UNIVERSITY OF CALIFORNIA

Los Angeles

Incorporating World Model Knowledge into Event Parsing, Prediction, and Reasoning

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Computer Science

by

Baoxiong Jia

2022

ABSTRACT OF THE DISSERTATION

Incorporating World Model Knowledge into Event Parsing, Prediction, and Reasoning

by

Baoxiong Jia
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2022
Professor Song-chun Zhu, Chair

Event understanding is one of the most fundamental problems in artificial intelligence and computer vision. Rooted in the field of neuroscience, the study and analysis of human motion perception have long suggested that we perceive human activities as *goal-directed* behaviors. As an essential capability of humans, we interpret others' goals and learn tasks through the endless video stream of daily activities. To endow machines with the same intelligent behaviors, the challenges of emerging such a capability lie in the difficulty of generating a detailed understanding of world model knowledge including situated actions, their effects on object states (*i.e.*, state changes), and their causal dependencies. These challenges are further aggravated by the natural parallelism in human multi-tasking, and partial observations originated from both the egocentric perception and uncertainties in estimating others' beliefs in multi-agent collaborations.

In this dissertation, we propose to study this missing gap from both the data and the modeling perspective by incorporating knowledge of the world model for proper event parsing, prediction, and reasoning. First, we propose three datasets, RAVEN, LEMMA, and EgoTaskQA, to study the event understanding problem from both the abstract and real

domain. We further devise three benchmarks to evaluate models' detailed understanding of events with (1) intelligence tests for spatial-temporal reasoning in RAVEN, (2) compositional action recognition and prediction in LEMMA, and (3) task-conditioned question answering in EgoTaskQA. Next, from the modeling side, we decompose the problem of event understanding into a unified framework that involves three essential modules: grounding, inference, and the knowledge base. To properly solve the problem of detailed event understanding, we need to focus on (1) the perception problem for grounding, (2) the knowledge representation problem, and (3) the inference problem. For the perception problem, we discuss the potential in existing models and propose the BO-QSA for the unsupervised emergence of object-centric concepts. For the inference problem, we discuss ways to initialize the overall framework with (1) PrAE which makes use of probabilistic abductions given logical rules, and (2) GEP which leverages stochastic context-free grammars for modeling. We conduct experiments to show their effectiveness on various tasks and also discuss the limitations of each proposed work to highlight immediate next steps for possible future directions.

The dissertation of Baoxiong Jia is approved.

Demetri Terzopoulos

Kai-Wei Chang

Ying Nian Wu

Song-chun Zhu, Committee Chair

University of California, Los Angeles

2022

To my parents and Ying

for their unconditional support along the journey.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First, I want to express my deepest gratitude to my advisor Prof. Song-Chun Zhu, for introducing me to the field of computer vision and artificial intelligence. I still remember my first day at UCLA as a graduate student who was confused about his future path. His lectures, *Pattern Recognition and Machine Learning*, attracted me to the field with his handwritten notes and detailed illustrations. In addition to all the machine learning knowledge I have acquired, I learned from his visionary thoughts that connect methods as pieces to form a big picture. These insights shaped my understanding of AI and continue to inspire me to this day. His passion for research has shown me a fine example of how to be a good researcher and encouraged me to tackle the most challenging topics.

I also treasure the support from my committee members, Prof. Ying Nian Wu, Prof. Demetri Terzopoulos, and Prof. Kai-Wei Chang. I have learned from Prof. Wu how to be rigorous on mathematical derivations as a statistician, from Prof. Terzopoulos the field of computer graphics, and from Prof. Chang his inspiring works in natural language understanding. Their insightful ideas and expertise across fields broadened my horizon and helped me develop good intuitions, habits, and research attitudes.

I'm very fortunate to be a member of the VCLA. This is the first research lab that made me feel like a family and I sincerely thank all my lab members for building this incredible working environment. I would like to thank especially the following people:

Dr. Siyuan Qi, for being the best mentor and friend one could ever wish to have when entering a new field. He showed me how to become a good researcher and led me through every step that a student researcher needs to face. He is also an interesting and passionate person outside of research, who introduced me to guitars, which later helped me pass numerous nights during the COVID quarantine.

Dr. Feng Gao, for being the most caring and energetic long-term friend and collaborator. He showed me how to live a life and let go of the pressures by introducing many new things

and places to me. He has also supported me during my hard times by providing various help and cares at very critical moments. To that, I'm always grateful and the experiences we have shared shall always be valuable to me.

Prof. Yixin Zhu, for being an amazing leader and mentor inside and outside our lab. He is one of the magic glues that shaped the friendly and caring atmosphere at VCLA and I'm amazed by his management skills. His research insights on cognitive psychology also helped me a lot when I was a junior student. There is always something to learn from him.

Dr. Siyuan Huang, for being a wonderful friend and patient mentor. I met Siyuan first in DMAI and was fascinated by his research style. He showed me how to enjoy research and shaped my research habit. We have had discussions and debates regarding numerous topics inside and outside research, and, with his acute opinions, I'm always learning new things.

Mr. Sirui Xie, for sharing his always insightful ideas regarding life and research and for being a caring friend. I still remember the late-night talk at Weyburn where we discussed various topics and felt mentally connected. It is nice to have Sirui as a friend from whom you can always expect deep thoughts with a sharp mind.

Dr. Chi Zhang, for being a hardworking and pleasant friend and collaborator. I'm always learning from his self-discipline in research and life. He is also shy to share details in his life, leaving us a huge space for imagination which later contributed to many enjoyable anecdotes. I sincerely hope that at least some of the stories we made up for him could become true.

Dr. Yixin Chen, Dr. Tao Yuan, and Dr. Qing Li, for being awesome friends and collaborators. These guys helped me in my early Ph.D. studies and also provided a lot of support during the initial COVID quarantine. They were some of my only social connections back then that helped me feel connected with other people.

Ms. Shuwen Qiu and Ms. Yuxing Qiu, for their caringness and help. Shuwen is a nice and caring person who shared the same interest in music. She also introduced me to my girlfriend, Ying, who later supported me during my Ph.D. study. Yuxing and I started our Ph.D. studies

at the same time and shared a lot of common friends. She and Feng's apartment is like a second home to me in LA. I enjoyed playing with their cat, Mianhua, who is always interested in my socks.

Dr. Xu Xie, Dr. Mark Edmonds, Dr. Xiaojian Ma, Dr. Hangxin Liu, Mr. Ziyuan Jiao, Mr. Zeyu Zhang, and Mr. Muzhi Han at the robotics lab in Slitcher hall for being the best labmates that I have ever expected. I will always remember the nights at green house, blaze, and yoshinoya where we discussed numerous topics joyfully.

I also appreciate the helpful discussions with Dr. Zilong Zheng, Dr. Tengyu Liu, Dr. Lifeng Fan, Dr. Ruiqi Gao, Dr. Luyao Yuan, Dr. Arjun Arkula, Dr. Xiaofeng Gao, Dr. Tianmin Shu, Dr. Keze Wang, Dr. Zhixiong Nan, Dr. Erik Nijkamp, Dr. Liang Qiu, Mr. Shu Wang, Mr. Hanlin Zhu, Ms. Yining Hong, Mr. Pan Lu, and Mr. Ran Gong at VCLA. It is also a pleasure to work with Dr. Qing Ping who hosted my internship at Amazon Alexa AI.

Finally, I would like to express my deepest thanks to my parents for their selfless support at every stage of my life. Their never-ending love and precious character guide me to become the person I am. Of that, I'm always in debt. I thank my girlfriend, Ying, for her love and companionship throughout my Ph.D. study. Her caring helped me survive the arduous COVID quarantine.

VITA

2014–2018	B.S. (Computer Science), Peking University, Beijing.
2018–2019	M.S. (Computer Science), UCLA, Los Angeles, California.
2020	Ph.D. Candidate in Computer Science, UCLA, Los Angeles, California.
2017-2019	Graduate Student Researcher, UCLA, Los Angeles, California
2020-2021	Teaching Assistant, Computer Science, UCLA, Los Angeles, California.
2021	Applied Scientist Intern, Amazon, Virtual.
2021-present	Research Scientist Intern, Beijing Institute for General Artificial Intelligence (BIGAI), Beijing.

PUBLICATIONS

* denotes equal contribution.

Baoxiong Jia, Ting Lei, Song-Chun Zhu, Siyuan Huang. EgoTaskQA: Understanding Human Tasks in Egocentric Videos. NeurIPS 2022.

Chi Zhang*, Sirui Xie*, **Baoxiong Jia***, Yixin Zhu, Ying Nian Wu, Song-Chun Zhu. Learning Algebraic Representation for Systematic Generalization in Abstract Reasoning. ECCV 2022.

Peiyu Yu, Sirui Xie, Xiaojian Ma, **Baoxiong Jia**, Bo Pang, Ruiqi Gao, Yixin Zhu, Song-Chun Zhu, Ying Nian Wu. ICML 2022.

Chi Zhang*, **Baoxiong Jia***, Song-Chun Zhu, Yixin Zhu. Abstract Spatial-Temporal Reasoning via Probabilistic Abduction and Execution. CVPR 2021.

Chi Zhang, **Baoxiong Jia**, Mark Edmonds, Song-Chun Zhu, Yixin Zhua. ACRE: Abstract Causal REasoning Beyond Covariation. CVPR 2021.

Baoxiong Jia, Yixin Chen, Siyuan Huang, Yixin Zhu, Song-chun Zhu. LEMMA: A Multiview Dataset for LEarning Multi-agent Multi-task Activities. ECCV 2020.

Siyuan Qi, **Baoxiong Jia**, Siyuan Huang, Ping Wei, Song-chu Zhu. A Generalized Earley Parser Human Activity Parsing and Prediction. TPAMI 2020.

Chi Zhang*, **Baoxiong Jia***, Feng Gao, Yixin Zhu, Hongjing Lu, Song-chun Zhu. Learning Perceptual Inference by Contrasting. NeurIPS 2019.

Chi Zhang*, Feng Gao*, **Baoxiong Jia**, Yixin Zhu, Song-chun Zhu. RAVEN: A Dataset for Relational and Analogical Visual rEasoNing. CVPR 2019.

Siyuan Qi*, Wenguan Wang*, **Baoxiong Jia**, Jianbing Shen, Song-chun Zhu. Learning Human-Object Interactions by Graph Parsing Neural Networks. ECCV 2018.

Siyuan Qi, **Baoxiong Jia**, Song-chun Zhu. 2018. Generalized Earley Parser: Bridging Symbolic Grammars and Sequence Data for Future Prediction. ICML 2018.

CHAPTER 1

Introduction

1.1 Goal-oriented Human Activities

As the most readily available learning source, videos of daily human activities could be used to train intelligent agents and, in turn, to assist humans. However, compared to recent progress in learning from static images [AAL15, HZR16, HGD17, RHG15], current machine vision's ability to understand activities from videos still falls short. Admittedly, event understanding is inherently more challenging, which requires reason about the complex structures in activities along the additional temporal dimension; but there are more profound reasons that we must look back to the origin of activity understanding.

The study and analysis of human motion perception are rooted in the field of neuroscience [TCS08]. Using a dot-representation of human motions, Johansson [Joh73] adopted a method to produce proximal patterns (*i.e.*, the moving light display experiment), which demonstrated that human perception of events is not tightly coupled with *pixel-based features*; human subjects can still perceive the semantics of activities from *sparse* representations of motions. Evidence from developmental psychology, the classic Heider-Simmel experiment, further suggests that we perceive human activities as **goal-directed** behaviors [Woo98, BBS01, GBK02, CG07]; it is the underlying intent, rather than the surface pixels or behavior, that matters when we observe motions [BB01].

Following this line of thought, cognitive studies suggest that we approach this goal-attribution problem through three major mechanisms: a) action-effect association, b) em-

bodied simulation and c) teleological reasoning [CG07]. From the perspective of action-effect association, goals or desired effects automatically activate the corresponding action, while the activation of an action elicits the anticipation of the distal effect associated with it [HMA01, Woo98]. The embodied simulation theory conjectures that people imagine themselves in the other's position and simulatively generate mental states (beliefs, desires, intentions) in the other's "shoes" for understanding and predicting behaviours. This hypothesis is further boosted by the discovery of mirror neuron areas in humans [BSS07, GEM07, GG98]. At last, teleological reasoning emphasizes that people infer the goals of others by first reasoning on the accessible goal states given current situational constraints. The principle of rational action is then adopted for evaluating the efficiency of each approach toward the goal, which further leads to goal prediction and future action prediction [CG98, Csi03, GNC95]. Despite their differences, all three mechanisms require detailed knowledge of **action dependencies and effects**. With such knowledge playing crucial roles in human cognitive development, learning them from visual observation is pivotal for building more intelligent agents.

Finally, daily human activities are intrinsically multi-tasked [Mon03, RME01]. Understanding activity naturally demands a learning system to interpret concurrent interactions. As agents' decision-making processes are deeply affected by their unique social values, task scheduling is significantly affected by interactions (*e.g.*, cooperation, competition, subordination) among multi-agents [KHA16]. These observations implicate that the machine vision system must objectively understand how a given task should be decomposed into atomic-actions, how multi-tasks should be executed and coordinated in parallel among multi-agents, and take the perspective from human agents to understand why the observed human activities are optimal solutions. Such a **decompositional, multi-task, multi-agent, diagnostic-driven, social perspective** of event understanding is critical for an intelligent agent to understand human behavior and team with humans collaboratively.

1.2 Evaluating Spatial-Temporal Understanding

In spite of the importance of event understanding, the key factors as discussed in Section 1.1 have been largely left untouched in current video-related research. The majority of the recent progress in video understanding has been focusing on action recognition, captioning, and future anticipation, especially in an embodied egocentric view [SZS12, KTS14, CEG15, CZ17, FKE18, SGS18, LLR18, JCH20, DDF22, GWB22]. However, these tasks merely cover the tip of the iceberg, considering how humans learn from visual observations to obtain knowledge for profound tasks like learning world models, planning for desired goals, and building beliefs about others.

Motivated by the deficiencies of existing works, we start from a synthetic abstract domain, RAVEN [ZGJ19], to evaluate the model's capabilities on the abstract spatial-temporal understanding and reasoning task through intelligent quotient tests. When viewing from the video perspective, columns in these tasks for reasoning could be treated as consecutive time frames inside the videos. With simple shapes, this task offers a clean environment as a test bed for evaluating different sequence modeling methods without perceptual level difficulties. Next, we introduce the LEMMA dataset [JCH20] to go a step further toward the innate difficulties in real-world human activity modeling. By quantifying the scenarios to up to two multi-step tasks with two agents, we strive to address detailed human multi-task and multi-agent interactions in our videos for task-oriented event understanding. Finally, we present EgoTaskQA [JLZ22], a challenging egocentric, goal-oriented video question-answering benchmark built on top of LEMMA. By extending the LEMMA dataset with annotations consisting of object status, human-object and multi-agent relationships, and causal dependency structures between actions, we design different questions that target various aspects of event and task understanding. We provide the details of the proposed evaluation benchmarks in Part I.

1.3 Modeling Sequential Events

For sequence modeling, we use the POMDP [KLC98] framework to illustrate existing problems that need to be solved. The basic elements of the formulation could be defined by the tuple $\langle S, A, T, V, \Omega, O, G\rangle$ where S is a set of states representing the world; A is a set of actions that an agent can perform in the environment; $T : S \times A \mapsto \Pi(S)$ is the state-transition function, giving for each world state and agent action, a probability distribution over world states (we use $T(s, a, s')$ or $P_{\text{env}}(s'|s, a)$ to denote this probability); $V : S \mapsto \mathbb{R}$ is the value function, giving the value of states that agents can reach; Ω is a set of observation the agent can experience or observe; $O : S \mapsto \Pi(\Omega)$ is the observation function, which gives, for each action and resulting state; and $G \subseteq S$ is a set of goal state that an agent is trying to achieve, or more intuitively, the final world state an agent is trying to arrive at.

For a real-world problem, we are provided with the observed sequence $o_{1:T} = \{o_1, o_2, \cdots, o_t\}$ and history action sequence $a_{1:T} = \{a_1, a_2, \cdots, a_t\}$. We use g to denote the potential goal state of an agent. The probability of the event with latent dynamics and observations of both scene configurations and human actions, *i.e.*, $\tau = \{o_1, a_1, o_2, a_2, \cdots, o_{t-1}, a_{t-1}, o_t\}$, can be formulated as:

$$\begin{aligned} P(\tau, I_{0:T}; \Delta) =& P(s_0, a_0, o_0, ..., s_T, a_T, o_T; \Delta) \\ =& P(o_0|s_0; \Delta)P(s_0)\prod_{t=1}^{T} P(a_t|s_t; \Delta)P(s_t|s_{t-1}, a_t; \Delta)P(o_t|s_t; \Delta) \\ \propto& P(o_0|s_0; \Delta)\prod_{t=1}^{T} P(a_t|s_t; \Delta)P(s_t|s_{t-1}, a_t; \Delta)P(o_t|s_t; \Delta) \\ =& P(o_{0:T}|s_{0:T}; \Delta)\prod_{t=1}^{T} P(a_t|s_t; \Delta)\ P(s_t|s_{t-1}, a_t; \Delta) \end{aligned} \tag{1.1}$$

where Δ contains knowledge of goal set G, values R, as well as needed parameters for the underlying representation of states and actions. Eq. (1.1) points out the three key challenges toward good activity understanding and learning from demonstration:

(1) What is a good representation space of $\mathcal{S}$ such that fits for both visual generation and projection ($P(o|s)$) and latent world dynamics building ($P(s'|s,a)$)?
(2) How do we capture world dynamics ($P(s'|s,a)$) from raw pixels with limited supervision?
(3) How do we model action policies of humans ($P(a|s)$)?

In this dissertation, we focus on the first two problems. The challenge in solving these problems resides in the efficiency in properly representing both world states and dynamics. To answer the question of what is a good representation for knowledge, we have some key properties that such representations must possess: (i) the representation should have a compositional syntax and semantics, (ii) the operations defined over these representations are sensitive only to their syntax. This hypothesis is also referred to as the Language of Thought Hypothesis (LOTH), which states that human thinking occurs in a mental language that has the systematic generalization ability to generalize from interrelations. Important as it is, disentangling such concepts from visual stimuli is an exceedingly difficult task to accomplish with limited supervision [GVS20] and requires proper inductive biases [SLB21]. Therefore, we propose Bi-level Optimized Query Slot Attention (BO-QSA) at the beginning of Part II to study the architectural inductive biases that a model should possess for learning groundable concepts from static images. Following this discussion, we aim to address the second problem by making basic assumptions on video representations and incorporating world dynamics knowledge of different forms for sequential event modeling. We show that with three different knowledge representations (logic, grammar) and their corresponding inference module (PrAE [ZJZ21], Generalized Earley Parser (GEP) [QJH20]), we can improve the capability of existing models on modeling sequential events. We leave the discussion on this topic to the second half of Part II.

Part I

Evaluating Spatial-Temporal Understanding

CHAPTER 2

RAVEN: A Dataset for Relational and Analogical Visual Reasoning

Computer vision has a wide spectrum of tasks. Some computer vision problems are clearly purely visual, "capturing" the visual information process; for instance, filters in early vision [CR68], primal sketch [GZW07] as the intermediate representation, and Gestalt laws [KK79] as the perceptual organization. In contrast, some other vision problems have trivialized requirements for perceiving the image, but engage more generalized problem-solving in terms of relational and/or analogical visual reasoning [HHT96]. In such cases, the vision component becomes the "basis for decisions about our thoughts and actions". With dramatic progress made in relational reasoning through the task of Visual Question Answering (VQA) [AAL15, JHV17, RKZ15, YWG18, ZGB16], the reasoning capability required in such tasks lies only at the periphery of the cognitive test circle [CJS90], especially when compared to core reasoning capabilities like spatial-temporal and analogical reasoning. To push the limit of computer vision towards the center of cognitive ability test circle, we need a test originally designed for measuring human intelligence on these reasoning problems to challenge, debug, and improve the current artificial systems.

A surprisingly effective ability test of human visual reasoning has been developed and identified as the Raven's Progressive Matrices (RPM) [KMG13, Rav38, SCS13], which is widely accepted and believed to be highly correlated with real intelligence [CJS90]. Unlike VQA, RPM lies directly at the center of human intelligence [CJS90], is diagnostic of abstract and structural reasoning ability [EKM84], and characterizes the defining feature of high-level

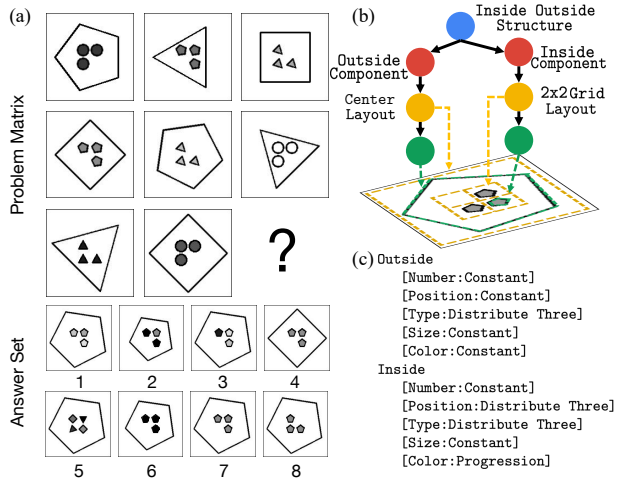


Figure 2.1: A visualization of RPM (a), structural information (b), and compositional rules (c).

intelligence, *i.e.*, *fluid intelligence* [JBJ08].

Figure 2.1 shows an example of RPM problem together with its structure representation. Provided two rows of figures consisting of visually simple elements, one must efficiently derive the correct image structure (Figure 2.1(b)) and the underlying rules (Figure 2.1(c)) to jointly reason about a candidate image that best completes the problem matrix. In terms of levels of reasoning required, RPM is arguably harder compared to VQA:

- Unlike VQA where natural language questions usually imply what to pay attention to in the image, RPM relies merely on visual clues provided in the matrix and the *correspondence problem* itself, *i.e.*, finding the correct level of attributes to encode, is already a major factor distinguishing populations of different intelligence [CJS90].
- While VQA only requires spatial and semantic understanding, RPM needs joint spatial-temporal reasoning in the problem matrix and the answer set. The limit of *short-term memory*, the ability of *analogy*, and the discovery of the *structure* have to be taken into consideration.
- Structures in RPM make the compositions of rules much more complicated. Unlike VQA whose questions only encode relatively simple first-order reasoning, RPM usually includes more sophisticated logic, even with recursions. By composing different rules at various

levels, the reasoning progress can be extremely difficult.

We, therefore, generate the dataset with RPMs and refer to the generated dataset as the Relational and Analogical Visual rEasoNing dataset (RAVEN) in homage to John Raven for the pioneering work in the creation of the original RPM [Rav38]. In summary:

- RAVEN consists of 70K RPM problems, equally distributed in 7 distinct configurations.
- Each problem has 16 tree-structure annotations, totaling up to 1.12M structural labels.
- We design 5 rule-governing attributes and 2 noise attributes. Each rule-governing attribute goes over one of 4 rules, and objects in the same component share the same set of rules, making in total 440K rule annotations and an average of 6.29 rules per problem.

The RAVEN dataset is designed inherently to be light in visual recognition and heavy in reasoning. Each image only contains a limited set of simple gray-scale objects with clear-cut boundaries and no occlusion. Meanwhile, rules are applied row-wise, and there could be one rule for each attribute, attacking visual systems' major weaknesses in *short-term memory* and *compositional reasoning* [JHV17].

An obvious paradox is: in this innately compositional and structured RPM problem, no annotations of structures are available in previous works (*e.g.*, [BHS18, WS15]). Hence, we set out to establish a semantic link between visual reasoning and structure reasoning in RPM. We ground each problem instance to a sentence derived from an Attributed Stochastic Image Grammar (A-SIG) [Fu74, LWP09, PZ15, WXZ07, ZWZ16, ZM07] and decompose the data generation process into two stages: the first stage samples a sentence from a pre-defined A-SIG and the second stage renders an image based on the sentence. More importantly, the data generation pipeline naturally provides us with abundant dense annotations, especially the structure in the image space. This semantic link between vision and structure representation opens new possibilities by breaking down the problem into image understanding and tree- or graph-level reasoning [KW16, TSM15]. In the following sections, we discuss related work in visual reasoning and computational efforts in RPM, provide detailed descriptions of the RAVEN dataset generation process, design models that could leverage the important

structural information, and provide human as well as machine performance on RAVEN with analyese. The notable gap between human subjects (84%) and vision systems (59%) calls for further research into this problem.

2.1 Related Work

Visual Reasoning Early attempts were made in the 1940s-1970s in the field of logic-based Artificial Intelligence (AI). Newell argued that one of the potential solutions to AI was "to construct a single program that would take a standard intelligence test" [New73]. There are two important trials: (i) Evans presented an AI algorithm that solved a type of geometric analogy task in the Wechsler Adult Intelligence Scale (WAIS) test [Eva62, Eva64], and (ii) Simon and Kotovsky devised a program that solved Thurstone letter series completion problems [TT41]. However, these early attempts were heuristic-based with hand-crafted rules, making it difficult to apply to other problems.

