Understanding Manipulation Contexts by Vision and Language for Robotic Vision

by

Chen Jiang

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

In Activities of Daily Living (ADLs), humans perform thousands of arm and hand object manipulation tasks, such as picking, pouring and drinking a drink. Interpreting such tasks and grasping the underlying concepts of manipulation from vision is straightforward for humans, but difficult for robotics. Recent years, fusing computer vision with natural language processing has aided in many visual understanding tasks, such as action recognition and video captioning. Despite the advances in natural image tasks, applying visual understanding methods in robotic vision has proven to be challenging.

Given the visual observations of the manipulation scene over time, we aim to estimate their visual attentions and describe the internal relational structures of all presenting manipulation concepts into a dynamic knowledge graph. In this thesis, we propose a framework to fuse an attention-based vision-language model with an ontology system. A convolutional neural network (CNN) with a spatial attention mechanism is invoked for weight feature extraction. A sequence-to-sequence structure with recurrent neural networks (RNN) is then followed, encoding temporal information and mapping from vision to command language. An ontology system, which defines the properties and attributes over various concepts of manipulation in a taxonomic manner, is inferred at last, converting command language into the intended dynamic knowledge graph and governing manipulation concepts with commonsense knowledge.

To evaluate the effectiveness of our framework, we construct a specialized

RGB-D dataset with 100K images spanning both robot and human manipulation tasks. The dataset is constructed under a strictly constrained knowledge domain for both robot and human manipulations, with annotated concepts and relations by frame. The performance of our framework is evaluated on our constructed *Robot Semantics Dataset*, plus an additional public benchmark dataset. Furthermore, ablation studies and online experiments with real-time camera streams are conducted. We demonstrate that our framework works well under the real world robot manipulation scenario, allowing the robot to attend to important manipulation concepts in the pixels and decompose manipulation relations using dynamic knowledge graphs in real time.

The study serves as a fundamental baseline to process robotic vision along with natural language understanding, thus mimicking human-like intentional behaviors and represent the evolution of an intended manipulation procedure. In future, we aim to enhance this framework further for knowledge-guided assistive robotics.

Preface

Parts of this work, authored by Chen Jiang, and supervised by Dr. Martin Jagersand, have been published and presented at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) and IEEE International Conference on Automation Science and Engineering (CASE) as:

• C. Jiang, M. Dehghan, and M. Jagersand. Understanding Contexts Inside Robot and Human Manipulation Tasks through Vision-Language Model and Ontology System in Video Streams. In International Conference on Intelligent Robots and Systems (IROS), October 2020

• C. Jiang, and M. Jagersand. Bridging Visual Perception with Contextual Semantics for Understanding Robot Manipulation Tasks. In International Conference on Automation Science and Engineering (CASE), August 2020

Acknowledgements

Firstly, I'd like to take this opportunity to thank Dr. Martin Jagersand for his guidance, for making this research possible. It has been my greatest honor to meet with such a brilliant supervisor that support my research interests. Thank you for your trust in my ability and your support in my research work.

Secondly, I'd like to express my deep appreciation to my parents, especially my father, who has been another role figure at the start of my research career. Thank you for your encouragement and support, which has been invaluable to aspects of my work and my life.

I also wish to show my thanks to everyone in University of Alberta Computer Vision and Robotics Research Group. Over the course of my master study, you guys have been amazingly inspiring. All of your helps have been of great significance to me. And I wish all of you well for years to come.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and Problem

Visual understanding of contexts in manipulation tasks is fundamental to enable human-like intelligent robots. Understanding contexts is important because humans express intention through hand motions and gestures during the process of manipulation task execution. To understand specific object manipulation contexts, we need to observe conceptual changes and describe on-scene manipulation behaviors. For example, in a liquid pouring manipulation task, its manipulation context involves a sequence of hand motions and grasps executed with a bottle and a cup. Those actions, performed by a human actor, can include: (a) grasp an object that contains the liquid, (b) move the object over an empty container, (c) pour the liquid into the empty container, (d) release the object after finishing the pouring. A core aspect of human intelligence lies in the capability to semantically utilize manipulation information, as humans process the ability to fundamentally perceive, interpret and utilize the knowledge of manipulations in the following ways:

- As a result of hand-eye coordination, humans will intentionally focus their eye gaze onto relevant regions where actions like "grasping" or "pouring" occur. The cognitive operations that select the conspicuous parts from the manipulation scene are attributed as the process of visual attention [1], directly initiated from human eye gazes. That gaze characterizes how humans perceive the visual world for manipulation

knowledge.

- The consecutive actions happening throughout the manipulation procedure form structured relationships, initiated from the actor, and mapped to associated objects on scene. The structured knowledge evolves through patterns as the manipulation task continues. Therefore, the structured knowledge of manipulation can be summarized into a taxonomy. Using a taxonomy, all manipulation tasks can be classified, sorted and organized distinctively by the involvements of various actors, actions and objects.

- Learned experience over specific scenarios will serve as commonsense knowledge that describes the necessary instructions or patterns to repeat a previously encountered manipulation task successfully. The commonsense knowledge known to humans from past experience can explain "why a human can successfully grasp this object", as it is known from common senses that "this object is graspable" without "getting hurt by hot temperature", or "simply crushing the object".

The above process can be deciphered in Figure 1.1. Eventually, for an intelligent robot to assist or to mimic humans in Activities of Daily Life (ADLs), a similar procedure of decomposing and analyzing manipulation contexts needs to be carried out first by the robot before carrying out its own motions.

Understanding manipulation contexts for robots to enact manipulation actions like humans has direct applications in building intelligent robots for daily life assistance. For example, 23% of Canadians require daily assistance with their living independence [2], and some of them might suffer nil difficulties even in grasping daily life objects. A well-versed assistance robot can for example greatly increase the efficiency of grasping for them, with the help of automatic planning based on on-scene visual observation. Cooking robots are another potential development. With robotic vision and external commonsense knowledge of cooking, robots can grasp food ingredients, observe the progress of cooking and make decisions whether to simply stir food, or to flavor the food with a specific ingredient in time.

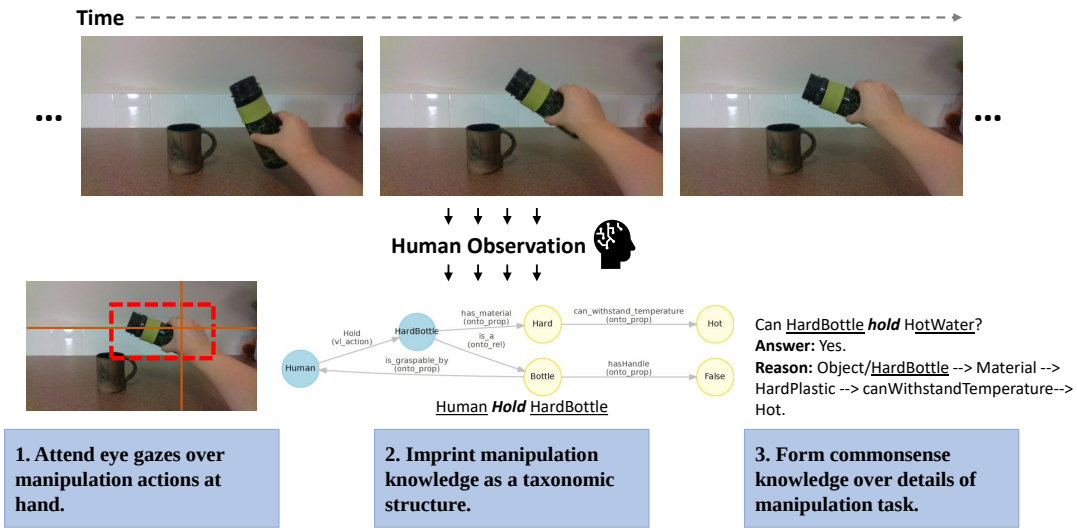


Figure 1.1: The human process of interpreting knowledge of manipulation. Firstly, from hand-eye coordination, eye gaze will provide visual attention over the manipulation scene. Then, by describing relationships among the presenting manipulation actor, action and objects, a taxonomic structure can represent the current manipulation context. At last, commonsense knowledge will allow humans to reason about manipulation concepts and deduct facts based on past experience.

Nowadays, techniques in fields of computer vision and natural language processing have provided promising tools for intelligent robots to better understand manipulation tasks and assist humans in their daily life environment. Still, intelligent robots are far from perfect. As shown in Figure 1.2, intelligent robots face many challenges when trying to perform human-like task understanding. The first challenge is the problem of robustly modeling the dynamics of any visual scene spatio-temporally. This means that, a vision model needs to capture any salient action happening in the scene throughout time. This can further be proven difficult in situations involving fast-changing or cluttered backgrounds, occlusions, or viewpoint variations. The second challenge is the problem of bridging vision and language to represent contextual information in a manipulation scene. Video captioning [3]–[10] is proven successful to compress salient actions and interacting objects into captions. However, simple captions usually lack concept-level modeling. Visual Genome [11] was proposed in later years, allowing researchers to represent the context of an entire

image scene into a structured scene graph. However, most scene graphs are intended to describe geometric location relationships like "A next to B" or "A on top B", while manipulation tasks are more action-object oriented with relationships like "pour A into B" or "hold A". Another challenge that needs to be considered can come from computational resources, where on-board processing in light-weighted robots cannot easily access heavy computational powers like methods in computer vision can. Various published works analyze human behaviors for robotics, from attending to human eye gazes [12]–[19], to structuring ways of manipulations [20]–[25]. Still, none of the studies have offered a general solution to understanding context for daily life robotics. While the process of understanding context seems rather straightforward for humans, how is it possible for robots to capture distinctive phenomenon in observation and carry out a similar process of understanding manipulation context?

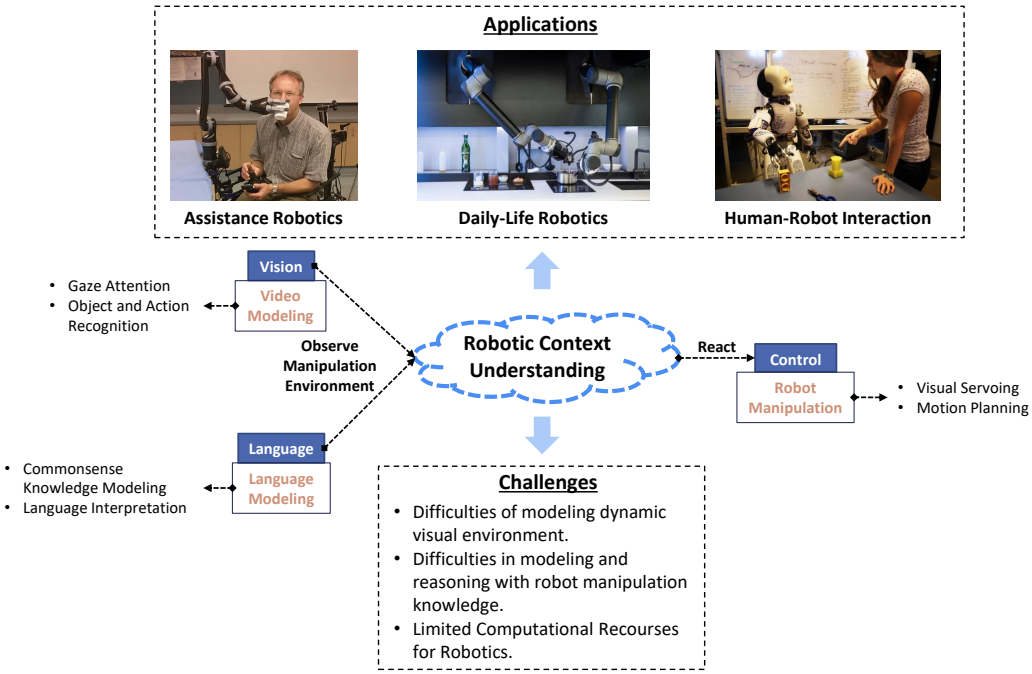


Figure 1.2: Mindmap of context understanding for robotics. Initiated from vision and language, intelligent robots need to focus on context understanding before reacting to decisions and controls.

1.2 Research Goal

On the basis of the problem and challenges discussed above, we propose to investigate the fundamental problem of visually perceiving robot and human object manipulations and interpreting these into structured knowledge. To achieve this, interpretable and explainable cues that emerge from the process of manipulation tasks need to be captured. More specifically, given the visual perception of the real-time manipulation scene and an actor, either a human or a robot, using visual attention and dynamic knowledge graph, we aim to describe three specific cues for manipulation contexts understanding: (a) what manipulation actions the actor is performing; (b) where the manipulation actions take place in the image frame; and (c) what objects do the actor takes interests in during the execution of a specific manipulation action. The process is visually available in Figure 1.3.

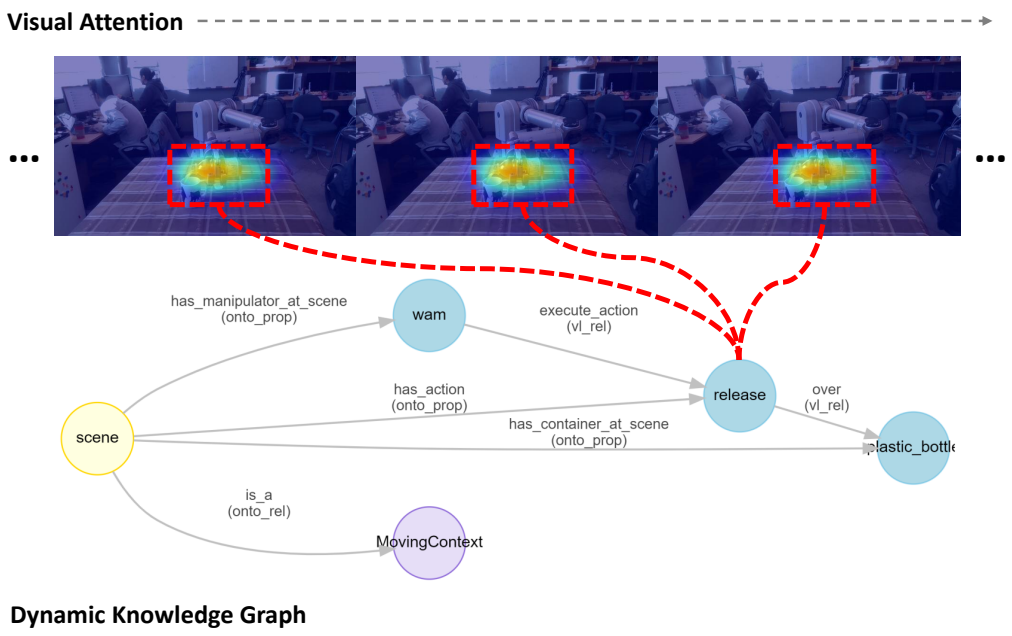


Figure 1.3: Given visual observation of a manipulation scene over time, visual attention can attend to salient manipulation actions happening throughout the scene, while a dynamic knowledge graph can describe the relational structure among concepts of actor, action and object.

Modeling modeling human eye-gaze-like with attention models allow direct and explainable visualizations over regions corresponding to salient ac-

tions and events in tasks of visual understanding. A proper attention model can intuitively explain which part of the manipulation scene a robot attends to particularly for decision making. For any manipulation task, the evolution of the task itself can be encoded by manipulation actors, actions and objects. Manipulation actors and objects are semantically connected by manipulation actions, thus allowing a manipulation task to be summarized using a semantic structure like knowledge graphs. A knowledge graph dynamically predicted by vision-language models and constrained by a knowledge domain can be helpful to structurally represent activities and events, as they evolve over time. By connecting vision and language, we can represent manipulation events through graphical connections between actions and objects from visual inputs to a dynamic knowledge graph. Furthermore, an ontology system is capable of storing pre-existing reasonable concepts, and therefore can be used to specify the underlying properties and attributes of any actions and objects, ensuring the logical correctness of our dynamic knowledge graph and allowing us to interact with external high-level knowledge, thus mimicking "human-like common senses" on robots.

1.3 Research Contents and Methodologies

Based on the research goal proposed above, the main research contents and methodologies of the thesis deal with the following parts:

- Propose a fundamental framework to capture manipulation concepts and assess manipulation intention with visual attention and dynamic knowledge graph for real time robot and human manipulation tasks.
- Construct a dataset specifically tailored to record full demonstrations of robot and human manipulation tasks, highlight key manipulation activities and events, and develop a sampling mechanism to allow training of Vision-Language models.
- Investigate the training and deploying of an attention-based Vision-Language model capable of performing spatio-temporal with the con-

structed dataset, and assess the capability of visual encoding over real-time camera streams.

- Develop an ontology system to represent commonsense knowledge over a domain of robot and human manipulation tasks, and develop an algorithm to allow real-time fusion with a pre-trained Vision-Language model and the ontology system.

- Design experiments to verify the effectiveness of our framework for robot manipulation environment, and evaluate the versatility of the generated visual attention and dynamic knowledge graph.

1.4 Significance of Research

The first important factor to represent commonsense knowledge in real-time robot manipulation lies in the connection between taxonomies and ADLs. While daily life can be composed of a significant amount of tedious, repetitive actions, those actions can actually be sorted into taxonomies. Furthermore, collaborations from robots for daily tasks can be supported by different taxonomies from specific robotic domains, such as task taxonomy for arm manipulation [26]. Those taxonomies can be used to effectively classify and describe everyday activity tasks, allowing us to capture actions of high importance and their associating motion data. As such, when developing techniques for robot manipulation, supporting specific manipulation contexts with robot taxonomies can greatly benefit our understanding of robot manipulation activities in general.

As robotic hardware develops, many robot platforms now integrate high frame-rate cameras with RGB-D capability, allowing real-time 3D point cloud rendering. How newer hardware settings influence a robotic vision system is an interesting factor to explore. Studies in computer vision have also enabled neural networks to detect objects, recognize actions or extract poses from images. Still, applications to robot context understanding suffer from factors like noisy environmental interference, restricted real-time computational re-

sources, or even human to robot knowledge transferability. This motivates us to investigate computer vision methods that integrate seemingly onto robotics.

Learning robust action state representations is another important motivation for this thesis, and why the methods presented are conditioned on real-time camera streams. A successful vision-guided robotic control policy should carefully encode visual inputs in a way that visual disturbances like occlusion or lighting variation do not significantly distract network decisions, and most ideally, the generated features should be useful to state representation for a reinforcement learning algorithm. In successful cases, the activations or the attentions of those visual encoders should focus to localize salient actions or objects that are task dependent, or have significant implications over the intention of manipulation task in time. As such, we are motivated to investigate architectures that can perform spatio-temporal encoding in real time.

1.5 Thesis Organization

The organization of the thesis is as follows: Chapter 2 summarizes the advances in the fields of computer vision and robotics for context understanding. Chapter 3 presents our framework and discusses the associated enabling methods. Descriptions of dataset and formulations of the methodology are presented in Chapter 4. Experimental details and analysis are conducted in Chapter 5. and we draw the final conclusion and summarize possible future works in Chapter 6. Our code and collected dataset are publicly available at: `https://github.com/cjiang2/rs_concepts`.

