75	12.0	38.0
Breakfast										
KG Baseline [17]	5.44 / 8.22	4.95 / 9.10	4.22 / 9.66	3.98 / 10.02	6.02 / 7.90	5.15 / 8.77	4.86 / 9.21	4.51 / 9.78	7.22	12.31
Video-Llama [108]	- / 11.20	- / 12.24	- / 13.62	- / 13.82	- / 11.08	- / 12.04	- / 12.98	- / 13.22	5.39	9.80
GPT4-V [1]	- / 4.56	- / 6.04	- / 6.93	- / 7.26	- / 5.12	- / 6.08	- / 7.26	- / 7.62	19.27	22.45
CNN [31]	5.76 / 6.98	5.52 / 7.22	5.45 / 7.98	4.80 / 8.43	7.84 / 6.48	6.62 / 6.95	6.02 / 7.44	5.17 / 8.13	11.45	18.90
RNN [31]	6.16 / 6.76	5.60 / 7.05	5.53 / 7.69	4.96 / 8.09	7.67 / 6.67	6.73 / 6.90	6.15 / 7.44	5.22 / 8.12	12.02	19.96
FUTR [38]	9.54 / 1.63	7.24 / 2.07	6.42 / 2.40	5.58 / 3.02	14.70 / 1.41	12.55 / 1.76	12.10 / **2.06**	11.71 / 2.62	23.97	30.05
NeSCA ($T = 0$)	9.69 / 1.65	7.20 / 2.04	6.55 / 2.40	5.62 / 3.06	15.30 / 1.43	13.23 / 1.82	12.24 / 2.22	11.65 / 2.68	20.55	25.47
NeSCA ($T = 1$)	**9.91** / **1.60**	**7.95** / **2.02**	**6.86** / **2.34**	**5.88** / **2.98**	**15.53** / 1.41	**13.52** / 1.76	**13.07** / 2.09	**11.94** / 2.63	**25.25**	26.45
NeSCA ($T = 2$)	9.75 / 1.62	7.60 / 2.05	6.70 / 2.38	5.76 / 3.00	15.52 / **1.36**	13.46 / **1.72**	12.68 / 2.15	11.84 / **2.60**	23.32	**30.35**

Table 3.1: NeSCA performance compared to the current state-of-the-art in long-term action anticipation for different horizons of $\alpha - \beta$ (top row). The numbers in boldface and underlined indicate the highest and the second-highest accuracy, respectively.

Finally, we retrieve confidence $\mathbf{c}_{pred}$ for each predicted action. We quantify the certainty of the model's prediction using negative entropy of the predicted distribution of actions, i.e., $\mathbf{c}_{pred} = \sigma(\mathbf{a}_{pred}) \log(\sigma(\mathbf{a}_{pred}))$, where $\sigma(.)$ represents the softmax function.

3.3.3 Knowledge-Guided Attention Mechanism

So far, we have discussed how relevant domain knowledge is retrieved from a symbolic KG, as well as the general action anticipation pipeline. In this section, we describe our main contribution: Altering the multi-head attention layers of the encoder $f_e(\dots)$ and $f_d(\dots)$ to improve contextual prediction by leveraging our extracted domain knowledge $\mathbf{c}_{KG}$. Intuitively, the extracted domain knowledge establishes a connection between the objects in the scene and their respective affordances, improving the predicted actions' relevance by boosting or attenuating the attention between different features. To this end, we introduce a rectification matrix $\mathbf{R}$ inside the multi-head attention equation. We obtain a separate knowledge-guided rectification matrix for our encoder and decoder, namely $\mathbf{R}_e$ and $\mathbf{R}_d$, with which we modify the attention mechanism:

$$\text{KG-Attn}_{e/d}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax\big(\frac{\mathbf{Q}\mathbf{R}_{e/d}\mathbf{K}^T}{\sqrt{d_k}}\big)\mathbf{V} \tag{3.2}$$

Boosting or suppressing features using the rectification matrix allows our model to prioritize the features associated with objects having relevant affordances, giving them higher importance than those not present in the scene. The rectification matrix is presented as a diagonal matrix for which we retrieve the diagonal by predicting it from $\mathbf{c}_{KG}$. Particularly, we a utilize $\mathbf{R}_{e/d} = f^R_{e/d}(\mathbf{c}_{KG})$ to predict each diagonal, where $f^R_{e/d}(...)$ is implemented as an LSTM.

Note that $f_e^R(\dots)$ and $f_d^R(\dots)$ are separate networks that do not share parameters amongst themselves.

3.3.4 Human-Robot Collaboration using Anticipated Actions

Having access to a sequence of likely future actions as well as their durations and confidences, we define a policy $a_r = \pi(f_\theta(\mathbf{F}), \mathbb{S})$ that chooses an appropriate assistive robot action a_r from a set of possible skills $\mathbb{S}$ (see Fig 3.3). Selecting the appropriate action $a_r \in \mathbb{S}$ is a challenging task as, upon selection of an action, the robot is committed to performing it. This commitment requires time and utilizes objects in the environment that could have been used otherwise by the human partner.

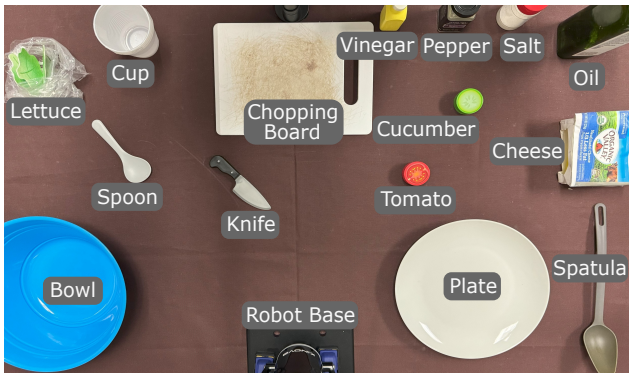


Figure 3.4: Our *Dummy Kitchen* setup and available objects for creating salads in an HRC task involve cutting/peeling vegetables, preparing dressing, and mixing/serving the salad.

Thus, we define four selection criteria to choose an appropriate action or not to choose an action at all and continue to observe the user. First, the cumulative duration of actions $d_s = \sum_{i=0}^{i=r-1}(d_i)$ for any action candidate a_r, where $0 \leq r \leq |\mathbf{a}|$ must be larger than the average length d_r of action candidate a_r (obtained from the training dataset). This constraint ensures that the human collaborator would not have done or needed to do the chosen task before the robot can complete it. Secondly, we ensure that all objects needed for a chosen action a_r, as defined in our skill library $\mathbb{S}$, are observed in our set of active concepts $\mathbf{c}_{KG}$ and that all objects have the appropriate affordances. For example, if we consider the action of cutting a tomato, the robot requires a knife, cutting board, and tomato, but also that the tomato has the affordance of being *cuttable* (i.e., is not already in a diced state, which would not afford the ability to be cut it further), and hence, these concepts should be part of the list of active concepts. Thirdly, we verify whether the prerequisites for the specific task have already been fulfilled; for instance, the action of *placing tomato in bowl* necessitates that the *cut tomato* action precedes it. Lastly, we consider the confidences $\mathbf{c}_{pred}$ for the candidate action a_r. Specifically, we only consider actions for which the estimated confidence is above a pre-defined threshold to ensure that the robot only executes the most likely actions.

With these four constraints, we define policy $\pi(\dots)$ that, given the predicted action sequence for horizon β over an observed time-horizon α, selects a single action a_r that

should be executed by the robot. However, note that if no such action that satisfies all four constraints can be found, policy π will return a no-op action. In such cases, the policy will continue attempting to identify an appropriate action as further video frames are available. Similarly, when the robot is currently committed to executing a previously selected action a_r, the robot will ignore action choices made by policy π until the prior action is completed.

3.4 Experiments

In this section, we evaluate `NeSCA` on two common benchmarks for action anticipation – *50Salads* and *Breakfast* – and demonstrate how action anticipation can be used for human-robot collaboration in a real-world task. Our benchmarks (see Sec. 3.4.1) extensively evaluate the ability to utilize short video contexts while predicting long-horizon future actions. In our real-world setup (see Sec. 3.4.2), we utilize the ability to correctly anticipate actions to facilitate the collaborative creation of a salad.

Datasets We evaluate the effectiveness of `NeSCA` using two publicly available benchmark datasets for action anticipation for in-home environments, particularly kitchen scenarios, as well as one real-world robotics dataset: 1) The *50Salads* dataset [87] with its five splits, densely annotated with 17 fine-grained action labels and three high-level activities; 2) The *Breakfast* dataset [53] with four splits, categorizing each frame into one of 10 breakfast-related activities using a comprehensive set of 48 fine-grained action labels; and 3) a dataset of 20 videos collected from our dummy kitchen setup (see Fig **??**). Among these dummy kitchen videos, we designate half of them for fine-tuning the model, while the remaining half are reserved for assessing the performance of the fine-tuned model.