The reasoning ability of modern vision systems was first systematically analyzed in the CLEVR dataset [JHV17]. By carefully controlling inductive bias and slicing the vision systems' reasoning ability into several axes, Johnson *et al*. successfully identified major drawbacks of existing models. A subsequent work [JHM17] on this dataset achieved good performance by introducing a program generator in a structured space and combining it with a program execution engine. A similar work that also leveraged language-guided structured reasoning was proposed in [HAR17]. Modules with special attention mechanisms were latter proposed in an end-to-end manner to solve this visual reasoning task [HM18, SRB17, ZZH17]. However, superior performance gain was observed in very recent works [CLL18, MTS18, YWG18] that fell back to structured representations by using primitives, dependency trees, or logic. These works also inspire us to incorporate structure information into solving the RPM problem.

More generally, Bisk *et al*. [BSC18] studied visual reasoning in a 3D block world. Perez *et al*. [PSD18] introduced a conditional layer for visual reasoning. Aditya *et al*. [AYB18]

proposed a probabilistic soft logic in an attention module to increase model interpretability. And Barrett *et al*. [BHS18] measured abstract reasoning in neural networks.

Computational Efforts in RPM The research community of cognitive science has tried to attack the problem of RPM with computational models earlier than the computer science community. However, an oversimplified assumption was usually made in the experiments that the computer programs had access to a symbolic representation of the image and the operations of rules [CJS90, LF17, LFU10, LTF09]. As reported in Section 2.3.3, we show that giving this critical information essentially turns it into a searching problem. Combining it with simple heuristics provides us an optimal solver, easily surpassing human performance. Another stream of AI research [LLG12, MG14, MKG14, MSD18, SG18b] tries to solve RPM by various measurements of image similarity. To promote fair comparison between computer programs and human subjects in a data-driven manner, Wang and Su [WS15] first proposed a systematic way of automatically generating RPM using first-order logic. Barrett *et al*. [BHS18] extended their work and introduced the Procedurally Generating Matrices (PGM) dataset by instantiating each rule with a relation-object-attribute tuple. Hoshen and Werman [HW17] first trained a CNN to complete the rows in a simplistic evaluation environment, while Barrett *et al*. [BHS18] used an advanced Wild Relational Network (WReN) and studied its generalization.

2.2 Creating RAVEN

Our work is built on the prior work aforementioned. We implement all relations in Advanced Raven's Progressive Matrices identified by Carpenter *et al*. [CJS90] and generate the answer set following *the monotonicity of RPM's constraints* proposed by Wang and Su [WS15].

Figure 2.2 shows the major components of the generation process. Specifically, we use the A-SIG as the representation of RPM; each RPM is a parse tree that instantiates from

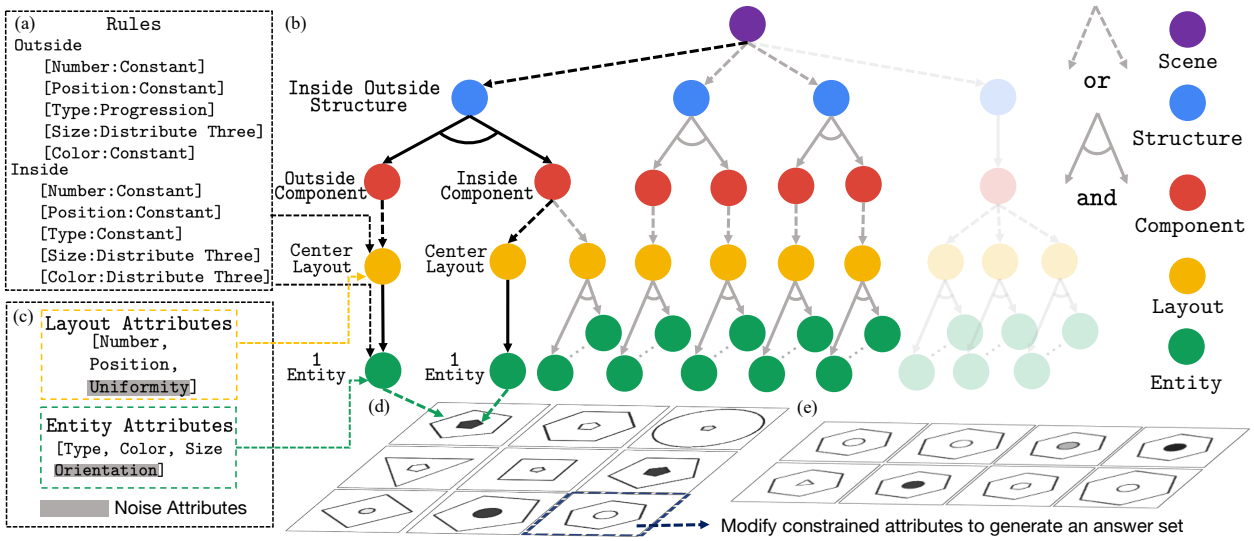


Figure 2.2: An illustration the A-SIG (b). Given a sampled set of rules (a), we prune the grammar tree and sample values of attributes (c) from (b) to generate a row of images.

the A-SIG. After rules are sampled, we prune the grammar to make sure the relations could be applied to any sentence sampled from it. We then sample a sentence from the pruned grammar, where rules are applied to produce a valid row. Repeating such a process three times yields a problem matrix. To generate the answer set, we modify attributes on the correct answer such that the relationships are broken. Finally, the structured presentation is fed into a rendering engine to generate images. We elaborate on the details in the following subsections.

2.2.1 Defining the Attributed Grammar

We adopt an A-SIG as the hierarchical and structured image grammar to represent the RPM problem. Such representation is advanced compared with prior work (*e.g.*, [BHS18, WS15]) which, at best, only maintains a flat representation of rules.

See Figure 2.2 for a graphical illustration of the grammar production rules. Specifically, the A-SIG for RPM has 5 levels—`Scene`, `Structure`, `Component`, `Layout`, and `Entity`. Note that each grammar level could have multiple instantiations, *i.e.*, different categories or

types. We list all production rules in Table 2.1. The `Scene` level could choose any available `Structure`, which consists of possibly multiple `Components`. Each `Component` branches into `Layouts` that links `Entities`.

Table 2.1: Grammar production rules used in RAVEN.

Level	Production Rules

`Scene`	`Scene` → `Singleton`
	`Scene` → `Left-Right`
	`Scene` → `Up-Down`
	`Scene` → `Out-In`

`Structure`	`Singleton` → `Grid`
	`Left-Right` → `Left` · `Right`
	`Up-Down` → `Up` · `Down`
	`Out-In` → `Out` · `In`

`Layout`	`Layout`* → `Entity` · `Layout`* `Layout`* → ∅

`Component`	`Grid` → `Center`
	`Grid` → 2×2`Grid`
	`Grid` → 3×3`Grid`
	`Left` → `Center`
	`Right` → `Center`
	`Up` → `Center`
	`Down` → `Center`
	`Out` → `Center`
	`In` → `Center`
	`In` → 2×2`Grid`

`Entity`	`Entity` → Entity

Another important construct in A-SIG is the attribute. We only have attributes in `Layout` and `Entity`, as summarized in Table 2.2. Note that all the symbols in the same level have the same set of attributes. In Table 2.2, `Uniformity` and `Orientation` are noise attributes and are not governed by rules. `Uniformity`, set false, will not constrain `Entities` in a `Layout` to look the same, while `Orientation` allows an `Entity` to self-rotate. The other attributes share their naming semantic where `Number` and `Position` indicates the number of entities and possible object slots in a given layout. Each `Entity` has its own `Type`, `Size`, `Color`, and `Orientation`.

Table 2.2: Attributes in different levels of the grammar.

Level	Attributes
`Layout`	`Number`, `Position`, `Uniformity`
`Entity`	`Type`, `Size`, `Color`, `Orientation`

This grammatical design of the image space allows the dataset to be very diverse and easily extendable. In this dataset, we manage to derive 7 configurations by combining different `Structures`, `Components`, and `Layouts`.

2.2.2 Applying Rules

Carpenter *et al.* [CJS90] summarized that in the advanced RPM, rules were applied row-wise and could be grouped into 5 types. Unlike Berrett *et al.* [BHS18], we strictly follow Carpenter *et al.*'s description of RPM and implement all the rules, except that we merge `Distribute Two` into `Distribute Three`, as the former is essentially the latter with a null value in one of the attributes.

Specifically, we implement 4 types of rules in RAVEN: `Constant`, `Progression`, `Arithmetic`, and `Distribute Three`. Different from [BHS18], we add internal parameters to certain rules (*e.g.*, `Progression` could have increments or decrements of 1 or 2), resulting in a total of 8 distinct rule instantiations. Rules do not operate on the 2 noise attributes. As shown in Figure 2.1 and 2.2, they are denoted as `[attribute:rule]` pairs. These 4 rules operate on 5 rule-governing attributes: `Constant`, `Progression`, `Arithmetic`, and `Distribute Three` where `Constant` indicates attributes governed by this rule would not change in the row. `Progression` indicates that attribute values monotonically increase or decrease in a row. `Arithmetic` uses mathematical summation or subtraction between numbers of objects in the first two panels to obtain the third panel. `Distribute Three` samples 3 values of an attribute in a problem instance and permutes the values in different rows. To make the image space even more structured, we require each attribute to go over one rule and all `Entities` in the same `Component` to share the same set of rules, while different `Components` could vary.

Given the tree representation and the rules, we first prune the grammar tree such that all sub-trees satisfy the constraints imposed by the relations. We then sample from the tree and apply the rules to compose a row. Iterating the process three times yields a problem matrix.

2.2.3 Generating the Answer Set

To generate the answer set, we first derive the correct representation of the solution and then leverage the monotonicity of RPM constraints proposed by Wang and Su [WS15]. To

break the correct relationships, we find an attribute that is constrained by a rule as described in Section 2.2.2 and vary it. By modifying only one attribute, we could greatly reduce the computation. Such a modification also increases the difficulty of the problem, as it requires attention to subtle difference to tell an incorrect candidate from the correct one.

2.3 Comparison and Analysis

In this section, we fill in two missing pieces in a desirable RPM dataset, *i.e.*, structure and hierarchy (Section 2.3.1), as well as the human performance (Section 2.3.2). We also show that RPM becomes trivial and could be solved instantly using a heuristics-based searching method (Section 2.3.3), given a symbolic representation of images and operations of rules.

2.3.1 Introduction of Structure

A distinctive feature of RAVEN is the introduction of the structural representation of the image space. Wang and Su [WS15] and Barrett *et al.* [BHS18] used plain logic and flat rule representations, respectively, resulting in no base of the structure to perform reasoning on. In contrast, we have in total 1.12M structure annotations in the form of parsed sentences in the dataset, pairing each problem instance with 16 sentences for both the matrix and the answer set. These representations derived from the A-SIG allow a new form of reasoning, *i.e.*, one that combines visual understanding and structure reasoning. As shown in [LF17, LFU10, LTF09] and our experiments in Section 2.5, incorporating structure into RPM problem solving could result in further performance improvement across different models.

2.3.2 Human Performance Analysis

Another missing point in the previous work [BHS18] is the evaluation of human performance. To fill in the missing piece, we recruit human subjects consisting of college students from a

subject pool maintained by the Department of Psychology to test their performance on a subset of representative samples in the dataset. In the experiments, human subjects were familiarized by solving problems with only one non-`Constant` rule in a fixed configuration. After the familiarization, subjects were asked to answer RPM problems with complex rule combinations, and their answers were recorded. Note that we deliberately included all figure configurations to measure generalization in the human performance and only "easily perceptible" examples were used in case certain subjects might have impaired perception. The results are reported in Table 2.3. The notable performance gap calls for further research into this problem. See Section 2.5 for detailed analysis and comparisons with vision models.

2.3.3 Heuristics-based Solver using Searching

We find that the RPM could be essentially turned into a search problem, given the symbolic representation of images and the access to rule operations as in [LF17, LFU10, LTF09]. Under such a setting, we could treat this problem as constraint satisfaction and develop a heuristics-based solver. The solver checks the number of satisfied constraints in each candidate answer and selects one with the highest score, resulting in perfect performance. Results are reported in Table 2.3. The optimality of the heuristic-based solver also verifies the well-formedness of RAVEN in the sense that there exists only one candidate that satisfies all constraints.

2.4 Dynamic Residual Tree for RPM

The image space of RPM is inherently structured and could be described using a symbolic language, as shown in [CJS90, LF17, LFU10, LTF09, Rav38]. To capture this characteristic and further improve the model performance on RPM, we propose a simple tree-structure neural module called Dynamic Residual Tree (DRT) that operates on the joint space of image understanding and structure reasoning.

In the DRT, given a sentence S sampled from the A-SIG, usually represented as a serialized n-ary tree, we could first recover the tree structure. Note that the tree is **dynamically** generated following the sentence S, and each node in the tree comes with a label. With a structured tree representation ready, we could now consider assigning a neural computation operator to each tree node, similar to Tree-LSTM [TSM15]. To further simplify computation, we replace the LSTM cell [HS97] with a ReLU-activated [NH10] fully-connected layer f. In this way, nodes with a single child (leaf nodes or OR-production nodes) update the input features by

$$I = \text{ReLU}(f([I, w_n])), \tag{2.1}$$

where $[\cdot, \cdot]$ is the concatenation operation, I denotes the input features, and w_n the distributed representations of the node's label [MSC13, PSM14]. Nodes with multiple children (AND-production nodes) update input features by

$$I = \text{ReLU}\left(f\left(\left[\sum_c I_c, w_n\right]\right)\right), \tag{2.2}$$

where I_c denotes the features from its child c.

In summary, features from the lower layers are fed into the leaf nodes of DRT, gradually updated by Equation 2.1 and Equation 2.2 from bottom-up following the tree structure, and output to higher-level layers. Inspired by [HZR16], we make DRT a **residual** module by adding the input and output of DRT together, hence the name Dynamic Residual Tree (DRT)

$$I = \text{DRT}(I, S) + I. \tag{2.3}$$

2.5 Experiments

2.5.1 Baselines

We adopt several representative models suitable for RPM and test their performances on RAVEN [BHS18, HZR16, KSH12, XCW15]. In summary, we test a simple sequential learning model (LSTM), a CNN backbone with an MLP head (CNN), a ResNet-based [HZR16] image classifier (ResNet), the recent relational WReN [BHS18], and all these models augmented with the proposed DRT.

LSTM The partially sequential nature of the RPM problem inspires us to borrow the power of sequential learning. Similar to ConvLSTM [XCW15], we feed each image feature extracted by a CNN into an LSTM network sequentially and pass the last hidden feature into a two-layer MLP to predict the final answer. In the DRT-augmented LSTM, *i.e.*, LSTM-DRT, we feed features of each image to a shared DRT before the final LSTM.

CNN We test a neural network model used in Hoshen and Werman [HW17]. In this model, a four-layer CNN for image feature extraction is connected to a two-layer MLP with a softmax layer to classify the answer. The CNN is interleaved with batch normalization [IS15] and ReLU non-linearity [NH10]. Random dropout [SHK14] is applied at the penultimate layer of MLP. In CNN-DRT, image features are passed to DRT before MLP.

ResNet Due to its surprising effectiveness in image feature extraction, we replace the feature extraction backbone in CNN with a ResNet [HZR16] in this model. We use a publicly available ResNet implementation, and the model is randomly initialized without pre-training. After testing several ResNet variants, we choose ResNet-18 for its good performance. The DRT extension and the training strategy are similar to those used in the CNN model.

Table 2.3: Mean and per-category testing accuracy of each model against human subjects under different figure configurations.

Method	Acc	Center	2x2Grid	3x3Grid	L-R	U-D	O-IC	O-IG
LSTM	13.07%	13.19%	14.13%	13.69%	12.84%	12.35%	12.15%	12.99%
WReN	14.69%	13.09%	28.62%	28.27%	7.49%	6.34%	8.38%	10.56%
CNN	36.97%	33.58%	30.30%	33.53%	39.43%	41.26%	43.20%	37.54%
ResNet	53.43%	52.82%	41.86%	44.29%	58.77%	60.16%	63.19%	53.12%
LSTM+DRT	13.96%	14.29%	15.08%	14.09%	13.79%	13.24%	13.99%	13.29%
WReN+DRT	15.02%	15.38%	23.26%	29.51%	6.99%	8.43%	8.93%	12.35%
CNN+DRT	39.42%	37.30%	30.06%	34.57%	45.49%	45.54%	45.93%	37.54%
ResNet+DRT	**59.56%**	**58.08%**	**46.53%**	**50.40%**	**65.82%**	**67.11%**	**69.09%**	**60.11%**
Human	84.41%	95.45%	81.82%	79.55%	86.36%	81.81%	86.36%	81.81%
Solver*	100%	100%	100%	100%	100%	100%	100%	100%

WReN We follow the original paper [BHS18] in implementing the WReN. In this model, we first extract image features by a CNN. Each answer feature is then composed with each context image feature to form a set of ordered pairs. The order pairs are further fed to an MLP and summed. Finally, a softmax layer takes features from each candidate answer and makes a prediction. In WReN-DRT, we apply DRT on the extracted image features before the relational module.

For all DRT extensions, nodes in the same level share parameters and the representations for nodes' labels are fixed after initialization from corresponding 300-dimension GloVe vectors [PSM14]. Sentences used for assembling DRT could be either retrieved or learned by an encoder-decoder. Here we report results using retrieval.

2.5.2 Experimental Setup

We split the RAVEN dataset into three parts, 6 folds for training, 2 folds for validation, and 2 folds for testing. We tune hyper-parameters on the validation set and report the model accuracy on the test set. For loss design, we treat the problem as a classification task and train all models with the cross-entropy loss. All the models are implemented in PyTorch [PGC17] and trained with ADAM [KB14] before early stopping or a maximum

number of epochs is reached.

2.5.3 Performance Analysis

Table 2.3 shows the testing accuracy of each model trained on RAVEN, against the human performance and the heuristics-based solver. Neither human subjects nor the solver experiences an intensive training session, and the solver has access to the rule operations and searches for the answer based on a symbolic representation of the problem. In contrast, all the computer vision models go over an extensive training session, but only on the training set.

In general, human subjects produce better testing accuracy on problems with simple figure configurations such as `Center`, while human performance reasonably deteriorates on problem instances with more objects such as `2x2Grid` and `3x3Grid`. Two interesting observations:

1. For figure configurations with multiple components, although each component in `Left-Right`, `Up-Down`, and `Out-InCenter` has only one object, making the reasoning similar to `Center` except that the two components are independent, human subjects become less accurate in selecting the correct answer.
2. Even if `Up-Down` could be regarded as a simple transpose of `Left-Right`, there exists some notable difference. Such effect is also implied by the "inversion effects" in cognition; for instance, inversion disrupts face perception, particularly sensitivity to spatial relations [CM09, LMM01].

In terms of model performance, a counter-intuitive result is: computer vision systems do not achieve the best accuracy across all other configurations in the seemingly easiest figure configuration for human subjects (`Center`). We further realize that the LSTM model and the WReN model perform only slightly better than random guess (12.5%). Such results contradicting to [BHS18] might be attributed to the diverse figure configurations in RAVEN. Unlike LSTM whose accuracy across different configurations is more or less uniform, WReN

achieves higher accuracy on configurations consisting of multiple randomly distributed objects (`2x2Grid` and `3x3Grid`), with drastically degrading performance in configurations consisting of independent image components. This suggests WReN is biased to grid-like configurations (majority of PGM) but not others that require compositional reasoning (as in RAVEN). In contrast, a simple CNN model with MLP doubles the performance of WReN on RAVEN, with a tripled performance if the backbone is ResNet-18.

We observe a consistent performance improvement across different models after incorporating DRT, suggesting the effectiveness of the structure information in this visual reasoning problem. While the performance boost is only marginal in LSTM and WReN, we notice a marked accuracy increase in the CNN- and ResNet-based models (6.63% and 16.58% relative increase respectively). However, the performance gap between artificial vision systems and humans is still significant (up to 37% in `2x2Grid`), calling for further research to bridge the gap.

2.5.4 Effects of Auxiliary Training

Barrett *et al.* [BHS18] mentioned that training WReN with a fine-tuned auxiliary task could further give the model a 10% performance improvement. We also test the influence of auxiliary training on RAVEN. First, we test the effects of an auxiliary task to classify the rules and attributes on WReN and our best-performing model ResNet+DRT. The setting is similar to [BHS18], where we perform an OR operation on a set of multi-hot vectors describing the rules and the attributes they apply to. The model is then tasked to both correctly find the answer and classify the rule set with its governing attributes. The final loss becomes

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{target}} + \beta \mathcal{L}_{\text{rule}}, \tag{2.4}$$

where $\mathcal{L}_{\text{target}}$ denotes the cross-entropy loss for the answer, $\mathcal{L}_{\text{rule}}$ the multi-label classification loss for the rule set, and β the balancing factor. We observe no performance change on

WReN but a serious performance downgrade on ResNet+DRT (from 59.56% to 20.71%).

Since RAVEN comes with structure annotations, we further ask whether adding a structure prediction loss could help the model improve performance. To this end, we cast the experiment in a similar setting where we design a multi-hot vector describing the structure of each problem instance and train the model to minimize

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{target}} + \alpha \mathcal{L}_{\text{struct}}, \tag{2.5}$$

where $\mathcal{L}_{\text{struct}}$ denotes the multi-label classification loss for the problem structure, and α is the balancing factor. In this experiment, we observe a slight performance decrease in ResNet+DRT (from 59.56% to 56.86%). A similar effect is noticed on WReN (from 14.69% to 12.58%).

2.5.5 Test on Generalization

One interesting question we would like to ask is how a model trained well on one figure configuration performs on another similar figure configuration. This could be a measure of the models' generalizability and compositional reasoning ability. Fortunately, RAVEN naturally provides us with a test bed. To do this, we first identify several related configuration regimes:

- Train on `Center` and test on `Left-Right`, `Up-Down`, and `Out-InCenter`. This setting directly challenges the compositional reasoning ability of the model as it requires the model to generalize the rules learned in a single-component configuration to configurations with multiple independent but similar components.
- Train on `Left-Right` and test on `Up-Down`, and vice-versa. Note that for `Left-Right` and `Up-Down`, one could be regarded as a transpose of another. Thus, the test could measure whether the model simply memorizes the pattern in one configuration.
- Train on `2x2Grid` and test on `3x3Grid`, and vice-versa. Both configurations involve multi-object interactions. Therefore the test could measure the generalization when the number

(a) Trained on `Left-Right`/`Up-Down`.

	`Left-Right`	`Up-Down`
`Left-Right`	41.07%	38.10%
`Up-Down`	39.48%	43.60%

(b) Trained on `2x2Grid`/`3x3Grid`.

	`2x2Grid`	`3x3Grid`
`2x2Grid`	40.93%	38.69%
`3x3Grid`	39.14%	43.72%

(c) Trained on `Center`.

`Center`	`Left-Right`	`Up-Down`	`Out-InCenter`
51.87%	40.03%	35.46%	38.84%

Table 2.4: Generalization test results. The columns indicates the transfer subset used.

of objects changes.

The following results are all reported using the best-performing model, *i.e.*, ResNet+DRT.

Table 2.4c, 2.4a and 2.4b show the result of our model generalization test. We observe:

- The model dedicated to a single figure configuration does not achieve better test accuracy than one trained on all configurations together. This effect justifies the importance of the diversity of RAVEN, showing that increasing the number of figure configurations could actually improve the model performance.
- Table 2.4c also implies that a certain level of compositional reasoning, though weak, exists in the model, as the three other configurations could be regarded as a multi-component composition of `Center`.
- In Table 2.4a, we observe no major differences in terms of test accuracy. This suggests that the model could successfully transfer the knowledge learned in a scenario to a very similar counterpart when one configuration is the transpose of another.
- From Table 2.4b, we notice that the model trained on `3x3Grid` could generalize to `2x2Grid` with only minor difference from the one dedicated to `2x2Grid`. This could be attributed to the fact that in the `3x3Grid` configuration, there could be instances with object distribution similar to that in `2x2Grid`, but not vice versa.

2.6 Conclusion

We present a dataset for Relational and Analogical Visual Reasoning in the context of Raven's Progressive Matrices (RPM), called RAVEN. Unlike previous work, we apply a systematic and structured tool, *i.e.*, Attributed Stochastic Image Grammar (A-SIG), to generate the dataset, such that every problem instance comes with rich annotations. One distinguishing feature of RAVEN is the introduction of the structure. We also recruit quality human subjects to benchmark human performance on the RAVEN dataset. These aspects fill two important missing points in previous works.

We further propose a neural module called Dynamic Residual Tree (DRT) that leverages the structure annotations for each problem. Extensive experiments show that models augmented with DRT enjoy consistent performance improvement, suggesting the effectiveness of using structure information in solving RPM. However, the difference between machine algorithms and humans clearly manifests itself in the notable performance gap, even in an unfair situation where machines experience an intensive training session while humans do not. We also realize that auxiliary tasks do not help performance on RAVEN. The generalization test shows the importance of the diversity of the dataset, and indicates current computer vision methods do exhibit a certain level of reasoning ability, though weak.

The entire work still leaves us with many mysteries. How could we the combine top-down and bottom-up methods into a model for solving RPMs? What is the correct way of formulating visual reasoning? Is it model-fitting? Is deep learning the ultimate way to visual reasoning? How could we improve the models? We hope these unresolved questions would call attention to this challenging reasoning problem.