Chapter 2

Related Works

Our work on understanding manipulation contexts in robotic vision integrates areas of computer vision, natural language processing and knowledge representation. More specifically, the task of understanding manipulation contexts can be summarized as:

- Integrating vision and language.
- Attending to visual cues.
- Representing knowledge in robotics.

Moreover, related datasets of video understanding must be categorized first against different ways of vision and language integration. In this chapter, we summarize and discuss prior work done in the computer vision and robotics community for context understanding.

2.1 Datasets for Understanding Manipulation Contexts

A significant number of computer vision methods rely on a fully annotated dataset to enable joint vision and language understanding. Numerous datasets of instructional human activities and tasks [27]–[29], [12]–[15], [30], [31], [8], [32]–[34] have been proposed over the past years, categorized by fields of visual understanding tasks. We review the main typical datasets and highlight their challenges in this section.

The most dominant field has been action recognition. There are a handful of datasets specifically tailored to recognize instructional actions. One typical dataset of action recognition is Something-Something dataset, proposed in Goyal et al [27], where short clips of 220,847 videos with 174 action labels are available spamming basic actions that occur in the daily world. EPIC-KITCHEN dataset, proposed in Damen et al [28], [29], is another typical dataset of recognizing actions and activities. EPIC-KITCHEN dataset contains 90K vocabulary of human kitchen actions in 700 variable-length videos, recorded in first-person basis. Apart from focusing on the instructional nature of human activities for construction, GTEA dataset series [12]–[15] shared some interesting insights for human hand-eye coordination at that time, when eye gaze data is captured by eye tracking devices, along with the recorded instructional videos in egocentric view. The proposal of the GTEA dataset series enabled interesting applications for attention modeling. Other typical instructional video datasets include, YouCook2 dataset [30], HowTo100M [31], etc. Additionally, Kinetics Dataset [35] is widely recognized as the go-to dataset for pre-trained video action models before experimenting with the above datasets.

Applying video captioning in instructional task videos can be seen as an extension of the action recognition, as captions required for predictions contain the salient actions while capturing more attribute or descriptive details of the scene and objects. IIT-V2C dataset, proposed in Nguyen et al [8], was constructed on top of the Breakfast dataset [32] with 419 cooking videos. While the Breakfast dataset was originally used for action recognition, the author re-purposed the dataset for video captioning. By watching videos and using predicted captions as instructional commands, a humanoid robot could indirectly mimic to perform various manipulation tasks in human way.

While datasets like Visual Genome [11] accelerated the developments of techniques that allow fusion between computer vision and knowledge modeling, modeling knowledge base and performing high-level visual grounding from instructional task videos has still been a rather new rising field for video understanding. The most recently proposed dataset requiring knowledge base modeling on human activities is the Action Genome dataset, proposed in Ji et

al [33]. The dataset is built on top of Charades dataset [34] with nearly 10k videos, and it spans challenges of action recognition, few-shot action recognition and spatio-temporal scene graph prediction. There are 0.4M objects and 1.7M visual relationships annotated in total, making Action Genome dataset the largest dataset with rich annotations of human-object relationships known at this point.

The choices of datasets for human video understanding have been rich, thanks to numerous contributions from the computer vision community. While those datasets made significant contributions under natural imaging tasks, they are weakly tied to robotic vision. Human factors have played a huge role in those datasets, but little traces of robot or human-robot interaction elements can be presented. Also, those datasets are usually annotated simply with semantic class labels. And few of those datasets are annotated with knowledge bases that ground important features for robot manipulation, or represent manipulation actions and activities spatio-temporally.

2.2 Integrating Vision and Language

There are mainly three popular fields of research that integrate vision and language for video task and context understanding: Action Recognition, Video Captioning, Visual Relationship Detection with the extensive Scene Graph Generation. In this section, we start by discussing the relevant studies under those fields and then denoting their relevance in robotics.

2.2.1 Action Recognition

Action recognition is one of the most well-studied topics in computer vision, where a global action label is required for prediction for a given sequence of video. To capture the instructive nature of manipulation action, videos of manipulation tasks are usually recorded in egocentric view. The earliest methods focused on recurrently encoding spatio-temporal features over short clips of those egocentric manipulation actions. A typical work was Sudhakaran et al [36], where a ConvLSTM module coupled with class activation maps was

proposed to encode videos of egocentric activities with spatial attention cues. The class activation maps generated served as soft masks over the manipulation background and highlighted manipulation actions. The ConvLSTM encoded clips of video frames recurrently and outputted the egocentric actions to be recognized. In Lu et al [18], LSTM was integrated into the main network architecture, where for each consecutive video frame inputted, the CNN network along with the LSTM module could capture both spatial and temporal information from both appearance and motion stream. Some other works involving modified LSTM modules were Furnari and Farinella [37], Sudhakaran et al [38], etc. Li et al [39] was another typical work where LSTM could be used extensively to assess the quality of manipulation spatially and quantitatively with attention maps and metric scores.

Over the years, computer vision community has moved from simple recurrent networks to 3D convolutional neural networks [40]–[42], due to the fact that LSTMs are usually harder to train on video data and temporal information can be better modelled through 3D convolutions. With the support of large datasets for instructional human activities [31], training by joint video-action embedding method with large network architectures like S3D [42] became possible. However, those 3D CNN architectures are large, and take extensive hardware resources for training and inference. TinyVIRAT proposed by [43] was an interesting work to tackle the intensive computational resource demanding situation, where 3D encoder-decoder based CNN architecture was applied over low resolution action data.

Transformer [44] is an attention-based encoder-decoder architecture proven adaptive not only on natural language tasks, but on image recognition tasks as well. One typical work done combining action recognition with transformers was Gridhar et al [45] where transformer was adapted to not only predict, but to localize human actions and actors by pixels. TimeSformer proposed in Bertasius et al [46] was a brand new line of work, where transformer was proven effective in encoding spatio-temporal information.

While action recognition serves as a well-studied topic in the computer vision community, however, there has been a very limited number of works on

action recognition from robotic vision. One work was Pohlt et al [47], where an I3D network was used to monitor the process of human-robot interaction. Mason et al [48] explored the connection of human and robot activities and was able to identify key elements in human activities for robotic control. One reason for a significantly lower popularity in researching action recognition for robotics could be the overwhelming human factors involved in action recognition datasets, leaving little robotic elements in the participation of daily life activities.

2.2.2 Video Captioning

Video captioning can be seen as a more fine-grained extension of both image captioning and action recognition, where a vision-language model is required to compress salient video information into descriptive language. Here, we survey the typical works of video captioning done.

Due to the sequential nature of video and caption sequences, encoder-decoder architectures were widely explored for video captioning in the computer vision community. By encoder, recurrent modules like LSTM, or 3D CNN modules were utilized to encode spatio-temporal information from the video inputs. Then the decoder accepted the state representations from the encoder, decoding visual scenes into linguistic sentences that describe the actions on scene, along with the highlighting objects that enrich the description. Donahue et al [49] proposed to apply image CNN networks as universal visual feature extractor, followed by a decoding LSTM for caption generation. Adapted from NLP tasks, Venugopalan et al [50], [51] proposed to generate video descriptions of the event from a video clip using sequence-to-sequence (seq2seq) architecture [3]. Gao et al [4] applied a joint video and language model for learning robust embedding features. With the success of applying attention mechanism on top of visual encoding processing [52], similar works of generating attended spatio-temporal cues were applied maintaining good performance for video captioning in general. A typical work was Zhao et al [5], where additive attention mechanism was applied in depth of encoder-decoder architecture. Transformer also made its efforts in video captioning. Zhou et al

[6] proposed to use transformer for end-to-end video captioning training. Fang et al [7] proposed to generate commonsense-enriched descriptions.

Utilizing video captioning for robotic manipulation is an interesting research perspective that emerged in recent years. In Nguyen et al [8], [9], a sequence-to-sequence based video captioning method was first adapted to translate videos to commands, allowing the robot to understand various manipulation tasks and perform them simply by watching an input video. Yang et al [10] improved the effectiveness of video2command, assisting a dual-arm robotic system to imitate more complex and grounding skills from human demonstrations. Still, vision-language models need to be evaluated under a more realistic context, where immediate feedback is constantly requested from real-time robotic vision.

2.2.3 Visual Relationship Detection and Scene Graph Generation

Visual relationship detection involves the localization of a pair of objects and the detection of a predicate relationship between the object pair on a visual scene. The predicted visual relationship is summarized as a triplet of (object1, predicate, object2). In Lu et al [53], a visual module was first used to generate object proposals. The proposed objects would be sorted into pairs and inputted into a language module, predicting the most probable relationship. Zhuang et al [54] proposed to build an adaptive classifier for predicate classification, based on the global context of an image. Materzynska et al [55] proposed to composite action with objects which can be treated as an adaptation of visual relationships for human action recognition.

Scene graph generation can be seen as a more complex extension of visual relationship detection, where triplets of visual relationships are collected into a scene graph $G = \{B, O, R\}$, with B representing the bounding boxes of the localized objects, O representing the collection of objects, and R representing the collection of predicates. With the proposal of Visual Genome dataset [11] and the development of object detection, research on scene graph generation for natural images became more organized and generic. A typical work was

Neural Motifs, proposed in Zellers et al [56], where predicate classification could be done by detecting objects using an object detector, then encoding object and global context with BiLSTMs. Aditya et al [57], [58] proposed to construct scene graphs using semantic parser, filling scene graphs with commonsense knowledge. However, most of the works on scene graph generation focused on images only. It was not until the proposal of Action Genome [33] that the problem of scene graph generation was more thoroughly considered on videos. In Action Genome, a spatio-temporal scene graph was required for prediction, where objects with their category labels and bounding box locations, and human-object relationship instances are involved.

Compared to other widely researched video understanding methods, the field of visual relationship detection is relatively new. Some studies have been conducted where visual relationships are utilized for robotic visual understanding. One example is Zhang et al [59], where scene graphs are constructed over robotic grasping scenes specifically. Still, visual relationships are considered to be more action-oriented for robotics. The most typical example was manipulation action tree banks proposed in [60], which could be seen as an early adaptation of representing visual relationships specifically for manipulation actions. Later studies [61]–[63] involved using object detection systems to compose action manipulation trees from manipulation action videos. Those action manipulation trees could be inputted into a humanoid robot as direct manipulation commands. Other slightly relevant studies in robotics included processing visual relationships to ground for important spatial information, letting the visual relationships be bounded by trees, triplets or even captions. Works like Hatori et al [64], Shridhar and Hsu [65], Thomason et al [66], and Yan et al [67] discussed the importance of grounding manipulation scene in general for robotic control or human-robot interaction. However those works focused more on the fields of grounding manipulation scenes with the already available semantic information rather than detecting and summarizing semantic information into visual relationships.

2.3 Attending to Visual Cues

Attention mechanisms enable models to encode spatial information and provide transparent visual explanations for deep neural networks. Studying attention is important to understand the psychological characteristics of human vision, and to generalize those characteristics onto robotic vision. There are mainly two divisions in applying attention: implicitly-learned attention models and attention models with class activation maps. In this section, we summarize the development of attention models in those two major fields and denote their applications in robotics.

2.3.1 Implicitly-learned Attention Models

The implicitly-learned attention models first came into popularity in neural machine translation tasks. By implicitly-learned, a mechanism f_{att} is usually defined, which applies an grid weight of alignment over any inputted network feature, thus giving "attention" to the specific grid regions. Bahdanau et al [68] proposed an additive mechanism to implicitly learn to attend to words of importance. Later in Xu et al [69], the additive attention model was proven to be adaptive on image tasks, where attention weight could be "softly" placed among spatial regions corresponding to the decoding word. There were studies [39], [5] that adapted the additive attention model onto the specific problems of action recognition, video captioning, etc. The learnable attention models could also be replaced by more complex architecture like 2D convolutional modules, one being most typical was CBAM [70]. Some other examples of applying convolutional modules for video understanding were Meng et al [71] and Fan et al [72].

Integrating human gaze data with attention models is a popular variant of learning attention in general. Yu et al [16] proposed to supervise the attention model with human gaze data, enhancing the performance of video captioning in general. Works like Li et al [17], Lu et al [18], and Min and Corso [19] embodied more probabilistic approaches to learn attention from human gaze data. Vision-based transformer [45], [46] is a natural and more advanced progression

of the attention models due to the stacking of self-attention modules, and they have been proven as powerful in extracting spatio-temporal information when compared to traditional attention models.

In robotics, some works can be seen where attention models enable denoising of a more cluttered environment, and therefore more focused robotic grasping. Abolghasemi et al [73] proposed to generate task-focused attention which significantly enhances the performance of its visuomotor network against the visual disturbance. Ramachandruni et al [74] proposed a multi-level attention module to learn task-focused features for robot imitation.

2.3.2 Attention Models with Class Activation Maps

Another type of attention model is formulated for the purpose of explaining the internal mechanisms of convolutional neural networks. Those attention weights are denoted as class activation maps (CAMs). Any CAM covers regions of object of interest that are discriminative for the decision of a convolutional neural network. Studies like CAM [75], Grad-CAM [76], Grad-CAM+++ [77] and Score-CAM [78] have extensively studied the formulation of acquiring such class activation maps from network activation features. The implicitly-learned attention models have also exerted their own influences over the CAM methods, where studies have proposed to enable CAMs to be differentiable and transferable with respect to the visual understanding task at hand. Li et al [79] proposed to make the network's CAMs trainable in an end-to-end fashion. Ramanishka et al [80] proposed to generate caption related saliency maps for videos in a similar fashion to Grad-CAM. Sudhakaran et al [36] proposed to use CAMs to guide recurrent modules to focus on egocentric actions in videos. While CAMs offer good performance of explaining network decisions in visual cues, it is more utilized as an offline evaluation tool. In an online learning environment, the implicitly-learned attention models are usually more preferred.

2.4 Representing Knowledge in Robotics

While complex actions can be decomposed by vision models, robots still need to enact and plan for actions with some generic schemes that can take advantage of the knowledge of manipulation actions, objects and tasks. Furthermore, methods are required to automatically associate executable robotic signals with linguistic actions and skills. In this section, we first investigate specifically how the fields of robotics represent the task evolution when performing complex manipulation tasks. We then investigate some studies that model commonsense knowledge of manipulation tasks in robotics.

2.4.1 Commonsense Knowledge for Robotics

By representing commonsense knowledge over a set of known manipulation objects, robots can utilize those pre-stored knowledge and execute manipulation tasks according to the constraints or relations imposed. Various methods have studied the effect of introducing commonsense knowledge for robotic behaviors. RoboBrain, proposed in Saxena et al [81], stored different sources of robot manipulation information as knowledge bases. Data is captured, including symbols, natural language, haptic senses, robot trajectories, visual features and many others. Misra et al [82] defined task instructions using logical forms. Paulius et al [83], [84] constrained robot manipulation tasks as knowledge graphs and defined taxonomies for generic manipulation tasks. In Petrich et al [26], taxonomies were investigated in ADLs, giving insights for arm manipulation with various types of manipulation objects and actions. While representing commonsense knowledge enables understanding of robotic behavior, how to combine vision with common senses to enable robot cognition is still a very challenging problem. And few of those studies have formally discussed a universal strategy in modeling common senses for robot intelligence.

2.4.2 Task Evolution in Robotics

The evolution of manipulation tasks were widely studied in fields of Learning from Demonstration, Imitation Learning, etc, where robots were usually re-

quired to execute a task of hierarchical nature with a set of sub-actions and skills. It should be denoted that, representing taxonomies of manipulation task evolution itself can be seen as modeling task-specific common senses.

Task evolution can be bounded by semantic structures. The earliest semantic structure widely adapted for direct robotic execution control was semantic trees or graphs, as in studies like Yang et al [85], Zhang et al [86], Welschehold et al [87], and [20]. The common point of those studies is that, methods of visual relationship detection are usually applied here, as the process of generating semantic trees or graphs is regarded as describing entities and relations into sets of visual relationships. Additionally, studies like Fox et al [21] parsed semantic trees from human demonstrations for robot imitation learning. Strudel et al [22] hierarchically arranged policies to achieve a task with different skill levels with reinforcement learning. State transition graphs were another type of widely adapted semantic structure, where a series of robotic decisions were usually assumed to satisfy Markov property. Lee et al [88] discussed a scheme to extract transition graphs from human activities. Takayanagi et al [23] formulated the transitions of actions and states to complete a task. Behavior trees [24], [25] were developed more recently, where manipulation tasks can be semantically composed and executed. While studies in robotics have no problem applying semantic structures to represent task evolution, few studies have demonstrated formal systems to interpret task evolution with real-time robotic vision. By interpreting task evolution online on scene, we can enable a robot system to perform dynamic decision making, based on the current manipulation action.

2.5 Summary

In this chapter, we reviewed and discussed the relevant literature in fields of computer vision, robotics and knowledge representation. We discussed studies in action recognition, video captioning, and visual relationship detection, while their adaptation to interpret instructional manipulation tasks were denoted. We then investigated attention models and denote their contributions

in the increasing explanaibility for modeling manipulation tasks from vision. Knowledge representation in Robotics was discussed at last, where we denoted important semantic structures utilized by popular research for robotic controls. Various studies have supported the fact, that understanding semantics and contexts in manipulation actions is the key to allow robots to learn the executions of identical or similar manipulation tasks intelligently.

Chapter 3

Framework

Given a visual observation of a robot or human manipulation scene, we aim to capture its visual attention and describe the manipulation concepts and their internal relational structures into a dynamic knowledge graph. In this chapter, we formulate the fundamental definitions for understanding manipulation contexts in robotic vision, and discuss how to associate visual observations with commonsense knowledge. We then propose our framework and discuss its enabling methods.

3.1 Definitions

3.1.1 Visual Observations

To capture visual observations of a robot or human manipulation scene, a camera stream is involved. A camera stream CS_{T_1}, from start time T_1, observes a scene of a robot or human performing a sequence of actions $A = \{a_1, a_2, ..., A_m\}$ with a set of objects to complete a manipulation task. The camera stream produces an indefinite sequence of image frames $I_{T_1}, I_{T_2}, ...$ until the camera stream stops observing at time T_t. Consequently, we denote the stop time as T_t, and a video Vid_t of length t can be preserved in the form $Vid_t = \{I_{T_1}, I_{T_2}, ..., I_{T_t}\}$, which captures a full demonstration of the manipulation task involving the sequence of actions A.

3.1.2 Manipulation Knowledge

As the product of human thinking, manipulation knowledge can be treated as logical associations of different concepts in a knowledge domain. Those concepts and logical associations are describable by relational language. Specifically, for any two entities $e_i, e_j \in E$, a relation $r \in R$ can be imposed over them, forming a labelled directed graph $lc \in LC$:

$$e_i \xrightarrow{r} e_j \in LC \tag{3.1}$$

where $E = \{e_1, e_2, ..., e_n\}$ defines the concepts of manipulation, and $R = \{r_1, r_2, ..., r_m\}$ defines the relations of manipulation concepts.