Metrics To evaluate the efficacy of our approach, we calculate the *Mean over Classes* (MoC) accuracy [31]. This metric is computed by comparing the predicted actions to the ground-truth actions for all future frames within the horizon window defined by β, making it the most comprehensive metric as it captures action sequence and action durations. To quantify the ability of our model to identify the sequence of the next actions without considering their durations, we employ a metric that computes the minimum number of addition, deletion, or substitution operations required to exactly match the predicted to the ground truth action sequence. While neglecting action durations, this metric captures the semantic understanding of the task composition. Derived from this metric, we also employ immediate single next-action prediction as a metric. Finally, in our real-world setup, we utilize the accuracy of completing an action, i.e., anticipating the right action and executing it, as our primary

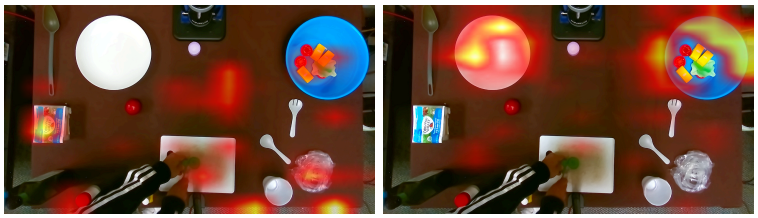


Figure 3.5: Attention to visual features relevant to our task, as attended to by FUTR (Left) and NeSCA (Right). With our re-focusing approach, attention is heightened for areas having objects relevant to tasks after the current *cutting lettuce*.

evaluation metric.

3.4.1 Action Anticipation Benchmark

Action Anticipation Performance We evaluate NeSCA by comparing the performance on all metrics averaged across all splits against long-term action anticipation baselines [31, 38], depicted in Table 3.1. [31] uses action labels extracted from the action segmentation model, while our work and the most recent state-of-the-art [38] use visual features from the observed video segments. In addition, we conduct a comparative analysis with two additional baseline methods. The first is a KG-only approach proposed by [17], which aims to extract all objects along with their associated affordances in each p^{th} frame of the video. This method incorporates a decay mechanism with a rate of γ to account for the diminishing importance of active nodes over time. Finally, we also utilize a set of multimodal fusion models, namely Video-Llama [108] and GPT4-V [1], where we begin by providing a comprehensive explanation of the entire scenario and subsequently prompt it to produce predictions for future actions from a predefined list of possibilities. As can be seen in Table 3.1, NeSCA outperforms the current state-of-the-art in long-term action anticipation using short context in all the metrics on the *50Salads* dataset and on nine out of the ten metrics we used on the *Breakfast* dataset. On the MoC metric, NeSCA outperforms the baseline by up to 9% on *50Salads* and 1% on *Breakfast*.

As our method relies on a fixed number of iterations T during CGS, we also evaluated varying numbers with $0 \leq T \leq 2$. The most favorable outcome was observed when T was set to 1. In the case of $T = 0$, no graph propagation was performed, and the model relied solely on objects detected by our object detector. As a result, its performance resembled that of [38], which lacks information about associated object affordances. On the other hand, when $T = 2$, the list of concepts considered by the model expanded beyond the context relevant to the video which resulted in the model receiving information that was redundant or unnecessary, thereby, confusing the model. Empirically, we chose a propagation of $T = 1$

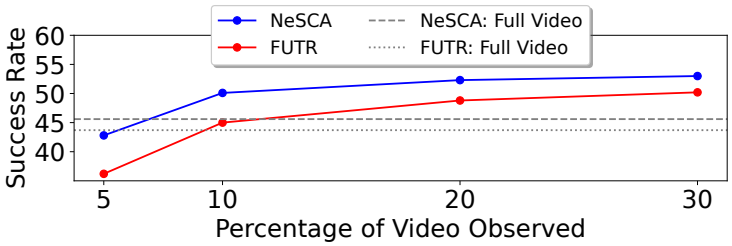


Figure 3.6: Success rate of NeSCA in our kitchen setup with varying context lengths. The observed percentage is reported with respect to the average length of finetuning videos.

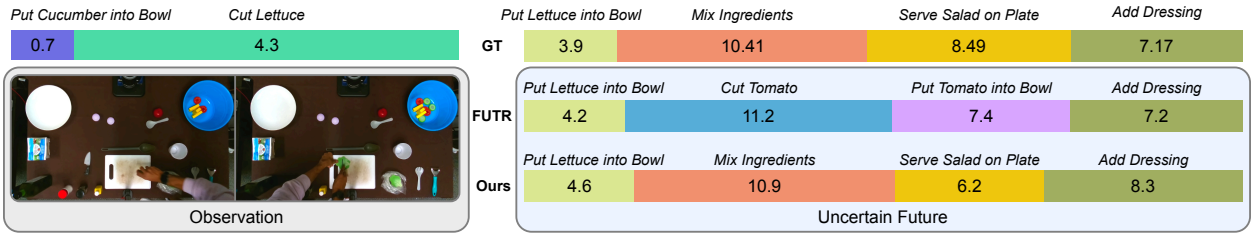


Figure 3.7: Sample result of NeSCA on the real-world kitchen setup, observing 5% and predicting another 30% into the future, along with the predictions made by FUTR [38] and the ground truth action labels.

for all our experiments.

Qualitative Evaluation We showcase an example to compare NeSCA against [38] by evaluating the time-series segmentation of the predicted future actions. Figure 3.7 depicts an example from our kitchen setup where the model observes two actions in the $\alpha = 5\%$ (≈ 6 seconds) observed segment of the video and then predicts what actions take place in the next $\beta = 30\%$ (≈ 36 seconds) of the video. While our model accurately identifies the sequence of all four ground-truth future actions and their approximate durations, the baseline approach failed to identify two out of the four actions correctly. We attribute our approach's improved performance to our model's ability to focus on the objects currently in use and objects that could be used later by extracting their associated affordances.

The re-focusing of our model is demonstrated in Figure 3.5, highlighting the areas our approach (right) and [38] (left) focuses on. Our model directs attention to both the bowl and the plate, even in scenarios where the subject is not directly interacting with them. This capability enables our model to accurately anticipate future actions, such as *put cheese into bowl* and subsequently *serve salad onto plate*. In contrast, the baseline approach indiscriminately focuses on many objects in the scene, neglecting to discern the relevant objects based on their affordances and their potential utility in the context of ongoing and completed actions.

Approach	Finetuning	Confidence	Success		MoC	
			$\alpha = 5\%$	$\alpha = 10\%$	$\alpha = 5\%$	$\alpha = 10\%$
Autoregressive			13.0	17.4	6.2	7.4
Autoregressive	✓		27.3	36.4	8.9	12.2
FUTR			16.6	20.8	6.7	9.2
NeSCA			19.2	23.1	6.9	9.9
NeSCA	✓		33.7	41.8	12.4	18.1
NeSCA (Full)			35.2	43.6	14.4	20.2
NeSCA	✓	✓	42.8	50.1	-	-

Table 3.2: Performance of the action anticipation pipeline, NeSCA, for human-robot collaboration on our kitchen setup. *Success* values represent the real-time joint performance of anticipating the sequence of actions and performing the actions in the kitchen setup, while the *MoC* values represent the accuracy of framewise prediction of actions over the collected trajectories from the kitchen setup. The average length of sequences (according to which the percentages are calculated here) is 120 seconds.

3.4.2 Real-World Human-Robot Collaboration

After showing the effectiveness of our NeSCA approach on two common baselines, we utilize it in an HRC scenario of preparing a salad in a joint task between a robot and a human user. To bridge the domain gap that arises due to the shift in physical attributes (for example, lighting conditions, the color of prevalent objects, etc.) of the real-world kitchen setup and the trained dataset, we finetune our trained model to a dataset comprising of both the original videos and 10 videos collected on our kitchen setup.

Transfer Learning on Kitchen Environment We assess the effectiveness of our fine-tuned model on our kitchen environment depicted in Table 3.2, which utilizes the same action space as the *50Salads* dataset. During inference, we provide access to a pre-defined skill library $\mathbb{S} = \{S_0, S_1, S_2 ... S_m\}$ where each high-level skill S_i corresponds to a specific sequence of low-level control inputs, conditioned on the placement of the objects. We use a top-down RGB camera (see Figure **??**) to track our objects using Scale-Invariant Feature Transform (SIFT) [63] and color feature detection. Given the skill library and video stream, our action anticipation module operates in real time to deduce future actions and their associated confidences. When the four action criteria mentioned in Section 3.3.4 are met, the respective instruction is sent to the robot for execution.

The skills in the library are broadly categorized into three "grasp types": a top-down grasp, suitable for pick-and-place actions with items like vegetables; a sideways grasp, ideal for picking up and pouring objects such as olive oil or vinegar bottles; and an aligned grasp, designed for handling oriented tools like knives and spatulas. The aligned grasp feature is

engineered to bring and hand over tools to a human collaborator. In this process, the robot brings the instructed tool near the potential area of use for easy accessibility.

For real-world experiments, we define success as the robot correctly identifying future actions and executing the respective target action. The observed % of video, denoted by α, is computed by comparing the duration of human action observed by our model with the average duration of a video in our finetuning dataset. This evaluation involves comparing its performance against several baselines, namely: (1) a non-finetuned model, (2) an autoregressive classifier that predicts the next action by considering extracted video features in addition to prior action predictions, and finally, (3) a model with the same architecture but trained from scratch on 25 videos collected in our dummy kitchen environment. While training from scratch on our dummy kitchen environment only uses 25 videos as compared to the original *50Salads* dataset, we find that providing further videos does not improve the performance of the model any further. In addition to our approach, we also compare against the best performing state-of-the-art method in long-term action anticipation, FUTR [38]. In Table 3.2, we have observed a significant performance improvement when fine-tuning the model using a few videos from our kitchen setup. Moreover, `NeSCA` consistently outperforms autoregressive baselines, underscoring the significance of leveraging not only the visual-temporal features of the video but also exploiting information about objects in the scene and their associated affordances. In comparison to a model trained on the complete dataset (see `NeSCA` (Full) in Table 3.2), our fine-tuned approach demonstrates a superior success rate and comparable frame-wise action prediction accuracy. Note that we have not presented the MoC values for our model with confidence estimation, since this is specifically incorporated into the model for real-time evaluation and is not applied in the assessment using our collected set of videos.