CHAPTER 3

LEMMA: A Benchmark for Learning Multi-agent Multi-task Activities

In this chapter, we extend the event understanding problem to the real-world domain where agents collaboratively work together towards certain real-world goals. In contrast to RPMs, the difficulty of real-world event understanding resides in the innate complexity of both human activities and noisy perceptions. As a result, as described in Section 1.2, understanding and interpreting human actions has been a long-standing challenge for artificial intelligence. To study this problem, a few imperative components of daily human activities are largely missed in prior literature, including goal-directed actions, concurrent multi-tasks, and collaborations among multi-agents. Therefore, we introduce the LEMMA dataset to provide a single home to address these missing dimensions with meticulously designed settings, wherein the number of tasks and agents varies to highlight different learning objectives. In addition, we focus on the compositionality of human actions since the semantics of human actions are intrinsically ambiguous when described in natural language. For instance, although both "opening the fridge" and "opening a book" use the action verb "open," their semantics of the actions are utterly different. We take the stance of Grice's influential work on language act [Gri75]—technical tools for reasoning about rational action should elucidate linguistic phenomena [GF16]. Specifically, the compositional relations between the verbs and nouns could reveal the functionality of the object and the patterns of human-object interactions, which subsequently facilitate the understanding of the observed human activities and the language that describes them.

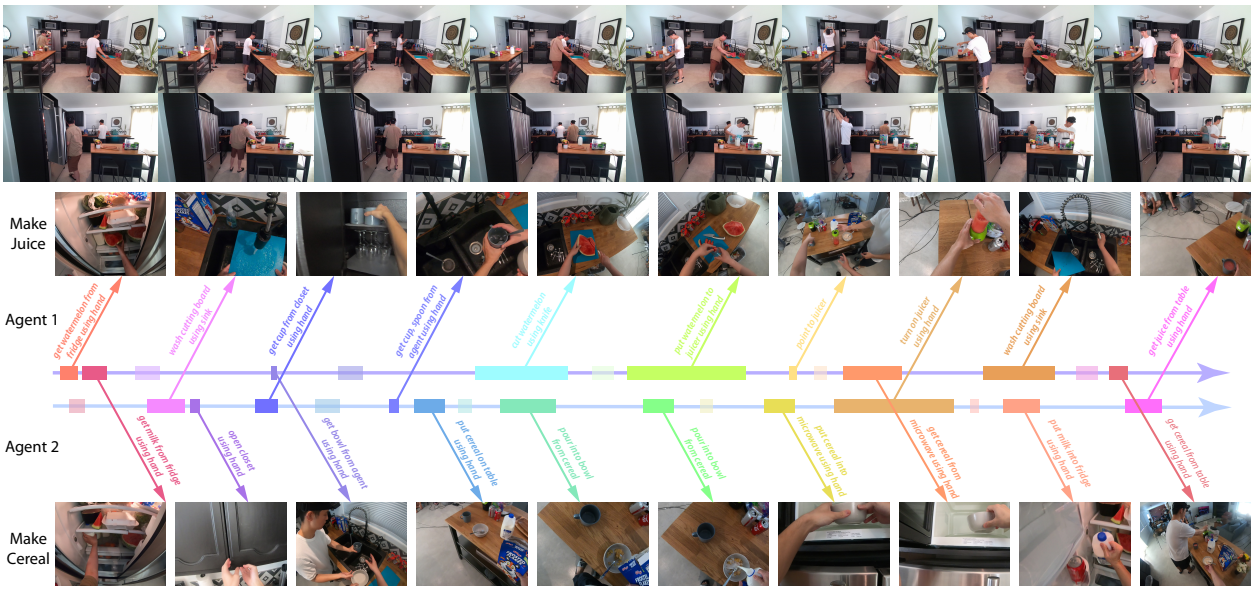


Figure 3.1: Illustrations of the proposed multi-view dataset with annotations.

In the following sections, we first discuss related efforts in video understanding. We then describe in details about the data collection process for the multi-view video dataset, LEMMA, that captures multi-agent, multi-task activities with goal-directed daily tasks; see Fig. 3.1 for an overview. Next, we list all annotations collected on LEMMA, focusing on the compositionality of actions and the governing task for each atomic-action. Finally, we provide compositional action recognition and action/task anticipation benchmarks by considering the aforementioned features, and provide comparisons and analyses of multiple baseline models to promote future research on human activity understanding.

3.1 Related Work

In this section, we review and compare prior indoor activity datasets on the basis of tasks and captured video contents; see a detailed summary in Table 3.1.

Crowd-sourced from online videos and movie-sharing platforms, typical large-scale video datasets [SZS12, KTS14, CEG15, CZ17, FKE18] focus on **video-level summarization and classification**. Although activity classes exhibit a large inter-class variability, spanning

from outdoor sports activities to indoor household activities, they generally lack sequential, goal-directed activities. Notably, they suffer from a major drawback [GR20]; activities are highly correlated to the general scene and object context, possessing a strong dataset bias for activity understanding.

Some datasets tackle the **human atomic-actions** using short clips or limited tasks, with a focus on the semantics of action verbs and objects [GKM17], 3D action analysis [LZL10, IPO13, SCH16], and action grounding with multi-modality inputs [MAZ19]. Although such datasets are suitable for atomic-actions, they are intrinsically impaired at studying the long-term reasoning of goal-directed human activities.

Recently, **concurrent actions** have been taken into consideration. For instance, Charades [SVW16] is a large-scale benchmark for household activities, and Charades-Ego [SGS18] steps further with both FPVs and TPVs. However, the activities involved are mostly unrelated to specific goals due to the crowdsourced script generation process. Similarly, although Multi-THUMOS [YRJ18] and AVA [GSR18] focus on highly paralleled activities, and some datasets look at the temporal order of activities [BLB14, TZS16], the unnaturally scripted activities result in the lack of meaningful goal-directed tasks exhibited in our daily life.

Conversely, **instructional video** datasets [ABA16, SM13, KAS14, KGS13, RRR16] tackle goal-directed multi-step tasks, mostly in cooking, repairing, and assembling activities. In spite of their relevance, they fail to account for multi-agent or multi-task problems.

Table 3.1: Comparisons between LEMMA and relevant indoor activity datasets.

Dataset	Task Annotation	Multi-agent	Multi-task	Multi-view	Samples	Frames	Action Classes	Action Segments	Actions per Video	Modality	Year
MPII Cooking [RAA12]	✓	✗	✗	✗	273	2.9M	88	14,105	51.7	RGB	2012
ADL [PR12]	✗	✗	✓	✗	20	1.0M	32	436	13.6	RGB	2012
50Salads [SM13]	✓	✗	✗	✗	50	0.5M	17	966	19.3	RGB-D	2013
CAD-120 [KGS13]	✗	✗	✗	✗	120	0.1M	10	1,175	9.8	RGB-D	2013
Breakfast [KAS14]	✓	✗	✗	✓	433	3.0M	50	3,078	7.1	RGB	2014
Watch-n-Patch [WZS15]	✓	✗	✗	✗	458	0.1M	21	2978	6.5	RGB-D	2015
Charades [SVW16]	✗	✗	✓	✗	9,848	7.4M	157	67,000	6.8	RGB	2016
Something-Something [GKM17]	✗	✗	✗	✗	108,499	-	174	108,499	1.0	RGB	2017
EGTEA GAZE+ [LLR18]	✓	✗	✗	✗	86	2.4M	106	10,325	120.1	RGB	2018
EPIC-KITCHENS [DDM18]	✗	✗	✓	✗	432	11.5M	149	39,596	91.7	RGB	2018
LEMMA (proposed)	✓	✓	✓	✓	324	4.6M	641	11,781	36.4	RGB-D	2020

EPIC-KITCHENS [DDM18] is perhaps the only exception; it records naturally paralleled task execution of agents in kitchen environments, but with no task specification or multi-agent interactions. Additionally, prior instructional video datasets have either drastic view perspective changes [ZXC18, ABA16, TDR19, TCH17] or limited egocentric view with severe occlusions [PR12, LLR18], hindering the activity understanding.

Another related stream of work is the learning of group-level activities in a **multi-agent** setting [IMD16], such as detecting key actors [RHA16], predicting future trajectories [PES09, LCL07], and recognizing collective activities [CSS09, OHP11, SXR15]. However, such coarse-grained multi-agent interactions leave the latent subtlety of collaboration and task assignment untouched. Although simulation-based multi-agent environments [BKM20, VBC19, BBC19] can partially address such an issue, learning from noisy and real visual input in physical work is still essential for understanding collaborative planning behaviors of agents in the context of complex daily tasks.

The collected LEMMA dataset strives to address the shortcomings of the aforementioned works, capturing goal-directed, decompositional, multi-task activities with multi-agent collaborations. As shown in Table 3.1, the size, annotation, and actions per video of LEMMA are at a comparable scale to state-of-the-art benchmarks.

3.2 The LEMMA Dataset

This section describes the design, data collection, and data annotation process of the LEMMA dataset. The dataset is profiled by various statistics from diversified perspectives to highlight its potential in activity understanding.

3.2.1 Activities and Scenarios

We first build a task pool of 15 common tasks in the kitchen (*e.g.*, "make juice," "make cereal") and the living room (*e.g.* "watch TV," "water plant"). On top of these tasks, we design four

types of scenarios (with a different focus) to study goal-directed multi-step multi-task indoor activities in multi-agent settings.

- **Single-agent Single-task (1×1):** Each participant was first asked to perform all tasks from the task pool independently; this ensured participants are clear with the goal of each task and could schedule and assign tasks efficiently in later multi-task or multi-agent scenarios. Participants were asked to read the instructions and walk around to get familiarized with the new environments.
- **Single-agent Multi-task (1×2):** Each participant was then asked to simultaneously perform two tasks, randomly sampled from the task pool. The participants determined the order of task executions without any restrictions.
- **Multi-agent Single-task (2×1):** Two participants were asked to perform a single task cooperatively; the task is randomly selected from the task pool. To emulate human-robot teaming accurately, only one participant (leader) was provided with task instructions; the other participant (helper), with no knowledge of the task, was asked to collaborate with the leader agent to finish the task efficiently. Only nonverbal communication (*e.g.*, gestures) were allowed between two participants; this design would open up new venues for nonverbal communication and the emergence of language in real-world environments.
- **Multi-agent Multi-task (2×2):** Both participants were provided with task instructions. Since both participants were asked to accomplish two complex multi-step tasks collaboratively, this scenario has the most natural activity/task patterns and richest mechanisms for learning task scheduling and assignment.

In total, the LEMMA dataset includes 37 unique task combinations in the multi-task scenarios. Participants were explicitly instructed to perform tasks efficiently and provided with a brief task instruction with basic environment information. Except for the specification of the goal states for each task, we add no additional constraint to the order of task execution; participants perform tasks naturally and freely. Fig. 3.2 shows a sample instruction for the 2×1 scenario.

In this task, you are asked to **make watermelon juice**. Here are things to know before your start: **Leader**
- All the items needed for this task can be found either in the **fridge**, on the **table**, or in one of the **drawers** or **closets**.
- Please **cut** the **watermelon** into pieces before blending it with the **juicer**.
- Please keep the kitchen clean; wash all the **tools/objects** you used.
- You will have an additional **helper** to collaborate with you.
  - Do **Not** speak with them. They do **NOT** know anything about the task you are working on.
  - Feel free to ask them for help, but only using **non-verbal** communication (e.g., gestures). For instance, you may point to something, or any other gestues you think may help instruct them.

In this task, you are asked to **collaborate** with your friend to finish a task in the kitchen. **Helper**
Here are things to know before your start:
- All the items needed for this task can be found either in the **fridge**, on the **table**, or in one of the **drawers** or **closets**.
- Please keep the kitchen clean; wash all the **tools/objects** you used.
- As only your friend knows the task instruction, please try to infer what the task is and offer helps.
- **You may not speak with your friend**. You can only use **non-verbal** communication (e.g., gestures).

Figure 3.2: An example instruction of making juice in a multi-agent single-task (2×1) scenario.

3.2.2 Data Collection

We recorded the data in 7 different Airbnb houses, performed by 8 individuals in 14 unique kitchens/living rooms. To provide different views of performing daily activities and avoid occlusion in narrow spaces, we set up two Kinect Azure cameras to capture RGB-D videos of the global scene and human bodies. In addition, each participant was instructed to wear a head-mounted GoPro camera to capture detailed agent-specific actions in an egocentric view.

In post-processing, we synchronize the camera recordings of all views at a frame rate of 24 FPS. Fig. 3.2 shows an example of a scene with a point cloud merged from two Kinects and four RGB views from both Kinects and GoPros. Combining TPVs and FPVs captures most of the details of performing daily activities, provides sufficient data for understanding human activities and benefits future research in embodied vision. The additional depth information and 3D human skeletons captured by Kinects can also be adopted for future 3D

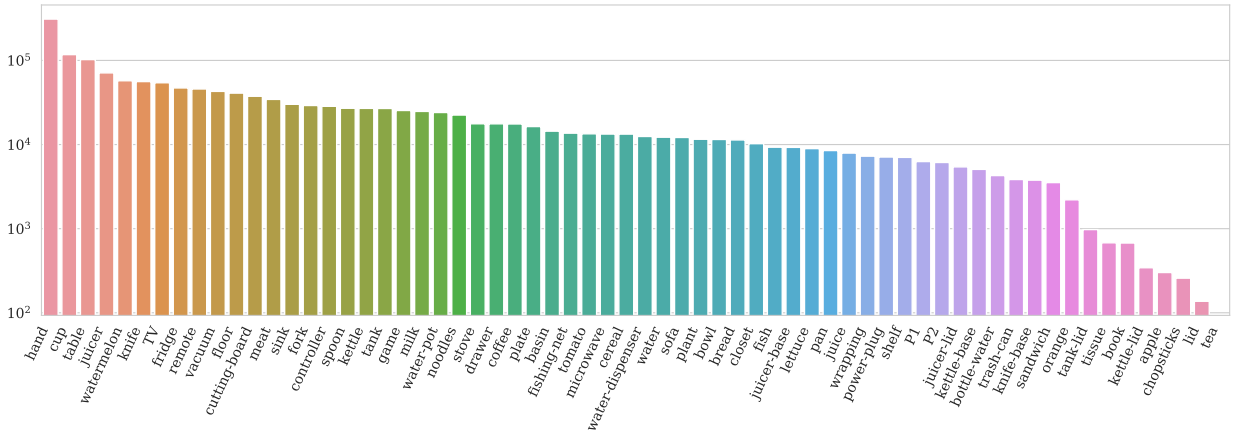
understanding tasks.

3.2.3 Ground-truth Annotation

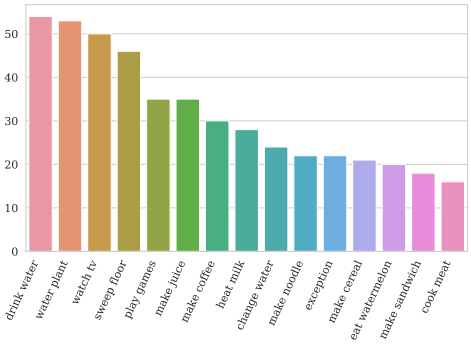
We used the Amazon Mechanical Turk (AMT) to annotate both human bounding boxes and action information in the recordings. Specifically, action information includes the temporal localization of segments, semantic labels, and the governing task of each atomic-action. The semantic labels of atomic-actions are composed of verbs and nouns, representing flexible compositional relations to describe human actions. Additional details are provided below.

Bounding Boxes and Segments: Bounding boxes of humans are annotated on the primary view of TPVs. Skeletons captured by Kinects are used to provide initial estimations of bounding boxes. Next, we use Vatic [VPR13] to adjust bounding boxes and annotate the segments of atomic-actions. The segments of atomic-actions are defined by verbs without corresponding nouns, for example, "put __ to __ using __," "pour into __ from __." Each video was first annotated by two AMT workers; task-irrelevant actions (*e.g.*, "walking," "holding") are ignored. We then compute the Intersection over Union (IoU) of both bounding boxes and temporal segments. A third AMT worker is asked to fine-tune the annotations if the IoU of bounding boxes or segments annotated is lower than 0.5.

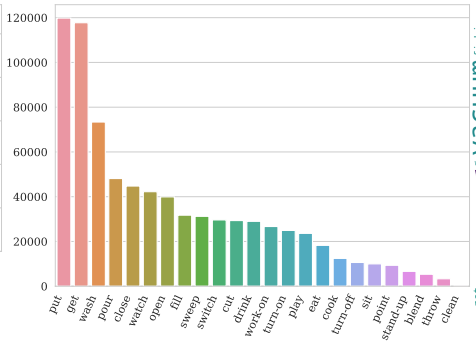
Atomic-actions and Activities: Given the verbs of the atomic-action segments, two AMT workers were asked to fill in the blanks of the verb patterns and annotate the governing tasks in multi-task scenarios with a self-developed interactive annotation tool. We allow concurrent actions for each agent with multiple nouns for the same verb; for example, "get spoon, cup from table using hand." As there might exist ambiguities in describing the atomic-actions with natural languages, such as the possible annotations of "wash cup using water" *vs.* "wash cup using sink," we manually go through all the annotations and resolve the ambiguous action annotations following a uniform criterion.



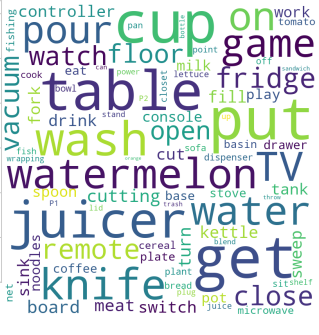
(a) Frequency of annotated noun classes across all frames



(b) Frequency of recorded tasks



(c) Frequency of annotated verb



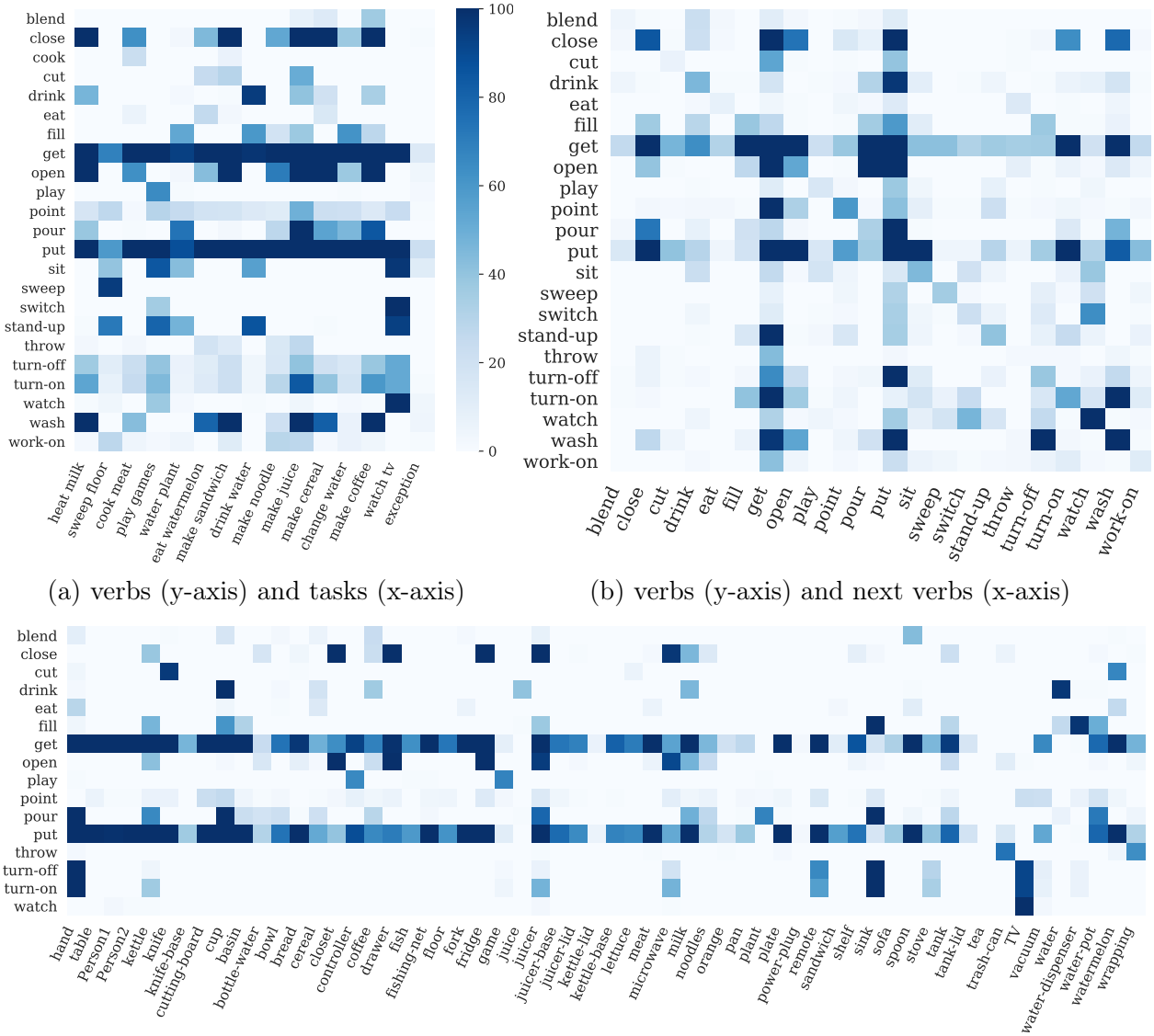
(d) Action wordle

Figure 3.3: Statistics of the LEMMA dataset.

3.2.4 Dataset Statistics

In total, we recorded 324 activities, generating 324 × 2 TPV videos (from both Kinects) and 445 FPV videos. Among them, 136 activities were performed in kitchens, and the remaining 188 in the living rooms. The collected LEMMA dataset consists of 127 1 × 1 activities, 76 1 × 2 activities, 66 2 × 1 activities, and 55 2 × 2 activities. The frequency of the recorded tasks is shown in Fig. 3.3b. The total duration of all the activities is 10.1 hours, with an average duration of 2 minutes per video and the longest activity of 7 minutes.

We retrieved a total of 4.6 million images during post-processing, including 2.9 million RGB images captured by both GoPros and Kinects and 1.7 million depth images captured by Kinects. We annotated 0.9 million RGB frames captured by the primary view Kinect and



(a) verbs (y-axis) and tasks (x-axis)

(b) verbs (y-axis) and next verbs (x-axis)

(c) verbs (y-axis) and nouns (x-axis)

Figure 3.4: The co-occurrence statistics for verbs, nouns, and tasks in LEMMA.

gathered 0.8 million annotated frames with one or more actions performed by each of the agents (if multiple).

After resolving annotation ambiguities, we collected 24 verb classes and 64 noun classes, resulting in 862 compositional atomic-action labels, of which 641 appear more than 50 times. We show the frequencies of annotated verbs and nouns in Figs. 3.3a and 3.3c; both distributions roughly follow the Zipf's law.

Co-occurrence relations among annotated verbs, nouns, and tasks are shown in Fig. 3.4. As we can see from Figs. 3.4a and 3.4c, verbs like "get" and "put" co-occur with various nouns in almost all of the tasks, which aligns with our intuition that moving objects around consists a large portion of our daily activities. Interactive actions between participants are captured by verbs (*e.g.*, "point-to") and nouns (*e.g.*, "P1," short for "participant 1") in the form of annotations like "get knife from P1 using hand" or "point-to sink."

3.3 Benchmarks

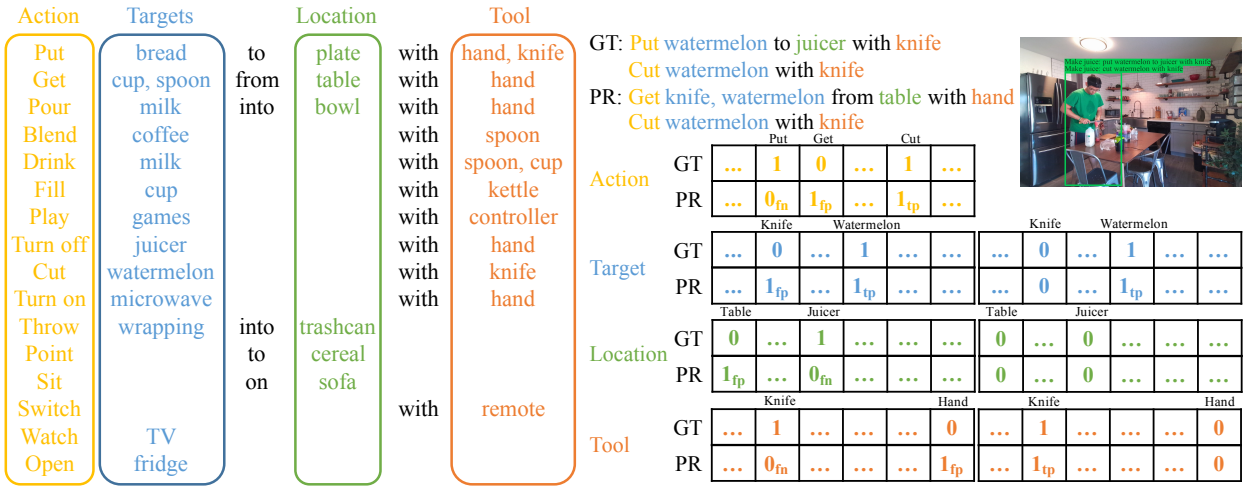
Aligned with our motivations, two tasks are constructed to evaluate indoor human activity understanding on the collected LEMMA dataset: (i) recognizing atomic-actions and their semantics; and (ii) predicting possible future steps for goal-directed activities, especially in multi-agent scenarios. Specifically, we define two challenging benchmarks to test the capability of understanding complex goal-directed activities for computer vision algorithms.