The set of linguistic entities E covers the complete linguistic vocabulary over the manipulation domain knowledge. This includes human, robot, objects and any other concepts of importance that can be used to describe any piece of manipulation knowledge.

The set of linguistic relations R logically associates concepts of manipulation among each other. In summary, relations can be composed of two divisions. First division is governed by performing actions and their associating concepts during the interactive manipulation scene, for example, $robot \xrightarrow{hold} plastic\_bottle$, $human \xrightarrow{pour} water$, etc. In this case, a labelled directed graph represents Entity-Relation-Entity (E-R-E) knowledge. Second division is governed by hierarchical or relational definitions between concepts, for example, $plastic\_bottle \xrightarrow{isA} Bottle$, $water \xrightarrow{canPresentIn} PourScene$, and attributes or properties of any individual concept, for example, $milk\_can \xrightarrow{hasMaterial} Paper$. In this case, a labelled directed graph represents Entity-Attribute-Value (E-A-V) knowledge.

3.1.3 Dynamic Knowledge Graph

For any visual observation initiated from a camera stream CS_{T_1}, a set of objects, spanning the set of entities E, will be presented on scene over time. A robot or human interacts with those objects according to a sequence of k actions $A = \{a_1, ..., a_k\}$ for the specific manipulation task at hand, producing

action relations spanning LC. As such, a relational structure can naturally be used to logically describe the interactive nature of the commonsense knowledge in total. More specifically, a dynamic knowledge graph associated with time attributes can be invoked. A dynamic knowledge graph $G_{T_1,END} = (N_G, E_G)$, initiated from time T_1, is a form of spatially connected labeled directed graph, where nodes $N_G \subseteq E$ and edges $E_G \subseteq LC$. A dynamic knowledge graph is the "blood" of information processing in robotic context understanding, and it is capable of describing the status or the evolution of manipulation knowledge over any time period.

Originated from visual observations of the manipulation scene, a dynamic knowledge graph should fundamentally be composed by action relations in order to capture current manipulation events and to assess manipulation intention with semantic knowledge. Therefore, E-R-E knowledge should serve as the skeleton of the dynamic knowledge graph. However, a skeleton is incomplete without details or characteristics of any manipulation concepts presented, and it is necessary to capture the properties and attributes for concepts associated by action relations. As such, sets E-A-V knowledge need to be appended in order to complete the skeleton into a fully-grown dynamic knowledge graph.

3.2 Framework and Executive Logic

Guided by visual observation and manipulation knowledge, we propose our framework to generate a dynamic knowledge graph over a time period of visual observations. Figure 3.1 presents the overview of our framework. The workflow of the framework is as follows:

- **Encode Visual Perception From Real-Time Robotic Vision:** A live camera stream is set up to observe the scene, where a robot or human performs a sequence of actions to complete a manipulation task. A vision-language model is pre-trained on a dataset of manipulation task videos, where observation clips are generated as training samples by a video stream sampling algorithm offline. During the online inference process, the pre-trained vision-language model is inferred, encoding visual

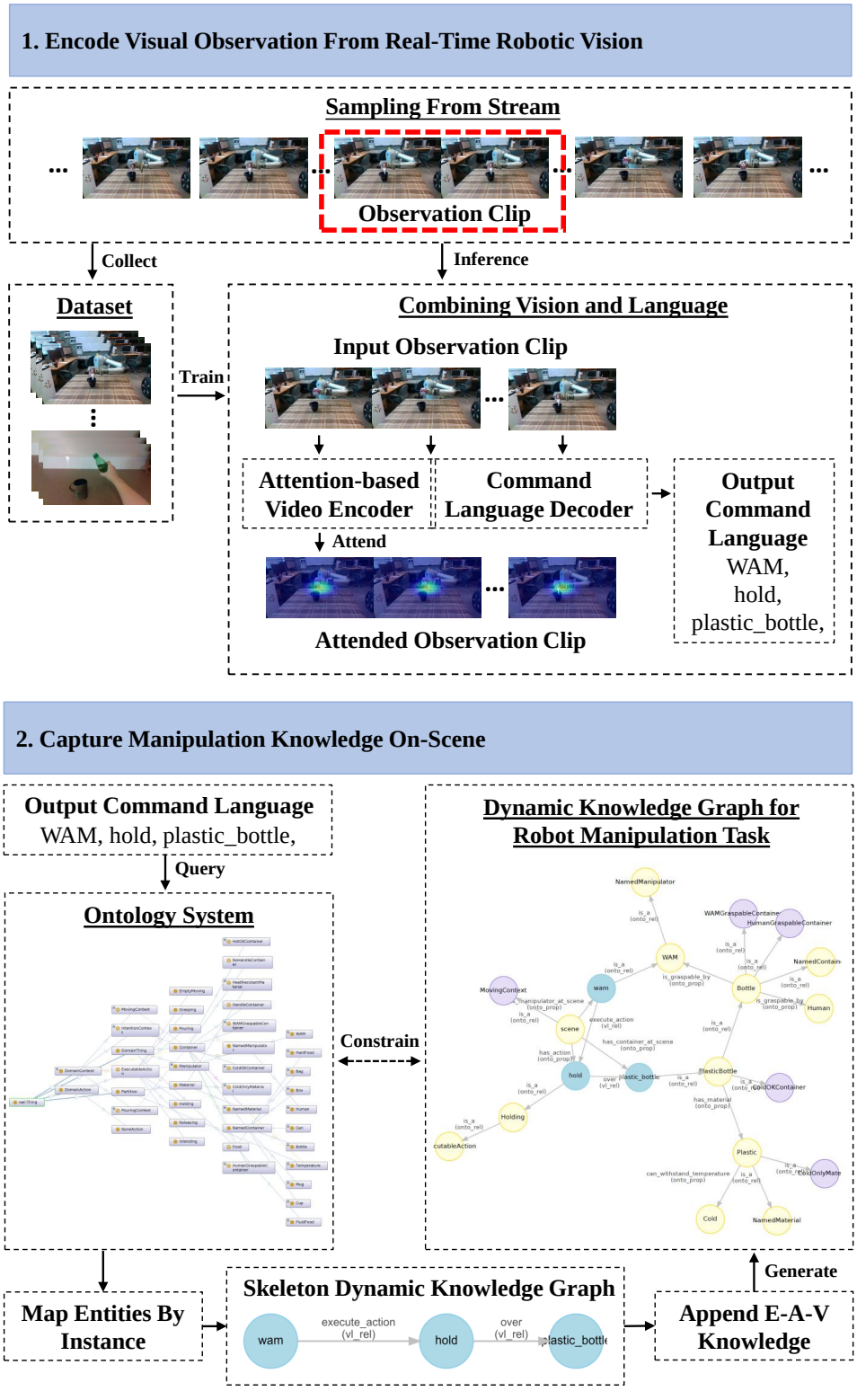


Figure 3.1: Overview of our framework.

observations to detect entities with their associating action relations, while generating visual attention maps by frame.

- **Capture Manipulation Knowledge From Dynamic Scene:** By decoding entities and their associating action relations, a skeleton of the dynamic knowledge graph can be constructed initially. An ontology system is pre-constructed, storing concepts of manipulation as commonsense knowledge. Mapping from the command language predicted by the vision-language model, objects and action relations will be parsed as instances of the ontology tree. Furthermore, given a keyword of entity, sets of commonsense knowledge are queried from the ontology system, generating triples of manipulation knowledge and thus completing the dynamic knowledge graph.

3.3 Enabling Methods

In order to implement the framework mentioned above, the following key-enabling methods need to be developed.

3.3.1 Dataset

A dataset of robot manipulation activities needs to be constructed in order to be useful for real robot manipulation learning. To fully interpret the manipulation context presenting throughout time, changes to objects being manipulated and the interactions between objects and manipulator need to be annotated by the performing action.

3.3.2 Sampling from Stream

In real-time application scenarios, robotic vision is defined as a continuous data stream. The timeliness of the camera stream demands sampling from a moving window of recent video frames while flushing out-dated information. As such, it is important to adapt a suitable stream sampling method for online streaming applications.

3.3.3 Combining Vision and Language

A suitable vision and language model is required to encode sequences of visual observations over the manipulation scene into robust vectors of visual state representations. From those visual representations, linguistic entities and action relations need to be decoded, describing the observable manipulating events happening at the moment. Visual attentions need to be captured, providing localizing capability to describe the salient manipulation actions to the pixels.

3.3.4 Ontology System

Humans are capable of learning from linguistic experience. The experience serves as commonsense knowledge which can be inferred at any time. To store commonsense knowledge for robots and allow deductive and inductive reasoning over manipulation concepts, a system of ontological structure is required.

3.3.5 Generating Dynamic Knowledge Graph

To generate dynamic knowledge graphs for a robot manipulation task, integration between vision-language model and ontology system needs to be regulated while allowing individual-based reasoning. To adapt our framework of understanding manipulation context and allow generating dynamic knowledge graphs in real-time robotic applications, an algorithm for online inference needs to be developed, enabling the extraction of visual attention maps by frame and the generation of dynamic knowledge graphs by demand. A dynamic knowledge graph also needs to be governed by the ontology system constructed, therefore ensuring the logical correctness of any reasonable concept.

3.4 Experiment for Proposed Framework

In order to validate the effectiveness of our framework and to assess the performance of the associating enabling methods, the following experiments will be conducted.

3.4.1 Experiment with Datasets

While the specific dataset of robot manipulation activities is to be constructed, specialized evaluation categories need to be set up in order to inspect the robustness of the trained vision-language model. In specific, the evaluation categories will quantitatively evaluate the scores of linguistic entities and action relations being correctly decoded by any vision-language model. Moreover, public dataset will also be utilized to further validate the designs of the attention vision-language models.

3.4.2 Experiment with Dynamic Knowledge Graph

Under a specific scenario where a robot arm performs a list of manipulation actions over some designated objects, we wish to inspect whether the fusion of vision-language model and the ontology system can successfully map the visual manipulation scene into a fully-fledged dynamic knowledge graph, and whether the manipulation context of the current scenario can be correctly reasoned by the ontology system.

3.4.3 Experiment with Visual Attention

Given a pre-trained attention vision-language model, under a real-time camera stream where video frames are constantly produced, we wish to inspect whether the vision-language model can successfully encode the frames to generate visual attention maps that indicate where the salient manipulation actions take place in the pixel. Moreover, we wish to evaluate the correctness of the generated attention maps based on the network decision, and to inspect the scenario where the network decision is known to be incorrect.

3.5 Summary

In this chapter, we formally introduced the system framework to incorporate understanding manipulation contexts for robotic vision. The workflow of our framework was discussed in detail, while the enabling methods associated and the list of experiments to be done were highlighted.

Chapter 4

Methods

Given a visual observation of robot or human manipulation scene in time period $T_i, .., T_j$, we aim to capture its visual attention and describe the manipulation concepts and their internal relational structures into a dynamic knowledge graph G_{T_i,T_j}. In this chapter, following the definitions and the proposed framework for robot context understanding, we first discuss each of the enabling methods in detail. We present our Robot Semantics dataset used to specifically study joint robot and human manipulation context understanding. Then, we discuss and describe a sampling method against offline videos. Our attention-based vision-language model is next formulated. The scheme to construct and use the ontology system for robot commonsense knowledge is then discussed. Finally, we describe an algorithm to generate visual attention and time-attributed dynamic knowledge graphs for online camera streams.

4.1 Robot Semantics Dataset

While many datasets exist for manipulation tasks and human intentions, few span both robot and human manipulation tasks. We propose the *Robot Semantics Dataset*, where videos demonstrating complete particular manipulation tasks such as "pouring water to a cup" are collected. Figure 4.1 presents an overview over our benchmark data with command language annotations and robot way-point trajectories.

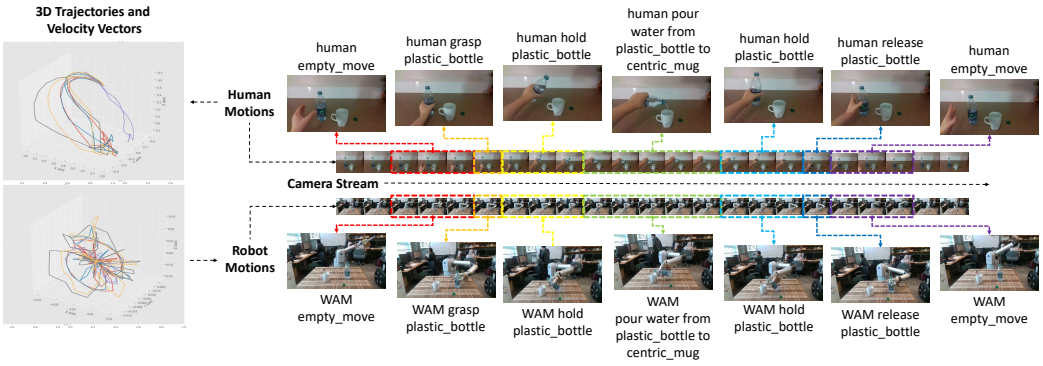


Figure 4.1: Samples from our *Robot Semantics dataset*. Left: Ten of the 3D trajectories and the velocity vectors from the robot motions used in video collection. Right: Videos with annotated entities and actions relations.

4.1.1 Collection

The manipulation videos are collected using an Intel RealSense D435i Camera. For each video collected, objects are first placed at random locations, and a manipulator (robot or human) executes a sequence of motions to complete the full manipulation task. During the process, actions such as "grasp", "release", "pour", "hold" and "intent" are enlisted. Two types of manipulators are presented in total: human subjects and a Barrett Whole Arm Manipulator (WAM) robot.

Human: The camera was set up with an egocentric view in a kitchen table environment. Human subjects were asked to use one hand to perform a series of actions to complete a manipulation task. For manipulation tasks performed by a human, 94 videos - 42,681 images are collected in total.

WAM: A Barrett WAM robot was used to perform the same set of manipulation tasks as the human subjects. The camera was setup on the side to view the majority of the WAM and a kitchen table top with objects on it. The experimental protocol originating from *IVOS benchmark* of Siam et. al. [89] is used to control the WAM robot. A human operator guided the WAM to reach the target and perform the intended manipulation actions. Robotic waypoint trajectories, in the form of quaternions over the 7 joints poses, were recorded during the kinesthetic teaching. The WAM robot then executed the manipu-

lation actions by following the teaching trajectories. For the WAM robot arm, 46 videos - 69,368 images and 43 recorded trajectories are collected in total.

4.1.2 Command Language

The command language is annotated to describe the occurring action and the objects involved in the action over a period of time. An example of the command language for pouring action is visualized in Figure 4.2. In the example, the salient action "pouring water" is highlighted, with WAM robot holding the "plastic bottle" where the pouring liquid "water" comes from. The "water" goes into "centric mug", concluding the end purpose of the action.

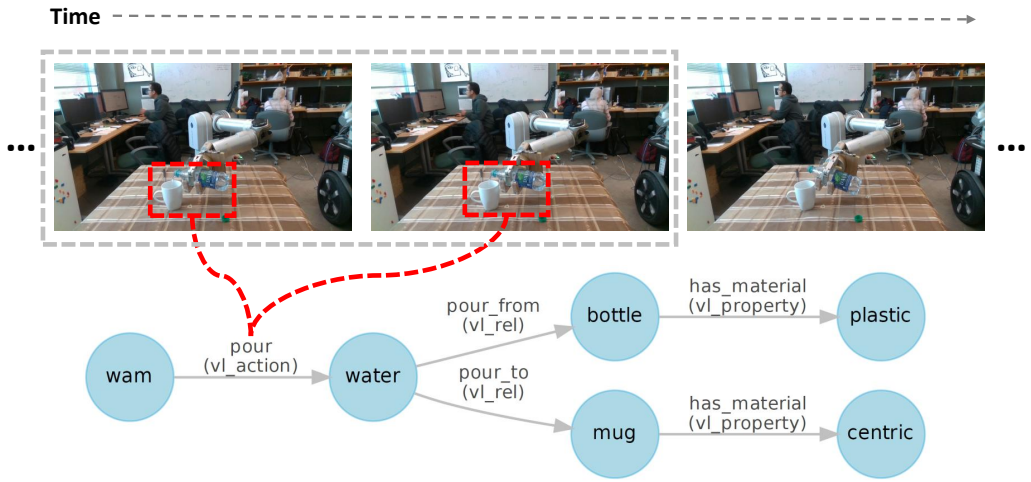


Figure 4.2: Visualization of command language for pouring action in a WAM robot.

Mathematically speaking, A command language $S_{T_i...T_j}$ over the time period $T_i...T_j$ is represented as:

$$e_1 \xrightarrow{r_1} e_2 \xrightarrow{r_2} ... \xrightarrow{r_{n-1}} e_n \tag{4.1}$$

where $e_1, e_2, ..., e_n \in E$ represents the concepts of manipulation, including manipulators and objects involved in the manipulation, $r_1, r_2, ..., r_{n-1} \in R$ jointly describes the salient actions presenting in the manipulation context, and S has a length of n. Edges inside $S_{T_i...T_j}$ are sequentially composed.

For any entity e_i perceived inside a command language, commonsense knowledge associated with the entity can be denoted as a labelled directed graph G_{e_i}. A dynamic knowledge graph $G_{T_i...T_j}$ can be seen as the union over the command language and labeled directed graphs over all associated entities:

$$G_{T_i...T_j} = S_{T_i...T_j} \cup G_{e_1} \cup ... \cup G_{e_n}. \tag{4.2}$$

In summary, a single graph of command language can be attributed as Entity-Relation-Entity (E-R-E) knowledge, where things and their probable action relationships are associated together. Entities are usually denoted by classes of objects or concepts, such as "Mug", "Centric", "Robot", etc. Relations are denoted as semantic relations that are able to logically associate any two entities, such as action verbs like "hold", "pour", or logical terms like "from", "to", "on", "with", etc. A command language serves as a basic skeleton form to generate the final dynamic knowledge graph.

4.2 Sampling from Stream

Uniformly sampling a fixed number of frames from an entire video to represent has been the standard practice for action recognition. However, in online applications, camera vision is provided as a stream of data, where new data is queued while outdated data is flushed. As such, continuous feedback can be requested at any time of streaming. However, manipulation actions do not need to be executed in a strict logical sequential manner for a task of manipulation. For example, suppose we need to pour water from a bottle to two different mugs, the manipulation task can be executed in two different flavors:

- Grasp the bottle, pour to the first mug, then move to pour to the second mug, release the bottle.
- Grasp the bottle, pour to the first mug, release the bottle, shake your hand meaninglessly for a while, grasp the bottle again, pour to the second mug, release the bottle.

Abruptly encoding long-term video sequences recurrently can cause difficulties in learning. Suppose our objective here is to recognize the actions of the manipulation task in different stages. Stages of pouring to the first and second mug can happen concurrently, or in a more separate manner. Especially for the second scenario, memorizing the context of meaningless shaking seems unnecessary. For the purpose of recognizing the pouring action for the second mug, a short time of look-back should be enough to maintain temporal consistency while avoiding the accumulation of out-dated information. In this section, we present a sampling mechanism to specifically deal with this short-term sampling problem for camera streaming.