Further, we also evaluate the dependence of `NeSCA` and FUTR on the percentage of video observed on our kitchen setup (see Figure 3.6). As is expected, the performance of both approaches increases as the percentage of video increases, but the difference is much more pronounced when the context window is shorter. Further, the dashed line represents an approach that, instead of employing a sliding video window focusing on a specific fixed context, utilizes the entire video up to that point. By observing only 10% of the video, `NeSCA` outperforms the non-sliding window approach. This underscores the ability of `NeSCA` to draw meaningful inferences with a very short context window and highlights the impact of using uncertainty-based thresholding to improve the success rate in real-world scenarios.

Finally, we also evaluate the efficacy of our approach to encode the set of active concepts in the rectification matrix. For this, we take an example where our model accurately predicts the appropriate sequence of future actions given the fulfillment of preconditions. Subsequently,

we deliberately remove one of the crucial connections or edges within the active knowledge graph and observe the resulting change in the proposed action. In the example, our model correctly anticipates the action of *cutting a tomato*. However, upon removing the connection between the *tomato* node and the node representing the affordance *cut*, we observe a shift in the model's prediction from *cutting the tomato* to *mixing ingredients in a bowl*, reflecting the effective encapsulation of relevant concepts using the rectification matrix.

Chapter 4

Conclusion

In summary, our work emphasizes the significant potential of neurosymbolic architectures in overcoming the current limitations of deep learning models. By integrating a structured form of domain knowledge, via the integration of knowledge graphs, with neural networks, we demonstrated the ability to enhance model interpretability, generalization, and flexibility. This approach not only improves the robustness and reliability of AI systems but also enables them to handle novel entities and concepts more effectively. Through extensive experimentation, we have shown that our method outperforms existing models in tasks such as few-shot classification and action anticipation, highlighting its practical utility and effectiveness in real-world scenarios.

The integration of neurosymbolic methods into AI systems represents a promising avenue for bridging the gap between data-driven learning and structured reasoning. By leveraging the strengths of both neural networks and symbolic reasoning, we can create more versatile and interpretable AI systems capable of nuanced environmental understanding and seamless human interaction. Our findings underscore the necessity of incorporating structured domain knowledge into AI architectures, paving the way for more reliable and transparent performance in complex, human-centric environments. Ultimately, this work contributes to the advancement of AI, making it more adaptable and trustworthy for applications that require a deep understanding of the environment and effective collaboration with humans.

Chapter 5

Discussion, Limitations, and Future Work

To address our neurosymbolic concept identification model's limitation in handling numerous instances of the same concept within a scene, we propose augmenting the knowledge graph with a scene graph. This extension will enable our graph search algorithm to operate on both graphs, allowing us to consider different instances of the same concept within a scene. For example, a scene containing a refrigerator, an oven, and a stove is likely a kitchen. However, if these concepts appear in large numbers, it is more likely to be an IKEA store.

While our method of exaction of novel concepts is capable of learning to recognize various objects, abstract concepts, and affordances in a sample-efficient manner, it is dependent on the comprehensiveness of the underlying knowledge graph Additionally, the reachability of the desired target class from the initially detected concept $\mathbb{F}_{\boldsymbol{I}}$ depends on the number of propagation steps T. Further, the accuracy of the model depends on the initial object detections of Faster R-CNN (see Appendix A.10 for a brief analysis). Another factor adding to this is that `RelaTe` requires any potentially related knowledge with respect to a novel concept to be expanded by $\mathbb{I}_{SME}$ due to the prohibitive computational complexity of checking against every node in $\mathcal{G}$. In future work, we plan to address these issues by choosing the number of propagation steps dynamically, allowing for further expansions, while also exploring the options of allowing SMEs to review proposed connections that `RelaTe` introduces.

Coming to our neurosymbolic approach to action anticipation for human-robot collaboration, while our experiments demonstrate the value of action anticipation in human-robot collaboration, it is crucial to acknowledge that real-world human behavior is highly unpredictable. This necessitates the ability of action anticipation approaches to quickly and accurately predict actions from only short observations of task-relevant behavior. However,

exploring more complex methods that incorporate additional factors such as gaze, behavior patterns, or personalized action anticipation tailored to individual differences could be promising avenues for future research. Additionally, we demonstrated that augmenting action anticipation with symbolic knowledge greatly benefits the model's performance; however, our approach relies on the availability of a hand-crafted knowledge graph that encompasses relevant scene objects and their respective affordances. To address this issue, we plan on generating relevant knowledge graphs in a data-driven manner.

Additionally, we aim to enhance the human-robot interaction system with advanced zero-shot grasping capabilities [57, 58]. This will enable the model to grasp objects with complex geometries [6] and perform actions anticipated from human actions, without requiring prior training on those specific objects.

Appendix A

Supplementary: Sample-Efficient Learning of Novel Visual Concepts

A.1 Cross-Modal Attention Mechanism in `RelaTe`

In this section, we elaborate further on the cross-modal attention mechanism to fuse the linguistic concept representation $\boldsymbol{w}_c \in \mathbb{R}^{F_w}$ with the image representation $\boldsymbol{I}_P \in \mathbb{R}^{P \times F_P^2 C}$. Fundamentally, this is a standard cross-attention approach in which the word embedding is considered as query and image patch embedding as key and value. However, for completeness, we outline the process as follows. Particularly, we define

$$\boldsymbol{e}_c = f_{MC}\left(\boldsymbol{I}_P, \boldsymbol{w}_n\right) \tag{A.1}$$

The transformer encoder architecture is built as L sequential layers each composed of a multi-head cross-attention and multi-layer perceptron block where each block is preceded by layer normalization and followed by a residual connection.

The initial input to the encoder is a sequence $\mathbf{Z}^0$ of length P where each element $\boldsymbol{z}_p \in \mathbb{R}^{F_l}$ of size F_l is the computed as follows for each patch in $\boldsymbol{I}_{P[i,:]}$:

$$\boldsymbol{z}_i^0 = \boldsymbol{I}_{P[i,:]}\boldsymbol{E}_{[i,:]} + \boldsymbol{P}_{[i+1,:]} \tag{A.2}$$

where $\boldsymbol{E} \in \mathbb{R}^{(F_p^2 C) \times F_l}$ is a learnable projection matrix and $\boldsymbol{P} \in \mathbb{R}^{(P+1) \times F_l}$ is a learnable positional embedding for each patch in $\boldsymbol{I}_P$. Further, we insert a `CLS` token at the beginning of the list $\mathbf{Z}_1^0 = \boldsymbol{P}_{[0,:]}$. Provided that the word embedding $\boldsymbol{w}_c$ for the concept c, for each layer

Approach	Affordances	Attributes
CLIP (0-shot)	28.4	35.0
Flamingo (0-shot)	0	0
MiniGPT (0-shot)	18.4	24.4
Flamingo (5-shot)	30.8	47.8
Ours (5-shot)	**61.2**	**69.8**

Table A.1: Novel non-visual concept prediction in comparison to free-form text generation models.

$l \in [1, \ldots, L]$, the embedding $\boldsymbol{z}_i$ is given by the following equations:

$$\boldsymbol{z}_i^{l'} = f_{CA}\big(\text{layernorm}(\boldsymbol{z}_i^{l-1}), \boldsymbol{w}_c\big) + \boldsymbol{z}_i^{l-1} \tag{A.3}$$

$$\boldsymbol{z}_i^{l} = \text{MLP}\big(\text{layernorm}(\boldsymbol{z}_i^{l'})\big) + \boldsymbol{z}_i^{l'} \tag{A.4}$$

In the above equation, the cross-attention is computed by querying the concept embedding $\boldsymbol{w}_c$ against the patchwise encoding of the previous layer, initialized by the patches from image $\boldsymbol{I}_P$. The cross-attention module, $f_{CA}(\ldots)$, is a multi-head approach encompassing h heads. Following the standard transformer architecture, we compute $f_{CA}(\ldots)$ as follows:

$$f_{CA}(\boldsymbol{k}, \boldsymbol{v}, \boldsymbol{q}) = \text{softmax}\Big(\frac{\boldsymbol{q}\boldsymbol{k}^\mathsf{T}}{\sqrt{D_A}}\Big)\boldsymbol{v} \tag{A.5}$$

The key $\boldsymbol{k}$, query $\boldsymbol{q}$, and value $\boldsymbol{v}$ for the individual cross-attention heads are given by:

$$\Big[\boldsymbol{k} \quad \boldsymbol{v}\Big] = \text{layernorm}(\boldsymbol{z}_i^{l-1})\boldsymbol{W}_{kv}, \quad \boldsymbol{q} = \boldsymbol{w}_c\boldsymbol{W}_q \tag{A.6}$$

where $\boldsymbol{W}_{kv}$ and $\boldsymbol{W}_q$ are trainable weights and $D_A = \frac{F_l}{\beta}$, where β is a hyper-parameter. The final embedding for concept $\boldsymbol{e}_{c_n}$ is obtained by extracting the representation corresponding to the 1^{st} element in sequence $\mathbf{Z}$ after all L layers.

$$\boldsymbol{e}_c = f_{MC}\left(\boldsymbol{I}_P, \boldsymbol{w}_c\right) = \text{layernorm}(\mathbf{Z}_0^L) \tag{A.7}$$