3.3.1 Compositional Action Recognition

Human indoor activities are composed of fine-grained action segments with rich semantics. As mentioned by Goyal *et al.* [GKM17], interactions with objects are highly purposive. From the simplest verb of "put," we can generate a plethora of combinations of objects and target places, such as "put cup onto table," "put fork into drawer." Situations could become even more challenging when objects were used as tools, *e.g.*, "put meat into pan using fork."

Motivated by the above observation, we propose the compositional action recognition benchmark on the collected LEMMA dataset with each object attributed to a specific semantic position in the action label. Specifically, we build 24 compositional action templates. In these action templates, each noun could denote an interacting object, a target or a source location, or a tool used by a human agent to perform certain actions.

The proposed compositional action recognition benchmark is challenging; it requires



(a) Compositional action templates (b) Prediction of verbs and nouns

Figure 3.5: Compositional action labels in LEMMA.

computational models to correctly detect the ongoing concurrent action verbs as well as the nouns at their correct semantic positions. We evaluate model performances by metrics on compositional action recognition in both FPVs and TPVs. Specifically, the model is asked to predict (i) multiple labels in verb recognition for concurrent actions (*e.g.*, "watch tv" and "drink with cup" at the same time), and (ii) multiple labels in noun recognition for each semantic position given verbs, representing the interactions with multiple objects using the same action (*e.g.*, "wash spoon, cup using sink"). Fig. 3.5b shows the schematics of the evaluation process. For training and testing on TPVs, we provide ground-truth bounding boxes of humans as additional information on spatial localization.

3.3.2 Action and Task Anticipation

As emphasized throughout this chapter, the most significant factor of human activities is the goal-directed, teleological stand. An in-depth understanding of goal-directed tasks demands a predictive ability of latent goals, action preferences, and potential outcomes. To tackle these challenges, we propose the action and task anticipation benchmark on the collected LEMMA dataset. Specifically, we evaluate model performances for the anticipation (*i.e.*, predictions

for the next action segment) of action and task with both FPV and TPV videos.

This benchmark provides both the training and testing data in all four scenarios of activities to study the goal-directed multi-task multi-agent problem. As there is an innate discrepancy of prediction difficulties among these four scenarios, we gradually increase the overall prediction difficulty, akin to a curriculum learning process, by setting the percentage of training videos to be 3/4, 1/4, 1/4, and 1/4 for 1×1, 1×2, 2×1 and 2×2 scenarios, respectively. Intuitively, with sufficient clean demonstrations of tasks in 1×1 scenario, interpreting tasks in more complex settings (*i.e.*, 1×2, 2×1, and 2×2) should be easier, thus requiring less learning samples; such a design encourages the model to generalize. The model performance is evaluated individually for each scenario.

3.4 Experiments

In this section, we conduct experiments on the two proposed benchmarks with details on evaluation metrics, experimental settings, and baseline results. We further discuss the results to highlight the underlying challenges of each task.

3.4.1 Compositional Action Recognition

Experimental Setup: We randomly split all the video samples into training and test sets with a ratio of 3:1, resulting in 243 recorded activities for training and the remaining 81 for testing. Due to the multi-agent setup, each activity may have multiple FPVs; 333 (out of 445) FPV videos are split into training. In TPVs, the recordings of the primary view with the ground-truth human bounding box annotations are given for both training and testing videos. Results are evaluated on two separate sources of inputs: FPVs and TPVs.

Evaluation Metrics: Model performances are evaluated separately for verbs, nouns, and compositional action recognition. Verb and compositional action recognition are treated as

multi-label classifications with 25 verb classes and 863 compositional action classes (including a "null" action). After generating multi-hot labels for each semantic position in the presented verb, noun recognition is evaluated as multi-label classification (64 object classes). Average precision, recall, and F1-score for all predictions are reported on testing sets. During the evaluation, we sample image frames at 5 FPS and evaluate models on these frames.

Methods: We adopt two recent 3D-CNN networks, I3D [CZ17] and SlowFast Network [FFM19], as the baseline models. The baseline models predict the compositional action directly. Considering the compositionality of verbs and nouns, we propose two variants of the baseline models: (i) a multi-branch network (branching model) that builds on the bottleneck layer of the backbone models to leverage both verb and noun supervision, and (ii) a multi-step inference model (sequential model), wherein verbs are first inferred with a beam search and then fed into object inference with their verb embeddings for joint learning.

Implementation Details: The training procedure utilizes all annotated segments in the training set. Additionally, we re-scale all the images with the short side to 256 pixels. To feed data into 3D-CNN models, 4 frames are first sampled for each action segment as center frames, and an additional 8 frames are then uniformly sampled around center frames with a window length of 32. We train each model on 8 Titan RTX GPUs on a single computing node for 50 epochs (20k iterations) with a batch size of 96. We use the warm-up strategy and perform large mini-batch batch normalization, as suggested in [GDG17]. The learning rate is initially set to 0.0125 for each parallel branch and decays with a cosine annealing. Other settings of the backbone models are the same as in [FFM19]. For the proposed sequential model, we use the beam search with a size of 5 for action inference. We extract bounding box features of humans with ROIAlign [HGD17] for frames in TPVs.

For the implementations of the two proposed models, "branching" and "sequential", we build both models on top of the backbone 3D CNN model and use a multi-branch network to

train verbs, nouns, and their correspondences. We start from the "sequential" model as the "branching" model is a variant of the "sequential" model; see an illustration in Fig. 3.6.

For the verb branch, we propose 3 verb candidates for each segment and extract verb visual features for verb recognition. Specifically, the verb visual features $f_{\text{verb}} = \{F^{(i)}_{\text{verb}}(f_{\text{vis}})\}$ are generated using three different linear projections $\{F^{(i)}_{\text{verb}}\}_{i=1,2,3}$ applied onto the feature f_{vis} extracted by 3D CNN. We sort ground-truth action labels according to their index in the verb vocabulary and use cross-entropy loss $\mathcal{L}_{\text{verb}}$ as the supervision for verb recognition.

For the noun branch, we utilize the embeddings of each verb provided by GloVe [PSM14] as additional features. The embedding of each verb is passed into a linear projection layer and concatenated with the extracted visual features to generate noun feature vectors $f_{\text{noun-vis}}$. Next, we use three different linear projections $\{F^{(i)}_{\text{noun}}\}_{i=1,2,3}$ to generate features for each of the noun visual feature vectors and obtain noun semantic features $f_{\text{noun-sem}} =$

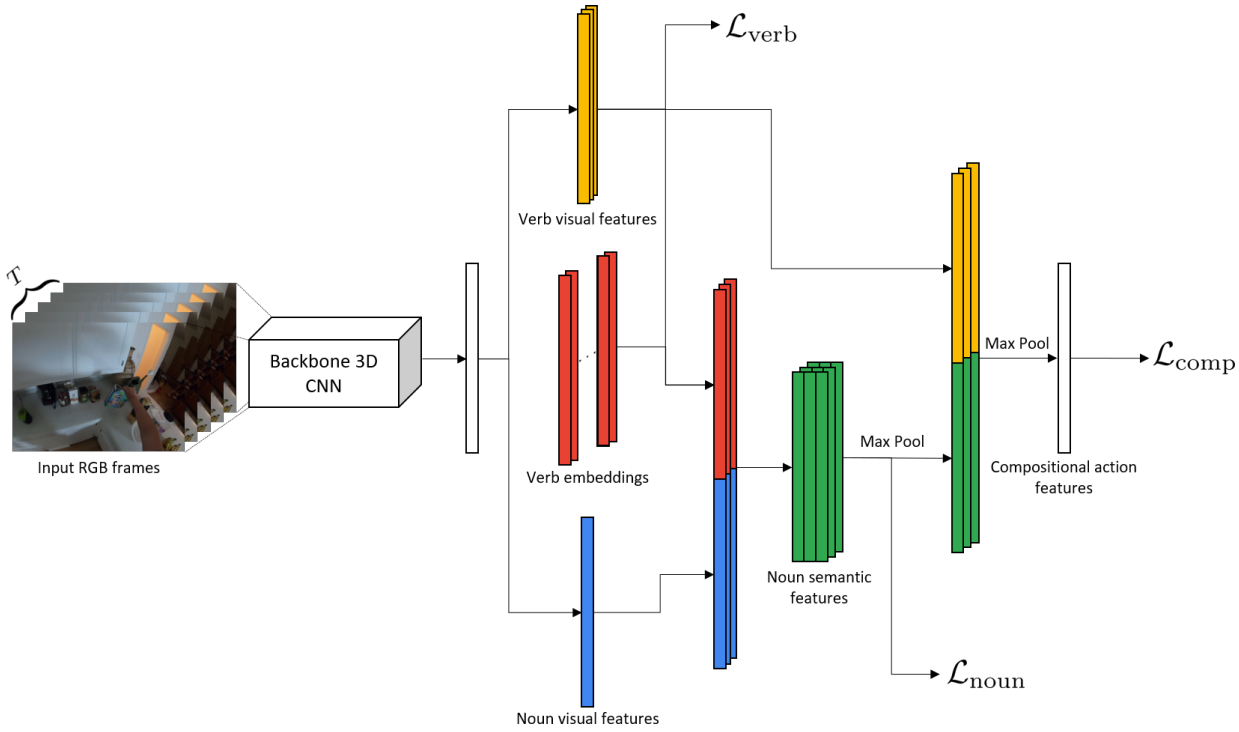


Figure 3.6: An illustration of the proposed "sequential" model, which predicts verbs, nouns, and compositional actions jointly.

$\{[F^{(i)\text{noun}}(f^{(j)}_{\text{noun-vis}})]_{i=1,2,3}\}_{j=1,2,3}$. As we generate ground-truth labels following the same scheme, we use binary cross-entropy loss $\mathcal{L}_{\text{noun}}$ as the supervision for recognizing nouns at their correct semantic positions using $f_{\text{noun-sem}}$. During training, the embeddings of the ground-truth verbs are fed into the network. During testing, we use the embedding of the predicted top-3 verbs.

We use max-pooling to summarize the noun semantic features and concatenate it with verb visual features. We use another layer of max pooling to generate the final compositional action feature and use binary cross-entropy loss as $\mathcal{L}_{\text{comp}}$ to provide supervision for compositional action recognition. The joint loss is

$$\mathcal{L} = \mathcal{L}_{\text{verb}} + \mathcal{L}_{\text{noun}} + \mathcal{L}_{\text{comp}}.$$

For the "branching" model, we follow the same basic scheme of the "sequential" model but remove the connection between the verb branch and the noun branch by discarding the additional verb embeddings. The remaining details of the architecture, as well as the optimizing objectives, remain the same.

Results and Discussion: Table 3.2 shows quantitative results of predicting verbs, nouns, and compositional actions for the compositional action recognition task. For FPVs, rather than directly predicting the compositional actions (baseline models), predicting the verbs and nouns with their semantic positions boosts the performance on all metrics, indicating that understanding the compositional structures of human actions indeed supports the prediction. We also observe that the results of compositional action recognition in the sequential models are slightly lower than the branching model due to the aggregated error brought in by a relatively low precision ($\sim$25%) of the verb recognition.

In comparison, the results of compositional action recognition in TPVs are significantly lower than those in the FPVs due to severe occlusion. It also shows that predicting the

Table 3.2: Comparisons of compositional action recognition on LEMMA.

View Type	Method	Verb			Noun			Compositional Action		
		Avg.Prec	Avg.Rec	Avg.F1	Avg.Prec	Avg.Rec	Avg.F1	Avg.Prec	Avg.Rec	Avg.F1
FPV	I3D	17.09	43.89	24.60	3.42	16.15	5.72	11.07	39.49	17.30
	Slowfast	22.27	56.42	31.94	4.31	20.60	7.13	18.68	**50.65**	27.3
	I3D sequential	25.04	**57.00**	34.80	**19.36**	**75.29**	**30.80**	18.00	50.04	26.47
	Slowfast sequential	24.30	49.71	32.64	17.95	59.11	27.54	26.80	38.41	31.57
	I3D branching	25.73	55.62	**35.8**	18.63	69.76	29.41	22.29	48.46	30.53
	Slowfast branching	**26.16**	56.33	35.73	18.18	73.46	29.15	**27.97**	48.87	**35.58**
TPV	I3D	14.18	36.34	20.40	2.29	11.05	3.79	6.85	**23.82**	10.64
	Slowfast	14.28	**37.38**	20.66	2.32	11.14	3.83	**7.76**	23.25	**16.31**
	I3D sequential	16.17	30.17	21.05	7.79	**25.41**	11.93	2.23	12.67	3.79
	Slowfast sequential	15.31	28.84	20.00	6.37	22.39	9.92	3.27	9.16	4.82
	I3D branching	12.92	32.09	18.43	12.75	17.70	14.82	4.67	20.76	7.6
	Slowfast branching	**16.64**	33.40	**22.21**	**17.29**	18.36	**17.81**	6.52	21.55	10.01

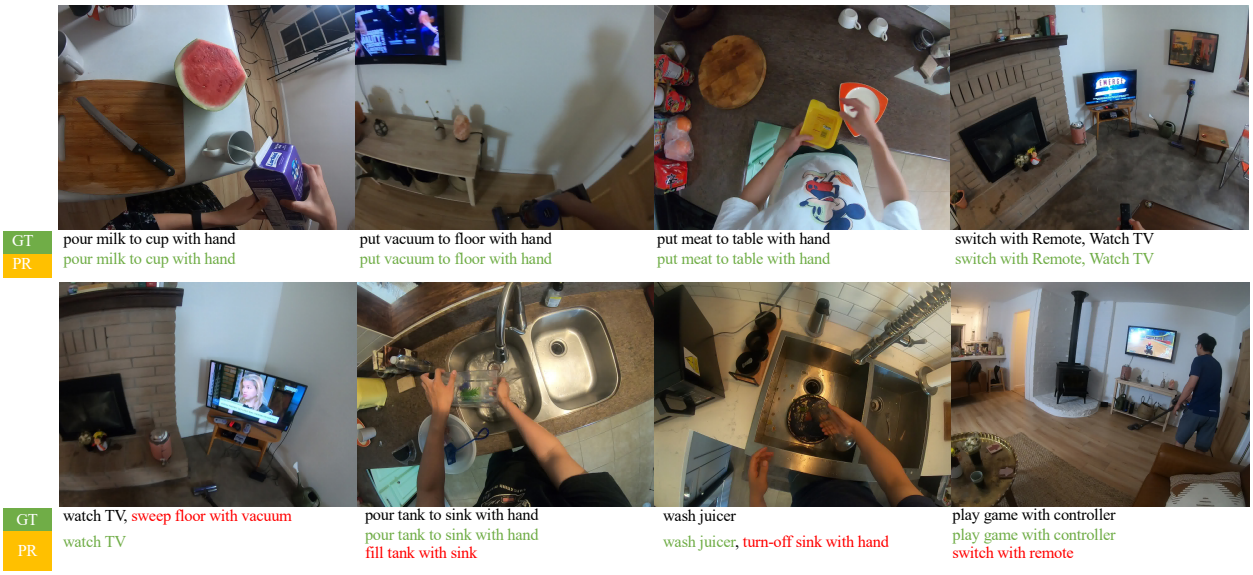


Figure 3.7: Qualitative results of compositional action recognition on LEMMA, we show correct predictions (in green) and failure examples (in red).

composition of verbs and nouns makes no significant improvement compared with predicting compositional action directly. Such a result implies that current models could not capture the details of compositions between verbs and nouns from TPVs. Taken together, the results indicate that fusion among the representations of visual embodiment between TPVs and FPVs might be a crucial ingredient to tackle this problem in the future.

Fig. 3.7 shows qualitative results for the compositional action recognition task.

3.4.2 Action and Task Anticipations

Experimental Setup: We split the training and test sets with ratios $3:1$, $1:3$, $1:3$, $1:3$ for the four scenarios 1×1, 1×2, 2×1, 2×2, respectively. Such a spit results in a training set with (96, 19, 16, 13) activities and a test set with (31, 57, 50, 42) activities in four scenarios. During training and testing, the computational models have access to both FPVs and TPVs, together with the ground-truth human bounding boxes annotations of the TPV primary view.

Evaluation Metrics: Model performances are evaluated individually (per agent) for the action and task anticipations task. Specifically, both action and task anticipations are evaluated as multi-label classifications with 863 compositional action classes (including a "null" action) and 15 task classes. Average precision, recall, and F1-score are reported individually for each of the four scenarios on the testing sets. Similar to the protocol used in the above compositional action recognition task, we re-sample image frames at 5 FPS and evaluate these sub-sampled frames during the testing phase.

Methods: We leverage the visual features extracted by the pre-trained SlowFast model in compositional action recognition for baseline models. Specifically, we compare two backbone models: (i) using segment-level recognition feature (SF) directly by adding an MLP on top of the features, and (ii) using long-term feature bank (LFB) with max pooling [WFF19]. For activities with multi-agent interactions, we use the other agent's FPV features together with their own's to capture the joint task execution progress for learning and inference; these variants are denoted as M-SF (FPV) and M-LFB (FPV) For comparison, we also use the concatenation of the FPV feature and primary TPV feature as the input; the corresponding models are denoted as M-SF (TPV) and M-LFB (TPV).

Implementation Details: For the LFB model, we use a history window size of 10 and aggregate the features using max-pooling, as described in [WFF19]. For the multi-agent variants, we use max-pooling to fuse features of two views and process them with a different branch as another temporal inference module. We train models on a single Titan Xp GPU for 50 epochs with a learning rate of 0.001.

For scenarios where two agents collaborate, we incorporate the egocentric features of another agent (denoted as Ego in Table 3) or TPV features (denoted as TPV in Table 3) through a pooling mechanism, similar to [GJF18]. We use these pooled features to incorporate global task execution information to each agent. Specifically, we concatenate the extracted global features to features extracted by the backbone 3D CNN models from the target agent's egocentric view for training and inference. For TPV, we use ROIAlign to extract visual features corresponding to each agent's bounding box. An illustration of the pipeline with TPV features as additional features is shown in Fig. 3.8.

Results and Discussion: Table 3.3 shows quantitative results of action and task anticipation. The proposed multi-agent variants (M-) of baseline models perform the best among all models. For single-agent activities (1×1, 1×2), we have the following observations. First, models that consider temporal relations between frames generally perform better than the models using segment features. Second, adding additional TPV features to single-agent activities slightly helps interpret the task being executed and therefore promotes anticipation. This result matches the intuition that models having access to both FPVs and TPVs would perceive more holistic scene information. We also find that the performances of task anticipation in the 1×1 single-task scenario are better than the one in the 1×2 multi-task scenario, matching what we would expect from more complicated task execution patterns.

For multi-agent activities (2×1, 2×2), we observe that the aggregation of FPV and TPV features generally performs better. It supports our hypothesis that observing the other agents' actions helps models to "understand" task scheduling and assignment. We also observe

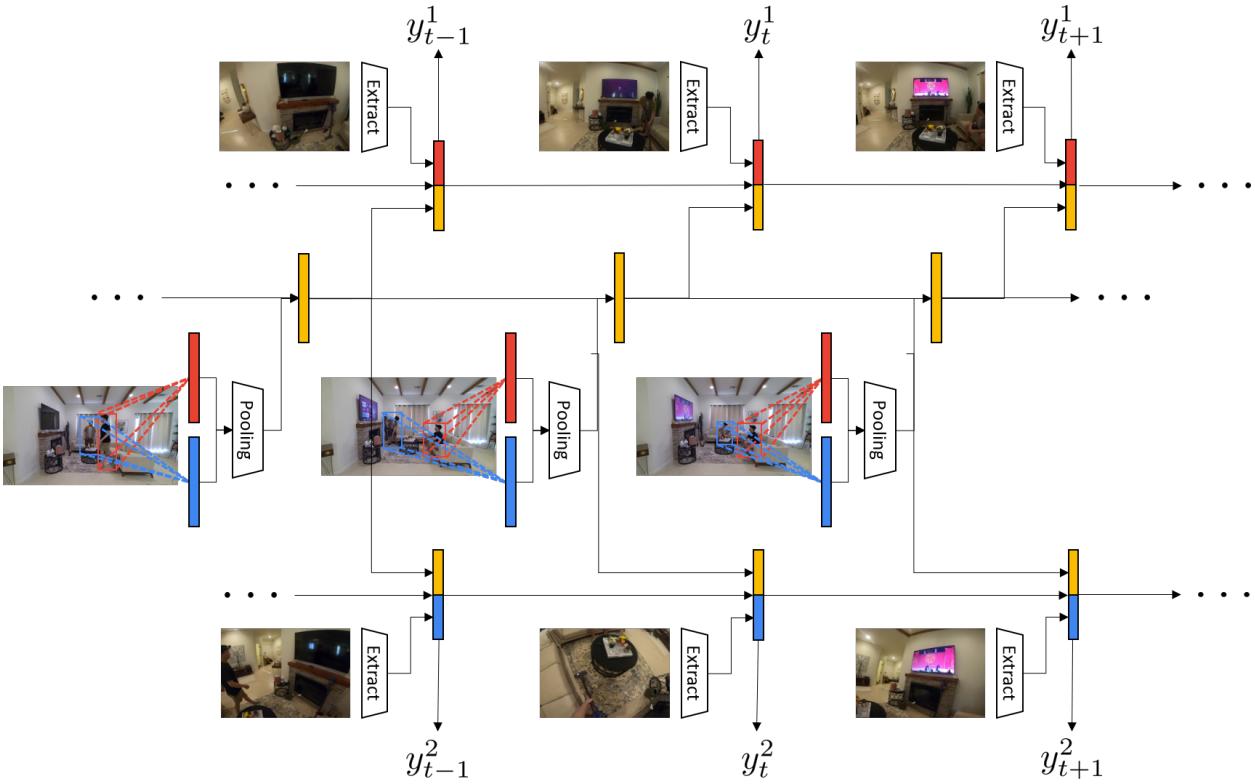


Figure 3.8: An illustration for the multi-agent variants of the original sequential model with TPV features as additional features.

Table 3.3: Comparisons of the action and task anticipations on LEMMA.

Scenario	Method	1 × 1			1 × 2			2 × 1			2 × 2		
		Avg.Prec	Avg.Rec	Avg.F1	Avg.Prec	Avg.Rec	Avg.F1	Avg.Prec	Avg.Rec	Avg.F1	Avg.Prec	Avg.Rec	Avg.F1
Compositional action	SF	23.42	22.25	22.82	20.13	20.06	20.10	18.89	19.22	19.05	18.31	16.67	17.45
	LFB	23.03	28.67	25.54	20.48	25.4	22.67	18.31	22.30	20.11	18.53	20.97	19.68
	M-SF (TPV)	**24.22**	28.05	25.99	20.10	24.48	22.08	19.15	16.71	17.85	19.64	15.18	17.12
	M-LFB (TPV)	23.54	**37.81**	**29.01**	21.10	**31.86**	25.39	19.67	21.03	20.33	**20.11**	20.30	**20.15**
	M-SF (FPV)	23.30	25.41	24.31	**21.34**	23.18	22.22	**19.70**	17.46	18.51	19.82	15.8	17.58
	M-LFB (FPV)	23.26	31.07	26.60	20.78	27.40	23.63	19.42	**21.73**	**20.51**	19.49	20.12	19.8
Task	SF	50.53	79.08	61.66	48.07	67.78	56.25	39.05	57.43	46.49	44.88	62.09	52.1
	LFB	57.57	**84.31**	68.42	52.12	68.94	59.36	38.40	53.08	44.56	48.17	64.61	55.19
	M-SF (TPV)	58.61	79.96	67.05	55.45	67.24	60.78	**45.73**	58.98	**51.51**	**49.66**	64.47	56.10
	M-LFB (TPV)	**60.27**	82.19	**69.54**	**56.2**	**72.46**	**63.30**	43.94	61.41	51.23	48.85	**67.48**	**56.67**
	M-SF (FPV)	51.12	79.18	62.13	48.42	69.04	56.92	41.00	58.11	48.08	46.04	65.97	54.24
	M-LFB (FPV)	55.56	82.83	66.51	52.22	70.01	59.82	41.33	**64.49**	50.38	46.65	69.59	55.86

that models' performances in 2×1 activities are slightly worse than in 2×2 activities. We hypothesize that task plans in the 2×2 scenarios change less frequently, with a clear task assignment coordinating the individual tasks. In comparison, in the 2×1 scenarios, the sequential ordering of the task requires more frequent communication between agents to coordinate. Such a performance gap calls for better modeling of multi-agent task assignments.

3.5 Conclusions

In this chapter, we introduce the LEMMA dataset with a focus on natural multi-agent multi-task daily activities. Dense annotations are provided on both compositional action and task for learning and inference on four different activity scenarios with increasing difficulty. Additionally, we propose two challenging tasks on LEMMA to measure existing models' competence in action understanding and temporal reasoning: (i) compositional action recognition, and (ii) action/task anticipations. The current performance of existing state-of-the-art models suggest efforts should be continually put into natural and realistic goal-directed human activities understanding, especially with complex activities and fine-grained compositional actions.