4.2.1 Clip Observation

Ideally, a camera stream CS_{T_1} produces an indefinite sequence of image frames $\{I_{T_1}, I_{T_2}, ..., I_{T_i}, ...\}$. At any given time T_i, a command language sequence S can be generated by estimating the conditional probability $p(S|I_{T_1}, ..., I_{T_i})$. The process can be denoted as recurrently mapping from visual to language space $\{I_{T_t}\} \rightarrow S$.

The goal of sampling from such a camera stream CS_{T_1} can therefore be defined as maintaining a uniform random sampling of L frames over the most recent frames arrived from a stream. This segment of L frames can be considered as an intermediate, short time of visual observation over a manipulation scene. We further denote this segment as a small observation clip of video frames $C_{T_i} = \{..., I_{T_i}\}$, persisting for a total length of L frames.

4.2.2 Sampling and Temporal Skipping

Fundamentally, a queue of maximum L length is maintained, serving as the observation window to sample a series of overlapping clip observations from the camera stream CS_{T_1}. As the number of frames sampled accumulate, streams of image frames continuously arrive into the observation window while the out-dated frames are flushed. Therefore, the sampled observation clip will be in form:

$$C_{T_i} = \{I_{T_i-L}, ..., I_{T_{i-1}}, I_{T_i}\} \tag{4.3}$$

Still, a fixed sampling strategy does not account for temporal variations. Temporal variations involve the situation, when for example, robotics carry out motions with a much slower speed compared to humans, or sudden frame drops occur in camera stream. To remedy this, we can define the sampled observation clip with temporal skipping:

$$C_{T_i} = \{I_{T_i-L_\gamma}, ..., I_{T_{i-\gamma}}, I_{T_i}\} \tag{4.4}$$

Between two subsequal frames I_{T_i} and $I_{T_{i+\gamma}}$, a skip size K can be defined, representing the maximum number of frames to possibly skip between ith and $i + \gamma$th. Furthermore, a probability p_i is associated with each incoming image frame I_{T_i}, representing the probability that the specific ith frame will be collected to the observation window. The probability increases if a subsequal frame I_{T_i+1} is dropped. The probability ensures that a future frame has a higher chance to be collected should skip be considered, until the frame I_{T_i+K} is collected indefinitely should all of its predecessors be dropped.

The sampling algorithm with temporal skipping is presented in Algorithm 1. A clip observation is collected when: (1) the observation window is filled for the first time; or (2) $\lfloor \frac{L}{2} \rfloor$ of the observation window is flushed with newer images.

4.3 Combining Vision and Language

Given an observation clip C_{T_i}, our goal is to acquire the related visual attentions and estimate the conditional probability $p(S|C_{T_i})$ over the occurring command language sequence $S = \{s_1, ..., s_n\}$ at the time period $T_{i-L_\gamma}...T_i$. To achieve this, we propose to train an end-to-end attention-based sequence-to-sequence (seq2seq) model. The details of the model are discussed next.

Algorithm 1: Sample observation clips from a stream of images

Inputs: A camera stream CS_{T_1}. Maximum clip size N. Maximum number frames to skip K. Step size L.

Result: Observation clips {..., C_N, ...} of length N.

initialize CS_{T_1};
initialize an empty *Queue* of size N;
$n\_queued \leftarrow 0$;
$k = 1$;
while *True* **do**
    $I_{T_i} \leftarrow$ CAMERA_CAP(CS_{T_i});
    $p = k/(K+1)$;
    if $p < rand(0,1)$ **then**
        $Queue.add(I_{T_i})$;
        $n\_queued \leftarrow n\_queued + 1$;
        $k \leftarrow 1$;
    else
        // Next frame is more likely to be sampled. ;
        $k \leftarrow k + 1$;
    end
    if $(Queue.isFull())$ & $(n\_queued \bmod L == 0)$ **then**
        $C_{T_L} \leftarrow Queue.retrieve()$;
    end
end

4.3.1 Sequence-To-Sequence

The seq2seq model [3] is an encoder-decoder architecture where an encoding vector representation v is learned by a sequential encoder, and a sequential decoder learns to generate the command language sequence S conditioned on the encoding vector:

$$p(s_1..., s_n|x_{i-L_\gamma}, ..., x_{T_i}) = \prod_{k=1}^{n} p(s_k|v, s_1, ..., s_{k-1}) \tag{4.5}$$

where $x_{T_i-L_\gamma}, ..., x_{T_i}$ is the sequence of visual features extracted from the frames of clip observation, v is the vector representation from encoding the sequence of visual features $x_{T_i-L_\gamma}, ..., x_{T_i}$, and $S = (s_1, ..., s_n)$ is the corresponding output command sequence with a maximum length of n. $p(s_k|v, s_1, ..., s_{k-1})$ is the probability of the next probable command token s_k, represented with a softmax over all the tokens in the command vocabulary. The seq2seq structure is optimized by maximizing the log likelihood objective:

$$\underset{\theta}{\text{argmax}} \sum_{(X,S)} \log p(S|X;\theta) \tag{4.6}$$

where θ is the model parameters. Figure 4.3 presents an attention-based seq2seq architecture for vision-language modeling.

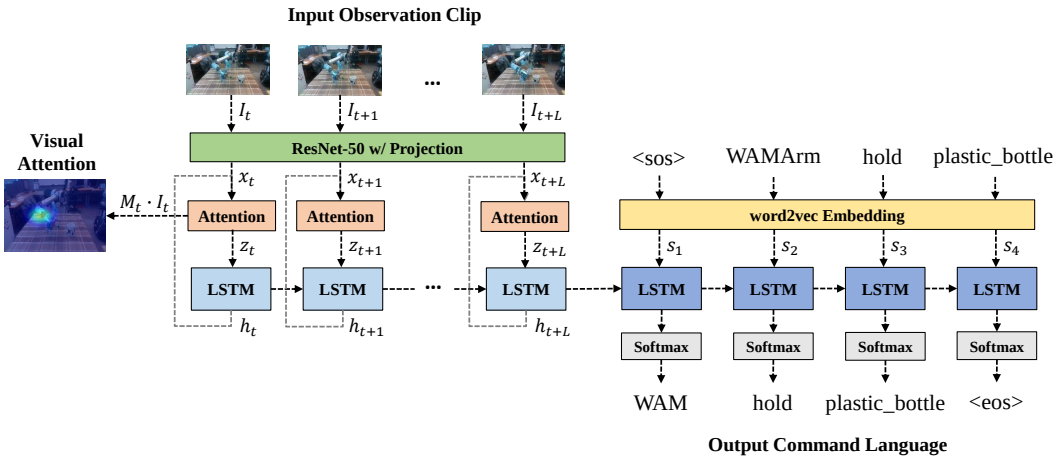


Figure 4.3: Architecture for an attention-based seq2seq.

4.3.2 Convolutional Feature Extraction

Video appearance features $X_{T_i} = \{x_{T_i - L_\gamma}, ..., x_{T_{i-\gamma}}, x_{T_i}\}$ of a clip observation C_{T_i} are extracted using a backbone CNN network such as ResNet. The visual appearance features provide fine-grained appearance details of the objects and attributes in the entire manipulation scene. The features from the last convolutional layer of the ResNet are used.

4.3.3 Spatial-temporal Encoding

The visual encoder is used to obtain a weight of alignment over the spatial resolution and a fixed-dimensional representation vector v_t which encodes the spatio-temporal information, given input sequence X_t. The convolutional features are attended spatially and temporally for every frame in the duration of the entire clip observations. And the fixed-dimensional representation vector can be acquired from the hidden states h_t of a recurrent neural network (RNN).

Attention Mechanism

Figure 4.4 illustrates the architecture of the attention mechanism. The attention mechanism f_{att} determines the amount of attention allocated to different regions of image feature, conditioned on the hidden states h_{t-1} of the encoder networks. Given a context vector a_t at timestamp t which is a dynamic representation of the relevant salient part of the image feature x_t, the attention mechanism implicitly generates a positive scalar weight of "soft" alignment $alpha_{i,t}$, interpreted as the relative importance to give to a pixel location i:

$$\begin{aligned} a_{i,t} &= f_{att}(x_t, h_{t-1}) \\ &= \omega^T[\tanh\left(W_{xa} * x_t + W_{ha} * h_{t-1} + b_{ha}\right)] \\ \alpha_{i,t} &= softmax(a_{i,t}) \\ \tilde{x_t} &= \alpha_t \odot x_t \end{aligned} \tag{4.7}$$

where f_{att} is an additive mechanism that determines the amount of attention allocated to different regions of the image feature, conditioned on the previous hidden state h_{t-1} of the encoding recurrent network. $i = 1, 2, ..., H \times W$. "*"

denotes 1D convolutional operations, W_{xa}, W_{ha}, b_{ha} are learn-able weights and bias. The attended visual feature $\tilde{x}_t$ is acquired by element-wise multiplication.

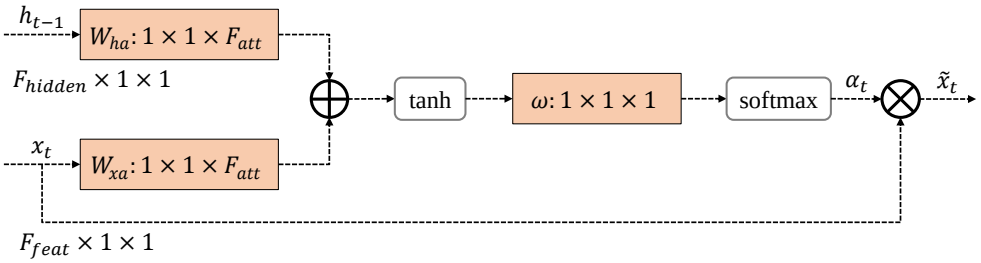


Figure 4.4: Architecture of the visual attention mechanism.

Recurrent Neural Network

Given the attended sequence of visual features $\tilde{X}_t$ applied on observation clip, one layer of Recurrent Neural Network (RNN) aggregates those frame-based convolutional features across the entire attended clip observation, encoding the sequence into hidden states and cell states $v_t = (h_t, c_t)$. The hidden state vectors implicitly memorize contextual information from all previous time steps from $t - L_\gamma$, up to t. We empirically investigate two types of RNN for visual encoding: plain LSTM and ConvLSTM.

For plain LSTM, a spatial pooling is first applied over the attended feature input $\tilde{x}_t$ at timestamp t. Given the spatially pooled visual feature input z_t and the previous hidden and cell states $v_{t-1} = (h_{t-1}, c_{t-1})$, the hidden state h_t and the memory cell state c_t at the next timestamp t are computed as:

$$
\begin{aligned}
z_t &= \sum_{i=1}^{H \times W} \tilde{x}_{i,t} \\
i_t &= \sigma(W_{xi} z_t + W_{hi} h_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} z_t + W_{hf} h_{t-1} + b_f) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tanh(W_{xc} z_t + W_{hc} h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo} z_t + W_{ho} h_{t-1} + b_o) \\
h_t &= o_t \circ \tanh(c_t)
\end{aligned}
\tag{4.8}
$$

where σ is the sigmoid function, i_t, f_t and o_t represent the input state, forget state and output state over the current timestamp t, "$\circ$" denotes the Hadamard product.

The convolutional LSTM is proposed later in Shi et. al. [90] to extend gated inputs of the LSTM network into 3D tensors, allowing the recurrent module to preserve spatial information of image inputs. Given the attended visual feature input $\tilde{x_t}$ at timestamp t, the unattended input feature map x_t is first added back, forming a skip connection. The hidden state h_t and the memory cell state c_t at the next timestamp t are then computed as:

$$\begin{aligned} z_t &= \tilde{x}_t + x_t \\ i_t &= \sigma(W_{xi} * z_t + W_{hi} * h_{t-1} + b_i) \\ f_t &= \sigma(W_{xf} * z_t + W_{hf} * h_{t-1} + b_f) \\ \mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh(W_{xc} * z_t + W_{hc} * h_{t-1} + b_c) \\ o_t &= \sigma(W_{xo} * z_t + W_{ho} * h_{t-1} + b_o) \\ \mathcal{H}_t &= o_t \circ \tanh(\mathcal{C}_t) \end{aligned} \tag{4.9}$$

where $*$ denotes 2D convolutional operation.

4.3.4 Language Decoding

Given the visual representational vector $v_t = (h_t, c_t)$, the decoder network computes the probability of $s_1..., s_n$, denoted by one-hot indicators $s_i \in \{0, 1\}^N$, with a word-to-vector embedding projection, followed by a standard LSTM. The word-to-vector embedding projection is employed as:

$$x_{s_i} = W_{embd} s_i \tag{4.10}$$

where $W_{embd} \in R^{|N| \times d_{embd}}$ is the embedding weight, d is the dimension of the embedding vector. Here Google word2vec [91] is used, with d=300 by default. The initial hidden state of the decoding LSTM is set to the last hidden state of the encoding LSTM. The final sequence of command language S is predicted by applying a softmax over the output layer of LSTM with a linear projection:

$$\begin{aligned} v_t &= Encoder(\tilde{X}_t, zeros) \\ p_{1...n} &= LSTM((x_{s_1}, ..., x_{s_n}), v_t) \end{aligned} \tag{4.11}$$

Additionally, to pass the 2D hidden states from the ConvLSTM, inspired by Abolghasemi et al [73], average-pooling and max-pooling are performed to squeeze spatial information. The two pooled features are summed together, serving as the initial hidden state for the decoding LSTM:

$$\begin{aligned} h_t &= AvgPool(\mathcal{H}_t) + MaxPool(\mathcal{H}_t) \\ c_t &= AvgPool(\mathcal{C}_t) + MaxPool(\mathcal{C}_t) \end{aligned} \tag{4.12}$$

4.4 Ontology System

4.4.1 Construction

An ontology is a well known way to store machine-interpretable definitions of concepts in a static knowledge domain. Figure 4.5 visualizes the constructed ontology tree for manipulation concepts in Protege [92].

The constructed ontology system consists of 4 main classes: (a) "DomainThing", where kitchen objects in the manipulation contexts, like manipulator, container, food, etc, are presented; (b) "Partition", where proportional partitions like temperature, or descriptive partitions like materials, etc, are presented; (c) "DomainAction", where executable manipulation actions like pouring, grasping, releasing, holding are presented; (d) "DomainContext", where configurations describing manipulation task scenarios are presented, for example, GraspingContext, PouringContext, etc. Additionally, 2 main properties are stored: (a) "topAttributeProperty" which describes internal object attributes such as hasMaterial, canWithstandTemperature, hasHandles, etc; (b) "topRelationProperty", which describes abstract relations over contexts such as hasAction, hasContainerAtScene, etc.

4.4.2 Usage

To query an ontology tree, we start by collecting hierarchies among linguistic vocabulary of entities $E = \{e_1, ..., e_n\}$ being stored. For any ontology tree, superclass-subclass hierarchy is fundamentally applied to specialize the required descriptions that define the hierarchical relation. For example, a **bottle** is simply a type of **Container**.

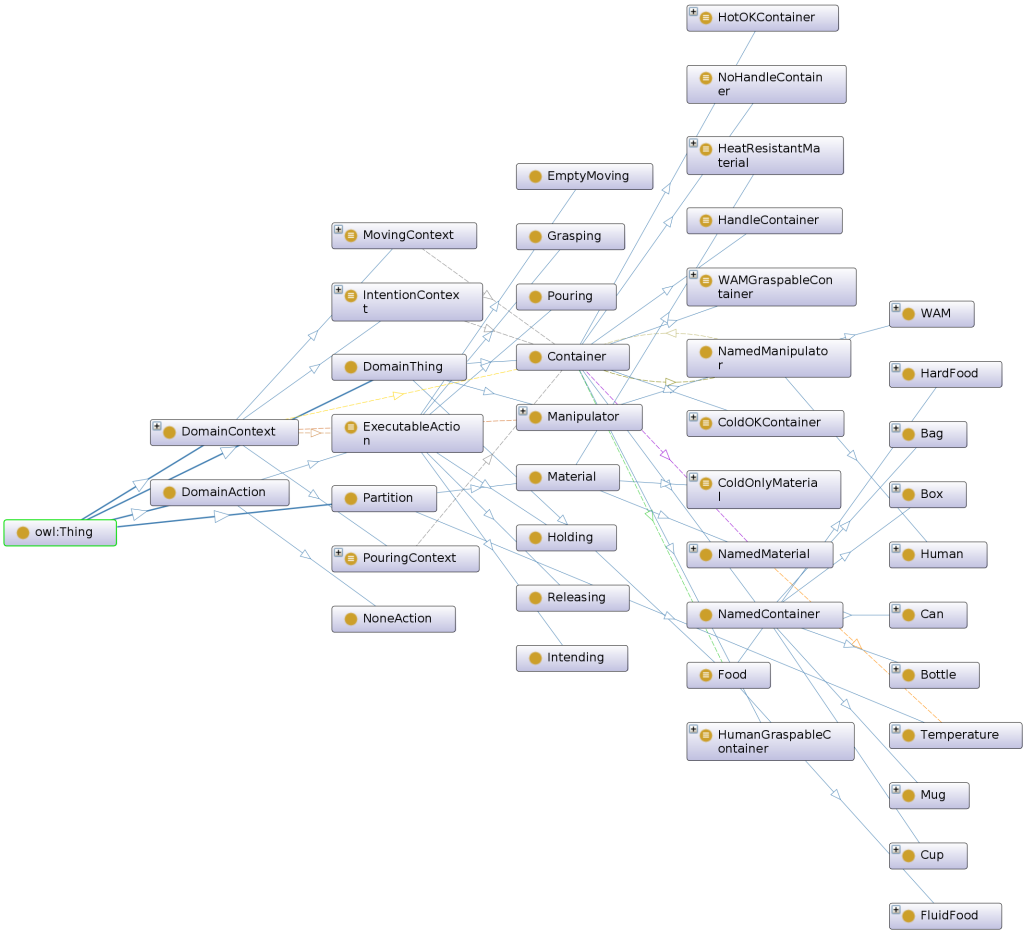


Figure 4.5: Visualization of the ontology tree for robot manipulation knowledge.

However, superclass-subclass hierarchy is not enough to cover up the graphical connectivity of commonsense knowledge. Furthermore, properties need to be collected linking two concepts, or defining characteristics that a specific concept holds with constraints. To describe the property or the attribute of an entity e_i, a labelled directed graph can be populated into the *onto* with an linguistic relation r and restriction re (Quantifier, Cardinality, and hasValue):

$$onto \leftarrow e_i \xrightarrow{r,re} e_j \tag{4.13}$$

We denote this entire process as growth of concept, similarly as humans describe an abstract concept by attributes and relations. For example, when robot describes a concept of "paper cup", superclass-subclass hierarchy can initiate the description, where a **PaperCup** is simply a child concept of **Cup**,

and the family of **Cup** concept is a concept of **Container**. Then the description of the concept can grow with its distinctive characteristics or attributes, where **PaperCup**, is a **cup** while possessing the property **hasMaterial** some **Paper** of some types of Material. Going further, distinctive characteristics for the concept **Paper** can be defined as well, for example, where any **Material** of **Paper canWithStandTemperature** of some temperature **Cold**, or the fact that **PaperCup isGraspableBy Human** only, as the gripper of WAM robot has a tendency of crushing the paper material with too much force, etc. As critical properties become available, reasoning becomes possible. For example, since **PaperCup isGraspableBy Human** only, therefore **PaperCup** is a type of **Container** which can be reasoned as **HumanGraspableContainer**. Figure 4.6 demonstrates such a representation over the commonsense knowledge for "PaperCup" as a labelled directed graph.