A.2 Evaluation of Novel Non-Visual Concept Extraction

Our model was assessed for its ability to predict non-visual concepts such as affordances and attributes, in comparison to the free-form language generation baselines, namely MiniGPT-4 [111], Flamingo [9], and CLIP [78], explained in Section 2.4.2. The methodology we utilize

to obtain the predictions for each baseline is mentioned below:

- CLIP (0-shot) [78]: We evaluated a standard CLIP model by tasking it with a multi-label classification task over our concepts. The language prompt for CLIP is the list of all 316 nodes plus the novel concept node and we considered the detection to be successful if the targeted concept was part of the N most confident classes, where N is the number of the respective image's ground-truth classes plus one. As CLIP does not provide an easy few-shot learning opportunity, we only evaluated the zero-shot case.
- Open-Flaming (0-shot) [9]: We utilize Open-Flamingo in lieu of the official Flamingo, as official models are not publicly available. However, in the zero-shot case, we provided our test images and prompted Flamingo with the following query: "Does the image show an item that can contain, display, feed, sit, or transport?" for the affordances and "Does the image show an item that is edible, electric, flora, sports, or wearable?" for the attributes. We then evaluated the generated text manually to determine whether or not Flamingo detected the concept correctly. For example, we counted a response like "Yes, the image shows hot dogs with cheese on them, which are edible." as successful identification of the concept *edible*.
- Open-Flamingo (5-shot) [9]: In the five-shot use-case, we provided further context to Flamingo by providing all 25 sample images (five for each class) with their respective label to Flamingo and then prompting for a single label for each of the novel test images.
- MiniGPT (0-shot) [111]: Finally, we also employed MiniGPT-4 in order to also utilize a multi-modal GPT baseline. Here, we provided the image as context and asked the same question as in Open-Flamingo while evaluating the generated response manually.

The results for the same are presented in Table A.1. Our approach outperforms the best baseline by an average score of 26.2% on the prediction of non-visual concepts. The superior performance of our model can be attributed to its capacity to deduce non-visual concepts by connecting them with visual concepts derived from the visual inputs.

A.3 Quantitative Evaluation of `RelaTe`

In addition to the ablations of Table 2.3, we provide a quantitative evaluation regarding the ability of `RelaTe` to restoring the ground-truth KG of the VGML dataset for the 16 novel classes. Recall that we intentionally removed the test classes from the KG used in our few-shot experiments. Ideally, `RelaTe` would restore or create an even better KG through the proposed edge-addition framework. Table A.2 presents a quantitative comparison between

	Fine-tuned		KG Configuration		All Classes			Novel Classes		
	GSNN	CLF	RelaTe	O-KG	**T-1**	**T-5**	**mAP**	**T-1**	**T-5**	**mAP**
1	-	✓	-	✓	90.6	70.2	38.8	91.4	72.2	67.6
2	✓	✓	-	✓	90.2	**72.8**	41.6	91.2	**73.7**	69.0
3	-	✓	✓	-	89.8	69.8	38.5	91.6	72.8	68.2
4	✓	✓	✓	-	**90.4**	72.4	**41.7**	**92.2**	73.6	**69.3**

Table A.2: Experimental results on Visual Genome dataset.

our proposed edge addition methodology, RelaTe , and using the original knowledge graph without removing nodes corresponding to the 16 novel classes. Rows 1 and 2 demonstrate the use of the original KG (O-KG) while rows 3 and 4 denote the models that use the KG populated by our RelaTe framework. All four of these models are trained without the use of MDES on a random selection of images from the original dataset. The results demonstrate that our approach effectively incorporates novel concepts into the KG. In fact, our method outperforms the model that utilized the original KG for some metrics. This is because our approach not only restores the previously removed edges but also introduces additional connections that are observed in the SME-provided images, thus, improving performance.

A.4 Qualitative Evaluation of RelaTe

In this section, we provide examples of connections recommended by our RelaTe framework. For each novel concept, we pick 4 images and pass them through our edge addition framework to demonstrate the edges that populate into the graph $\mathcal{G}$. In Figure A.1, each example starts with the concept that was removed from the knowledge graph (red) and its initial connections (purple). The suggested connections by our RelaTe framework are shown in the green box, which was generated when the system was given a set of 4 images. These results demonstrate the effectiveness of our relation prediction approach in adding back relevant connections that are prominent in the provided images. Moreover, our model suggests some additional relations that may not have been present in the original graph but are relevant and can provide significant information about the scene content.

A.5 Countering the Classifier Bottleneck

We demonstrate the crucial role of fine-tuning the propagation network and the node bias when adding novel concepts to the graph. In Figure A.2, we plot the mAP performance on the entire VGML dataset as a function of the number of novel classes added to the system

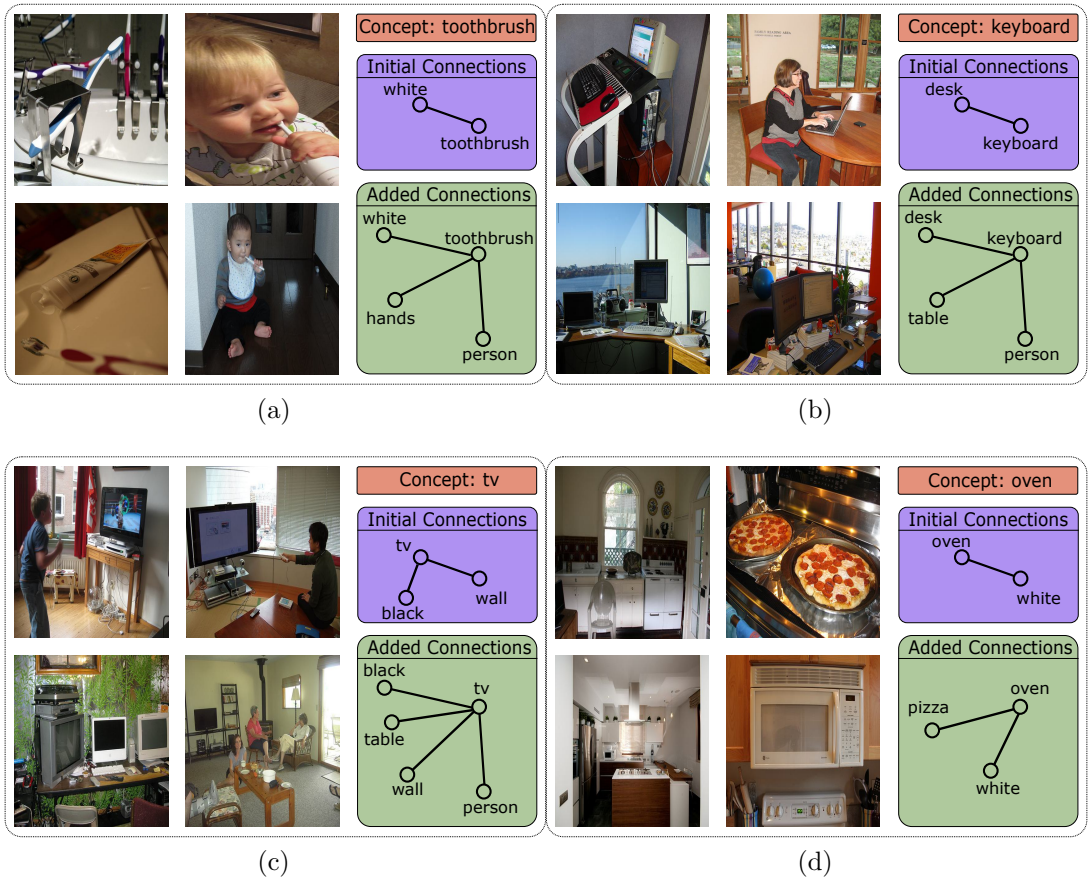


Figure A.1: Qualitative Evaluation of Edges added by the *relate* approach. In each example, we include the concept being added, the edges that were present in the knowledge graph originally, and the nodes that were suggested by *relate* for a set of images.

Steps of Expansion, T	Expansion %	mAP
2	39.5	39.1
3	93.4	**42.0**
4	100.0	41.3

Table A.3: Percentage of samples that required T steps of expansion and the corresponding mAP performance of our model with that T.

for the model where we fine-tune either only the classification or both classification and the propagation module including the node biases. During training with five images per concept, we utilize the one-by-one node addition strategy which showed improved performance (see Figure 2.5). Initially, for just a few nodes, not training the GSNN does not have a huge influence; however, the plot shows that the model in which we only fine-tune the classifier experiences a substantial performance drop which is proportional to the number of concepts added compared to the model in which both the modules are fine-tuned.

A.6 Ablation on Number of Propagation Steps

We experiment with different values for T that define the number of iterations between the propagation and importance network during inference of the GSNN module as described in Section 2.3.1. We aim to select the minimum possible value of T that ensures the complete expansion of most of the samples in our test dataset within the first T steps. In Table A.3, we report the performance of our model on all the classes of the VGML test dataset along with the percentage of samples that were expanded to full capacity by varying the number of expansion steps. The results we obtained in Table A.3 highlight that 3 is the optimal value of T since we start to obtain diminishing returns following steps greater than 3. Expanding less than two steps doesn't allow the model to experience many relevant connections while expanding beyond the third level makes it challenging for the model to identify concepts that are related to the original concepts $\mathbb{F}_{\boldsymbol{I}}$.