CHAPTER 4

EgoTaskQA: Understanding Human Tasks in Egocentric Videos

Stepping further into the fine-grained human activity understanding problem, we argue that the task of action localization or future prediction as an *indirect* metric is not enough for evaluating models on the capabilities that human possess. Specifically, these two tasks do not fully reveal the innate strong correspondences between actions, objects, and states. Therefore, to make a *direct* evaluation, we introduce the EgoTaskQA benchmark that builds on top of the egocentric videos collected in LEMMA to evaluate the crucial dimensions of task understanding for models through question-answering. By extending the LEMMA dataset with annotations consisting of object status, human-object and multi-agent relationships, and causal dependency structures between actions, We meticulously design questions that target three specific scopes: (1) actions with world state transitions and their dependencies, (2) agents' intents and goals in task execution, and (3) agents' belief about others in collaboration to provide an in-depth evaluation metric for task understanding. These questions are procedurally generated within four types: **descriptive**, **predictive**, **explanatory**, and **counterfactual**, to systematically test models' capabilities over **spatial**, **temporal**, and **causal** domains of goal-oriented task understanding. To avoid spurious correlations in questions, we include both direct and indirect references to actions and objects. We further balance the answer distribution by the reasoning type of questions and carefully design benchmarking train/test splits to provide a systematic test on goal-oriented reasoning and indirect reference understanding; see Fig. 4.1 for an example.

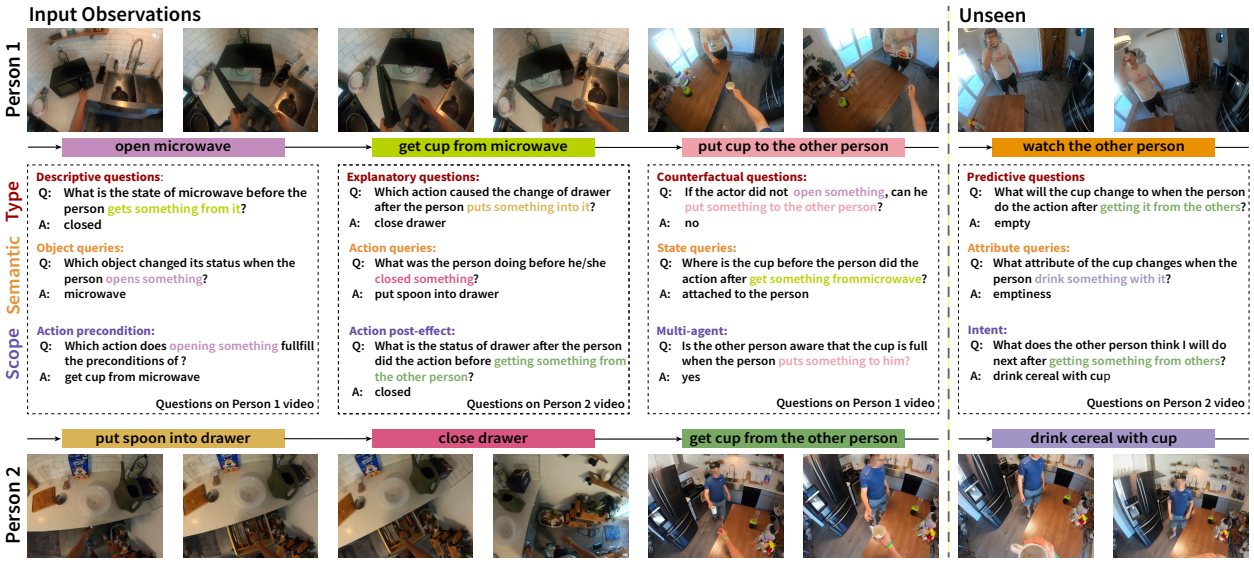


Figure 4.1: Example questions in EgoTaskQA.

The importance of adopting the egocentric view for perception lies in the essence of its usage in our daily life observations. Taking a closer look at how humans learn from interacting with the world, we locate objects, change their positions and manipulate them in various ways, all presumably under visual control from an egocentric perspective [LMR99]. This unique first-person experience provides essential visual cues for human attention and goal-oriented task understanding. Moreover, egocentric perception naturally reflects how humans reason and perform in a partially observable environment, making it the most available learning source for learning actions, tasks [NRK22], and belief modeling [FQZ21].

As shown in Table 4.1, EgoTaskQA complements existing video reasoning benchmarks on various dimensions. In the following sections, we first review all prior works on video-based question-answering. Next, we provide details of the extensions and additional annotations, especially the annotations of object status, human-object and multi-agent relationships, and causal dependency relationships between actions made on LEMMA. We also illustrate the question-answer generation pipeline with a focus on indirect reference and balancing. We further evaluate state-of-the-art video reasoning models on our benchmark and show their significant gaps between humans in understanding complex goal-oriented egocentric videos.

Table 4.1: A comparison between EgoTaskQA and existing video question-answering benchmarks. We use "world" for world model-related information, including action preconditions, post-effects, and dependencies. We use MC as short for multiple-choice question-answering, and OP for open-answer question-answering.

Dataset	Video		Question Scope			Question type				Answer Type	# questions
	View	Real-world	World	Intents & Goals	Multi-agent	Descriptive	Predictive	Explanatory	Counterfactual		
MarioQA [MSJ17]	TPV	✗	✓	✗	✗	✓	✗	✓	✗	OP	188K
Pororo-QA [KHC17]	TPV	✗	✓	✗	✗	✓	✗	✓	✗	MC	9K
CLEVRER [YGL20a]	TPV	✗	✗	✗	✗	✓	✓	✓	✓	OP+MC	282K
Env-QA [GWB21]	FPV	✗	✓	✗	✗	✓	✗	✗	✗	OP	85K
MovieQA [TZS16]	TPV	✓	✗	✗	✗	✓	✗	✓	✗	MC	14K
Social-IQ [ZCL19]	TPV	✓	✗	✓	✓	✓	✗	✓	✗	MC	7.5K
TVQA [LYB18]	TPV	✓	✗	✗	✗	✓	✗	✓	✗	MC	152.5K
TVQA+ [LYB20]	TPV	✓	✗	✗	✗	✓	✗	✓	✗	MC	29.4K
MSVD-QA [XZX17]	TPV	✓	✗	✗	✗	✓	✗	✗	✗	OP	50.5K
MSRVTT-QA [XZX17]	TPV	✓	✗	✗	✗	✓	✗	✗	✗	OP	243K
Video-QA [ZCC17]	TPV	✓	✗	✗	✗	✓	✗	✗	✗	OP	175K
ActivityNet-QA [YXY19]	TPV	✓	✗	✗	✗	✓	✗	✗	✗	OP	58K
TGIF-QA [JSY17]	TPV	✓	✗	✗	✗	✓	✗	✗	✗	MC	165.2K
How2QA [LCC20]	TPV	✓	✗	✗	✗	✓	✗	✗	✗	MC	44K
HowToVQA69M [YMS21]	TPV	✓	✗	✗	✗	✓	✗	✗	✗	OP	69M
AGQA [GKA21]	TPV	✓	✗	✗	✗	✓	✗	✗	✗	OP	3.6M
NExT-QA [XSY21]	TPV	✓	✗	✓	✗	✓	✓	✓	✗	OP+MC	52K
STAR [WYC21]	TPV	✓	✓	✗	✗	✓	✓	✗	✗	MC	60K
EgoVQA [Fan19]	FPV	✓	✗	✗	✓	✓	✗	✗	✗	OP+MC	520
EgoTaskQA (Ours)	FPV	✓	✓	✓	✓	✓	✓	✓	✓	OP	40K

With models exhibiting large performance gaps compared to humans, we devise diagnostic experiments to reveal both the easy and challenging spots in our benchmark. We hope such designs and analyses will foster new insights into goal-oriented activity understanding.

4.1 Related Work

Action as Inverse Planning Action understanding has been seen as an inverse planning problem on agents' mental states [BST09, SKL19]. Early studies formulate it as reasoning on the first-order logic formulae that describes actions' preconditions and post-effects [McC63, Rei91]. This symbolic formalism is later paired with domain-specific language and algorithms to become mainstays in robotics planning [FN71, MGH98]. In computer vision, similar attempts have been made to link visual observations with world states and actions [DPC12, ILA15, NG18]. Various methods treated actions as transformations on images to solve action-state recognition [FR13, FZ15, WFG16, ALS17, LWZ17] and video prediction [OGL15, VPT16, HS18]. With the emerging interest in language-grounded understanding, Zellers *et al.* [ZHP21] proposed PIGLeT to study the binding between images,

world states, and action descriptions. Padmakumar *et al.* [PTS22] further studies the problem of language understanding and task execution by designing an intelligent embodied agent that can chat during task execution. However, these works are mostly limited to atomic actions, missing the important action dependency in task execution. To tackle this problem, instructional videos [KAS14, ABA16, TDR19, MZA19] are studied with their goal-oriented multi-step activities. In these videos, external knowledge [PHM16, KW18] can be used as guidance for advanced tasks like temporal dynamics learning [EWS21] and visually grounded planning [CHX20, SHL22]. Unfortunately, these videos highlight the instructions and include no task-level noise, which is much simpler than the partially observable, highly paralleled, multi-agent environment that humans learn from and as presented in our benchmark. These complexities make the goal-oriented action understanding a challenging task remaining to be solved.

Egocentric Vision Egocentric vision offers a unique perspective for actively engaging with the world. Aside from traditional video understanding tasks like video summarization [LGG12, LG13], activity recognition [FFM19, WLM22, GSR22] and future anticipation [NLF20, FF20, QJZ18, QJH20, GG21], egocentric videos provide fine-grained information for tasks like human-object interaction understanding [NFG19, DLM16, CKS16, BLC15, QWJ18, GYB18, MFK16] and gaze/attention prediction [WLS18, LLR18]. With its natural reflectance of partial observability, egocentric videos are also used for social understanding tasks such as joint attention modeling [FHR12, SS15], perspective taking [YMY18, NXJ20] and communicative modeling [NZL20, FQZ21]. However, with various egocentric datasets curated over the last decade [PR12, LGG12, SGS18], data and detailed annotations for human tasks are still largely missing. Large-scale daily lifelog datasets like EPIC-KITCHENS [DDF22] and Ego4D [GWB22] cover certain aspects of action-dependencies, effects, and social scenarios in their recordings, but are unsuitable for detailed annotation due to their size. The other stream of datasets collects activities by providing coarse task instructions to both single actor [RCJ21]

and multiple agent collaborations [JCH20]. They annotate tasks and compositional actions to reveal agents' execution and collaboration process for multi-step goal-directed tasks. Despite all the preferred characteristics of these goal-oriented activity videos, none of them successfully addressed action-dependencies and effects, nor multi-agent belief modeling.

Video Question-Answering Benchmarks Visual question-answering can be designed to evaluate a wide spectrum of model capabilities, spanning from visual concept recognition and spatial relationship reasoning [TML14, AAL15, JHV17, HM19], abstract reasoning [BHS18, ZGJ19, NYM20, ZJZ21, ZJE21, ZXJ22], to common sense reasoning [PBM20, ZBF19]. In the temporal domain, synthetic environments are used for questions that involve simple action-effect reasoning [MSJ17, KHC17]. Crowdsourced videos [JSY17, YXY19, LYB18, YMS21] are used for collecting questions on basic spatial-temporal reasoning capabilities like event counting [JSY17], grounding [LYB20], and episodic memory [GWB22]. Recent advances in video question-answering aim for more profound reasoning capabilities. Gao *et al*. [GWB21] leverages an indoor synthetic environment to generate questions on spatial relationships and simple action-effect reasoning from an egocentric perspective. Xiao *et al*. [XSY21] designs NExT-QA containing questions about knowledge of the past, present, and future on both temporal and causal domains. Grunde-McLaughlin *et al*. [GKA21] programmatically generates questions for compositional spatial-temporal reasoning and generalization. Wu *et al*. [WYC21] focus on short atomic action clips for situated reasoning. Yi *et al*. [YGL20a] generates synthetic videos for studying counterfactual predictions on collisions. Zadeh *et al*. [ZCL19] collects questions for social intelligence evaluation. Nevertheless, none of these benchmarks addressed the aforementioned critical dimensions of goal-oriented activity understanding from a real-world egocentric perspective.

4.2 The EgoTaskQA Benchmark

The EgoTaskQA benchmark contains 40K balanced question-answer pairs selected from 368K programmatically generated questions generated over 2K egocentric videos. We target the crucial dimensions for understanding goal-oriented human tasks, including action effects and dependencies, intent and goals, and multi-agent belief modeling. We further evaluate models' capabilities to describe, explain, anticipate, and make counterfactual predictions about goal-oriented events. A detailed comparison between EgoTaskQA and existing benchmarks is shown in Table 4.1.

4.2.1 Data Collection

We select egocentric videos from the LEMMA dataset [JCH20] as base video sources. Compared to similar egocentric datasets, human activities in LEMMA are highly goal-oriented and multi-tasked. These activities contain rich human-object interactions and action dependencies in both single-agent and two-agent collaboration scenarios. We take advantage of these desired characteristics and augment LEMMA with ground truths of object states, relationships, and agents' beliefs about others. More specifically, we augment LEMMA on the following aspects:

World States We focus on world states consisting of object states, object-object relationships, and human-object relationships. First, we build the vocabulary of relationships and state attributes from activity knowledge defined in previous works [PHM16, JKF20]. We manually filter irrelevant relationships and attributes by removing dataset-specific (*e.g.*, under the car) and detailed numerical (*e.g.*, cut in three) relationships. Next, we gather similar relationships to obtain 48 relationships and 14 object attributes. This vocabulary covers spatial relationships (*e.g.*, on top of), object affordances (*e.g.*, openable), and time-varying attributes (*e.g.*, shape). We build on top of action annotations from LEMMA and use Amazon Mechanical Turk (AMT) to annotate this information before and after the changing action

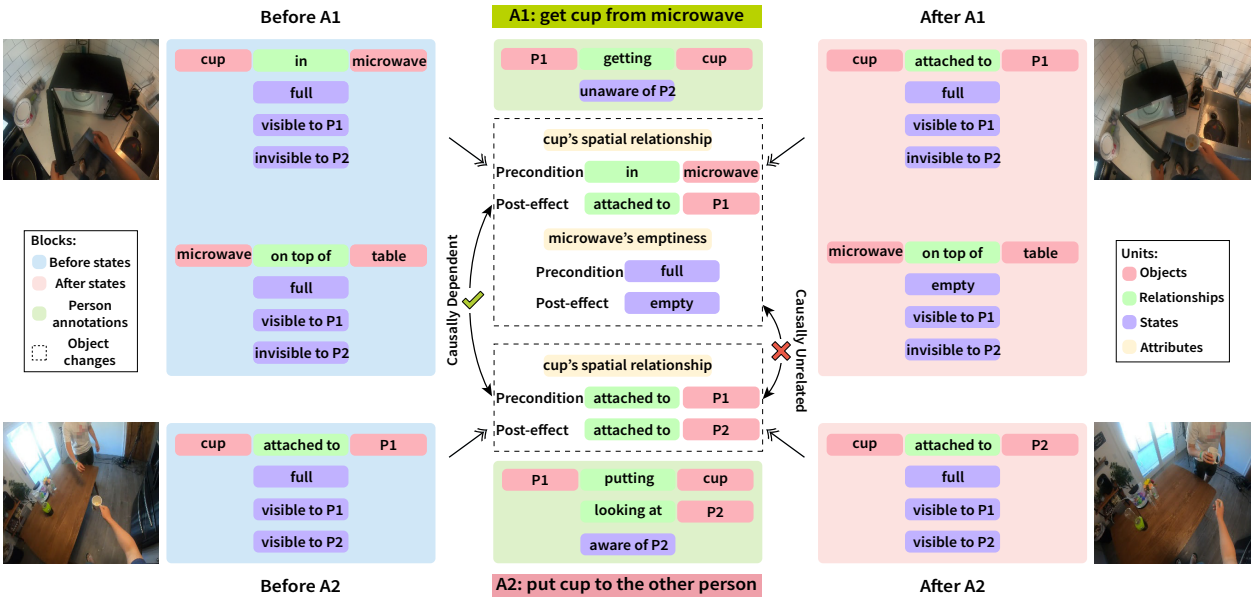


Figure 4.2: We use two actions A1:"get cup from microwave" and A2:"put cup to the other person" as an example to visualize annotations in EgoTaskQA. We annotate states and relationships for objects changed by actions as well as human-object and multi-agent relationships and decide the causal dependency between actions based on the "before" and "after" annotations.

for all time-varying objects. With these annotations, we reconstruct the transition chain for each interacted object and obtain their temporal status.

Multi-agent Relationships To capture how two agents (actor and helper) collaborate over the same task, we annotate basic information about objects' visibility and the actor's awareness of the helper. For each object that the actor operates on, we annotate its visibility to the helper by providing synchronized videos from both agents' views to AMT workers. For the actor's awareness of others, we instruct AMT workers to first go through the egocentric view video of both agents to get familiar with actions performed by the actor and the helper. Next, we ask AMT workers to replay the video of the actor and annotate, during each action segment, whether the actor can see the helper or whether the actor is aware of the helper's action if the helper is not in sight. As this annotation is usually subjective, we take the majority vote of three workers as ground truth.

Causal Trace Based on the annotated transition chain of objects, we generate causal traces for each action with rules. By checking whether the post-effect of one action fulfills the preconditions of another, we define the causal relationship between two actions into unrelated, related, and causally dependent; see Fig. 4.2 for an illustration.

Given two actions a_1 and a_2, and their state annotations s_1 and s_2, we determine the causal dependency between them as shown in Algorithm 1. We first collect all interactive object set O_1 and O_2 for a_1 and a_2, and see if there exists an overlap of objects. If no, we assume a_1 and a_2 is not *related*. Next, for each object o that is interacted in both actions, we check whether a_1 lead to the change of attribute s, which is a precondition of a_2's change on o. This condition is validated by checking if o changed the same attribute s in both a_1 and a_2, and the status after a_1 equals the status before a_2, *i.e.* $s_{1,o}^{\text{after}} = s_{2,o}^{\text{before}}$. We say that a_1 and a_2 are causally *dependent* if this condition is satisfied. If there exists an attribute s that was affected by a_1 and did not change during a_2, *i.e.* $(s_{1,o}^{\text{before}} \neq s_{1,o}^{\text{after}}) \wedge (s_{1,o}^{\text{after}} = s_{2,o}^{\text{before}})$, we say that these two actions are *related* since we can not determine whether this relationship is causal or not from the annotations. As currently, we do not use additional human resources for verifying each of these *related* actions, we limit our scope of question generation to the *dependent* and *unrelated* action pairs. After checking the causal dependency for all action pairs in the video, we recursively construct the dependency tree by taking each action as root and adding actions that are dependent on all dependants of the action to its dependants set. During the recursion, we update the dependency for a newly added action to *related* if there exist *related* dependency relationships in the path from the root action to it.

Algorithm 1: Causal Dependency Check

Input: two actions a_1 and a_2 and their object state annotation S_1 and S_2.

Output: the causal dependency relationships between a_1 and a_2.

Gather all interactive objects $O_1 = \{o_i^1\}_{i=1}^m$ and $O_2 = \{o_i^2\}_{i=1}^n$ in action a_1 and a_2.

if $O_1 \cap O_2 = \varnothing$ **then**
  return *unrelated*

else for $o \in O_1 \cap O_2$ **do**
  for $s_{1,o} \in S_1,\ s_{2,o} \in S_2$ **do**
    if $(s_{1,o}^{before} \neq s_{1,o}^{after}) \wedge (s_{1,o}^{after} = s_{2,o}^{before}) \wedge (s_{2,o}^{before} \neq s_{2,o}^{after})$ **then**
      return *dependent*
    else if $(s_{1,o}^{before} \neq s_{1,o}^{after}) \wedge (s_{1,o}^{after} = s_{2,o}^{before})$ **then return** *related*

return *unrelated*

Given a video, we run this dependency check for each pair of actions, resulting in a video-level dependency tree generated by recursively checking sequential depending relationships. We use it as the ground truth dependency structure for subsequent explanatory and counterfactual question generation.

In total, we augment LEMMA with 30K annotated before states, after states, and person annotation blocks as shown in Fig. 4.2. We then segment the videos in LEMMA into clips with lengths of around 25 seconds for question generation. This design helps generate interesting clips with partially observed environmental constraints (*e.g.*, the cup is already washed when the person pours juice), and visual hints for future actions (*e.g.*, cutting watermelon into dice instead of pieces for making juice rather than eating it directly). Meanwhile, we keep our videos reasonably long, with an average of 5 actions per clip to cover sufficient information for action dependency inference and future prediction.

As we can see from the histogram Fig. 4.3, spatial relationships of objects were annotated the most, followed by multi-agent relationships like "aware of others" and "looking at". This meets our expectation of the frequent changes in objects' spatial relationships during goal-

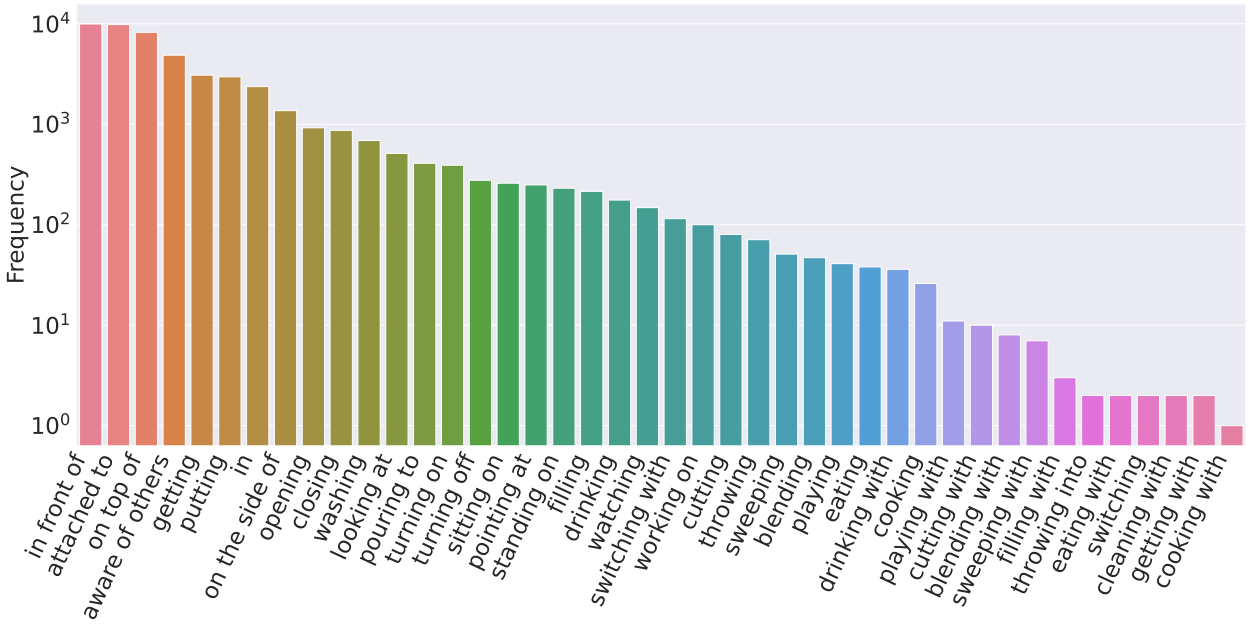


Figure 4.3: Statistics of relationships annotated during EgoTaskQA data collection.

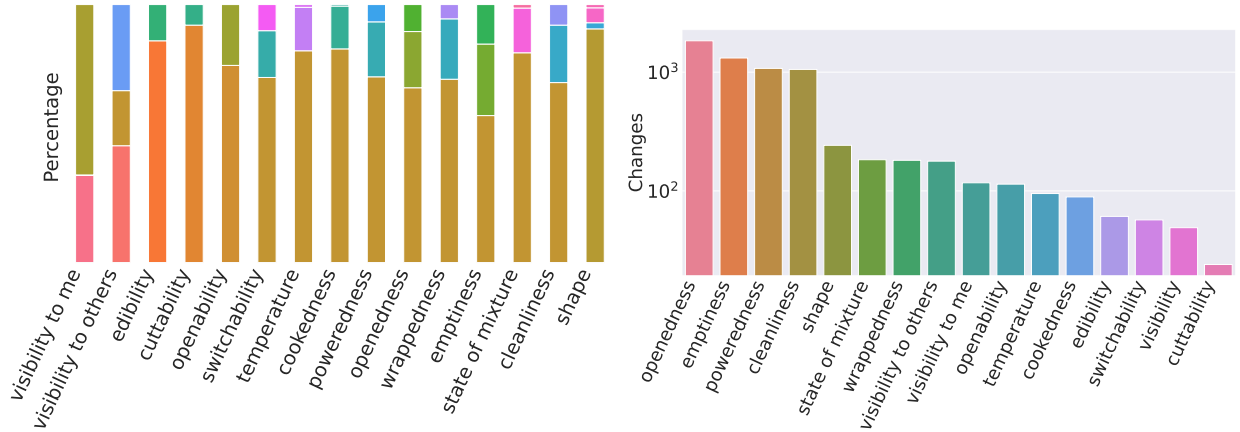


Figure 4.4: Statistics of object state/attribute change.

oriented task execution. Action-related relationships also make up a considerable portion of overall relationship annotations and describe detailed relationships between the person and the target object (*e.g.* getting, putting, pouring) or the tool object (*e.g.* getting-with, cutting-with, putting-with).