The generated labelled directed graph from the ontology system serves as a robot's commonsense knowledge over a single concept of "PaperCup", and can be generalized to different instances of paper-cup-like objects in real life. The usage of ontology system can also be denoted as querying and reasoning with tuples of Entity-Attribute-Value (E-A-V) knowledge, allowing us to capture and distinguish the properties of any entity and to assert certain restrictions given different attribute values, and thus completing the skeleton of the dynamic knowledge graph with commonsense knowledge.

4.5 Generating Dynamic Knowledge Graph

4.5.1 Mapping Manipulation Concepts By Instance

The process of constructing a dynamic knowledge graph can be summarized as a process of mapping by instance. This is achieved by inheriting grounded E-R-E knowledge captured inside a command language and instantiating entities and relations inside to their ontological parents, which in turn inheriting any E-A-V knowledge available for all grounded concepts. To begin, we assume that any dynamic knowledge graph starts by "scene", which is an instance of "DomainContext" under the ontology definition. The "scene" context is

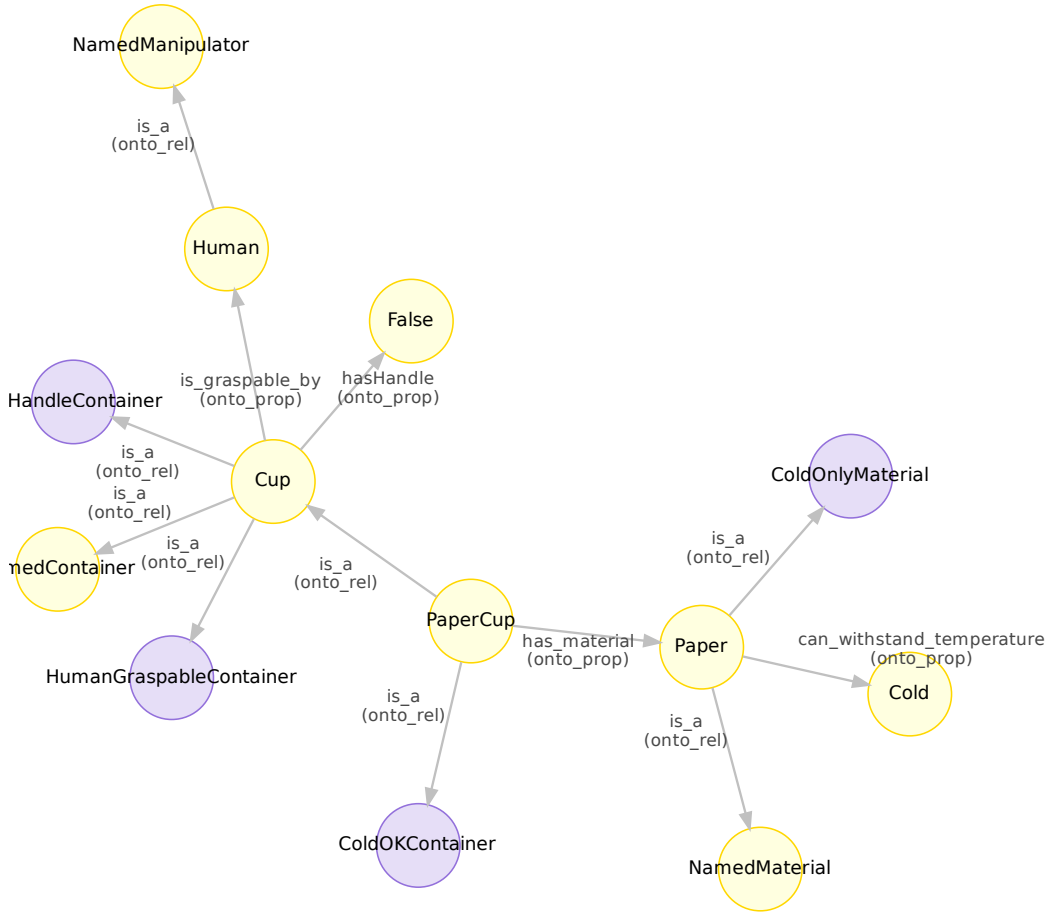


Figure 4.6: Commonsense knowledge for the object concept "PaperCup". Given a keyword concept, "PaperCup", we can query from the ontology tree, extract associating E-A-V knowledge into a Labeled Directed Graph.

generic and thus does not fall under any predefined contexts stored inside the ontology until further information becomes available. Next, we consider manipulator, manipulation actions, objects and their observable attributes which have been grounded by a Vision-Language model. Each word token in the command language can identify an object and be mapped to its conceptual counterpart in the ontology system easily. Therefore, a simple word-based NLP tagger can be invoked here, mapping manipulator, action and the associating objects into the ontology system as instances of the concepts defined. Finally, any ontological instance inherits all properties and attributes predefined and

reasoned in the ontology tree. Now, the process of capturing commonsense knowledge simply becomes the process of extracting the defined concepts in the ontology tree. As actions and objects are individualized, their presence will satisfy certain configurations and thus "scene" can be reasoned to be associated with a specific manipulation context like "PouringContext".

4.5.2 Algorithm for Online Inference

The pseudo-algorithm for online inference is shown in Algorithm 2. The algorithm can be segmented into three procedures: (1) attend to visual observations by frame; (2) predict command language by schedule; and (3) complete a dynamic knowledge graph by ontology system querying.

Attend: The attending procedure generates predictions of visual attention for every frame captured from a camera stream, much similar to frame-based visual tracking methods. In a real-time manipulation scene, a camera stream is CS_{T_1} setup at T_1 to observe the progression of the manipulation task. At any time T_i, a camera frame I_{T_i} is captured. The encoder of the vision-language model is inferred, encoding the appearance of the manipulation scene into an encoding vector representation V_{T_i} and generating the visual attention map α_{T_i} at time T_i.

Predict: For each L numbers of frames flushed, the encoding vector representation V_{T_l} at the specific time T_l is used to decode a command language $S_{T_{l-L+1}...T_l}$ for time period $T_{l-L+1}...T_l$. The predicted command language sequence is then converted into a labelled directed graph G_{T_l}. To convert from a command language, two steps are needed: (a) instantiate observed manipulators, objects and actions as instances of manipulation concepts stored in the ontology system, and (b) associate all concepts under an unknown instance of manipulation context denoted as "scene". The converted labelled directed graph serves as the skeleton of our dynamic knowledge graph.

Complete: Each entity e_i inside the command language is queried over the ontology tree *onto* by a word-based close matching. Triples of E-A-V knowledge are collected into a Labeled Directed Graph G_{e_i}, which are further merged with G_{T_l} and thus complete the dynamic knowledge graph. After-

wards, the LSTM states for the vision-language model are reset to zeros. This gives the vision-language model a chance to avoid accumulating model drifts and stabilize predictions. At last, G_{T_l} will be merged with $G_{T_1...T_{END}}$, recording the progression of the manipulation knowledge over time.

Algorithm 2: Attend Visual Observations and Generate a Progressive Dynamic Knowledge Graph in Real Time

Inputs: A camera stream CS_{T_1}. A Vision-Language Model $Model$. A static ontology tree *onto*. Step size L.
Result: Visual attention $\alpha_{T_1...T_{END}}$ and dynamic knowledge graph $G_{T_1...T_{END}}$ over time period $T_1...T_{END}$.

```
initialize CS_{T_1};
initialize an empty G_{T_1...T_END};
initialize an empty α_{T_1...T_END};
while True do
    I_{T_i} ← CAMERA_CAP(CS_{T_i});
    α_{T_i}, V_{T_i} ← ATTEND(I_{T_i}, Model);
    α_{T_1...T_END} ← APPEND(α_{T_i});
    if T_i mod L == 0 then
        S_{T_{i-L+1}...T_i} ← PREDICT(v_{T_i}, Model);
        G_{T_i} ← CONVERT(S_{T_{i-L+1}...T_i}, onto);
        for e_i in S_{T_{i-L+1}...T_i} do
            G_{e_i} ← QUERY(onto, e_i);
            MERGE(G_{e_i}, G_{T_i})
        end
        if Collect_Progressive then
            MERGE(G_{T_i}, G_{T_1...T_END});
        end
        RESET(Model);
    end
end
```

4.6 Summary

In this chapter, we formally discussed the methods that enable manipulation context understanding for robotic vision. Firstly, we proposed our Robot Semantics Dataset which is guarded under a strict knowledge domain with knowledge bases annotated to the frame. Next, we discuss the potential challenges we might face to sample from streams for manipulation scenes, and

we propose our probabilistic stream sampling algorithm to enable generating observation clips with temporal skipping. Followed by our stream sampling algorithm, our attention encoder-decoder architecture to model from vision to language, and to generate visual attention over salient manipulation actions are formulated. We then detailed the construction and usage of our ontology system to query and reason with manipulation knowledge. Lastly, we proposed our algorithm for inferring visual attention and dynamic knowledge with real-time camera vision.

Chapter 5

Experiments

In this chapter, we outline the details for our experiments. First, we specify details of implementation. Next, we highlight different settings for our experiments, including strategies to validate our framework and baseline model settings. Results and comparisons against state-of-art methods will be presented next. Performance of our manipulation context understanding framework will at last be discussed.

5.1 Experiment Details

5.1.1 Dataset

Robot Semantics Dataset

The majority of the videos collected for our *Robot Semantics dataset* will be used for the training of the vision-language models. To evaluate any trained vision-language model , three specialized evaluation categories are set up:

- **Stream**: Human operators significantly hinder the smoothness of task execution by slowing down or performing a number of task-invariant motions. There are 5 human videos - 6159 images for evaluation in this category.
- **Unknown**: Objects that are never presented during model training are collected into this category. There are 18 human videos - 8463 images and 15 WAM videos - 22821 images in this category.

- **Complex**: Multiple objects are presented at the robotic manipulation scene. One human first uses a finger to point to some objects of interest at specific locations. The manipulator then performs the action on the specified objects. 7 WAM videos - 9708 are available in this category.

Public Dataset

We also evaluate the designs of our Vision-Language models on a public dataset. The *IIT-V2C dataset* is originally proposed in Nguyen et al [8] to process fine-grained human action understanding. The videos in the dataset are 2 to 3 minutes long on average. Each video is randomly segmented into around 10 - 50 short observation clips, and a grammar-free command sentence is annotated per clip to describe the presenting human actions. We follow the same experiment protocols in Nguyen et al [8], where annotated clips of maximum 30 frames are extracted from the food manipulation videos. For any clip not reaching to 30 frames length, a synthetic mean image calculated from ImageNet dataset is padded.

5.1.2 Sampling from Stream

To train with our Robot Semantics Dataset under a normal, offline fashion, the proposed video sampling method with an overlapping size of 15 frames is adapted to generate observation clips, with a drop size of 0. The length of clip sequence is chosen as 30, equivalent to a one full sec of observing under a 30 frames per second (FPS).

Additionally, we assess the effectiveness of our probabilistic stream sampling algorithm in a semi-online fashion. To do this, we conduct a scheme involving 5 runs of experiments. For each experiment run, evaluation clips will be sampled from all evaluation categories of *Robot Semantics Dataset* with our probabilistic stream sampling algorithms. More specifically, evaluating skip sizes of 0, 1, 3, 6, 15 are employed. Then, the stream sampling algorithm will be employed on the training set of *Robot Semantics Dataset*. We choose the skip sizes for training observation clips to be 0, 1, 3 and 6, indicating that for each experiment run, 4 models will be trained with those different choices

of skip sizes correspondingly. Each model trained with a training skip size of choice will eventually be evaluated against the evaluation observation clips with all the five chosen evaluating skip sizes. An algorithm outline to perform evaluation experiment with our probabilistic stream sampling algorithm is available in 3.

Algorithm 3: Perform Evaluation with Probabilistic Stream Sampling Algorithm

Inputs: Training skip sizes $K_{train} = \{0, 1, 3, 6\}$, evaluating skip sizes $K_{eval} = \{0, 1, 3, 6, 15\}$, training dataset D_{train}, eval dataset D_{eval}.
Result: Average score over 5 runs of experiments.

$n_{exps} = 0$;
Initialize $scores_{exp}$ as an empty list;
while $n_{exps}<5$ **do**
  for k_{eval} *in* K_{eval} **do**
    $C_{k_{eval}} \leftarrow TRAIN(D_{eval}, k_{eval})$;
    for k_{train} *in* K_{train} **do**
      $C_{k_{train}} \leftarrow SAMPLE(D_{train}, k_{train})$;
      $Model_{k_{train}} \leftarrow TRAIN(C_{k_{train}})$;
      $scores \leftarrow EVAL(Model_{k_{train}}, C_{k_{eval}})$;
      $scores_{exp}.update(scores)$;
    end
  end
  n_{exps} += 1 ;
end

5.1.3 Vision-Language Model

Implementation Details

The implementation for Vision-Language models are done using PyTorch. For convolutional video feature extraction, we investigate different backbones pretrained on ImageNet without finetuning, including ResNet18, ResNet50 and MobileNetV3. For seq2seq, the weights for video encoder, language decoder and attention mechanism are initialized randomly with a hidden unit size of 256. Training is done with Adam optimizer for 50 epochs, with an initial learning rate of 0.0001, decaying by a factor of 0.1 after epoch 5 and epoch 35. For word embedding against the command language, Google Word2Vec

[91] is used, where we allow finetuning of the word embedding after the first learning rate decay. The maximum command sentence length is chosen as 10. The batch size is chosen as 16.

Ablation Study

To validate the effectiveness of our architectural design, we employ ablation studies with the following variations:

- **no_att vs. att**, where no visual attention mechanism is employed vs. with spatial attention during the stage of video feature encoding. This aims to observe the effects of attention mechanisms over the network decision.
- **ConvLSTM vs. LSTM**, where we inspect the effectiveness of spatial encoding with ConvLSTM network vs. a plain LSTM network where spatial resolutions are either pooled with attention weights or by an average pooling operation.
- **no_concat vs. concat**, where only the passing of the visual representational vector is employed vs. the last encoded hidden state h_t is also collected and concatenated along with the word embedding feature during the sequence decoding stage.

5.1.4 Ontology System

The ontology system is jointly constructed using Protégé [92] and owlready2 [93]. HermiT reasoner is invoked to assess the correctness of the constructed ontology tree. There are 70 classes and 11 relations on record.

5.1.5 Dynamic Knowledge Graph Generation

In addition to an offline evaluation setting, a real-time camera stream $CS_{T_0 \ldots T_{inf}}$ is set up to observe the manipulation scene in our WAM robotic grasping environment. The proposed online inference algorithm is employed to extract visual attention by frame over time. For any inference invoked, a command

language is generated. For each entity e_i inside the command language, a word-based close matching and a recursive tree searching algorithm are jointly employed to query over the ontology tree *onto*, returning a labelled directed graph of taxonomy. HermiT reasoner is invoked along with the querying process, padding logically reasoned facts into the queried labelled direct graph. The command language and labelled direct graph is merged at last into the final dynamic knowledge graph.

5.2 Results

5.2.1 Quantitative Results on Robot Semantics Dataset

The best experimental scores on *Robot Semantics Dataset* are shown in Table 5.1. The standard machine translation and language generation metrics are reported with coco-evaluation code [94]: BLEU 1-4, METEOR, CIDEr, and ROUGE-L, which quantify the grammar structures and the semantic meanings of the generated sentences.

In summary, the attn-seq2seq-ConvLSTM achieves superior performance against others, given its superior capability of encoding spatial information from visual inputs. The seq2seq models in general outperform traditional ConvNet-LSTM designs like EDNet, indicating that the sequential modeling strategy is more viable when dealing with a real-time camera stream. Interestingly, it is observed that the models with visual attention mechanisms converge faster in the training process while maintaining a steady performance, while performance of a simple seq2seq model is more variant, and spikes can be observed for BLEU scores (e.g. for model "seq2seq" with ResNet50 backbone). This might indicate that finetuning of learning rate for attention mechanisms need to be explored, or visual attention may cause difficulties for the decoder to look into fine-grained scene details, which we provide some analysis next.

5.2.2 Quantitative Results on IIT-V2C Dataset

The best experimental scores on IIT-V2C Dataset are shown in Table 5.2.

Table 5.1: Quantitative Evaluation Results on Robot Semantics Dataset. We report the standard machine translation and language generation metric scores, including BLEU 1-4, METEOR, CIDEr, and ROUGE-L. The highest scores achieved are highlighted.

Name	**Backbone**	**B-1**	**B-2**	**B-3**	**B-4**	**M**	**R**	**C**
attn-ConvLSTM-seq2seq	ResNet18	0.743	0.648	0.559	0.489	0.472	0.779	3.891
	ResNet50	**0.786**	**0.706**	**0.641**	**0.603**	**0.515**	**0.827**	**4.274**
	MobileNetV3_S	0.751	0.664	0.588	0.526	0.484	0.788	4.008
	MobileNetV3_L	0.761	0.681	0.615	0.567	0.496	0.800	4.128
attn-seq2seq	ResNet18	0.733	0.635	0.543	0.462	0.468	0.772	3.831
	ResNet50	0.759	0.677	0.611	0.580	0.494	0.803	4.038
	MobileNetV3_S	0.714	0.625	0.553	0.500	0.461	0.748	3.739
	MobileNetV3_L	0.745	0.657	0.582	0.531	0.483	0.789	3.937
attn-seq2seq-cat	ResNet18	0.717	0.618	0.529	0.456	0.464	0.763	3.741
	ResNet50	0.768	0.687	0.623	0.581	0.494	0.801	4.082
	MobileNetV3_S	0.710	0.616	0.533	0.476	0.451	0.758	3.672
	MobileNetV3_L	0.751	0.662	0.587	0.535	0.469	0.785	3.925
seq2seq	ResNet18	0.717	0.617	0.524	0.444	0.451	0.756	3.581
	ResNet50	0.772	0.685	0.624	0.593	0.476	0.786	3.966
	MobileNetV3_S	0.653	0.542	0.442	0.367	0.418	0.703	3.168
	MobileNetV3_L	0.719	0.627	0.543	0.486	0.460	0.757	3.690
EDNet[8]	ResNet18	0.712	0.612	0.522	0.455	0.448	0.751	3.592
	ResNet50	0.762	0.672	0.605	0.555	0.481	0.790	3.962
	MobileNetV3_S	0.674	0.558	0.460	0.335	0.405	0.696	3.137
	MobileNetV3_L	0.713	0.614	0.525	0.447	0.434	0.735	3.644

When invoking ResNet50 as backbone feature extraction, all seq2seq architectures significantly outperform the recent state-of-the-art methods, indicating the effectiveness of modeling video sequences recurrently. Models with visual attention present significant improvements compared to methods like V2CNet, where auxiliary action recognition and captioning are performed side-by-side. This indicates the importance of encoding spatial visual features to capture salient actions, and the fact that encoding fine-grained actions in the pixel with visual attention mechanism is more effective compared to traditional ways of implicitly "forcing" fine-grained action encoding through auxiliary classification tasks. The "attn-ConvLSTM-seq2seq" model significantly outperforms all methods, even with using much smaller networks like

Table 5.2: Quantitative Results on IIT-V2C Dataset. The best metric scores among the State-of-the-Art methods are highlighted.