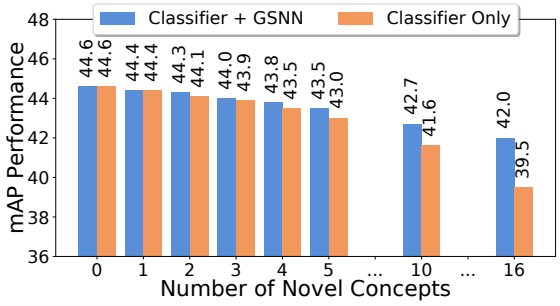


Figure A.2: Fine-tuning GSNN + Classifier vs Classifier Only

A.7 Significance of Image Conditioning on Node Embeddings

We explicitly enforce conditioning of the image content on the embeddings generated for each of the nodes during the graph propagation. These embeddings are utilized by both the importance and the context network and form the backbone of the entire graph expansion procedure. Unlike Marino et al. [67], which does not enforce this constraint, our model prioritizes expanding nodes that are relevant to the image content rather than simply expanding nodes that are only dependent on the initial class detection that would result in the same propagated nodes even for dissimilar images. We demonstrate the significance of our

(a) (b) (c)

Figure A.3: Samples from the dataset where the initial propagation begins with the concepts of *person* and *bench*.

proposed conditioning by selecting three vastly distinct images from the dataset in Figure A.3. The original GSNN fails to distinguish between these images in terms of expanded nodes in the KG, whereas our approach expands a unique set of nodes for each image. The following are the final classifications with and without image conditioning on the propagation network:

- Image 1:
  - w/o Conditioning: *person*, *bench*, *shirt*, *black*, *white*, *gray*
  - w Conditioning: *person*, *bench*, *shirt*, *wooden*, *brown*, *black*, *sunglasses*
- Image 2:
  - w/o Conditioning: *person*, *bench*, *green*, *sitting*, *shirt*, *white*
  - w Conditioning: *person*, *bench*, *jacket*, *green*, *visible*, *sitting*
- Image 3:
  - w/o Conditioning: *person*, *bench*, *shirt*, *sitting*, *pink*, *wooden*, *black*
  - w Conditioning: *person*, *bench*, *shirt*, *black*, *jacket*, *head*, *wooden*

While the model without image conditioning expands a generic list of nodes, our approach identifies image-specific concepts such as *sunglasses* for the first image and *jacket* for the second image, demonstrating the improvements imposed by this additional conditioning.

A.8 Ablation for Node types and Edge types

	Edge types	Node types	mAP
1	✓	-	42.3
2	-	-	42.1
3	-	✓	**44.6**

Table A.4: Experimental results with ablations of edge and node types.

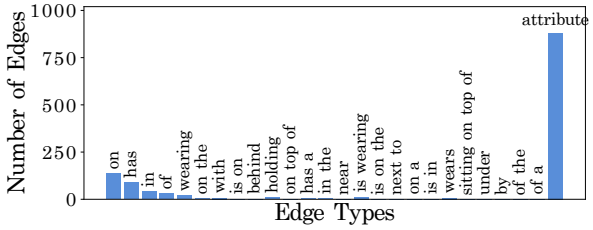


Figure A.4: Edge type distribution of the KG used by [67].

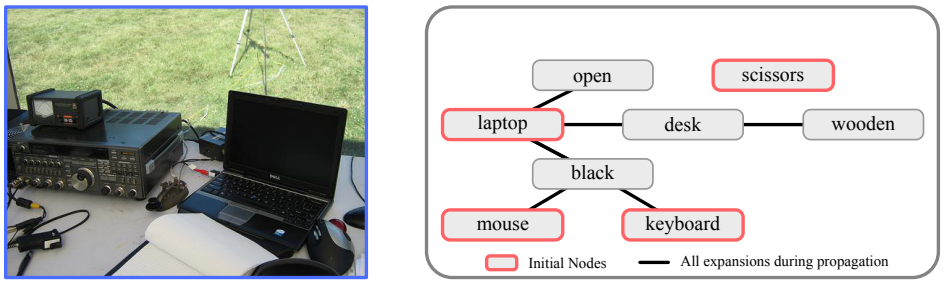
In Section 2.4.1, we introduced the changes to the KG as described in Marino et al. [67]. Here, we ablate these choices in greater detail. Table A.4 shows the performance of the algorithm with the original 26 edge types in line 1, no edge types (i.e. just a single unlabeled edge) in line 2, and our modified KG without edge types, but an additional one-hot indicating the node-type in line 3. The results show that edge types hinder the performance of the inference pipeline and indicating the node type improves performance. We hypothesize that this is due to the strong imbalance of the encoded edge types, as shown in Figure A.4, where the *has attribute*, comprising almost two-thirds of all edge types.

A.9 Maximally Diverse Expansion Sampling

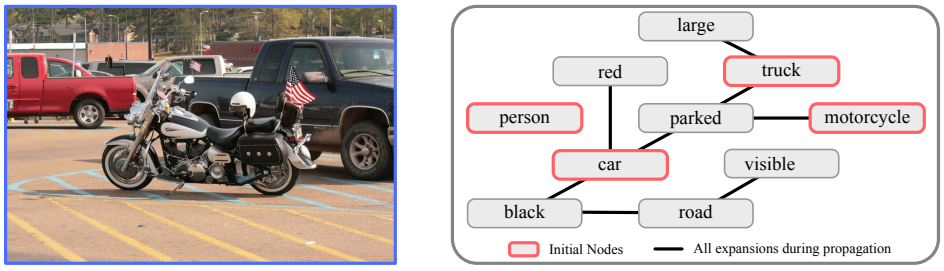
To select a small subset of the original dataset that allows us to maximize the diversity of expanded nodes in our KG, we adopt a binning-based approach. We begin with a single bin spanning all nodes and traverse the dataset to identify the image that can expand a node in the largest bin. Upon finding such an image, we use each expanded node in that image as the dividing line between the new bins. If an image does not expand a node that would divide the largest bin, the image is not added to our curated dataset $\mathcal{D}_C$. We only process the dataset once until either we have a set of images that expand all the possible nodes or all images have been either added to $\mathcal{D}_C$ or have been discarded. Under the assumption that rare classes are randomly distributed in the dataset, we ensure that at least some images containing that class are added to $\mathcal{D}_C$. As a result, we create a dataset $\mathcal{D}_C$ containing approximately 2% of the original VGML dataset.

A.10 Analysis of Dependence on Object Detectors

To test the resilience of our model against inaccuracies in the object detection module, we conducted an evaluation by replacing the detected objects with random concepts (that were not originally present in the respective example), and observing whether our model expands upon them. We conducted a small-scale experiment on 30 test images, where we introduced an additional random node that is unrelated to the actual image. We observed that the propagation and importance networks ignore these wrong nodes in 63% of the cases by not expanding them any further. Further, in 16.7% of the cases, the final classifier removes these nodes altogether. In the current work, the importance network can not remove previously added nodes; however, this capability could be explored in future work. Figure A.5 demonstrates two instances where Faster R-CNN mistakenly detects a non-existent object class in the image. Furthermore, we provide a few instances in Figure A.6 where we substitute one of the original object detections in the image with an entirely unrelated object, and our model refrains from further propagating the modified node, demonstrating its resistance to such potential issues.



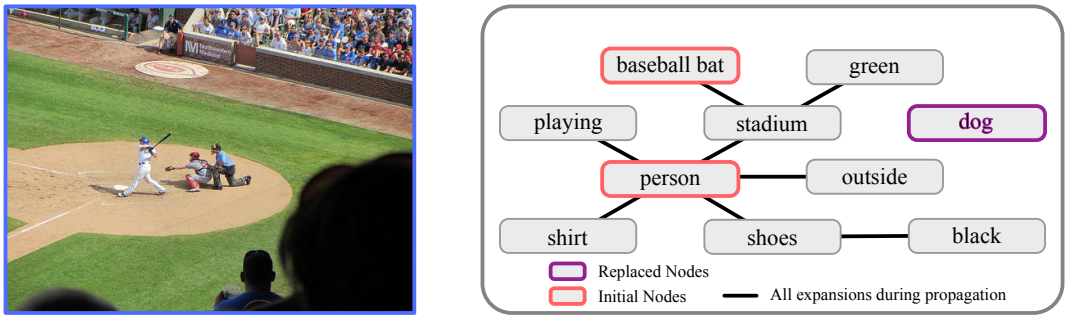
(a) Mitigated failure case: While Faster R-CNN detected a *scissors*, our propagation and importance network did not incorporate this node any further.



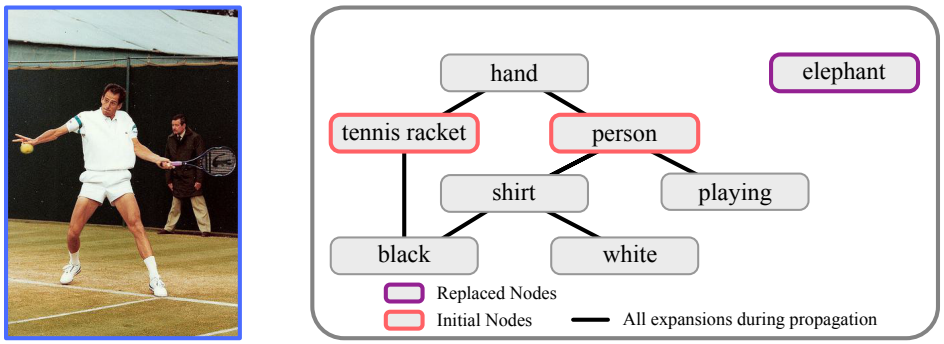
(b) Mitigated failure case: While Faster R-CNN detected a *person*, our propagation and importance network did not incorporate this node any further.