We list all annotated object attributes and their state values in Table 4.2, and visualize their statistics in Fig. 4.4. We add an option "unknown" to all attributes for annotating

Table 4.2: A full list of time-varying object attributes considered and their corresponding values.

Attribute	Type	Possible State Values
visibility to me	visibility	visible to me / invisible to me / unknown
visibility to the other person	visibility	visible to the other person / invisible to the other person / unknown
edibility	affordance	edible / can not be eaten / unknown
cuttability	affordance	cuttable / not cuttable / unknown
openability	affordance	openable / can not be opened / unknown
switchability	affordance	can be turned on / can not be turned on / unknown
temperature	status	boiled / in room temperature / unknown
poweredness	status	on / off / unknown
cookedness	status	cooked / raw / unknown
wrappedness	status	wrapped / unwrapped / unknown
emptiness	status	empty / full / unknown
state of mixture	status	mixing / not mixing / unknown
cleanliness	status	clean / dirty / unknown
shape	status	whole / part / diced / fluid / unknown

unclear scenarios and ignore this answer during question generation. As shown in Table 4.2 and Fig. 4.4, we consider various time-varying object attributes including visibility, affordance (*e.g.* cuttability, edibility), and task-dependent status (*e.g.* emptiness, shape). In Fig. 4.4 (right), we plot the number of changes for each object attribute. In addition to spatial relationship changes described previously, there is an increasing number of occurrences from affordance changes to visibility changes and, finally, task-dependent status changes. As LEMMA is recorded in indoor environments (kitchens and living rooms), we also observe a large amount of containment relationship changes ("open/close" and "emptiness"). We argue that this data is also potentially beneficial for the study of containment relationships [LZZ16].

4.2.2 Question-Answer Generation

We use machine-generated questions to evaluate models' task understanding capabilities. We focus on the transition chain of each interacted object, especially what actions caused changes in objects and how these changes contribute together to a multi-step task.

Question Design We design questions that pinpoint scopes, including (1) action preconditions, post-effects, and their dependencies, (2) agents' intents and goals, and (3) agents'

beliefs about others. Similar to [YGL20a], we categorize our questions over these three scopes into four types to systematically test models' capabilities over spatial, temporal, and causal domains of task understanding:

- **Descriptive** questions evaluate the understanding of detailed spatial-temporal information. We provide spatial-temporal references in the questions to identify a unique interval for answering queries on objects and actions. These properties include object states and changes, relationships, human actions, and multi-agent-related information. We generate this type of question by randomly sampling an interval in the video clip and then gathering all related annotations for question generation. Answers in this category are generated based on interval annotation and contain both open-ended queries and statement verifications.
- **Predictive** questions aim at understanding intents and task planning. Given a video clip, we ask about possible future object states and actions for both the actor and the helper. These predictions include both direct predictions on actions and objects, as well as more challenging task-dependent predictions such as the executability of actions and the desired states of objects. Questions and answers for predictive questions are generated by gathering the future action/object annotations in a fixed window size after the truncated interval (*i.e.* unseen future video) in the long original video. Answers in this category are open-ended action, object, and state queries.
- **Counterfactual** questions aim at understanding action preconditions and post-effects. Based on the causal trace of actions, we generate counterfactual questions with hypothetical conditions that certain actions in the sequence were not executed. Under this condition, we query both the affected and unaffected actions about their executability and whether the corresponding changes of object states associated with these actions will occur. We generate counterfactual questions by adding or removing actions in the causal trace and adjusting the depending actions' executability recursively. Answers in this category contain action executability verifications and object state queries.

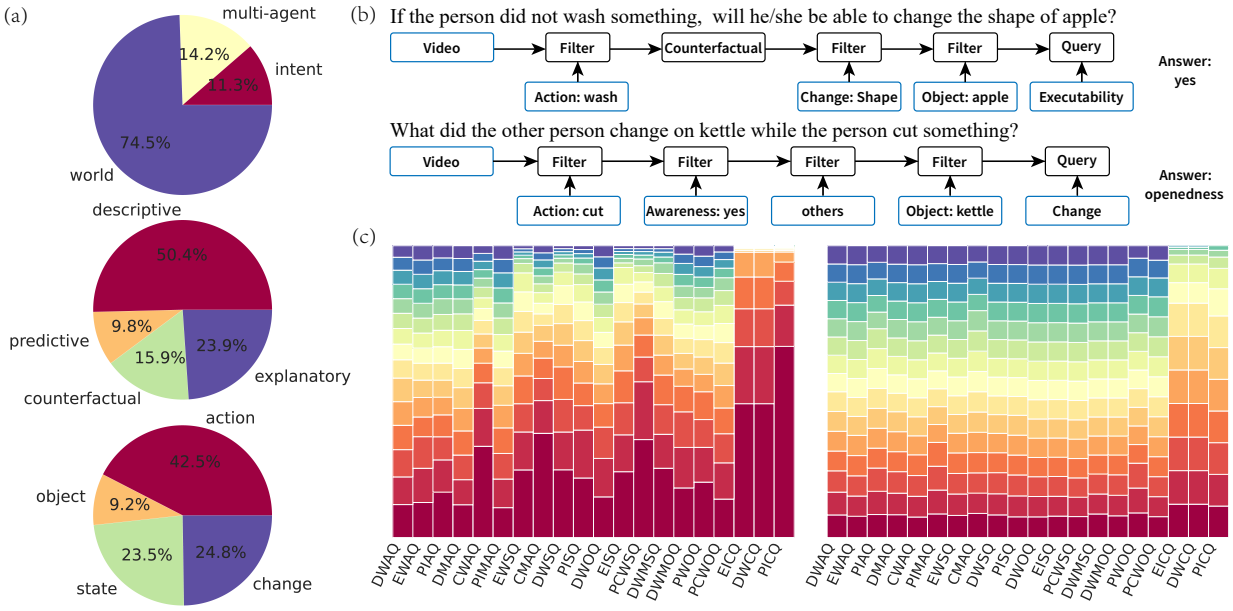


Figure 4.5: An illustration of the generation pipeline and statistics of the question-answer pairs. We balance questions by reasoning types and use the abbreviation with the concatenation of their initial letters (*e.g.*, DWAQ for descriptive, world, action, and query)

- **Explanatory** questions evaluate the understanding of task-related object changes as well as action preconditions and post-effects. Given the object state annotations and the causal trace, we query the cause of state changes, the leading factor that satisfies the preconditions of specific actions, as well as why would the post-effect of certain actions affects other actions in the video clip. We generate explanatory questions by querying both the annotations as well as the causal trace. Answers for explanatory questions contain both open-ended and verification queries.

Answer Generation In EgoTaskQA, we consider both open-answer queries and binary statement verifications. To generate answers, we first collect video intervals as mentioned in Section 4.2.1. These clips are cropped from original videos to contain 4∼5 actions on average. We further concatenate the next three actions performed by the actor that is unseen in videos into the intervals of interest for generating predictive questions. After generating these intervals of interest, we gather all corresponding annotations, including

Table 4.3: All program modules used for question-answer generation.

Query Operation	Parameter List	Return Type	Usage and Example
Filter	arg_1: conditions arg_2: intervals	intervals	Return the intervals that satisfies the conditions. `filter([obj@spoon, change@cleanliness], video)`
Only	arg_1: intervals	interval	Return the only interval from list, return `None` if $\|arg_1\| \neq 1$. `only(filter([action@putting]))`
Localize	arg_1: before/after arg_2: interval	intervals	Return all intervals before/after the interval provided in arg_2. `localize(before, only(filter(action@putting)))`
IterateUntil	arg_1:forward/backward arg_2: intervals	interval	Return the first interval of the interval list from the front/back. `iterate_until(forward, filter([change@emptiness], video))`
Query	arg_1: conditions arg_2: interval	value	Return the value from the interval identified by the conditions. `query([aware@yes, action$], only(filter([action@getting], video)))`
Verify	arg_1: conditions arg_2: interval	bool	Verify arg_1 in the interval arg_2, return "yes" if satisfied, "no" otherwise. `verify([change@openedness, obj@closet], only(filter([action@closing], video)))`
Pred	arg_1: intervals	intervals	Return the anticipating intervals within intervals arg_1 `pred(filter([action@pouring], video))`
Counterfactual	arg_1: conditions arg_2:intervals	intervals	Return the original intervals with executability of each interval adjusted according to the counterfactual query arg_2. `counterfactual([action@getting], video)`
Depend	arg_1: interval arg_2: interval	bool	Return "yes" if interval arg_1 and $interval_2$ are dependent, "no" otherwise. `depend(only(filter([action@opening], video)), only(filter([action@closing], video)))`

annotations for both the actor's action and the helper's (*i.e.* the other person's) simultaneous actions. We organize these annotations in a dictionary for convenience purposes. Next, to generate question-answer pairs for these questions, we design both text templates and the corresponding functional program templates as shown in Fig. 4.5 (b). More specifically, we design operators that work on the annotation dictionaries with different purposes. Inspired by previous works [JHV17, YGL20a, GKA21], we design nine basic operators for composing the logic for each program template. We provide the specification of each operator, and its usage with an example, in Table 4.3. The basis of these programs lies in the conditional query, similar to database queries. We use $A@B$ for filtering data with the attribute A equal B, and we use $A\$$ for querying the value of attribute A from data. The resulting programs consist of sequences of modules for querying the answers from the annotations and the causal traces. We exhaustively execute all possible program instantiations on videos to obtain answers by substituting arguments with instances in the available sample space. As all questions take action grounding as a prerequisite, we add indirect references (*e.g.*, the first..., the action before...) to actions and objects when making substitutions to reflect this challenge. Specifically, we make use of the provided templates and use indirect reference to substitute parameters for action ($\{a\}$) and object ($\{o\}$). Concretely, we substitute the corresponding positional arguments $\{a\}$ and $\{o\}$ with the substituting text and program templates. As this substitution could be easily adapted to have multi-step indirect references, we limit

Table 4.4: Question-answer pair statistics before and after balancing.

	World	Intent	Multi-agent	Descriptive	Predictive	Counterfactual	Explanatory	Action	Object	State	Change	Open	Binary
Before	299K	43K	53K	181K	22K	71K	102K	122K	14K	105K	126K	182K	186K
After	32K	5K	6K	21K	4K	6K	9K	17K	4K	9K	10K	26K	13K

the indirect references in our benchmark to 1-step indirect references to avoid generating questions that are difficult to understand. To facilitate models' understanding of indirect references, we add additional questions on these indirect queries for objects and actions. After these processes, we obtain 368K question-answer pairs over 2K videos as the full question set.

Answer Distribution Balancing We balance our answer distribution to avoid shortcuts from exploiting imbalances. Following the scheme introduced in [GKA21], we tag each question template with its scope, type, and the targeting semantic category (*e.g.*, actions, objects, states) and use the composition of all tags as the unique reasoning type for each question. We balance binary verification questions to have an equal proportion of each answer within each reasoning type. For open-answer questions, we use rejection sampling to ensure that the top 20% frequent answers for each reasoning type do not appear as answers for more than 33.3% of questions in the same type. After balancing by reasoning types, we proportionally sample questions to obtain a 40K diverse and balanced question set with a 1:2 ratio of binary and open-answer questions. We visualize the statistics of questions and answers and the effect of answer balancing in Fig. 4.5 . More specifically, we follow the algorithm provided by [GKA21] and adjust the open-answer problems to ensure that the top 20% answers of each reasoning type do not answer to more than 33% questions in the same type. We select this ratio to get a smoother answer distribution while not deleting too many questions in the whole set. To avoid overfitting to the binary answer distribution, we control the ratio between open-answer and binary questions to be 2:1. We show the statistics for each general question type before and after balancing in Table 4.4.

Benchmark Splits We provide two benchmarking splits *normal* and *indirect* for video question-answering on EgoTaskQA. For the *normal* split, we randomly sample questions according to their answer distribution and reasoning types to have a 3:1:1 split over training, validation, and test sets. The *indirect* split is motivated by the fact that during task execution, actions, objects, and their changes are often strongly correlated. It leaves the chance for the model to perform well by simply over-fitting these strong correlations without thorough task understanding; see Section 4.3.2 for a more in-depth discussion. We leverage the indirect references in our question to inspect the models' capability to use the learned knowledge for multi-step reasoning and generalize them to indirect references without over-fitting. More specifically, we filter questions without indirect references and simple indirect reference questions without multiple reasoning steps (*e.g.*, what is the first action this person did? what did the person do before action "putting something"?) from all question-answer pairs to form the training set, and split all indirect reference questions with multiple reasoning steps as validation and test sets. Under this setting, the *indirect* split has a portion of 2:1:1 for training, validation, and test sets, respectively. We leave the remaining discussion of the *indirect* split to Section 4.3.3.

4.3 Experiments

In this section, we evaluate and analyze the performance of video question-answering models on EgoTaskQA. We report how well models perform on different question scopes, types as well as targeting semantics on both *normal* and *indirect* splits. We also provide diagnostic experiments on the language modality to show the necessity of the *indirect* split.

Baselines In our experiments, we evaluate six state-of-the-art video question-answering models: VisualBERT [LYY19], PSAC [LSG19], HME [FZZ19], HGA [JH20], HCRN [LLV20], and ClipBERT [LLZ21]. VisualBERT is a VL-BERT model designed for vision-language

Table 4.5: Model performance on the EgoTaskQA *normal* split.

	Category	Most Likely	VisualBERT [LYY19]	PSAC [LSG19]	HME [FZZ19]	HGA [JH20]	HCRN [LLV20]	ClipBERT [LLZ21]	Human
Scope	world	18.62	39.73	40.76	41.91	38.82	**44.27**	42.15	74
	intent	2.54	44.51	46.19	48.92	42.12	**49.77**	40.94	82
	multi-agent	10.92	26.29	30.59	27.98	23.43	**31.36**	27.63	76
Type	descriptive	18.64	41.99	40.63	41.45	38.04	**43.48**	38.45	88
	predictive	1.57	30.37	31.98	35.88	25.57	**36.56**	31.50	88
	counterfactual	23.62	41.99	41.89	44.13	41.94	**48.00**	46.75	80
	explanatory	7.97	37.42	37.99	38.85	35.97	40.60	**42.39**	74
Semantic	action	10.05	15.02	14.75	14.99	15.08	14.92	**22.91**	70
	object	2.07	23.26	36.53	36.05	19.09	**45.31**	21.80	82
	state	6.05	59.20	61.89	63.44	55.65	**68.28**	54.36	80
	change	41.97	68.27	65.05	**68.87**	68.38	67.38	66.58	82
Overall	open	0.70	24.62	26.97	27.66	22.75	**30.23**	27.70	82
	binary	50.46	68.08	65.95	68.6	68.53	**69.42**	67.52	76
	all	15.4	37.93	38.90	40.16	36.77	**42.20**	39.87	80

tasks. PSAC uses positional self-attention and co-attention network blocks to fuse visual and language features. HME uses external memory blocks for both visual inputs and questions on top of an LSTM-based encoder-decoder structure. HGA formulates video question-answering by constructing graphs for both videos and questions and aligning them. HCRN adopts a hierarchical framework by stacking relational modules over motion, question, and visual features. ClipBERT leverages sparsely sampled video clips and grid features [JMR20] in a transformer architecture and achieves state-of-the-art results on video question-answering. We formulate question-answering in EgoTaskQA as a classification problem over all answer vocabulary and use models' accuracy as the evaluation metric under different settings.

4.3.1 Comparative Analysis

We provide experimental results of baseline models on the EgoTaskQA *normal* split in Table 4.5. Model performances are evaluated on question scopes, types, targeting semantics, and overall answer categories. To quantify the naturalness and correctness of questions and answers in the EgoTaskQA benchmark, we provide human evaluation following the consistency check introduced in [HM19, GKA21]. More specifically, we randomly sample 50 questions for each category and instruct AMT workers to evaluate the quality of the generated answer. Additionally, we compare all baseline models with a simple frequency-based baseline, namely "Most Likely" in Table 4.5, where we select the most likely answer for each category

to answer all questions in that category.

As shown in Table 4.5, the low performance of the most likely answer proves that our answer distribution is correctly balanced. For certain categories (*e.g.*, change), the most likely answer has relatively high accuracy (41.97%) as it covers both open-answer and binary questions. Next, we observe relatively low human performance in certain categories (*e.g.*, action and explanatory). This indicates that identifying causal dependency between actions and conducting multi-step reasoning is not a trivial task for humans as also discovered in [GKA21]. However, we still observe a large gap between state-of-the-art models and human performance. Among all models, we find HME, HCRN, and ClipBERT to perform the best. This result is reasonable since they leverage different ways to provide better visual representations and interactions between video and language. Among all question scopes, we recognize a relatively low accuracy on multi-agent-related questions among all question scopes. It implies that understanding other agents' actions during task execution is still difficult without explicit modeling. It is significant in egocentric vision as a person's view changes dramatically, and only glances can be taken to acquire others' information. Meanwhile, we notice that these models perform relatively well for questions on states and changing attributes. We conjecture that this is attributed to the task knowledge embedded in textual descriptions of questions since actions, objects, and state changes are strongly correlated, as mentioned in Section 4.2.2.

4.3.2 The Effectiveness of Language

Object information We found the object information in the texts to be highly beneficial for question-answering on task-related knowledge during initial experiments. Compared to the original LEMMA action annotation (*e.g.*, drinking [cereal] with [cup]), we use verbs to refer to actions in EgoTaskQA and obfuscate object information at different levels (*e.g.*, drink something with cup, drink something with something) as similarly done in [GKA21, WYC21]. While both types of action references localize to the same action interval, it contains different

levels of knowledge in the language modality. Intuitively, the combination of action verbs (*e.g.*, cut) and targeting objects (*e.g.*, watermelon) provide object state information (*e.g.*, diced) under certain scenarios. Therefore, we compare models' performance at different levels of object information obfuscation. As shown in Fig. 4.6, we recognize a significant performance gain for all models by gradually removing object information obfuscation in text, *i.e.*, substituting "something" with the original object. This result supports the hypothesis that with fine-grained action annotations, we can learn task-related knowledge reasonably well by simply exploiting texts. It shares the same conclusion with recent works on leveraging text-based knowledge for helping instructional video understanding [LPB22]. To further investigate the effectiveness of the language modality, we conduct ablative experiments on the EgoTaskQA *normal* split.

Language-Only Language has been shown to provide knowledge that helps visual question-answering [GKS17]. To study the role of language in EgoTaskQA, we design a text-only setting for VisualBERT and HCRN, testing BERT [DCL18] and HCRN without vision against their vision-language counterparts. As shown in Table 4.6, the performance for most question categories dropped significantly. For the task of video question-answering, we should expect that dropping the vision branch will significantly affect the models' performance. As shown in Table 4.6, we observe the general performance for the two models decreased as we expected. Among all categories, the models' performance for the objects decreased the most, which is consistent with the fact that the object queries highly depend on the situation provided in the videos (*e.g.*, which object changed its status in the video?). However, we observe a slight performance gain on object state change questions. This further suggests that the knowledge of world state change, *i.e.* which object attribute could change under actions, is embedded within question texts. Models could exploit question texts to learn simple associations between attribute types and action verbs (*e.g.*, cleanliness and wash, emptiness and pour, shape and cut, *etc.*).

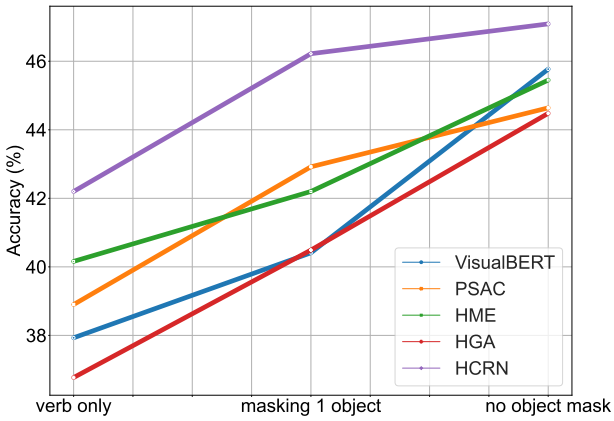


Figure 4.6: Ablative study on model performance with different levels of object information obfuscation on the EgoTaskQA *normal* split.

Table 4.6: Language-only question-answering results on the EgoTaskQA *normal* split.

Category	BERT [DCL18]		HCRN (w/o vision)	
	Acc.	Change	Acc.	Change
world	36.28	-8.7%	35.22	-20.4%
intent	35.02	-21.3%	34.93	-29.8%
multi-agent	20.58	-21.7%	19.17	-38.9%
descriptive	34.55	-17.7%	33.58	-22.8%
predictive	24.75	-18.5%	24.3	-33.5%
counterfactual	41.3	-1.6%	40.4	-15.8%
explanatory	31.78	-15.1%	30.57	-24.7%
action	15.72	+4.6%	15.64	-1.7%
object	7.43	-68%	6.33	-86.0%
state	45.03	-23.9%	42.51	-37.7%
change	69.87	+2.3%	68.77	+2.1%
all	33.92	-10.6%	32.51	-23.0%

Table 4.7: Model performance on the EgoTaskQA *indirect* split.

	Category	BERT	HCRN (w/o vision)	VisualBERT	PSAC	HME	HGA	HCRN	ClipBERT
Scope	world	34.96	33.61	40.00	**44.74**	35.91	31.29	44.04	26.51
	intent	23.56	23.98	36.02	**48.38**	31.73	20.42	47.02	14.66
	multi-agent	19.70	19.25	26.02	**35.37**	25.07	17.74	30.11	20.09
Type	descriptive	33.09	30.73	38.9	**43.36**	34.48	29.01	42.02	24.35
	predictive	15.58	13.68	31.37	29.11	27.79	15.16	**46.32**	10.32
	counterfactual	34.59	34.75	37.63	39.94	35.07	33.01	**43.64**	26.29
	explanatory	27.38	28.11	32.75	**42.53**	29.16	24.00	39.69	22.46
Semantic	action	26.91	28.18	27.49	**30.06**	25.12	26.15	29.61	25.25
	object	2.808	4.13	22.63	30.97	19.08	7.02	**32.20**	10.49
	state	21.96	21.24	32.02	**43.29**	31.60	17.67	41.81	15.29
	change	55.28	50.71	55.59	**57.20**	47.65	47.22	56.27	35.26
Overall	open	11.22	11.38	21.05	**28.23**	18.27	8.66	27.82	11.17
	binary	58.24	55.52	57.61	**60.30**	52.55	53.72	59.29	40.71
	all	31.78	30.76	37.01	**42.25**	33.06	28.36	41.56	24.08
Performance Change		-6.4%	-5.4%	-2.4%	+4.9%	-17.7%	-22.9%	-1.5%	-39.6%

4.3.3 Generalizing to indirect references

On the EgoTaskQA *indirect* split, we evaluate models' capability to leverage learned task knowledge for solving more complicated indirect reference tasks. With the *normal* split allowing for shortcuts on action-state associations, the *indirect* split forbids such exploitation by differentiating references during training and testing. As shown in Table 4.7, we observe

more significant performance drops in language-only models compared to their vision-language counterparts. More specifically, the performance of BERT and language-only HCRN dropped 20.8% and 26.3% on the "change" category, where we observed potential exploitation on question texts in Section 4.3.2. This serves as a shred of evidence that the *indirect* split helps reduce the possibility of exploiting simple associations in texts. As for baseline models, we recognize a common performance decrease shared by most models on the *indirect* split. Among them, we notice a significant performance drop for ClipBERT, which conflicts with the dominating role of large-scale pretrained vision-language models on various reasoning tasks. We suspect that this degeneration might originate from two lines of problems: (1) the model design on sampling fewer videos and aligning visual/text graphs directly, which conflicts with the intuition that detailed spatial-temporal information and reasoning is indispensable for grounding indirect references; and (2) adopting large-scale pre-trained models directly to a specific domain is non-trivial, especially with challenges in grounding knowledge to visual signals. Overall, our experiments on the EgoTaskQA *indirect* split further reveals the demand for better spatial-temporal reasoning modules that solve the problem of compositional goal-oriented reasoning with indirect references.

4.4 Conclusions

We introduce the EgoTaskQA benchmark to systematically evaluate models' understanding of goal-oriented activities from an egocentric perspective. We annotate object states, relationships, and agents' beliefs on the LEMMA dataset. We generate diverse questions covering different reasoning capabilities and target the crucial dimensions of task understanding: action dependencies and effects, agents' intents and goals, and belief modeling. We evaluate state-of-the-art video question-answering models and show their gaps compared with the human on two challenging splits, *normal* and *indirect*, to promote future study on indirect reference understanding and goal-oriented reasoning. As the next steps, we plan to

investigate the following two branches in the future: (i) explicit spatial-temporal grounding for modularized video QA models and (ii) prompting large-scale pre-trained models (both visual and language) for the domain-specific video QA challenges. Firstly, egocentric data can provide finer information and ease the challenge of grounding in modularized neuro-symbolic models. This could complement existing video reasoning methods and test the potential of neuro-symbolic models on complex reasoning tasks from a real-world, multi-agent, and causal perspective. Next, with increasing efforts in adapting large-scale pre-trained models for reasoning, our experiments suggest that adopting such models directly to a specific domain is non-trivial. Compared to their capabilities in commonsense reasoning, how to enable pre-trained models with the ability to fastly adapt to complex reasoning tasks still remains an interesting problem to be solved.