Name	Backbone	B-1	B-2	B-3	B-4	M	R	C
attn-ConvLSTM-seq2seq	ResNet18	0.425	0.301	0.233	0.185	0.208	0.427	1.695
	ResNet50	**0.452**	**0.331**	**0.262**	**0.218**	**0.222**	**0.452**	**1.875**
attn-seq2seq	ResNet18	0.376	0.258	0.194	0.146	0.180	0.377	1.384
	ResNet50	0.413	0.2967	0.230	0.187	0.200	0.416	1.638
attn-seq2seq-cat	ResNet18	0.371	0.254	0.192	0.145	0.177	0.371	1.370
	ResNet50	0.410	0.292	0.226	0.184	0.199	0.410	1.601
seq2seq	ResNet18	0.381	0.262	0.197	0.148	0.182	0.390	1.380
	ResNet50	0.413	0.292	0.226	0.179	0.200	0.423	1.654
S2VT[51]	ResNet50, AlexNet	0.397	0.280	0.219	0.177	0.196	0.401	1.560
SGC	InceptionV3	0.370	0.256	0.198	0.161	0.179	0.371	1.422
SCN	ResNet50, C3D	0.398	0.281	0.219	0.190	0.195	0.399	1.561
EDNet[8]	ResNet50	0.398	0.279	0.215	0.174	0.193	0.398	1.550
	InceptionV3	0.400	0.286	0.221	0.178	0.194	0.402	1.594
	VGG16	0.372	0.255	0.193	0.159	0.180	0.375	1.395
	ResNet18	0.365	0.248	0.185	0.145	0.174	0.370	1.265
V2CNet[9]	ResNet50	0.406	0.293	0.233	0.199	0.198	0.408	1.656
	InceptionV3	0.401	0.289	0.227	0.190	0.196	0.403	1.643
	VGG16	0.391	0.275	0.212	0.174	0.189	0.393	1.528

ResNet18 as the backbone, where frame feature representations are believed to be less robust compared to other choice of backbone. This signals the importance of learning spatial information in recurrent modeling.

5.2.3 Results for Dynamic Knowledge Graph

We demonstrate the dynamic evolution of the knowledge graph over time for a pouring action with our WAM robot in real time, along with the generated visual attentions, the predicted command languages and the knowledge graph without external commonsense knowledge in Figure 5.1. Blue parts of graphs are governed by predictions of the vision-language model, while yellow parts are queried from the ontology system, and purple parts are from ontological reasoning. The visualization of dynamic knowledge graphs in Figure 5.1 only present instances and fundamental context reasoning due to space limitation

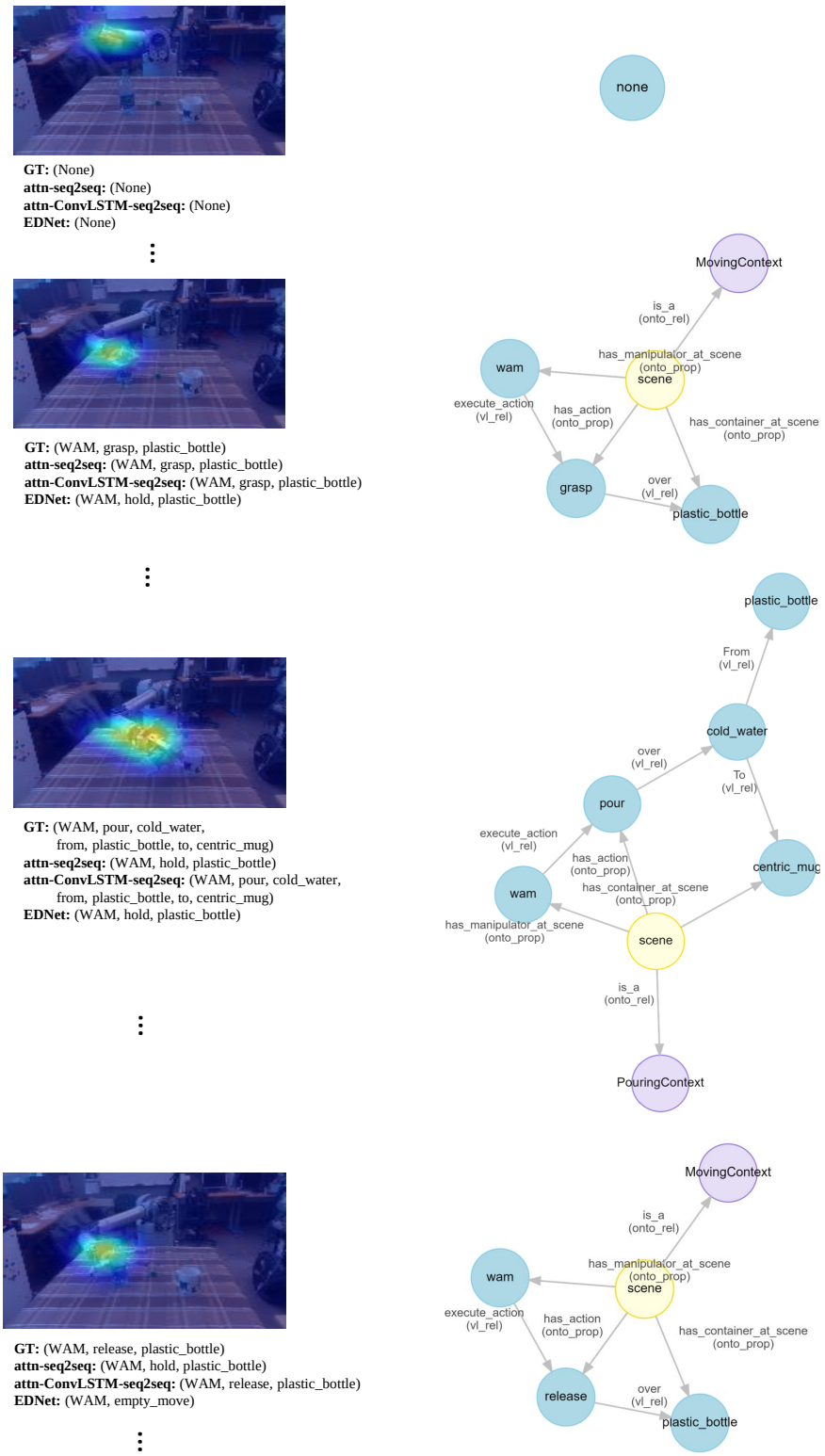


Figure 5.1: Visualization of predicted command language, generated visual attention and dynamic knowledge graph.

here. We attach the full queried version of dynamic knowledge graphs with more video examples in Appedix A.2. For each frame received from the real-time camera stream, our attention-based video encoder is able to successfully generate spatial attention maps focusing on the regions where manipulation actions present themselves. For every 30 frames passed, a command language is successfully summarized by our language decoder and is summed into the progressive dynamic knowledge graph. With the ontology system, commonsense knowledge is queried given any entity presenting under the visual observation, and the dynamic knowledge graph can be populated with rich taxonomy of manipulation knowledge. Furthermore, ontological reasoning can be performed during the querying process and populate our dynamic knowledge graph with logical fact. We distinguish the parts of the dynamic knowledge graph into parts where it is generated either by Vision-Language model prediction, static ontology querying, or by ontological reasoning. The combination of visual attention and the evolving dynamic knowledge graph fundamentally is able to reflect the intended manipulation knowledge over the robot pouring task.

5.2.4 Results for Probabilistic Stream Sampling

The mean and standard deviation of over 5 runs of evaluation on Robot Semantics Dataset with probabilistic stream sampling are plotted in Figure 5.2, using our best performing architecture "attn-ConvLSTM-seq2seq" with ResNet50 backbone. Each dot in the figure represents a model trained with a specific temporal skip size, and each model is evaluated against different choices of evaluating temporal skip size.

By observing horizontally and vertically, we can conclude that choices of temporal skipping can significantly influence the model's capability in temporal modeling. Horizontally, for a small training temporal skip size like 1, the performance drops in small amounts with each consecutively increasing evaluating temporal skip size. However, a larger choice of training temporal skip size like 6 can potentially increase the model's robustness against various choices of evaluating temporal skip size, thus maintaining a steadily linear-like path. Vertically, the model performance drops with each increasing training

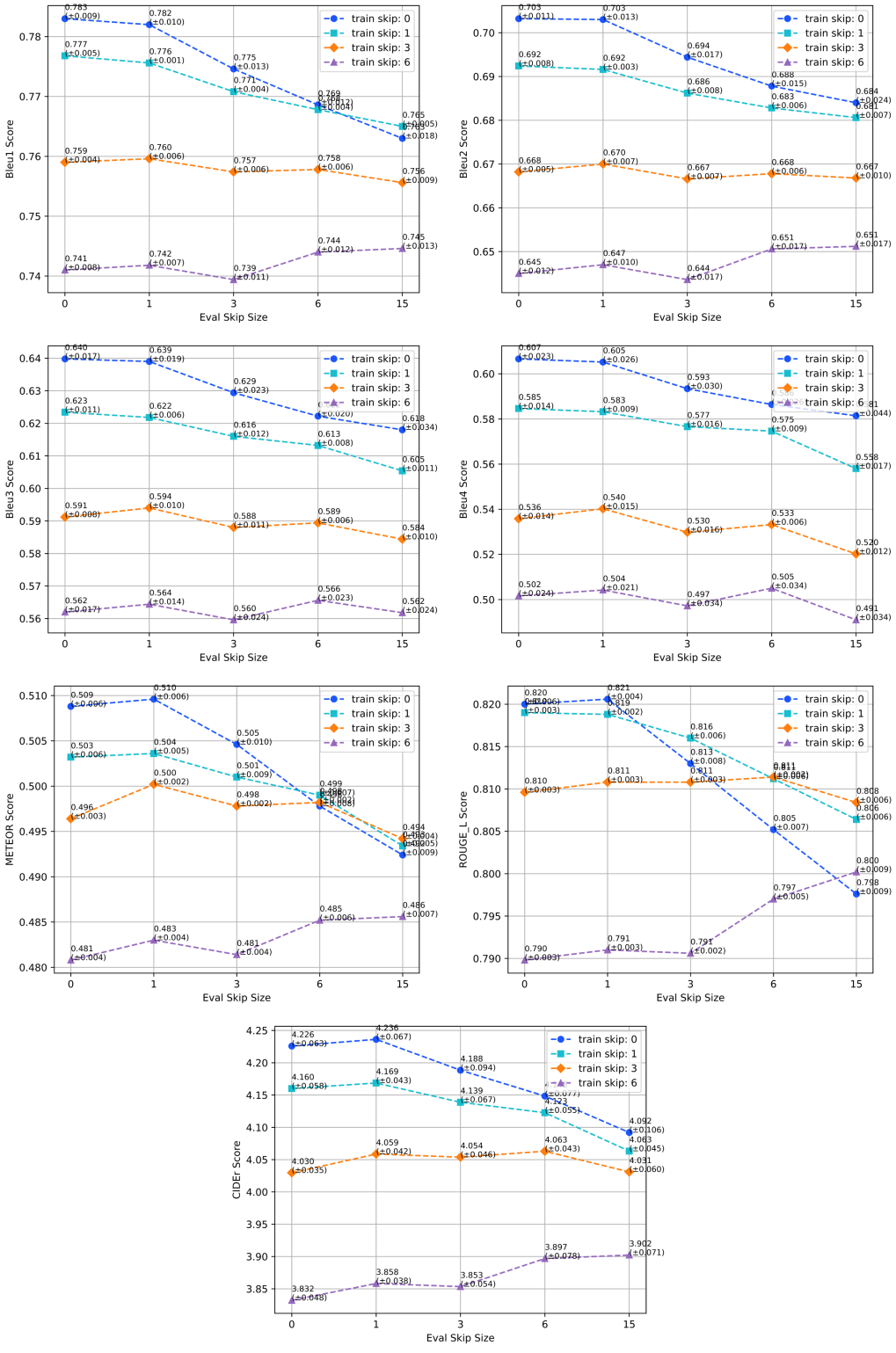


Figure 5.2: Plots of evaluation scores against various temporal skip sizes.

skip size in most cases. This again signals the fact that a model's capability in temporal modeling is associated deeply with how frames are sampled from streams.

5.3 Analysis

5.3.1 Analyzing Video Encoding Over-time

Figure 5.3: Visualization of T-SNE embedding over LSTM states from Vision-Language model.

To inspect the effectiveness of our video encoding procedure, a T-SNE embedding visualization of the representational hidden states produced for every single frame from Robot Semantics videos is shown in figure 5.3. As individual video progresses, the recurrent video encoder integrates consecutive

visual features over time, resulting in LSTM states having same or similar semantic contexts cluster together and extend recurrently in forms of long thin paths. This is important as it backs up the effects of temporal encoding of our video encoder. However, significant gaps can still be observed among different stages of semantic actions, which can possibly be a consequence of sub-sampling of videos into observation clips.

5.3.2 Analyzing Class Activation

In this section, we further evaluate the robustness of network decision making and analyze mis-classifications from our vision-language model with class activation maps (CAMs). Grad-CAM [76] is a gradient-based method to compute and visualize network decision through backpropagated gradients from convolutional activation features. In practice, last convolutional layers are used which maintain a good balance between high-level semantic and low-level spatial information. Here, we first introduce our technique to generalize Grad-CAM with our seq2seq vision-language model over clip observation.

Suppose an observation clip is composed by a sequence of convolutional video feature maps $X = \{x_1, ..., x_t, ..., x_T\}$ of T length, inferred from a convolutional neural network. $x_t \in R^{K \times u \times v}$ is a stack of K convolutional feature maps at timestamp t. A seq2seq vision-language model encodes the feature maps spatial information of the visual scene over time $1...T$, and translates from a sequence of video features into a sequence of semantic language $S = \{s_1, ..., s_j, ..., s_n\}$.

To obtain the neuron importance weights $\alpha_{k,t}^{s_j}$ for a word token s_j at timestamp t:

$$\alpha_{k,t}^{s_j} = \frac{1}{Z} \sum_u \sum_v \frac{\partial y^{s_j}}{\partial x_{t,uv}^k} \tag{5.1}$$

where y^{s_j} is the score for predicting the word token s_j (before the softmax), the gradients $\frac{\partial y^{s_j}}{\partial x_{t,uv}^k}$ can be acquired with backpropogation over time. A global average pooling operation is then followed by acquiring the final importance weight. A weighted combination of forward activation maps with a ReLU

can then be applied to obtain the final class-discriminative localization map $L_t^{s_j} \in R^{u \times v}$:

$$L_t^{s_j} = ReLU(\sum_k \alpha_k^{s_j} x_t^k) \tag{5.2}$$

The ReLU activation enforces that only pixels with positive influence on the word token of interest are collected. In theory, a total of $T \times n$ CAMs can be generated across all frame features maps. However, only the feature map x_T at end timestamp T is used to generate the localization map in our experiments, resulting in n CAMs for each observation clip.

Given a command language prediction generated from any vision-language model baseline after encoding 30 frames, at 30th timestamp, the gradient of its log probability w.r.t. the last convolutional feature map is computed and Grad-CAM visualizations are generated for each predicted word token in our command language prediction. We discuss some selected cases of Grad-CAM visualizations with specified model information or mis-classifications.

CAMs w/wo Visual Attention

Figure 5.4 shows up the Grad-CAM visualizations for seq2seq models with implicitly-learned visual attention vs. no attention mechanism in the video encoding process. It can be clearly observed that the distributions of activated regions are more organized and focused for models with visual attention, compared to plain seq2seq learning, where activated regions are more scattered around. This implies that implicitly-learned attention does help the video encoder to attend more on salient motions. Interestingly, for plain seq2seq models with comparable performance to ones with visual attention, the activation maps are distributed around the salient actions in a similar fashion of models with visual attention.

CAMs for Mis-classification

Figure 5.5 shows up the Grad-CAM visualizations for incorrect entities interpreted in the command language prediction. When an incorrect command to-

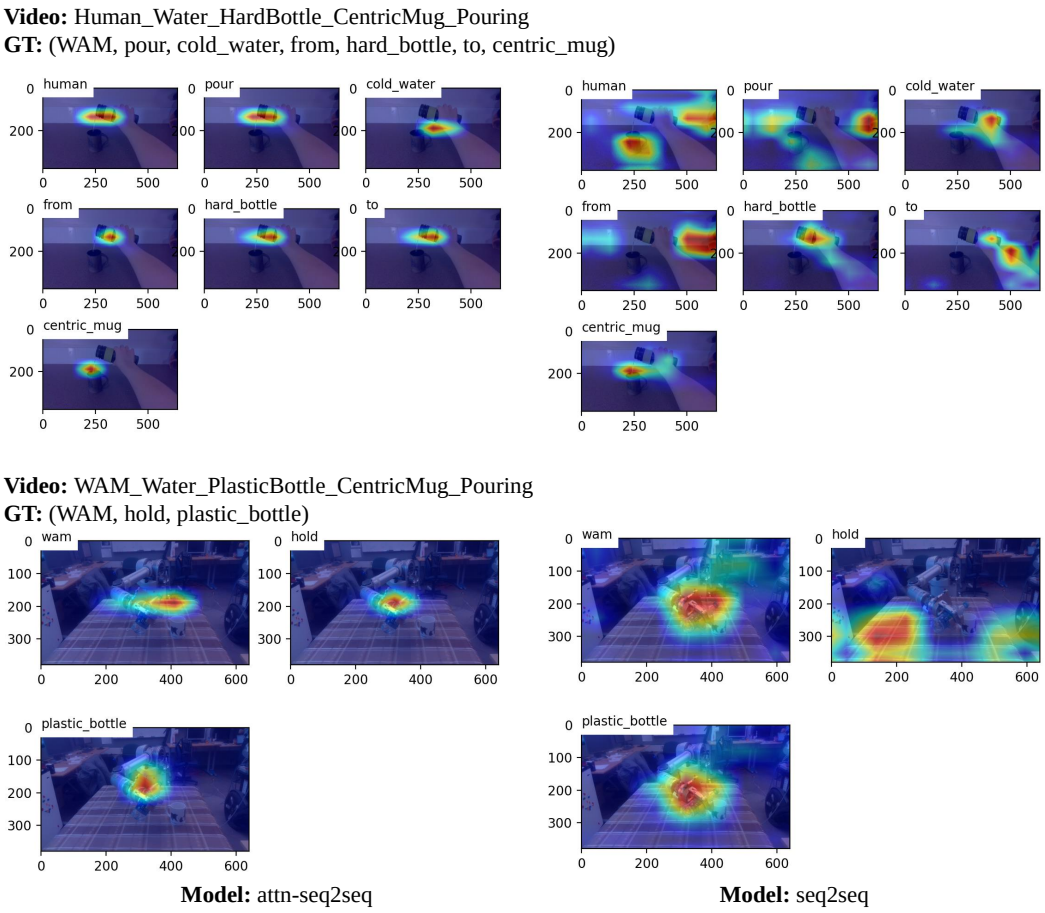


Figure 5.4: Grad-CAM visualizations for seq2seq models with visual attention vs. no attention mechanism.

ken is predicted, the associating CAM appears to be more random and chaotic, distributed across the scene in most cases. The CAMs can also become less indicative when, for example, the wrong attribute words like "centric" or "hard" is interpreted, where in this case the CAM does not fully cover the regions of the ground truth object "glass_mug" or "hard_bottle" on scene.