Figure A.5: Robustness to wrong graph initialization by Faster R-CNN detections.

Additionally, we also analyzed a potential failure case in which wrong edges exist in the graph. The only potential source for such edges is if `RelaTe` predicts wrong edges during novel concept addition. When testing the performance of `RelaTe` by removing a known node



(a) Node Replacement: Here, we removed the *sports ball* and replaced it with a *dog*, demonstrating how our approach does not incorporate the wrong node.



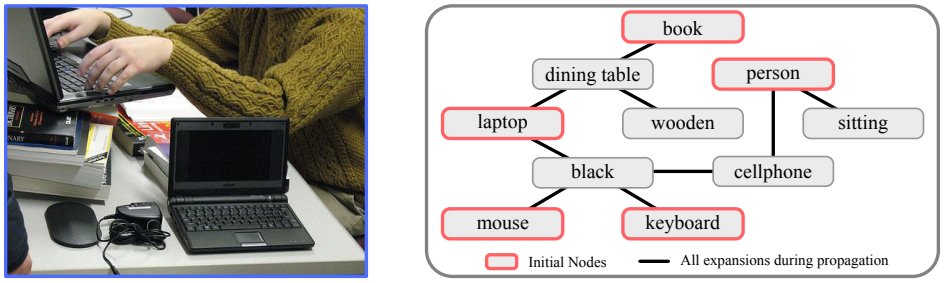
(b) Node Replacement: Here, we replaced the *sports ball* with an *elephant*. An interesting result here is that not only was the wrong node not expanded, but it was also removed from the final classification.

Figure A.6: Robustness to wrong graph initializations that are manually enforced.

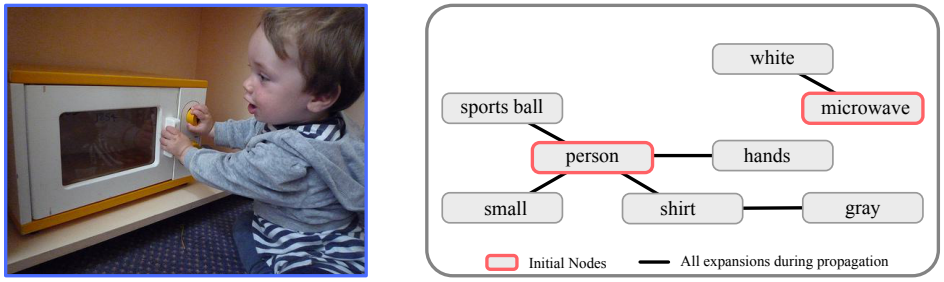
for which the desired edges are known, we observe that 84% of these edges are restored when re-adding the target node using our approach. However, while the remaining 16% of edges are not necessarily wrong, we analyzed the impact of potentially wrong edges by manually introducing them between the initially detected nodes and an arbitrary, unrelated node. This was evaluated on 30 images, as for the prior experiments. We observe that in 76.6% of the cases, the propagation and importance network ignore this wrong connection.

This highlights the robustness of our model to erroneous initialization. Moreover, we empirically observed that Faster R-CNN rarely introduces wrong nodes, thus further mitigating this potential error source.

A.11 Failure Analysis



(a) Failure case: The model identified an erroneously identified and further integrated a *cellphone*.



(b) Failure case: The model identified an erroneously identified and further integrated a *sports ball*.

Figure A.7: Failure cases of our model in which wrong nodes are integrated into the graph.

As part of our failure analysis, we highlight some examples where our model hallucinates non-existent concepts in the image. Although such misclassifications are not common, they offer valuable insights into how our approach functions and where it may be prone to errors. In the given scenario depicted in Figure A.7a, the model has incorrectly identified the concept of a *cellphone*. This error can be attributed to the model's tendency to associate objects with certain visual characteristics, which can result in confusion between objects that share common

properties. For instance, in this case, both the *laptop* and *cellphone* have a screen, and therefore the affordance of being able to display something, leading to the misidentification of the object as a *cellphone*. The second example in Figure A.7b demonstrates another instance where our model has made an incorrect prediction by identifying the object in the image as a *sports ball*. This error can be attributed to the model's tendency to rely on the way people interact with objects in the scene when identifying them. In this case, the child's hand gripping the knob of the *microwave* may resemble the way one would grip a ball, leading the model to mistakenly classify it as a *sports ball*.

Finally, we evaluate potential failure cases in which `RelaTe` may be tasked to add edges between contextually unrelated nodes. It is a key feature of `RelaTe` to automatically determine the nodes that are relevant for a novel concept while not adding edges to nodes that are contextually different. To evaluate this, we attempt to add edges between nodes from the *bedroom* context and nodes from the *stadium* context. In this case, we observe that `RelaTe` only adds an edge in 20% of the queried connections. However, it is important to note that some connections are in fact reasonable, as connections between the *person* node in the stadium context have a valid connection to *bed* in the bedroom context.

A.12 Runtime Complexity

We trained our model on a single RTX 6000 GPU for $\approx$ 100 hours of total training time. When adding a novel concept, the two-staged tuning of our model takes approximately 45 minutes. Finally, during inference, it takes approximately 30 seconds per image to obtain predictions using our approach.

Bibliography

[1] Gpt-4v(ision) system card. 2023. ??, ??, 3.4.1

[2] Yazan Abu Farha and Juergen Gall. Uncertainty-aware anticipation of activities. *ICCVW*, 2019. 3.1, 3.2

[3] Yazan Abu Farha, Qiuhong Ke, Bernt Schiele, and Juergen Gall. Long-term anticipation of activities with cycle consistency. *Pattern Recognition*, 2021. 3.1, 3.2

[4] Bilal Abu-Salih. Domain-specific knowledge graphs: A survey. *J. Netw. Comput. Appl.*, 185:103076, 2020. 2.1

[5] Henny Admoni and Brian Scassellati. Social eye gaze in human-robot interaction: A review. *J. Hum.-Robot Interact.*, 2017. 3.2

[6] Ananye Agarwal, Shagun Uppal, Kenneth Shaw, and Deepak Pathak. Dexterous functional grasping. In *Conference on Robot Learning*, 2023. 5

[7] Elahe Aghapour and Jay A. Farrell. Human action prediction for human robot interaction. In *ACC*, 2016. 3.2

[8] Mehwish Alam, Davide Buscaldi, Michael Cochez, Francesco Osborne, Diego Reforgiato Recupero, Harald Sack, Sebastian Monka, Lavdim Halilaj, Achim Rettinger, Mehwish Alam, Davide Buscaldi, Michael Cochez, Francesco Osborne, Diego Refogiato Recupero, and Harald Sack. A survey on visual transfer learning using knowledge graphs. *Semant. Web*, 13(3):477–510, jan 2022. ISSN 1570-0844. doi: 10.3233/SW-212959. URL https://doi.org/10.3233/SW-212959. 2.2

[9] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. *ArXiv*, abs/2204.14198, 2022. 2.2, 2.4.2, A.2

[10] Houda Alberts, Ningyuan Huang, Yash Deshpande, Yibo Liu, Kyunghyun Cho, Clara Vania, and Iacer Calixto. VisualSem: a high-quality knowledge graph for vision and language. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 138–152, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.mrl-1.13. URL https://aclanthology.org/2021.mrl-1.13. 2.2

[11] Amit Alfassy, Leonid Karlinsky, Amit Aides, Joseph Shtok, Sivan Harary, Rogério Schmidt Feris, Raja Giryes, and Alexander M. Bronstein. Laso: Label-set operations networks for multi-label few-shot learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6541–6550, 2019. 2.2, 2.2, 2.3.1, 2.4.1, 2.4.2, **??**, 2.4.2

[12] Paola Ardón, Éric Pairet, Ronald P. A. Petrick, Subramanian Ramamoorthy, and Katrin Solveig Lohan. Learning grasp affordance reasoning through semantic relations. *IEEE Robotics and Automation Letters*, 4:4571–4578, 2019. 2.2

[13] Qiang Bai, Shaobo Li, Jing Yang, Qisong Song, Zhiang Li, and Xingxing Zhang. Object detection recognition and robot grasping based on machine learning: A survey. *IEEE Access*, 8:181855–181879, 2020. doi: 10.1109/ACCESS.2020.3028740. 2.1

[14] Sarthak Bhagat, Vishaal Udandarao, Shagun Uppal, and Saket Anand. Discont: Self-supervised visual attribute disentanglement using context vectors. In Adrien Bartoli and Andrea Fusiello, editors, *Computer Vision – ECCV 2020 Workshops*, pages 549–553, Cham, 2020. Springer International Publishing. ISBN 978-3-030-65414-6. 2.3.1

[15] Sarthak Bhagat, Shagun Uppal, Zhuyun Yin, and Nengli Lim. Disentangling multiple features in video sequences using gaussian processes in variational autoencoders. In *ECCV*, 2020. 2.2, 3.2

[16] Sarthak Bhagat, Simon Stepputtis, Joseph Campbell, and Katia P. Sycara. Knowledge-guided short-context action anticipation in human-centric videos. *ArXiv*, abs/2309.05943, 2023. 3.1

[17] Sarthak Bhagat, Simon Stepputtis, Joseph Campbell, and Katia P. Sycara. Sample-efficient learning of novel visual concepts. *ArXiv*, 2023. 2.1, 3.1, 3.2, 3.3.1, **??**, **??**, 3.4.1

[18] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *ICLR*, 2022. 3.3.1

[19] Kaidi Cao, Maria Brbić, and Jure Leskovec. Concept learners for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2021. 2.2