Part II

Modeling Sequential Events

CHAPTER 5

Unsupervised Object-Centric Learning with Bi-level Optimized Query Slot Attention

Starting from this chapter, we focus on the problem of modeling sequential events. As discussed in Section 1.1 and Section 1.3, we believe the solution to fine-grained event understanding and reasoning lies in the usage of world model knowledge. However, such a design relies on the representation we adopt for world status. Therefore, the first crucial question to answer is how do we obtain representations with compositional syntax and semantics, especially given the challenge of disentangling concepts from visual stimuli with limited supervision? In fact, the ability to decompose complex natural scenes into meaningful object-centric abstractions lies at the core of human perception and reasoning. Objects, and their interactions, are the foundations of human cognition [SK07]. The endowment on making abstractions from perception and organizing them systematically empowers humans the ability to accomplish and generalize across a broad range of tasks, such as scene modeling [BFM20], visual reasoning [YGL20b], and simulating interactions [BFM20]. Therefore, we propose to use the unsupervised object-centric learning problem on the static image domain as a featured task to address the compositional representations required in modeling sequential events.

Motivated by the development of symbolic thought in human cognition, slot-based representations, instance [GVS17, GKK19, LWU20], sequential [GDG15, BMW19, EPP21, GLH21], or spatial [CP19, LWP20, JJD19], have been the key inductive bias to recent advances in unsupervised object-centric learning. Among them, the Slot-Attention module has received tremendous focus given its simple yet effective design [LWU20]. By leveraging the

iterative attention mechanism, Slot-Attention learns to compete between slots for explaining parts of the input, exhibiting a soft-clustering effect on visual signals. However, as revealed by recent studies, the Slot-Attention module comes with innate discrepancies for object-centric representation learning. First, with slots randomly initialized each time, the object-centric representations obtained by these models do not necessarily bind to object concepts [KEM22]. Intuitively, such randomness leads to undesired scenarios where slots with similar initializations compete for objects on different images. Such randomness challenges the iterative refinement procedure making it highly dependent on hyper-parameter tuning techniques.

To this end, we introduce an extension of the Slot-Attention module, BO-QSA, in this chapter to tackle the aforementioned problems. First, instead of sampling from a learnable Gaussian distribution, we propose directly learning the slot initializations as queries.With these learnable representations, we eliminate the ambiguous competitions between slots and provide a better chance for them to bind to specific object concepts. More importantly, we ease the difficulty in training Slot-Attention with learnable queries by formulating Slot-Attention as a bi-level optimization problem. Model-wise, we improve the training of query-initialized Slot-Attention with a straight-through gradient estimator (STE) by connecting our method with first-order approaches [FAL17, NS18, GZB21] in solving bi-level optimization problems. We provide experimental results to show that the proposed BO-QSA can achieve state-of-the-art results on both synthetic and real-world image datasets and validate the potential of our model for binding object concepts to slots with zero-shot transfer experiments. We hope these efforts can help foster new insights in the field of object-centric learning.

5.1 Related Work

Unsupervised Object-Centric Learning Our work falls into the recent line of research on unsupervised object-centric learning on images [GRB16, EHW16, GVS17, GKK19, BMW19, CP19, EKJ20, LWP20, BFM20, LWU20, ZKL21]. A thorough review and discussion on

this type of method can be found in [GVS20]. One critical issue of these methods is on handling complex natural scenes. [SDA21, LHG21] leverages a transformer-based decoder with Slot-Attention for addressing this problem. Similar attempts have also been made by exploiting self-supervised contrastive learning [CLR21, CTM21, WSH22, HKS22] and energy-based models [DLS21, YGW22]. Our work builds upon Slot-Attention by extending it with learnable queries and a novel optimization method for learning. Our compelling experimental suggests our model could potentially serve as a general plug-and-play module for a wider range of modalities where variants of Slot-Attention prosper [KEM22, EMS22, SWA22, YGW22, SDM22, SMP22].

Query Networks Sets of latent queries are commonly used in neural networks. These methods leverage permutation equivariant network modules (*e.g.* GNNs [SGT08] and attention modules [VSP17]) in model design for solving set-related tasks such as clustering [LLK19], outlier detection [ZKR17, ZHP19], *etc*. These learned latent queries have been shown to have good potential as features for tasks like contrastive learning [CMM20], object detection [CMS20], and data compression [JBA21, JGB21]. In contrast to the recent success of query networks in supervised or weakly-supervised learning [CMS20, ZGZ21, KEM22, EMS22, XDL22], [LWU20] demonstrates the detrimental effect of using independently initialized slots in Slot-Attention learning. However, we show that our BO-QSA method successfully overcomes this issue and generalizes the success of query networks to the domain of unsupervised object-centric learning.

Bi-level Optimization Our work is closely related to bi-level optimization methods with iterative fixed update rules for solving the inner objective. Specifically, methods are designed with implicit differentiation [AK17, BKK19] to stabilize the iterative update procedure. Similar formulations are also found when combined with meta-learning where [MKG21] train queries through recurrence in a meta-learning fashion and [RFK19] provides a unified view of the optimization problem with implicit gradients. Concurrent work from [CGL22]

formulate the Slot-Attention learning from an implicit gradient perspective with gradient-stopping derived from first-order hyper-gradient methods [GZB21]. However, they ignore the important role of slot initializations in generalization and concept binding. As our experiments suggest, such gradient stopping methods do not guarantee superior performance compared to the original Slot-Attention. We leave the details to Section 5.4.3 for an in-depth discussion.

5.2 Preliminaries

5.2.1 Object-Centric Representation Learning with Slot-Attention

Slot-Attention [LWU20] takes a set of N input feature vectors $\mathbf{x} \in \mathbb{R}^{N \times D_{\text{input}}}$ and maps them to a set of K output vectors (*i.e.*, slots) $\mathbf{s} \in \mathbb{R}^{K \times D_{\text{slots}}}$. It leverages an iterative attention mechanism to first map inputs and slots to the same dimension D with linear transformations $k(\cdot)$, $q(\cdot)$ and $v(\cdot)$ parameterized by ϕ^{attn}. At each iteration, the slots compete to explain part of the visual input by computing the attention matrix $\boldsymbol{A}$ with softmax function over slots and updating slots with the weighted average of visual values:

$$\tilde{\mathbf{s}} = f_{\phi^{\text{attn}}}(\mathbf{s}, \mathbf{x}) = \left(\frac{A_{i,j}}{\sum_{l=1}^{N} A_{l,j}} \right)^{\top} \cdot v(\mathbf{x}) \quad \text{where} \quad \boldsymbol{A} = \text{softmax}\left(\frac{k(\mathbf{x}) \cdot q(\mathbf{s})^{\top}}{\sqrt{D}} \right) \in \mathbb{R}^{N \times K}.$$

The slots are initialized from a learnable Gaussian distribution with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}$. They are refined iteratively within the Slot-Attention module by passing the updates into a Gated Recurrent Unit (GRU) [CVG14] and MLP parameterized by ϕ^{update} for T iterations:

$$\mathbf{s}^{(t+1)} = h_{\phi^{\text{update}}}(\mathbf{s}^{(t)}, \tilde{\mathbf{s}}^{(t)}), \quad \mathbf{s}^{0} \sim \mathcal{N}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma})), \quad \hat{\mathbf{s}} = \mathbf{s}^{(T)}. \tag{5.1}$$

The final prediction $\hat{\mathbf{s}}$ can be treated as the learned object-centric representation w.r.t. to input features $\mathbf{x}$. In the image domain, we take as input a set of images $\boldsymbol{I}$ and encode them with $f_{\phi^{\text{enc}}}$ to obtain features $\mathbf{x} \in \mathbb{R}^{HW \times D_{\text{input}}}$. After obtaining $\hat{\mathbf{s}}$ through the iterative refinement

procedure with $h_{\phi^{\text{update}}}$, images could be decoded from these object-centric representations with a mixture-based decoder or autoregressive transformer-based decoder. We provide details on the designs as follows:

Mixture-based Decoder The mixture-based decoder [WMB19] decodes each slot $\hat{\mathbf{s}}_i$ into an object image $\mathbf{x}_i$ and mask $\mathbf{m}_i$ with decoding functions $g^{\text{img}}_{\phi^{\text{dec}}}$ and $g^{\text{mask}}_{\phi^{\text{dec}}}$, which are implemented using CNNs. The decoded images and masks are calculated by:

$$\hat{\boldsymbol{I}}_i = g^{\text{img}}_{\phi^{\text{dec}}}(\hat{\mathbf{s}}_i), \quad \mathbf{m}_i = \frac{\exp\ g^{\text{mask}}_{\phi^{\text{dec}}}(\hat{\mathbf{s}}_i)}{\sum_{j=1}^{K} \exp\ g^{\text{mask}}_{\phi^{\text{dec}}}(\hat{\mathbf{s}}_j)}, \quad \hat{\boldsymbol{I}} = \sum_{i=1}^{K} \mathbf{m}_i \cdot \hat{\boldsymbol{I}}_i.$$

During training, a reconstruction objective is employed for supervising model learning. Despite its wide usage, mixture-based decoders showed limited capability at handling natural scenes with high visual complexity [SDA21].

Autoregressive Transformer Decoder Recently, [SDA21, SWA22] reveal the limitations of mixture decoder and leverage transformers and discrete VAE (dVAE)s [VV17, RPG21] for decoding slot-based object-centric representations. To obtain decoded images $\hat{\boldsymbol{I}}$, they learn a separate dVAE for first encoding $\boldsymbol{I}$ into a sequence of L tokens $\mathbf{z} = \{\mathbf{z}_1, \cdots, \mathbf{z}_L\}$ with dVAE encoder $f^{\text{dVAE}}_{\phi^{\text{enc}}}$. Next, they use a transformer decoder $g^{\text{transformer}}_{\phi^{\text{dec}}}$ to auto-regressively predict image tokens with learned slot representation $\hat{\mathbf{s}}$:

$$\mathbf{o}_l = g^{\text{transformer}}_{\phi^{\text{dec}}}(\hat{\mathbf{s}}; \mathbf{z}_{<l}) \quad \text{where} \quad \mathbf{z} = f^{\text{dVAE}}_{\phi^{\text{enc}}}(\boldsymbol{I}).$$

To train the entire model, we have the reconstruction objective supervising the learning of $\mathbf{z}$ with dVAE decoder $g^{\text{dVAE}}_{\phi^{\text{dec}}}$. Next, the objective for object-centric learning relies on the correct prediction from the auto-regressive transformer for predicting correct tokens:

$$\mathcal{L} = \mathcal{L}_{\text{dVAE}} + \mathcal{L}_{\text{CE}} \quad \text{where} \quad \mathcal{L}_{\text{dVAE}} = ||g^{\text{dVAE}}_{\phi^{\text{dec}}}(\mathbf{z}) - \boldsymbol{I}||_2^2, \ \mathcal{L}_{\text{CE}} = \sum_{l=1}^{L} \text{CrossEntropy}(\mathbf{z}_l, \mathbf{o}_l)$$

Under this setting, the model does not predict additional masks and relies on the attention $\boldsymbol{A}$ within the Slot-Attention module for obtaining slot-specific object masks. Although such models can achieve competitive results on real-world synthetic datasets, as our experiments suggest, they can be inferior to mixture-based decoders on segmentation in synthetic datasets. We suspect that this originates from the low resolution when discretizing images into tokens.

5.2.2 Bi-level Optimization with Fixed Point Iterations

The problem of bi-level optimization embeds the optimization of an inner objective within the outer objective. Normally, a bi-level optimization problem can be formulated as:

$$\min_{\theta,\phi} f(\theta,\phi) \quad s.t. \quad \theta \in \arg\min_{\theta'} g(\theta',\phi), \tag{5.2}$$

where we call $f(\theta,\phi)$ the outer objective function and $g(\theta,\phi)$ the inner objective function. To jointly optimize both objectives w.r.t. parameters θ and ϕ, a straightforward approach to solving Eq. (5.2) is to represent the inner solution of θ as a function of ϕ, *i.e.*, $\theta^*(\phi) = \arg\min_{\theta'} g(\theta',\phi)$. Then we can optimize the outer objective with gradient descent by approximating $\nabla_\phi f(\theta^*(\phi),\phi)$ as a function of ϕ. When the inner optimization objective could be solved by a fixed point iteration $\theta = F_\phi(\theta)$ [AK17, BKK19], the bi-level optimization problem could be solved by

$$\frac{\partial f(\theta^*(\phi),\phi)}{\partial \phi} = \frac{\partial f(\theta^*(\phi),\phi)}{\partial \theta^*} \cdot \sum_{i=0}^{\infty} \left(\frac{\partial F_\phi(\theta^*)}{\partial \theta^*}\right)^i \cdot \frac{\partial F_\phi(\theta^*)}{\partial \phi}. \tag{5.3}$$

For efficiency concerns, recent methods often use the first-order approximation of the infinite Neumann's series [SCH19, GZB21] for updating ϕ.

5.3 Bi-level Optimized Query Slot Attention

5.3.1 Query Slot Attention

As mentioned previously, the Slot-Attention module adopts a random initialization of slots and conducts iterative refinement to obtain object-centric representations $\hat{\mathbf{s}}$ as in Eq. (5.1). However, as argued by [KEM22], such random initializations provide no hint on the notion of object and no means for controllably probing concepts from the model. As shown by [CGL22], this random initialization plays a minimal role and could be detached from training. This indicates that the estimation of $\hat{\mathbf{s}}$ relies heavily on the task-specific iterative refining of slots over data, leaving a limited possibility for slots to bind to specific concepts and be leveraged as generalizable representations.

To address this issue, we focus on the Query Slot Attention (QSA), which initializes the slots in the Slot-Attention module with learnable queries $\mathbf{s}_0 = \phi^{\text{init}}$. Such a design is motivated by the success of recent query-based networks [VV17, JGB21]. It facilitates an object-centric model to learn general symbolic-like representations that could be quickly adapted by refining over task-specific requirements, as discussed in [KEM22]. Meanwhile, in contrast to the use of learnable queries in other encoder-decoder structures (*e.g.* dVAE), the slot initializations $\mathbf{s}_0$ are not necessarily required to encode image features since they were designed for separating them. This resembles recent discoveries in query networks [CMS20, YLL21] where queries could be generalizable probes for input properties.

5.3.2 Bi-level Optimization for Object-Centric Learning with Slot-Attention

Despite the good properties and potentials QSA presents, it is shown detrimental to initialize slots independently in Slot-Attention under unsupervised settings [LWU20]. We inspect from the bi-level optimization perspective to provide rationales for this discrepancy. Viewing from the bi-level optimization, Slot-Attention could be treated as solving for the following

objectives:

$$\min_{\mathbf{s},\Phi} \sum_{i=1}^{M} \mathcal{L}(\mathbf{x}_i, \mathbf{s}_i, \Phi) \quad s.t. \quad \mathbf{s}_i^* = \arg\min_{\mathbf{s}} \mathcal{L}_{\text{cluster}}(\mathbf{x}_i, \mathbf{s}, \Phi), \tag{5.4}$$

where $\mathbf{x}_i$ and $\mathbf{s}_i$ denote the input feature from the i-th image and its corresponding slot features, and $\Phi = \{\phi^{\text{init}}, \phi^{\text{attn}}, \phi^{\text{update}}\}$ denotes parameters for assigning input features $\mathbf{x}$ to different slots. Under this setting, the outer objective $\mathcal{L}$ could be for reconstruction or set prediction [LWU20] depending on the task. The inner objective could be viewed as a soft-clustering objective [LWU20] $\mathcal{L}_{\text{cluster}} = -\sum_p \sum_q \mathcal{K}_\Phi(\mathbf{x}_i^p, \mathbf{s}_q)$, where $\mathcal{K}_\Phi(\cdot,\cdot)$ denotes a pseudo distance measure defined by attention for pixel-slot similarity. In Slot-Attention, the inner objective is solved by iterative refinement, which could be formulated as solving for fixed-points [CGL22] of

$$\mathbf{s} = h_{\phi^{\text{update}}}(\mathbf{s}, \tilde{\mathbf{s}}) = h_{\phi^{\text{update}}}(\mathbf{s}, f_{\phi^{\text{attn}}}(\mathbf{s}, \mathbf{x})) = F_\Phi(\mathbf{s}, \mathbf{x}), \tag{5.5}$$

where $F_\Phi(\cdot,\cdot)$ is some fixed-point operation. This procedure could be viewed as finding optimal solutions to an objective. As introduced by [CGL22] in Implicit Slot-Attention (ISA), by leveraging Eq. (5.3), the instabilities through the iterative updates could be avoided by detaching gradients, treating slots in the final iteration as an approximation of $\mathbf{s}_i^*$, and computing first-order gradient approximations for updating Φ with $\mathbf{s}_i^*$. However, we demonstrate in Table 5.7 that this design is only beneficial for randomly initialized slots and detrimental for query-initialized Slot-Attention architectures since it relies heavily on the good approximation of the solution to the inner objective. With no randomness in slot initializations or gradient during training, starting from a fixed set of initialization points puts challenges on the learning of Slot-Attention update F_Φ as it will be difficult to provide a good approximation of s_i^* with only a fixed number of iterations (see in Section 5.4.4). This urges the need for information flow to the slot initializations for better fixed-point approximation.

Algorithm 2: BO-QSA

Input: input features `input`, learnable queries `init`, number of iterations T
Output: object-centric representation `slots`
Modules: stop gradient module SG$(\cdot)$, slot attention module SA$(\cdot, \cdot)$
`slots` $=$ `init`
for $t = 1, \cdots, T$ **do**
  `slots` $=$ SA(`slots`, `inputs`)
`slots` $=$ SG(`slots`) $+$ `init` - SG(`init`)
`slots` $=$ SA(`slots`, `inputs`)
return *slots*

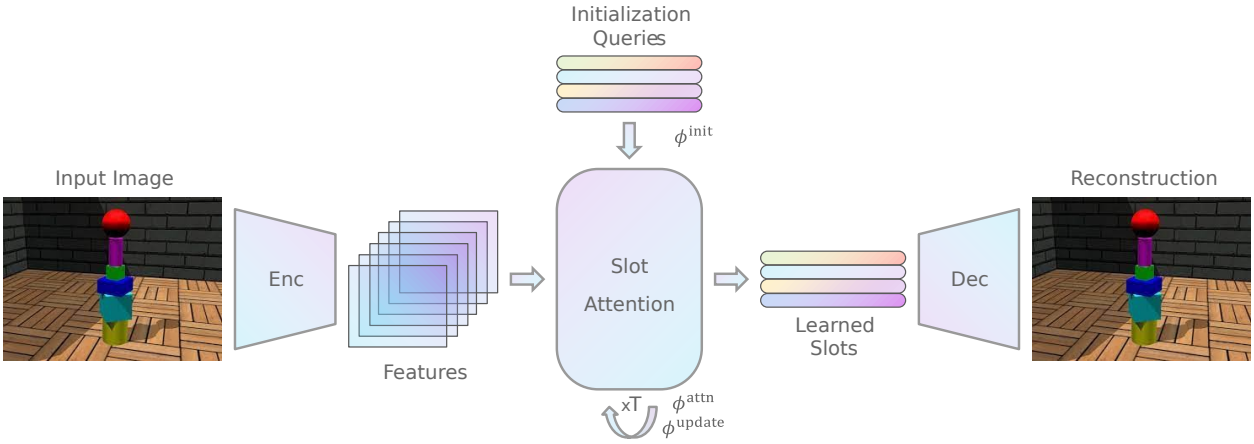


Figure 5.1: An illustrative visualization of our proposed BO-QSA slot-encoder.

5.3.3 Bi-level Optimized Query Slot Attention

We propose BO-QSA to address the learning problem of QSA. As shown in Algorithm 2 and Fig. 5.1, we initialize slots with learnable queries in BO-QSA and perform T steps of Slot-Attention update to obtain an approximation of $\mathbf{s}_i^*$. These near-optimal solutions of the inner objective are passed into one additional Slot-Attention step where gradients to all previous iterations are detached. In contrary to ISA, we use a STE [BLC13, VV17] to backpropagate gradients and also to slot initialization queries. Such designs help find good starting points for the inner optimization problem on clustering, alleviating the problem of bi-level optimization with QSA mentioned in Section 5.3.2. Similar to dVAE, the STE adds bias to the gradient of the initialization queries. However, since these learnable queries are

meant for disentangling image features, they do not have to maintain information about the approximated $\mathbf{s}^*$. Such bias could lead to learned queries which are better pivots for separating different image features, similar to anchors or filter queries learned for different tasks [CMS20, ZGZ21]. Note that we do not add constraints on the consistency between $\mathbf{s}_0$ and $\hat{\mathbf{s}}$ (*e.g.* $||sg(\hat{\mathbf{s}}) - \mathbf{s}_0||^2$) as done in dVAE since we find such constraints lead to a mean-representation of datasets that forbids better concept binding (see in Section 5.4.4). As shown in Table 5.7 and Fig. 5.4, our learned slot initialization queries do fulfill this goal by providing a more separable initialization space and can significantly facilitate model learning.

5.4 Experiments

In this section, we aim to address the following questions with our experimental results:

- How good is our proposed BO-QSA on both synthetic and complex natural scenes?
- How important is the query and the optimization method in BO-QSA?
- Does BO-QSA possess the potential for concept binding and zero-shot transfer?

Here we clarify the datasets and metrics selected for evaluating our model on each domain:

Synthetic Domain For the synthetic domain, we select three well-established challenging multi-object datasets Shapestacks [GFP18], ObjectsRoom [KBM19], and CLEVRTEX for evaluating our BO-QSA model. Specifically, we consider three metrics to evaluate the quality of object segmentation and reconstruction. Adjusted Rand Index (ARI) [HA85] and Mean Segmentation Covering (MSC) [EKJ20] for segmentation and Mean Squared Error (MSE) for reconstruction. Following the evaluation setting of recent works, we report the first two segmentation metrics over foreground objects (ARI-FG and MSC-FG).

Real-world Images For the real image domain, we use two tasks (1) unsupervised foreground extraction and (2) unsupervised multi-object segmentation for evaluating our

Table 5.1: Multi-object segmentation results on ShapeStacks and ObjectsRoom. We report ARI-FG and MSC-FG of all models with (mean ± variance) across 3 experiment trials. We visualize the best results in bold.

Model	ShapeStacks		ObjectsRoom	
	↑ ARI-FG	↑ MSC-FG	↑ ARI-FG	↑ MSC-FG
MONet-G [BMW19]	0.70±0.04	0.57±0.12	0.54±0.00	0.33±0.01
GENESIS [EKJ20]	0.70±0.05	0.67±0.02	0.63±0.03	0.53±0.07
Slot-Attention [LWU20]	0.76±0.01	0.70±0.05	0.79±0.02	0.64±0.13
GENSIS-V2 [EPP21]	0.81±0.01	0.67±0.01	0.86±0.01	0.59±0.01
SLATE [SDA21]	0.65±0.03	0.63±0.05	0.57±0.03	0.30±0.03
Ours (transformer)	0.68±0.02	0.70±0.02	0.68±0.03	0.72±0.03
Ours (mixture)	**0.93±0.01**	**0.89±0.00**	**0.87±0.03**	**0.80±0.02**

method. Specifically, we select Stanford Dogs [KJY11], Stanford Cars [KSD13], CUB200 Birds [WBM10], and Flowers [NZ10] as our benchmarking datasets for foreground extraction and YCB [CSB17], ScanNet [DCS17], COCO [LMB14] proposed by [YY22] for multi-object segmentation. We use mean Intersection over Union (mIoU) and Dice as metrics for evaluating the quality of foreground extraction and use the evaluation metrics adopted by [YY22] for multi-object segmentation.

5.4.1 Object Discovery on Synthetic Datasets

Experimental Setup We explore our proposed BO-QSA with two types of decoder designs, mixture-based and transformer-based, as discussed in Section 5.2.1. We follow the decoder architecture in Slot-Attention [LWU20] for mixture-based decoders and SLATE [SDA21] for transformer-based decoders. For both types of models, we use the Slot-Attention module with a CNN image encoder and initialize slots with learnable embeddings.

Results We report multi-object segmentation results on synthetic datasets in Table 5.1 and visualize qualitative results in Fig. 5.2. As shown in Table 5.1, our BO-QSA achieves the state-of-the-art results with large improvements over previous object-centric learning

Table 5.2: Multi-object segmentation results on CLEVRTEX. We report ARI-FG and MSE of all models in the form of (mean ± variance) across 3 experiment trials. We visualize the best results in bold.