5.4 Summary

In this chapter, we formally presented the experiments to validate the effectiveness of our manipulation context understanding framework for robotic vision. First, experiment setups were discussed, followed by metric scores for our vision-language models. Next, we demonstrated the results of capturing

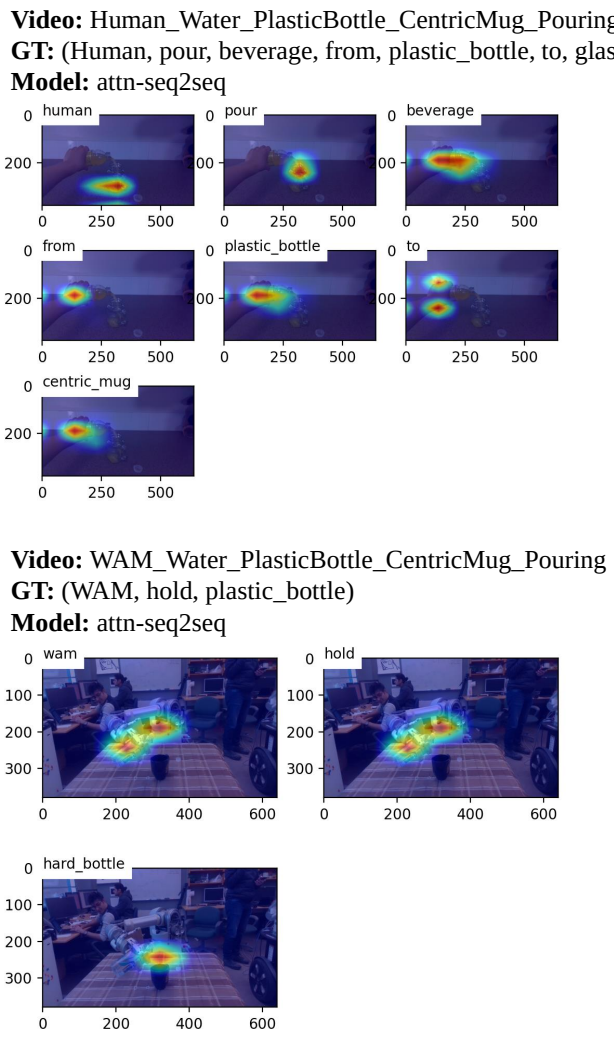


Figure 5.5: Grad-CAM visualizations for incorrectly interpreted entities.

visual attention and dynamic knowledge graphs in real time with our framework. Analysis over the pros and cons of our attention video encoding was discussed in detail.

Chapter 6

Conclusion

6.1 Contributions

In this thesis, we propose a framework and describe its key enabling methods to represent context in robot manipulation tasks by capturing visual attention and generating dynamic knowledge graphs in real time. The framework allows the fusion of any attention-based Vision-Language model with an ontology system to capture manipulation intention by pixel and by commonsense knowledge; this enables us to interpret any robot manipulation task visually and semantically in a timeliness manner. The contributions of our work is summarized as follows:

- We construct our *Robot Semantics Dataset*, which consists of manipulation tasks performed by both robots and humans. Ground truth object and action relations in forms of knowledge bases are annotated to the video frames. A specific sampling method is developed to sample observation clips with temporal skipping from videos.

- We apply a Vision-Language model using sequence-to-sequence structure with spatial attention mechanism to perform spatio-temporal encoding over real-time robotic camera stream. The Vision-Language model is able to implicitly learn spatial attention on the salient regions corresponding to the manipulation actions to the pixels, while grounding salient actions and the related objects into semantic language.

- We present a scheme to constrain manipulation knowledge into a time-independent knowledge domain using an ontology system. The ontology system stores objects and relations in a taxonomic structure, serving as the robot's commonsense knowledge over a particular domain of manipulation tasks.

- We present a framework to generate dynamic knowledge graphs over the manipulation context by combining a Vision-Language model with an ontology system. Predictions from the Vision-Language model are first instantiated, inheriting reasonable taxonomies governed by the ontology system. An algorithm for online inference is then proposed, allowing the robot to generate dynamic knowledge graphs filled with reasonable commonsense knowledge and interpreting the evolution of a manipulation task in real-time.

Apart from good performing metric scores, we analyze the key enabling methods of our framework by first inspecting the robustness of spatio-temporal encoding with feature visualizations, then presenting the correctness of our dynamic knowledge graph with reasoning capability. We successfully demonstrate that our framework works well under the real world robot manipulation scenario, allowing the robot to mimic human-like intentional behaviors and represent the evolution of an intended manipulation procedure in real time.

6.2 Future Work

The proposed framework is simple and adaptive. In future works, many aspects of the framework can be explored to enable more complex and straightforward grounding of manipulation knowledge in real time. Real time robotics are usually limited by their resources, and it is mandatory to explore network architectures with lower computational cost. With the development of new computer vision methods like transformers, the complexity of the vision-language model can be expanded to generate more robust visual attentions while ensuring spatio-temporal encoding with higher performance. Visual attention maps

themselves are also open for further interpretation, where connections between machine generated attention can possibly be associated with human eye gaze.

Apart from improving methods, a number of things still need to be explored to further evaluate the helpfulness of our framework for robotic decision making. Visual state representations can possibly be helpful with a reinforcement learning environment, encoding raw visual observations into highly robust state representations with semantic meanings. A captured dynamic knowledge graph reflects the manipulation intention at a given time period, and as such, it will be interesting to forecast future manipulation actions or interpolate key intentional visual frames from the dynamic knowledge graph. Ultimately, we hope our proposed framework can serve as our milestone to connect computer vision, robotics and knowledge representation, and it can be helpful in our future research of vision-guided robotics.

References

[1] S. A. McMains and S. Kastner, "Visual attention," in *Encyclopedia of Neuroscience*, M. D. Binder, N. Hirokawa, and U. Windhorst, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 4296–4302, ISBN: 978-3-540-29678-2. DOI: 10.1007/978-3-540-29678-2_6344. [Online]. Available: https://doi.org/10.1007/978-3-540-29678-2_6344. 1

[2] S. Morris, G. Fawcett, L. Brisebois, and J. Hughes, *Canadian survey on disability reports: A demographic, employment and income profile of canadians with disabilities aged 15 years and over, 2017 [website]*, 2018. 2

[3] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NIPS*, 2014. 3, 13, 35

[4] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. Shen, "Video captioning with attention-based lstm and semantic consistency," *IEEE Transactions on Multimedia*, vol. 19, pp. 2045–2055, 2017. 3, 13

[5] B. Zhao, X. Li, and X. Lu, "Cam-rnn: Co-attention model based rnn for video captioning," *IEEE Transactions on Image Processing*, vol. 28, pp. 5552–5565, 2019. 3, 13, 16

[6] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong, "End-to-end dense video captioning with masked transformer," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8739–8748, 2018. 3, 14

[7] Z. Fang, T. Gokhale, P. Banerjee, C. Baral, and Y. Yang, "Video2commonsense: Generating commonsense descriptions to enrich video captioning," *ArXiv*, vol. abs/2003.05162, 2020. 3, 14

[8] A. Nguyen, D. Kanoulas, L. Muratore, D. G. Caldwell, and N. G. Tsagarakis, "Translating videos to commands for robotic manipulation with deep recurrent neural networks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1–9. 3, 9, 10, 14, 47, 51, 52

[9] A. Nguyen, T.-T. Do, I. Reid, D. G. Caldwell, and N. G. Tsagarakis, "V2cnet: A deep learning framework to translate videos to commands for robotic manipulation," *arXiv preprint arXiv:1903.10869*, 2019. 3, 14, 52

[10] S. Yang, W. Zhang, W. Lu, H. Wang, and Y. L, "Learning actions from human demonstration video for robotic manipulation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1805–1811. 3, 14

[11] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, pp. 32–73, 2016. 3, 10, 14

[12] A. Fathi, X. Ren, and J. M. Rehg, "Learning to recognize objects in egocentric activities," in *CVPR 2011*, IEEE, 2011, pp. 3281–3288. 4, 9, 10

[13] A. Fathi, Y. Li, and J. M. Rehg, "Learning to recognize daily actions using gaze," in *European Conference on Computer Vision*, Springer, 2012, pp. 314–327. 4, 9, 10

[14] Y. Li, Z. Ye, and J. M. Rehg, "Delving into egocentric actions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 287–295. 4, 9, 10

[15] Y. Li, M. Liu, and J. M. Rehg, "In the eye of beholder: Joint learning of gaze and actions in first person video," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 619–635. 4, 9, 10

[16] Y. Yu, J. Choi, Y. Kim, K. Yoo, S. Lee, and G. Kim, "Supervising neural attention models for video captioning by human gaze data," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6119–6127, 2017. 4, 16

[17] Y. Li, M. Liu, and J. M. Rehg, "In the eye of beholder: Joint learning of gaze and actions in first person video," in *ECCV*, 2018. 4, 16

[18] M. Lu, Z. Li, Y. Wang, and G. Pan, "Deep attention network for egocentric action recognition," *IEEE Transactions on Image Processing*, vol. 28, pp. 3703–3713, 2019. 4, 12, 16

[19] K. Min and J. J. Corso, "Integrating human gaze into attention for egocentric activity recognition," *ArXiv*, vol. abs/2011.03920, 2020. 4, 16

[20] Z. Zhang, Y. Zhu, and S. Zhu, "Graph-based hierarchical knowledge representation for robot task transfer from virtual to physical world," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11 139–11 145, 2020. 4, 19

[21] R. Fox, R. Berenstein, I. Stoica, and K. Goldberg, "Multi-task hierarchical imitation learning for home automation," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, 2019, pp. 1–8. 4, 19

[22] R. A. M. Strudel, A. Pashevich, I. Kalevatykh, I. Laptev, J. Sivic, and C. Schmid, "Learning to combine primitive skills: A step towards versatile robotic manipulation §," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4637–4643, 2020. 4, 19

[23] T. Takayanagi, Y. Kurose, and T. Harada, "Hierarchical task planning from object goal state for human-assist robot," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, 2019, pp. 1359–1366. 4, 19

[24] M. Colledanchise, D. Almeida, and P. Ögren, "Towards blended reactive planning and acting using behavior trees," *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8839–8845, 2016. 4, 19

[25] K. French, S. Wu, T. Pan, Z. Zhou, and O. C. Jenkins, "Learning behavior trees from demonstration," *2019 International Conference on Robotics and Automation (ICRA)*, pp. 7791–7797, 2019. 4, 19

[26] L. Petrich, J. Jin, M. Dehghan, and M. Jagersand, "Assistive arm and hand manipulation: How does current research intersect with actual healthcare needs?" *arXiv preprint arXiv:2101.02750*, 2021. 7, 18

[27] R. Goyal, S. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fründ, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax, and R. Memisevic, "The "something something" video database for learning and evaluating visual common sense," *IEEE International Conference on Computer Vision (ICCV)*, pp. 5843–5851, 2017. 9, 10

[28] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "Scaling egocentric vision: The epic-kitchens dataset," in *European Conference on Computer Vision (ECCV)*, 2018. 9, 10

[29] D. Damen, H. Doughty, G. Farinella, A. Furnari, E. Kazakos, J. Ma, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "Rescaling egocentric vision," *ArXiv*, vol. abs/2006.13256, 2020. 9, 10

[30] L. Zhou, C. Xu, and J. Corso, "Towards automatic learning of procedures from web instructional videos," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018. 9, 10

[31] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips," in *ICCV*, 2019. 9, 10, 12

[32] H. Kuehne, A. B. Arslan, and T. Serre, "The language of actions: Recovering the syntax and semantics of goal-directed human activities," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 780–787, 2014. 9, 10

[33] J. Ji, R. Krishna, L. Fei-Fei, and J. C. Niebles, "Action genome: Actions as compositions of spatio-temporal scene graphs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 236–10 247. 9, 11, 15

[34] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding," *ArXiv*, vol. abs/1604.01753, 2016. 9, 11

[35] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, A. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," *ArXiv*, vol. abs/1705.06950, 2017. 10

[36] S. Sudhakaran and O. Lanz, "Attention is all we need: Nailing down object-centric attention for egocentric activity recognition," *ArXiv*, vol. abs/1807.11794, 2018. 11, 17

[37] A. Furnari and G. Farinella, "What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6251–6260, 2019. 12

[38] S. Sudhakaran, S. Escalera, and O. Lanz, "Lsta: Long short-term attention for egocentric action recognition," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9946–9955, 2019. 12

[39] Z. Li, Y. Huang, M. Cai, and Y. Sato, "Manipulation-skill assessment from videos with spatial attention network," *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 4385–4395, 2019. 12, 16

[40] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4489–4497, 2015. 12

[41] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4733, 2017. 12

[42] A. Miech, J.-B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman, "End-to-End Learning of Visual Representations from Uncurated Instructional Videos," in *CVPR*, 2020. 12

[43] U. Demir, Y. Rawat, and M. Shah, "Tinyvirat: Low-resolution video action recognition," *ArXiv*, vol. abs/2007.07355, 2020. 12

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *ArXiv*, vol. abs/1706.03762, 2017. 12

[45] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, "Video action transformer network," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 244–253, 2019. 12, 16

[46] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" *ArXiv*, vol. abs/2102.05095, 2021. 12, 16

[47] C. Pohlt, T. Schlegl, and S. Wachsmuth, "Weakly-supervised learning for multimodal human activity recognition in human-robot collaboration scenarios," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8381–8386, 2020. 13

[48] C. Mason, K. Gadzicki, M. Meier, F. Ahrens, T. Kluss, J. Maldonado, F. Putze, T. Fehr, C. Zetzsche, M. Herrmann, K. Schill, and T. Schultz, "From human to robot everyday activity," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8997–9004, 2020. 13

[49] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015. 13

[50] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Colorado: Association for Computational Linguistics, May 2015, pp. 1494–1504. DOI: 10.3115/v1/N15-1173. [Online]. Available: https://www.aclweb.org/anthology/N15-1173. 13

[51] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4534–4542. 13, 52

[52] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *ICML*, 2015. 13

[53] C. Lu, R. Krishna, M. S. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," in *ECCV*, 2016. 14

[54] B. Zhuang, L. Liu, C. Shen, and I. Reid, "Towards context-aware interaction recognition for visual relationship detection," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 589–598, 2017. 14

[55] J. Materzynska, T. Xiao, R. Herzig, H. Xu, X. Wang, and T. Darrell, "Something-else: Compositional action recognition with spatial-temporal interaction networks," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1046–1056, 2020. 14

[56] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, "Neural motifs: Scene graph parsing with global context," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5831–5840, 2018. 15

[57] S. Aditya and C. Baral, "Deepiu : An architecture for image understanding," 2016. 15

[58] S. Aditya, Y. Yang, C. Baral, Y. Aloimonos, and C. Fermüller, "Image understanding using vision and reasoning through scene description graph," *Comput. Vis. Image Underst.*, vol. 173, pp. 33–45, 2018. 15

[59] H. Zhang, X. Lan, X. Zhou, Z. Tian, Y. Zhang, and N. Zheng, "Visual manipulation relationship network for autonomous robotics," *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pp. 118–125, 2018. 15

[60] Y. Yang, A. Guha, C. Fermüller, and Y. Aloimonos, "Manipulation action tree bank: A knowledge resource for humanoids," *2014 IEEE-RAS International Conference on Humanoid Robots*, pp. 987–992, 2014. 15

[61] Y. Yang, A. Guha, C. Fermüller, Y. Aloimonos, and A. V. Williams, "A cognitive system for understanding human manipulation actions," 2014. 15

[62] Y. Yang, Y. Li, C. Fermüller, and Y. Aloimonos, "Robot learning manipulation action plans by "watching" unconstrained videos from the world wide web," in *AAAI*, 2015. 15

[63] H. Zhang and S. Nikolaidis, "Robot learning and execution of collaborative manipulation plans from youtube videos," *arXiv preprint arXiv:1911.10686*, 2019. 15

[64] J. Hatori, Y. Kikuchi, S. Kobayashi, K. Takahashi, Y. Tsuboi, Y. Unno, W. Ko, and J. Tan, "Interactively picking real-world objects with unconstrained spoken language instructions," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3774–3781, 2017. 15

[65] M. Shridhar and D. Hsu, "Interactive visual grounding of referring expressions for human-robot interaction," *ArXiv*, vol. abs/1806.03831, 2018. 15

[66] J. Thomason, A. Padmakumar, J. Sinapov, N. Walker, Y. Jiang, H. Yedidsion, J. W. Hart, P. Stone, and R. J. Mooney, "Improving grounded natural language understanding through human-robot dialog," *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6934–6941, 2019. 15

[67] F. Yan, D. Wang, and H. He, "Robotic understanding of spatial relationships using neural-logic learning," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8358–8365, 2020. 15

[68] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014. 16

[69] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057. 16

[70] S. Woo, J. Park, J.-Y. Lee, and I.-S. Kweon, "Cbam: Convolutional block attention module," in *ECCV*, 2018. 16

[71] L. Meng, B. Zhao, B. Chang, G. Huang, W. Sun, F. Tung, and L. Sigal, "Interpretable spatio-temporal attention for video action recognition," *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 1513–1522, 2019. 16

[72] D.-P. Fan, W. Wang, M.-M. Cheng, and J. Shen, "Shifting more attention to video salient object detection," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8546–8556, 2019. 16

[73] P. Abolghasemi, A. Mazaheri, M. Shah, and L. Bölöni, "Pay attention! - robustifying a deep visuomotor policy through task-focused visual attention," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4249–4257, 2019. 17, 39

[74] K. Ramachandruni, M. Vankadari, A. Majumder, S. Dutta, and S. Kumar, "Attentive task-net: Self supervised task-attention network for imitation learning using video demonstration," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4760–4766, 2020. 17

[75] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Computer Vision and Pattern Recognition*, 2016. 17

[76] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, pp. 336–359, 2019. 17, 57

[77] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 839–847, 2018. 17

[78] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 111–119, 2020. 17

[79] K. Li, Z. Wu, K. Peng, J. Ernst, and Y. Fu, "Tell me where to look: Guided attention inference network," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9215–9223, 2018. 17

[80] V. Ramanishka, A. Das, J. Zhang, and K. Saenko, "Top-down visual saliency guided by captions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7206–7215. 17

[81] A. Saxena, A. Jain, O. Sener, A. Jami, D. Misra, and H. Koppula, "Robobrain: Large-scale knowledge engine for robots," *ArXiv*, vol. abs/1412.0691, 2014. 18

[82] D. Misra, K. Tao, P. Liang, and A. Saxena, "Environment-driven lexicon induction for high-level instructions," in *ACL*, 2015. 18

[83] D. Paulius and Y. Sun, "A survey of knowledge representation in service robotics," *Robotics Auton. Syst.*, vol. 118, pp. 13–30, 2019. 18

[84] D. Paulius, Y. Huang, J. Meloncon, and Y. Sun, "Manipulation motion taxonomy and coding for robots," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5596–5601, 2019. 18

[85] Y. Yang, A. Guha, C. Fermüller, and Y. Aloimonos, "Manipulation action tree bank: A knowledge resource for humanoids," in *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 987–992. 19

[86] H. Zhang, X. Lan, S. Bai, L. Wan, X. Zhou, and N. Zheng, "A multi-task convolutional neural network for autonomous robotic grasping in object stacking scenes," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6435–6442, 2018. 19

[87] T. Welschehold, N. Abdo, C. Dornhege, and W. Burgard, "Combined task and action learning from human demonstrations for mobile manipulation applications," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4317–4324, 2019. 19

[88] S. U. Lee, A. Hofmann, and B. C. Williams, "A model-based human activity recognition for human–robot collaboration," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 736–743, 2019. 19

[89] M. Siam, C. Jiang, S. Lu, L. Petrich, M. Gamal, M. Elhoseiny, and M. Jagersand, "Video object segmentation using teacher-student adaptation in a human robot interaction (hri) setting," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 50–56. 29

[90] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *NIPS*, 2015. 38

[91] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *arXiv preprint arXiv:1310.4546*, 2013. 38, 49

[92] M. A. Musen, "The protégé project: A look back and a look forward," *AI matters*, vol. 1, no. 4, pp. 4–12, 2015. 39, 49

[93] J.-B. Lamy, "Owlready: Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies," *Artificial intelligence in medicine*, vol. 80, pp. 11–28, 2017. 49

[94] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, "Microsoft coco captions: Data collection and evaluation server," *arXiv preprint arXiv:1504.00325*, 2015. 50

Appendix A

Appendix

A.1 Attention Maps for IIT-V2C Dataset

Visualizations of attention maps for IIT-V2C dataset are shown in Figure A.1.