[20] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 3.3.2

[21] Jiaoyan Chen, Yuxia Geng, Zhuo Chen, Jeff Z. Pan, Yuan He, Wen Zhang, Ian Horrocks, and Hua zeng Chen. Zero-shot and few-shot learning with knowledge graphs: A comprehensive survey. 2021. 2.2

[22] Riquan Chen, Tianshui Chen, Xiaolu Hui, Hefeng Wu, Guanbin Li, and Liang Lin. Knowledge graph transfer network for few-shot recognition. In *AAAI Conference on Artificial Intelligence*, 2019. 2.2

[23] 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*, 44:1371–1384, 2020. 2.2, 2.2, **??**

[24] Ching-Yao Chuang, Jiaman Li, Antonio Torralba, and Sanja Fidler. Learning to act properly: Predicting and explaining affordances from images. *2018 IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 975–983, 2017. 2.2

[25] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018. 3.2

[26] Ana Claudia Akemi Matsuki de Faria, Felype de Castro Bastos, Jose Victor Nogueira Alves da Silva, Vitor Lopes Fabris, Valeska Uchôa, D'ecio Gonccalves de Aguiar Neto, and Claudio Filipi Gonçalves dos Santos. Visual question answering: A survey on techniques and common trends in recent literature. *ArXiv*, abs/2305.11033, 2023. 2.1

[27] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2.3.1

[28] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 2.1, 2.3.1, 2.3.2

[29] Nuno Ferreira Duarte, Mirko Raković, Jovica Tasevski, Moreno Ignazio Coco, Aude Billard, and José Santos-Victor. Action anticipation: Reading the intentions of humans and robots. *IEEE Robotics and Automation Letters*, 2018. 3.1

[30] Yuan Fang, Kingsley Kuan, Jie Lin, Cheston Tan, and Vijay Ramaseshan Chandrasekhar. Object detection meets knowledge graphs. In *International Joint Conference on Artificial Intelligence*, 2017. 2.2

[31] Yazan Abu Farha, Alexander Richard, and Juergen Gall. When will you do what? - anticipating temporal occurrences of activities. *CVPR*, 2018. 3.1, 3.2, **??**, **??**, **??**, **??**, 3.4, 3.4.1

[32] Basura Fernando and Samitha Herath. Anticipating human actions by correlating past with the future with jaccard similarity measures. *CVPR*, 2021. 3.1, 3.2

[33] Antonino Furnari and Giovanni Farinella. What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. *CVPR*, 2019. 3.1, 3.2

[34] Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Predicting the future: A jointly learnt model for action anticipation. *ICCV*, 2019. 3.1, 3.2

[35] Zhe Gan, Linjie Li, Chunyuan Li, Lijuan Wang, Zicheng Liu, and Jianfeng Gao. Vision-language pre-training: Basics, recent advances, and future trends, 2022. 2.3.2

[36] Sayontan Ghosh, Tanvi Aggarwal, Minh Hoai, and Niranjan Balasubramanian. Text-derived knowledge helps vision: A simple cross-modal distillation for video-based action anticipation. In *Findings*, 2022. 3.1

[37] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. *ICCV*, 2021. 3.1, 3.2

[38] Dayoung Gong, Joonseok Lee, Manjin Kim, Seong Jong Ha, and Minsu Cho. Future

transformer for long-term action anticipation. In *CVPR*, 2022. (document), 3.1, 3.2, 3.3.2, **??**, **??**, 3.4.1, 3.4.1, 3.7, 3.4.1, 3.4.2

[39] Ian J. Goodfellow, Mehdi Mirza, Xia Da, Aaron C. Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgeting in gradient-based neural networks. *CoRR*, abs/1312.6211, 2013. 2.2

[40] Andrzej Gretkowski, Dawid Wiśniewski, and Agnieszka Ławrynowicz. Should we afford affordances? injecting conceptnet knowledge into bert-based models to improve commonsense reasoning ability. In Oscar Corcho, Laura Hollink, Oliver Kutz, Nicolas Troquard, and Fajar J. Ekaputra, editors, *Knowledge Engineering and Knowledge Management*, pages 97–104, Cham, 2022. Springer International Publishing. ISBN 978-3-031-17105-5. 2.2

[41] Yue Guo, Joseph Campbell, Simon Stepputtis, Ruiyu Li, Dana Hughes, Fei Fang, and Katia Sycara. Explainable action advising for multi-agent reinforcement learning. *arXiv preprint arXiv:2211.07882*, 2022. 2.2

[42] Muhammad Hassan, Haifei Guan, Aikaterini Melliou, Yuqi Wang, Qianhui Sun, Sen Zeng, Wenqing Liang, Yi wei Zhang, Ziheng Zhang, Qiuyue Hu, Yang Liu, Shun-Dong Shi, Lin An, Shuyue Ma, Ijaz Gul, Muhammad Akmal Rahee, Zhou You, Canyang Zhang, Vijay Pandey, Yuxing Han, Yongbing Zhang, Ming Xu, Qi Huang, Jiefu Tan, Qinwang Xing, Peiwu Qin, and Dongmei Yu. Neuro-symbolic learning: Principles and applications in ophthalmology. *ArXiv*, abs/2208.00374, 2022. 2.1

[43] 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. 2.1

[44] Xuejiao Hu, Jingzhao Dai, Ming Li, Chenglei Peng, Yang Li, and Sidan Du. Online human action detection and anticipation in videos: A survey. *Neurocomputing*. 3.2

[45] He Huang, Yuan-Wei Chen, Wei Tang, Wenhao Zheng, Qingguo Chen, Yao Hu, and Philip S. Yu. Multi-label zero-shot classification by learning to transfer from external knowledge. *ArXiv*, abs/2007.15610, 2020. 2.2

[46] Yangqing Jia, Joshua T. Abbott, Joseph L. Austerweil, Thomas L. Griffiths, and Trevor Darrell. Visual concept learning: Combining machine vision and bayesian generalization on concept hierarchies. In *NIPS*, 2013. 2.2

[47] Licheng Jiao, Jing Chen, F. 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*, 4:2–22, 2023. 2.2

[48] Michael C. Kampffmeyer, Yinbo Chen, Xiaodan Liang, Hao Wang, Yujia Zhang, and Eric P. Xing. Rethinking knowledge graph propagation for zero-shot learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11479–11488, 2018. 2.2

[49] Qiuhong Ke, Mario Fritz, and Bernt Schiele. Time-conditioned action anticipation in one shot. In *CVPR*, 2019. 3.1, 3.2

[50] Taesup Kim, Sungwoong Kim, and Yoshua Bengio. Visual concept reasoning networks.

In *AAAI*, 2020. 2.2

[51] Hema S. Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE TPAMI*, 2016. 3.2

[52] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016. URL https://arxiv.org/abs/1602.07332. 2.4.1

[53] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *CVPR*, 2014. 3.1, 3.4

[54] Brenden M. Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua B. Tenenbaum. One shot learning of simple visual concepts. *Cognitive Science*, 33, 2011. 2.2

[55] L. Lamb, Artur S. d'Avila Garcez, Marco Gori, Marcelo O. R. Prates, Pedro H. C. Avelar, and Moshe Y. Vardi. Graph neural networks meet neural-symbolic computing: A survey and perspective. In *International Joint Conference on Artificial Intelligence*, 2020. 2.2

[56] Chung-Wei Lee, Wei Fang, Chih-Kuan Yeh, and Y. Wang. Multi-label zero-shot learning with structured knowledge graphs. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1576–1585, 2017. 2.2

[57] Samuel Li, Sarthak Bhagat, Joseph Campbell, Yaqi Xie, Woojun Kim, Katia P. Sycara, and Simon Stepputtis. Shapegrasp: Zero-shot task-oriented grasping with large language models through geometric decomposition. *ArXiv*, abs/2403.18062, 2024. URL https://api.semanticscholar.org/CorpusID:268723780. 5

[58] Samuel Li, Sarthak Bhagat, Joseph Campbell, Yaqi Xie, Woojun Kim, Katia P. Sycara, and Simon Stepputtis. Geometric shape reasoning for zero-shot task-oriented grasping. In *ICRA 2024 Workshop on 3D Visual Representations for Robot Manipulation*, 2024. URL https://openreview.net/forum?id=nwjsY2OIJO. 5

[59] Ziniu Li, Kelvin Xu, Liu Liu, Lanqing Li, Deheng Ye, and Peilin Zhao. Deploying offline reinforcement learning with human feedback. *ArXiv*, 2023. 3.2

[60] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. 2.3.1, 2.4

[61] Huihan Liu, Soroush Nasiriany, Lance Zhang, Zhiyao Bao, and Yuke Zhu. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. *ArXiv*, 2022. 3.2

[62] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv*, 2023. 3.3.1

[63] David G Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999. 3.4.2

[64] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes words and sentences from

natural supervision. *ArXiv*, abs/1904.12584, 2019. 2.2

[65] Junhua Mao, Xu Wei, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Loddon Yuille. Learning like a child: Fast novel visual concept learning from sentence descriptions of images. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2533–2541, 2015. 2.2

[66] K. Marino, R. Salakhutdinov, and A. Gupta. The more you know: Using knowledge graphs for image classification. In *CVPR*, 2017. 3.1, 3.2

[67] Kenneth Marino, Ruslan Salakhutdinov, and Abhinav Kumar Gupta. The more you know: Using knowledge graphs for image classification. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20–28, 2016. (document), 2.1, 2.2, 2.3, 2.3.1, 2.1, 2.3.1, 2.3.1, 2.4.1, 2.4.2, 2.4.2, A.7, A.4, A.8

[68] Lingjie Mei, Jiayuan Mao, Ziqi Wang, Chuang Gan, and Joshua B. Tenenbaum. FALCON: Fast visual concept learning by integrating images, linguistic descriptions, and conceptual relations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=htWIlvDcY8. 2.2

[69] Antoine Miech, Ivan Laptev, Josef Sivic, Heng Wang, Lorenzo Torresani, and Du Tran. Leveraging the present to anticipate the future in videos. In *CVPRW*, 2019. 3.1, 3.2

[70] Igor Mordatch. Concept learning with energy-based models. *ArXiv*, abs/1811.02486, 2018. 2.2

[71] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. 2.2

[72] Liangming Pan, Wenqiang Lei, Tat-Seng Chua, and Min-Yen Kan. Recent advances in neural question generation. *ArXiv*, abs/1905.08949, 2019. 2.1

[73] Maithili Patel and Sonia Chernova. Proactive robot assistance via spatio-temporal object modeling. *arXiv preprint arXiv:2211.15501*, 2022. 3.1

[74] Prajakta Ganesh Pawar and V Devendran. Scene understanding: A survey to see the world at a single glance. In *2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT)*, pages 182–186, 2019. doi: 10.1109/ICCT46177.2019.8969051. 2.1

[75] Zhimao Peng, Zechao Li, Junge Zhang, Yan Li, Guo-Jun Qi, and Jinhui Tang. Few-shot image recognition with knowledge transfer. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 441–449, 2019. 2.2

[76] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162. 2.1, 2.3.2

[77] Peng Qian, Luke B. Hewitt, Joshua B. Tenenbaum, and R. Levy. Inferring structured visual concepts from minimal data. In *Annual Meeting of the Cognitive Science Society*, 2019. 2.2

[78] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language

supervision. In *International Conference on Machine Learning*, 2021. 2.2, 2.4.2, A.2

[79] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015. 2.3.1

[80] Mohammad Rostami, Soheil Kolouri, James L Mcclelland, and Praveen K. Pilly. Generative continual concept learning. In *AAAI Conference on Artificial Intelligence*, 2019. 2.2

[81] Debaditya Roy and Basura Fernando. Action anticipation using pairwise human-object interactions and transformers. *IEEE Transactions on Image Processing*, 2021. 3.1, 3.2

[82] Fadime Sener and Angela Yao. Zero-shot anticipation for instructional activities. *ICCV*, 2019. 3.1, 3.2

[83] Fadime Sener, Dipika Singhania, and Angela Yao. Temporal aggregate representations for long-range video understanding. In *ECCV 2020*. 3.1, 3.1, 3.2

[84] Md Tanzil Shahria, Md Samiul Haque Sunny, Md Ishrak Islam Zarif, Jawhar Ghommam, Sheikh Iqbal Ahamed, and Mohammad H Rahman. A comprehensive review of vision-based robotic applications: Current state, components, approaches, barriers, and potential solutions. *Robotics*, 11(6), 2022. ISSN 2218-6581. doi: 10.3390/robotics11060139. URL https://www.mdpi.com/2218-6581/11/6/139. 2.1

[85] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2.3.1

[86] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. From show to tell: a survey on deep learning-based image captioning. *IEEE transactions on pattern analysis and machine intelligence*, 45(1): 539–559, 2022. 2.1

[87] Sebastian Stein and Stephen J. McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2013. 3.1, 3.4

[88] Simon Stepputtis, Joseph Campbell, Mariano Phielipp, Stefan Lee, Chitta Baral, and Heni Ben Amor. Language-conditioned imitation learning for robot manipulation tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 13139–13150. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/9909794d52985cbc5d95c26e31125d1a-Paper.pdf. 2.1

[89] Sandor Szedmak, Emre Ugur, and Justus Piater. Knowledge propagation and relation learning for predicting action effects. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 623–629, 2014. doi: 10.1109/IROS.2014.6942624. 2.2

[90] Ilaria Tiddi and Stefan Schlobach. Knowledge graphs as tools for explainable machine learning: A survey. *Artificial Intelligence*, 302:103627, 2022. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2021.103627. URL https://www.sciencedirect.com/science/article/pii/S0004370221001788. 2.2

[91] Shagun Uppal, Anish Madan, Sarthak Bhagat, Yi Yu, and Rajiv Ratn Shah. C3vqg: category consistent cyclic visual question generation. In *Proceedings of the 2nd ACM International Conference on Multimedia in Asia*, MMAsia '20, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383080. doi: 10.1145/3444685.3446302. URL https://doi.org/10.1145/3444685.3446302. 2.1

[92] Shagun Uppal, Sarthak Bhagat, Devamanyu Hazarika, Navonil Majumder, Soujanya Poria, Roger Zimmermann, and Amir Zadeh. Multimodal research in vision and language: A review of current and emerging trends. *Information Fusion*, 77:149–171, 2022. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2021.07.009. URL https://www.sciencedirect.com/science/article/pii/S1566253521001512. 2.1

[93] Shagun Uppal, Ananye Agarwal, Haoyu Xiong, Kenneth Shaw, and Deepak Pathak. Spin: Simultaneous perception, interaction and navigation. 2024. 3.2

[94] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. NIPS'17. 3.1

[95] Sai Vemprala, Shuhang Chen, Abhinav Shukla, Dinesh Narayanan, and Ashish Kapoor. Grid: A platform for general robot intelligence development. *ArXiv*, abs/2310.00887, 2023. URL https://api.semanticscholar.org/CorpusID:263605413. 3.2

[96] Sai H. Vemprala, Rogerio Bonatti, Arthur Bucker, and Ashish Kapoor. Chatgpt for robotics: Design principles and model abilities. *IEEE Access*, 12:55682–55696, 2024. doi: 10.1109/ACCESS.2024.3387941. 3.2

[97] Hai Wan, Jialing Ou, Baoyi Wang, Jianfeng Du, Jeff Z. Pan, and Juan Zeng. Iterative visual relationship detection via commonsense knowledge graph. In *Joint International Conference of Semantic Technology*, 2019. 2.2

[98] Hang Wang, Youtian Du, Guangxun Zhang, Zhongmin Cai, and Chang Su. Learning fundamental visual concepts based on evolved multi-edge concept graph. *IEEE Transactions on Multimedia*, 23:4400–4413, 2021. doi: 10.1109/TMM.2020.3042072. 2.2

[99] Jin Wang and Bo Jiang. Zero-shot learning via contrastive learning on dual knowledge graphs. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 885–892, 2021. doi: 10.1109/ICCVW54120.2021.00104. 2.2

[100] X. Wang, Yufei Ye, and Abhinav Kumar Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6857–6866, 2018. 2.2

[101] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.*, 53(3), jun 2020. ISSN 0360-0300. doi: 10.1145/3386252. URL https://doi.org/10.1145/3386252. 2.2

[102] Yudong Wang, Ma Chang, Qingxiu Dong, Lingpeng Kong, Zhifang Sui, and Jingjing Xu. Worst-case few-shot evaluation: Are neural networks robust few-shot learners?, 2023. URL https://openreview.net/forum?id=53yQBJNQVJu. 2.1

[103] Jiwei Wei, Yang Yang, Zeyu Ma, Jingjing Li, Xing Xu, and Heng Tao Shen. Semantic enhanced knowledge graph for large-scale zero-shot learning. *ArXiv*, abs/2212.13151,

2022. 2.2

[104] Lin Xie, Feifei Lee, Li Liu, Koji Kotani, and Qiu Chen. Scene recognition: A comprehensive survey. *Pattern Recognition*, 102:107205, 2020. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2020.107205. URL https://www.sciencedirect.com/science/article/pii/S003132032030011X. 2.1

[105] Yaqi Xie, Ziwei Xu, M. Kankanhalli, Kuldeep S. Meel, and Harold Soh. Embedding symbolic knowledge into deep networks. In *Neural Information Processing Systems*, 2019. 2.2

[106] Kun Yan, Chenbin Zhang, Jun Hou, Ping Wang, Zied Bouraoui, Shoaib Jameel, and Steven Schockaert. Inferring prototypes for multi-label few-shot image classification with word vector guided attention. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(3):2991–2999, Jun. 2022. doi: 10.1609/aaai.v36i3.20205. URL https://ojs.aaai.org/index.php/AAAI/article/view/20205. 2.2, 2.2, 2.4.2, ??

[107] Renos Zabounidis, Joseph Campbell, Simon Stepputtis, Dana Hughes, and Katia P Sycara. Concept learning for interpretable multi-agent reinforcement learning. In *Conference on Robot Learning*, pages 1828–1837. PMLR, 2023. 2.2

[108] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint*, 2023. ??, ??, 3.4.1

[109] Zhong-Qiu Zhao, Peng Zheng, Shou tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30:3212–3232, 2018. 2.1

[110] Zeyun Zhong, Manuel Martin, Michael Voit, Juergen Gall, and Jürgen Beyerer. A survey on deep learning techniques for action anticipation. *ArXiv*, 2023. 3.2

[111] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models, 2023. 2.2, 2.4.2, A.2

[112] Yuke Zhu, Alireza Fathi, and Li Fei-Fei. Reasoning about object affordances in a knowledge base representation. In *ECCV 2014*. 3.1

[113] Yuke Zhu, Alireza Fathi, and Li Fei-Fei. Reasoning about object affordances in a knowledge base representation. In *European Conference on Computer Vision*, 2014. 2.2