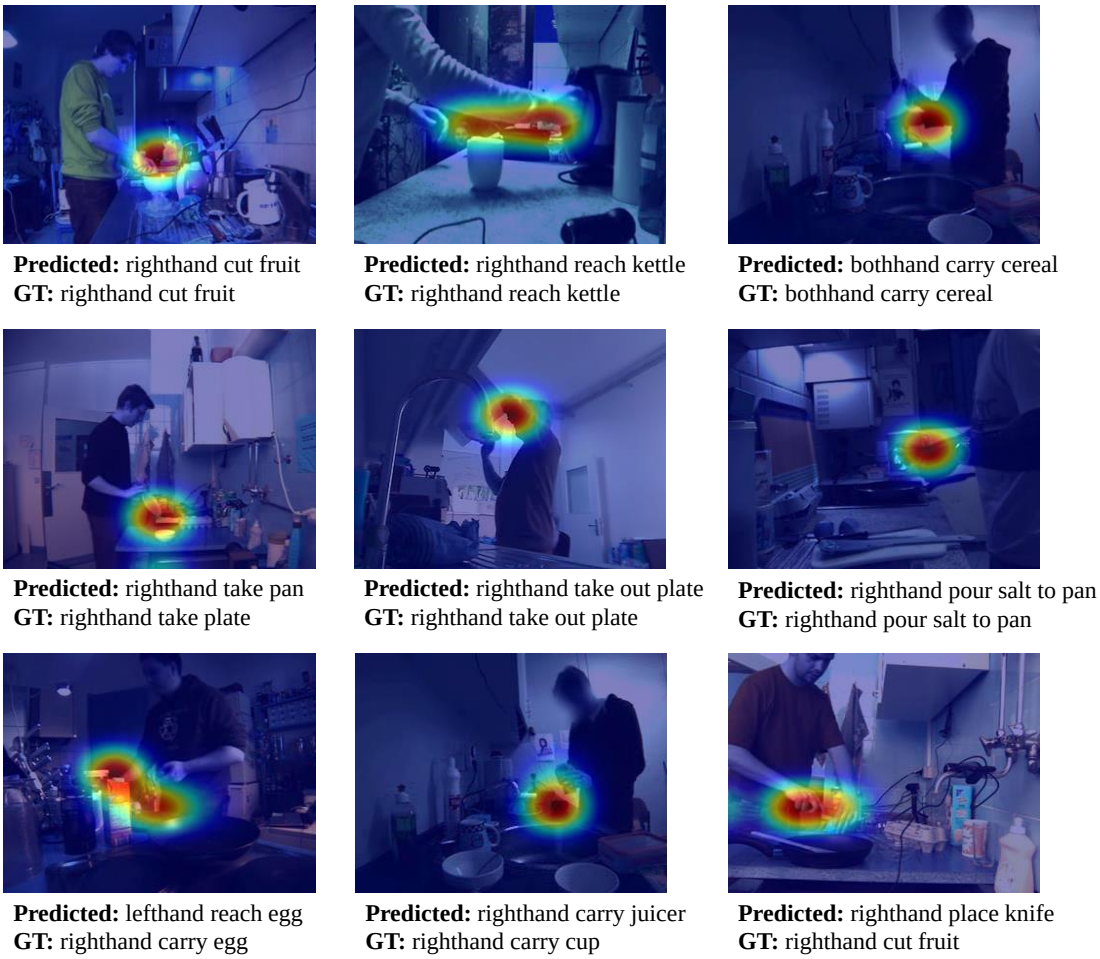
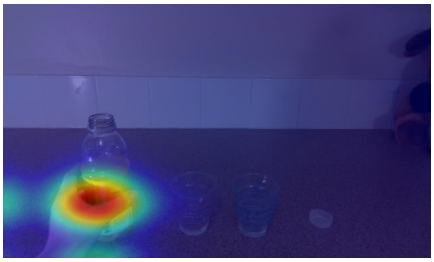
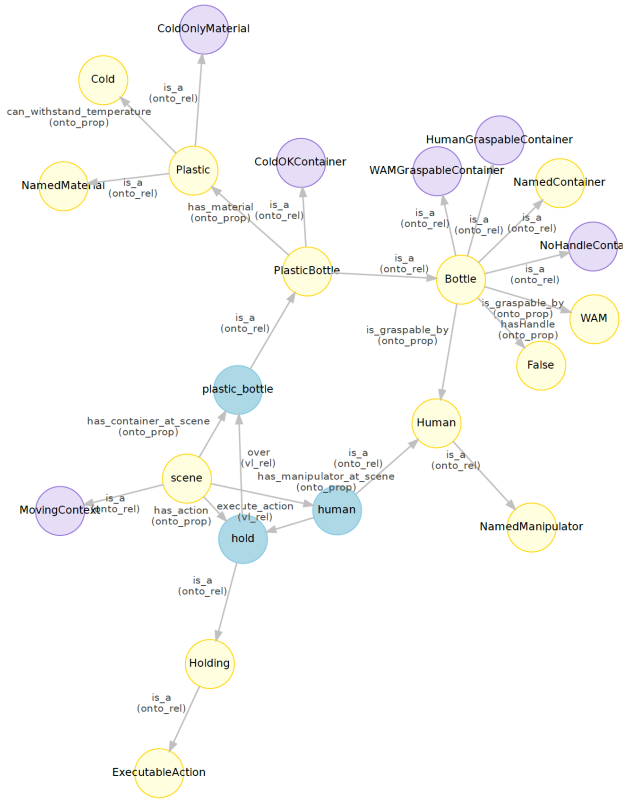


Figure A.1: Visualization of attention maps for IIT-V2C dataset.

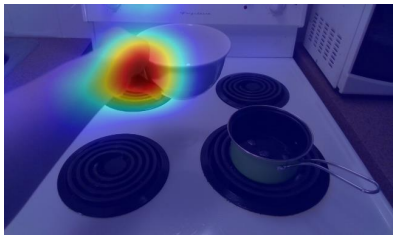
A.2 More Results for Dynamic Knowledge Graph

We attach additional visualizations of dynamic knowledge graphs for different manipulation tasks from our Robot Semantics Dataset. The visualized dynamic knowledge graph contains all queried and reasoned ontological facts, based on command language predictions from our vision-language model.

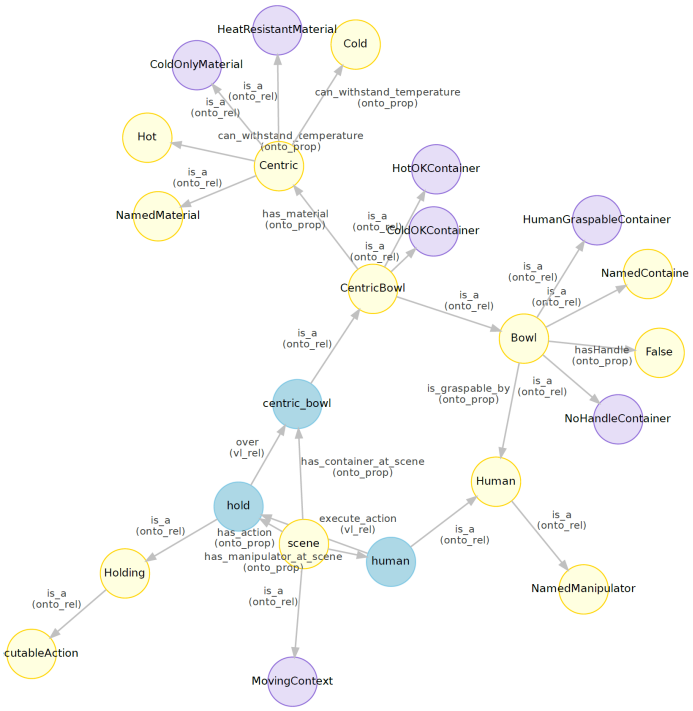


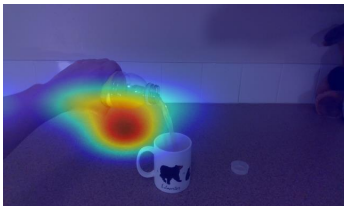


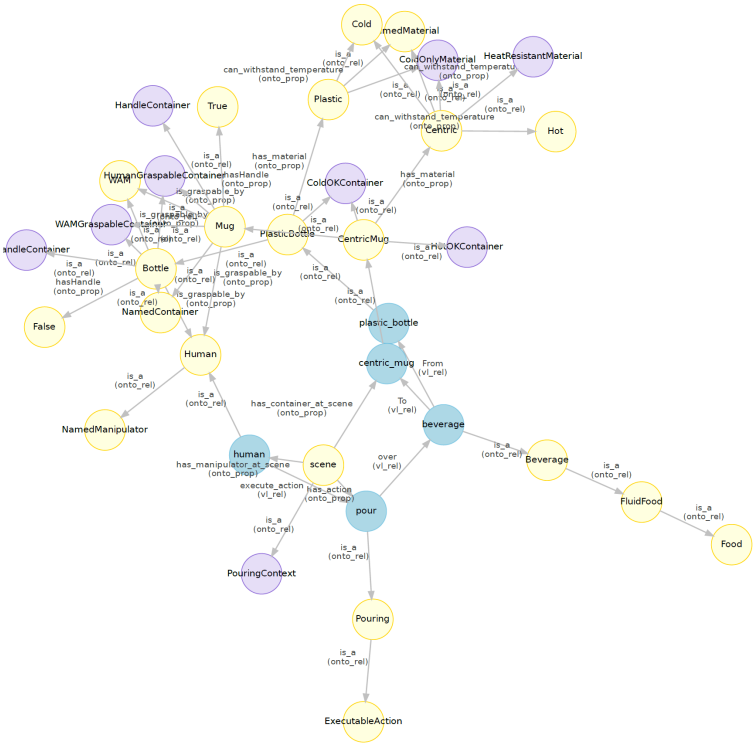
GT: (human, hold, plastic_bottle)
Predicted: (human, hold, plastic_bottle)



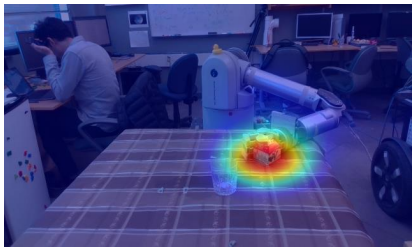
GT: (human, hold, centric_bowl)
Predicted: (human, hold, centric_bowl)

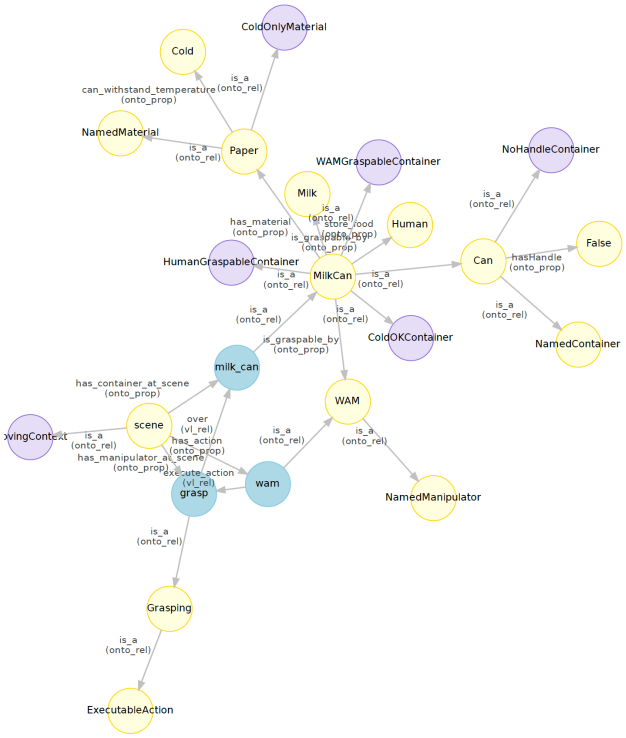




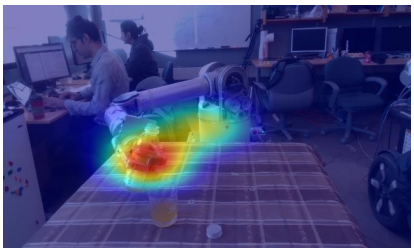


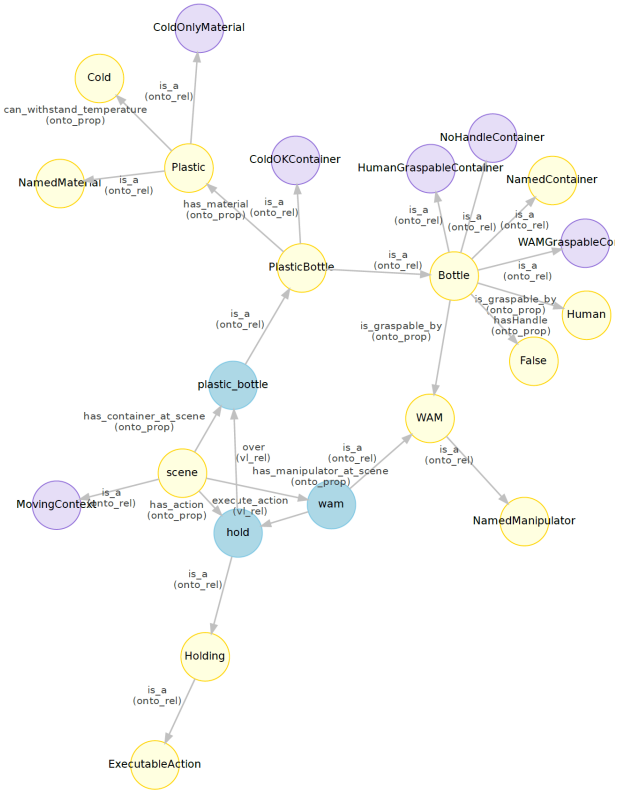
GT: (human, pour, beverage, from, plastic_bottle, to, centric_mug)
Predicted: (human, pour, beverage, from, plastic_bottle, to, centric_mug)



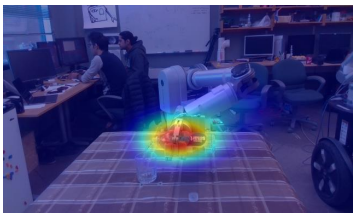


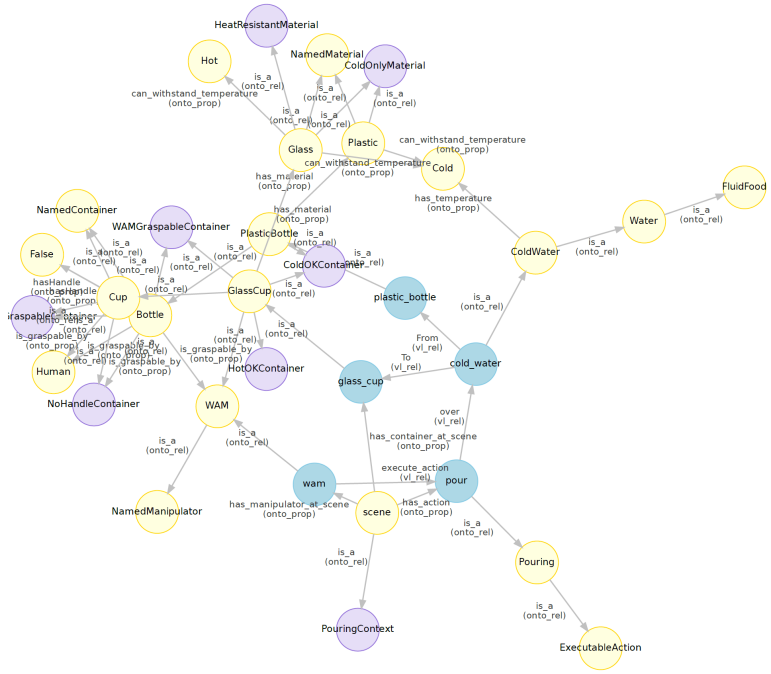
GT: (WAM, grasp, milk_can)
Predicted: (WAM, grasp, milk_can)





GT: (WAM, hold, plastic_bottle)
Predicted: (WAM, hold, plastic_bottle)





GT: (WAM, pour, cold_water, from, plastic_bottle, to, glass_cup)
Predicted: (WAM, pour, cold_water, from, plastic_bottle, to, glass_cup)