Model	CLEVRTEX-FULL		CLEVRTEX-OOD		CLEVRTEX-CAMO	
	ARI-FG (%) ↑	MSE ↓	ARI-FG (%) ↑	MSE ↓	ARI-FG (%) ↑	MSE ↓
MONet [BMW19]	19.78±1.02	**146±7**	37.29±1.04	409±3	31.52±0.87	265±1
Slot-Attention [LWU20]	62.40±2.33	254±8	58.45±1.87	487±16	57.54±1.01	**215±7**
GENSIS-V2 [EPP21]	31.19±12.41	315±106	29.04±11.23	539±147	29.60±12.84	278±75
DTI [MVP21]	79.90±1.37	438±22	73.67±0.98	590±4	**72.90±1.89**	377±17
Ours (mixture)	**80.47±2.49**	268±2	**86.50±0.19**	**265±25**	63.71±6.11	280±7

Table 5.3: Reconstruction results between mixture-based and transformer-based decoders.

Model	ShapeStacks	ObjectsRoom
Slot-Attention (mixture)	80.8	20.4
ours (mixture)	**72.0**	**8.1**
SLATE (transformer)	52.3	16.3
ours (transformer)	**49.3**	**14.7**

methods on all metrics in ShapeStacks and ObjectsRoom. We also observe more stable model performance, *i.e.* smaller variances in results, across different trials of experiments. Our model with mixture-based decoders obtains the best overall performance on all datasets. More specifically, our mixture-based BO-QSA significantly outperforms the vanilla Slot-Attention model (∼15%) with minimal architectural differences. This validates the importance of the learnable queries and our optimization method. We will continue this discussion in Section 5.4.3. As shown in Table 5.2, our model also achieves state-of-the-art results on the unsupervised object segmentation task in CLEVRTEX with consistent improvement over Slot-Attention on the CAMO and OOD generalization split. Interestingly, our model (1) shows larger reconstruction errors, (2) generalizes well in out-of-distribution scenarios, and (3) shows marginal improvement in camouflaged images. We attribute (1) and (3) to the simple architecture of encoders/decoders currently adopted and provide insights on (2) in Section 5.4.4.

Mixture-based vs. Transformer-based Decoder We observe inferior segmentation but superior reconstruction performance of transformer-based variants of Slot-Attention on

Table 5.4: Unsupervised multi-object segmentation results on YCB, ScanNet, and COCO.

Model	YCB (AP / PQ / Pre / Rec) ↑	ScanNet (AP / PQ / Pre / Rec) ↑	COCO (AP / PQ / Pre / Rec) ↑
AIR	0.0(0.1)/0.6(0.3)/1.1(0.4)/0.8(0.2)	2.7(1.4)/6.3(1.7)/15.6(2.8)/7.3(1.6)	2.7(0.1)/6.7(0.5)/14.3(2.6)/8.6(0.8)
MONet	3.1(1.6)/7.0(2.6)/9.8(3.6)/1.2(0.8)	24.8(1.6)/24.6(1.6)/31.0(1.6)/40.7(1.8)	11.8(2.0)/12.5(1.1)/16.1(0.9)/21.9(1.7)
IODINE	1.8(0.2)/3.9(1.3)/6.2(2.0)/7.3(1.9)	10.1(2.9)/13.7(2.7)/18.6(4.2)/24.4(3.8)	4.0(1.2)/6.3(1.2)/9.9(1.8)/10.8(2.0)
Slot-Attention	9.2(0.4)/13.5(0.9)/20.0(1.3)/26.2(6.8)	5.7(0.3)/9.0(1.5)/12.4(2.5)/18.3(2.7)	0.8(0.3)/3.5(1.2)/5.3(1.7)/7.3(2.2)
Ours (transformer)	**47.96 (1.84)**/**34.81 (1.30)**/**50.83 (1.05)**/**53.60 (0.68)**	**28.50 (2.36)**/**26.37 (2.01)**/**37.30 (2.00)**/**42.37 (1.91)**	**17.77 (0.61)**/**17.60 (0.64)**/**25.29 (0.61)**/**30.58 (0.87)**

synthetic datasets. Specifically, we compare the MSE of models on ShapeStacks and ObjectsRoom. As shown in Table 5.3, transformer-based methods provide better reconstruction results. We attribute the low segmentation performance to mask prediction in these methods, which relies on the attention matrix computed over input features. This leads to coarse object masks as a result of image tokenization. Nonetheless, we observe consistent improvement by applying our slot encoder to both mixture and transformer decoders.

5.4.2 Object Discovery on Real Datasets

Experimental Setup For real-world experiments, we use the same slot encoder design used in Section 5.4.1 with a 4-layer CNN image encoder and initialize slots with learnable queries. For unsupervised foreground extraction, we follow [YXM21] and report the best model performance on all datasets. During the evaluation, we select the slot's mask prediction that has a maximum intersection with the ground-truth foreground mask as our predicted foreground. For unsupervised multi-object segmentation, we follow [YY22] and report the models' performance on all datasets across trials with different random seeds.

Results We show quantitative experimental results in Table 5.5 and Table 5.4. We also visualize qualitative results in Fig. 5.2. For multi-object segmentation, as shown in Table 5.4, our model outperforms existing object-centric learning baselines by a large margin, especially on the YCB dataset where the segmented objects have clear semantic meanings. For foreground extraction, as shown in Table 5.5, our method significantly outperforms all existing baselines on the task of foreground extraction, achieving new state-of-the-art on all datasets. We recognize the discrepancy of mixture-based decoders in both Slot-Attention and

Table 5.5: Unsupervised foreground extraction results on CUB200 Birds (Birds), Stanford Dogs (Dogs), Stanford Cars (Cars), and Caltech Flowers (Flowers).

Model	Birds		Dogs		Cars		Flowers	
	↑ IoU	↑ Dice	↑ IoU	↑ Dice	↑ IoU	↑ Dice	↑ IoU	↑ Dice
ReDO [CAD19]	46.5	60.2	55.7	70.3	52.5	68.6	76.4	-
IODINE [GKK19]	30.9	44.6	54.4	67.0	51.7	67.3	-	-
OneGAN [BW20]	55.5	69.2	71.0	81.7	71.2	82.6	-	-
Slot-Attention [LWU20]	35.6	51.5	39.6	55.3	41.3	58.3	30.8	45.9
[VMB20]	68.3	-	-	-	-	-	54.0	-
DRC [YXM21]	56.4	70.9	71.7	83.2	72.4	83.7	-	-
[MRL21]	66.4	-	-	-	-	-	54.1	-
SLATE [SDA21]	36.1	51.0	62.3	76.3	75.5	85.9	68.1	79.1
Ours (mixture)	25.1	39.2	36.8	53.6	69.1	81.5	36.1	51.6
Ours (transformer)	**71.0**	**82.6**	**82.5**	**90.3**	**87.5**	**93.2**	**78.4**	**86.1**

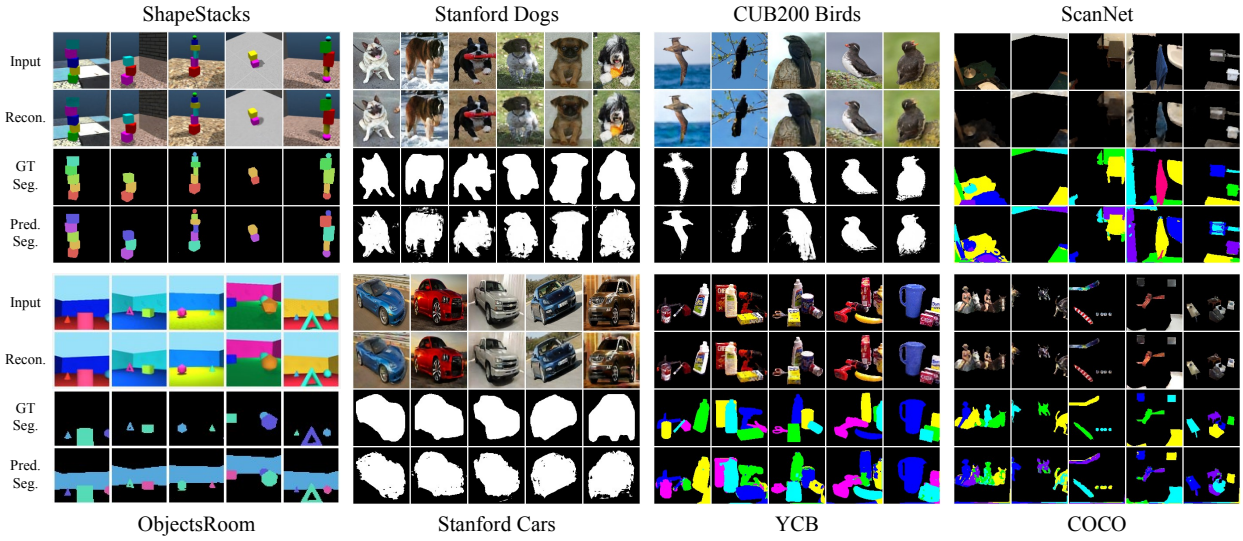


Figure 5.2: Visualization of our segmentations and reconstructions on synthetic and real-world images.

our mixture-based design in modeling real-world images, reflecting similar discoveries from recent works [SDA21] that mixture-based decoder struggles in modeling real-world images. On the other hand, our transformer-based model shows significant improvements over the vanilla version. Notably, our method outperforms a broad range of models, including GAN-based generative models (*i.e.* OneGAN, [VMB20]), and large-scale pre-trained contrastive methods

Table 5.6: Unsupervised segmentation results compared with contrastive learning methods which are pre-trained on ImageNet.

Model	Birds
MoCo v2 [CFG20]	63.5
BYOL [GSA20]	56.1
R2O [GKL22]	**71.2**
ours (BO-QSA+transformer)	**71.0**

Table 5.7: Ablative experiments on slot initialization and optimization methods. We visualize the best results in bold and underline the second-best results.

Method	Dogs		ShapeStacks	
	↑ IoU	↑ Dice	↑ ARI-FG(%)	↑ MSC-FG(%)
SA*	71.0	81.9	86.7	<u>84.8</u>
I-SA	80.8	89.2	<u>88.3</u>	76.8
BO-SA	<u>80.9</u>	<u>89.3</u>	87.7	66.6
QSA	64.5	72.9	88.1	76.1
I-QSA	59.3	77.6	84.6	81.8
BO-QSA (ours)	**82.5**	**90.3**	**92.9**	**89.2**

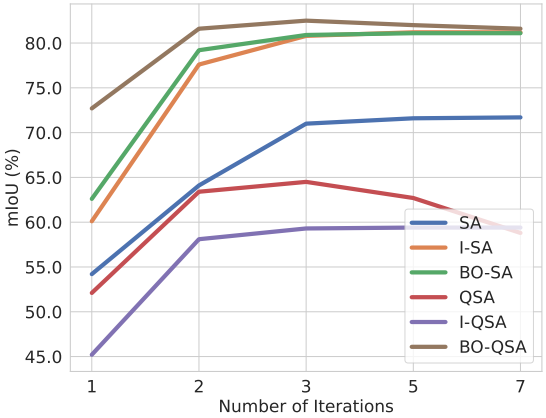


Figure 5.3: Effects of iterative updates in testing.

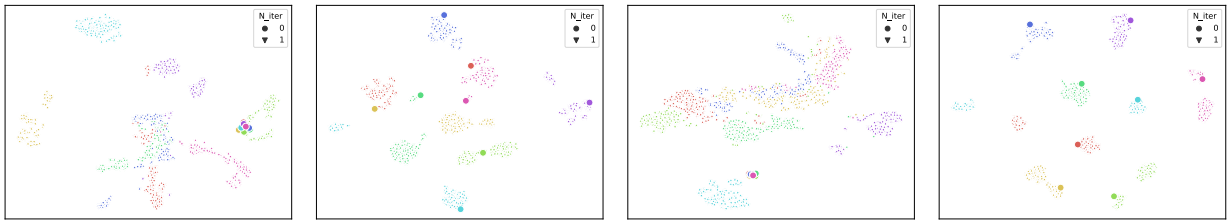


Figure 5.4: Visualization of learned slot initializations and post-iteration slots after the first iteration of Slot-Attention on ShapeStacks.

(*i.e.* MoCo-v2, BYOL, R2O). As shown in Table 5.6, our method achieves comparable results with state-of-the-art self-supervised contrastive learning methods without large-scale pre-training and data augmentation. This result sheds light on the potential of object-centric learning as a pre-training task for learning general visual representations.

5.4.3 Ablative Studies

Experimental Setup We perform ablative studies over our designs by comparing them with different design variants on ShapeStacks and Stanford Dogs. For slot initialization, we consider (1) the original Slot-Attention module's sampling initialization (SA), and (2) initializing with learnable queries (QSA). For optimization, we consider (1) the original optimization in Slot-Attention (*i.e.* w/o detach or STE), (2) the ISA optimization where gradients to slots in iterative updates are detached (*i.e.* w/ detach only), and (3) our optimization where we both detach the gradients into iterative refinement, and pass gradient to the initialization queries with STE (*i.e.* w/ detach and STE). For simplicity, we term these variants with prefixes (I-) for ISA and (BO-) for our full method. We run all ablations on each dataset with the same encoder-decoder architecture.

Results We show experimental results in Table 5.7 and Fig. 5.3. First, from Table 5.7, we observe that BO-QSA significantly outperforms other variants. For sample-based slot initializations, our method shows a similar effect compared with ISA on improving Slot-Attention learning. For query-based slot initializations, we validate the difficulty in training query-based Slot-Attention with its inferior performance. We further show the ineffectiveness of ISA for query-based Slot-Attention. The experiments on query-based Slot-Attention prove that both of our design choices are necessary and effective for superior performance. To study the effect of learned queries, we visualize in Fig. 5.3 where we set different numbers of iterative updates of Slot-Attention during inference on the Stanford Dogs dataset. We can see that our BO-QSA significantly outperforms other variants with only one iteration. This indicates that our query-based design can help ease training difficulties. In Fig. 5.4, we further visualize the learned initializations and post-iteration slots in the same feature space using t-SNE [MH08]. Our initializers provide a more separable space when differentiating image features, which validates the desired model behaviors mentioned in Section 5.3.3.

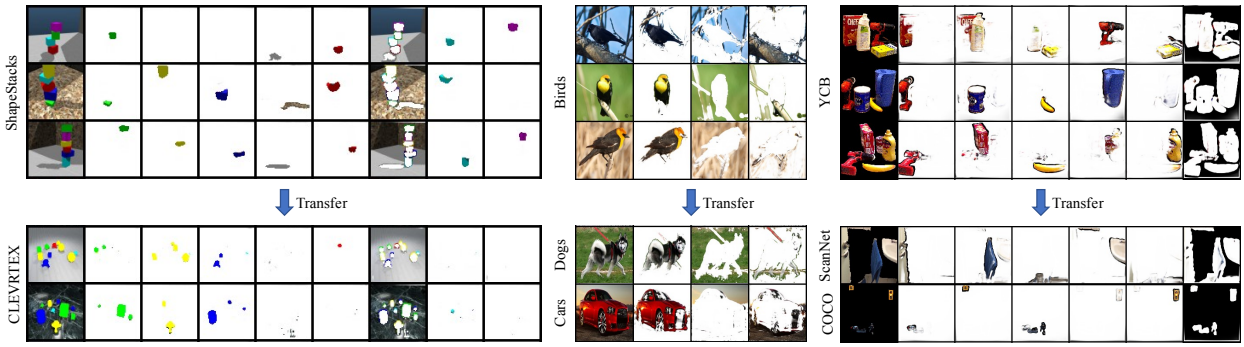


Figure 5.5: Visualization of learned concepts and attention maps in zero-shot transfer.

Table 5.8: Zero-shot transfer results of unsupervised multi-object segmentation on real images.

Model	YCB → ScanNet (AP / PQ / Pre / Rec)	YCB → COCO (AP / PQ / Pre / Rec)	ScanNet → YCB (AP / PQ / Pre / Rec)	ScanNet → COCO (AP / PQ / Pre / Rec)	COCO → YCB (AP / PQ / Pre / Rec)	COCo → ScanNet (AP / PQ / Pre / Rec)
SA	1.37/4.90/11.27/6.35	1.20/4.97/10.48/6.73	19.63/19.24/28.56/31.43	12.84/ 14.86/22.06/26.74	26.53/23.05/35.96/38.12	20.99/22.08/32.14/36.53
I-SA	21.62/21.81/32.32/34.19	18.39/18.47/27.23/30.38	18.66/18.56/28.97/30.82	11.83/14.14/20.70/25.42	26.72/22.90/35.89/37.98	19.34/20.00/29.44/33.18
BO-QSA(ours)	**28.24/25.93/36.68/39.62**	**24.23/21.65/30.20/35.79**	**21.85/19.96/31.51/33.45**	**13.95/16.04/23.35/28.49**	**31.21/25.44/38.90/41.35**	**24.21/23.59/34.07/38.49**

5.4.4 Additional Analyses

In this section, we provide additional analyses on the potential of our BO-QSA as a concept binder for generalizing to new examples. First, we qualitatively visualize our learned content for each slot (without additional clustering) in ShapeStacks, Birds, and YCB in Fig. 5.5. We observe high similarity within the learned content of each slot, indicating similar concepts learned by specific slots. This shows the potential of the slots in our BO-QSA for binding specific concepts on object properties (*e.g.* colors, contours, and spatial positions). Although we can not control which concepts to learn, these results are important indicators that our learned initialization queries could potentially be generalizable concept probes. We further provide quantitative evaluations where we use models trained on dataset X for zero-shot inference on dataset Y. For unsupervised multi-object segmentation, we report transfer results from ScanNet and COCO to all other real-image multi-object segmentation datasets in addition to the results on YCB. As shown in Table 5.8, our model shows consistent improvement over Slot-Attention and ISA during zero-shot transfer. For unsupervised foreground extraction, we report transfer results from Stanford Dogs and CUB200 Birds to all other real-image foreground extraction datasets. As we can see from Table 5.9, our

Table 5.9: Zero-shot transfer results on unsupervised foreground extraction (mIoU ↑).

Model	Dogs → Cars	Dogs → Flowers	Dogs → Birds	Birds → Dogs	Birds → Cars	Birds → Flowers
SA	57.96	57.96	45.06	74.68	58.79	62.02
I-SA	58.05	58.06	48.88	71.16	69.90	68.67
BO-SA	58.10	58.10	47.96	71.81	**70.75**	67.95
BO-QSA(ours)	**75.50**	**63.43**	**52.49**	**76.66**	66.74	**70.74**

model achieves the overall best results compared with other powerful Slot-Attention variants (models that achieve best or second-best results in our ablation studies as in Table 5.7 except for (Birds→Cars). However, our optimization method still helps improve zero-shot transfer for randomly initialized Slot-Attention.

Fixed-point approximation We further study whether a fixed point $\mathbf{s}^*$ could be reached by a fixed number of iterations during training as described in Section 5.3.2. Since we hypothesized that the low performance of I-QSA in Section 5.4.3 originated from the insufficient number of starting points for fixed-point approximation, we conduct experiments on increasing the number of Slot-Attention iterations during training for I-QSA on the Dog dataset. As shown in Table 5.10, increasing the number of Slot-Attention iterations during training for I-QSA significantly improves its performance. However, we found that adding more iterations after a threshold (*i.e.* 7 in this case) does not further improve the overall performance. This verifies the need for learning slot initialization vectors for better approximating the fixed point solution of the inner soft-clustering objective in Slot-Attention.

Table 5.10: Increasing the number of iterations during training for I-QSA.

Model	# of Training Iterations	Dogs			
		↑ IoU	Gain	↑ Dice	Gain
I-QSA	3	59.3	-	77.6	-
I-QSA	7	80.5	+35.8%	88.9	+14.6%
Ours	3	**82.5**	-	**90.3**	-

Design choices on slot initialization As described in Section 5.3.3, our method is connected with recent works on dVAE. However, we do not require the initialization queries

Table 5.11: Comparison between update methods for slot-initialization queries.

Metrics	RunningMean	RunningMean-M	KMeans	KMeans-M	VQ-constraint	Ours
ARI-FG (ShapeStacks)	7.5	51.4	21.0	70.6	88.6	**92.9**
MSC-FG (ShapeStacks)	3.7	15.4	4.2	60.4	85.3	**89.2**

to maintain information about the post-iteration slots $\hat{\mathbf{s}}$ as we found such constraints lead to the learning of the mean-representation of datasets which forbids disentanglement and concept binding. In this section, we provide experimental results to verify this argument. Specifically, we consider three different ways to update slot initialization queries in addition to our proposed method: 1) using the running mean of the post-iteration slots as initialization queries (RunningMean), 2) running K-Means clustering on post-iteration slots and updating the initialization queries using re-clustered centers by Hungarian matching (KMeans), 3) adding consistency loss between initialization queries and post-iteration slots as done in VQ-VAE (VQ-constraint). For (1) and (2), we empirically found such designs to be suffering from frequent updates and therefore use momentum updates to stabilize their training. We term these variants with the suffix (-M).

As shown in Table 5.11, our model achieves the best overall performance compared to other initialization methods. Specifically, we found that using the running mean of post-iteration slots or K-Means cluster centers re-clustered from post-iteration slots to be harmful to model performance. We attribute this effect to the learning of the mean-representation of datasets. This is further proved in experiments with VQ-VAE loss on consistency between slot initializations and post-iteration slots (*i.e.* $||\text{sg}(\hat{\mathbf{s}}) - \mathbf{s}_0||^2$), where the VQ-constraint variant showed inferior performance. We also found that the weight of this additional loss needs to be carefully tuned for the model to decompose objects. Empirically, most configurations of this hyperparameter will lead to bad reconstructions except for certain small weights (*e.g.* 0.01 reported here). Above all, we believe these experimental results verify the effectiveness of our design choices on initialization query learning. We provide additional visualizations on the learned contents of slots for each update method in Fig. 5.6.

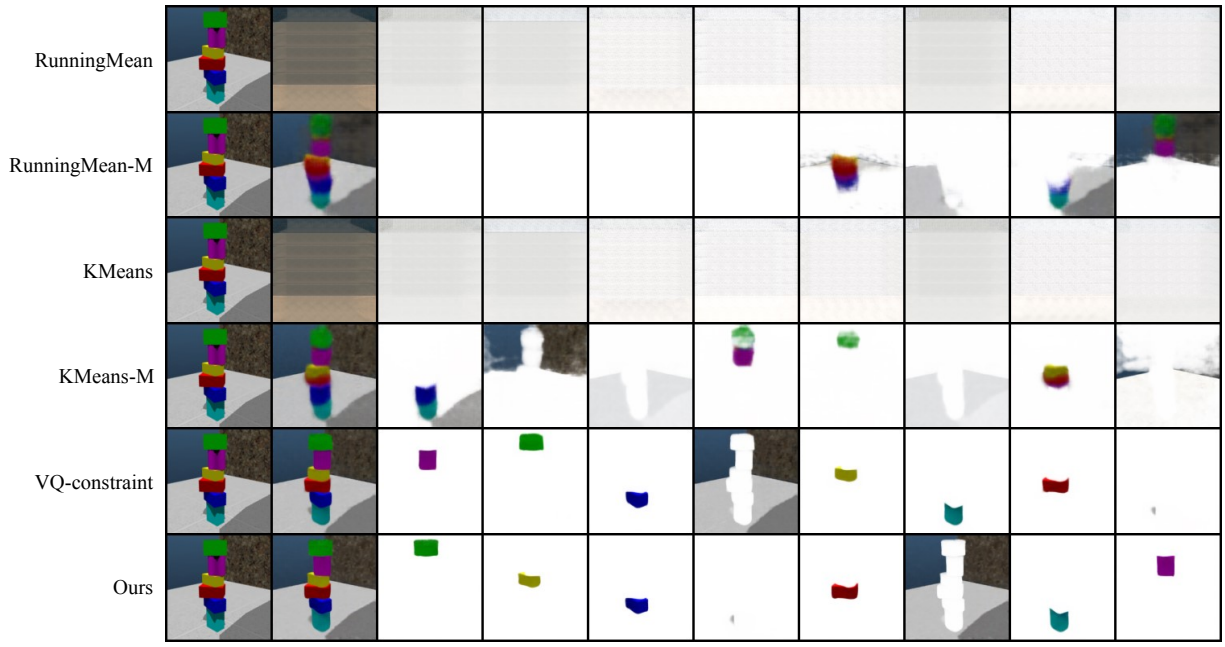


Figure 5.6: Visualizations per-slot reconstruction for different update methods.

5.5 Conclusions

We introduce BO-QSA for unsupervised object-centric representation learning. We initialize Slot-Attention with learnable queries, and combine bi-level optimization and straight-through gradient estimators to ease the difficulty in query-based Slot-Attention learning. With simple code adjustments on Slot-Attention, we obtain state-of-the-art model for unsupervised object segmentation in both synthetic and natural image domains, outperforming previous baselines by a large margin. More importantly, our learned model exhibits concept-binding effects where visual concepts are attached to specific slot queries. With a fixed number of initialized slots, our model is limited to handling a fixed maximum number of objects in the inputs. However, our queries could be learned to bind object attributes, which leads to meaningful segmentation of images by grouping similar properties (*e.g.* color, position, *etc.*). As a future direction, this connects our method with weakly-supervised contrastive learning methods that learn grounded visual representations with language.

CHAPTER 6

Abstract Spatial-Temporal Reasoning via Probabilistic Abduction and Execution

With the recent culmination of unsupervised object-centric learning, we have gone through powerful variants of models proposed for 3D scenes [YGW22, SDM22] and videos [KEM22, EMS22, SWA22]. With more works starting to tackle the concept binding problem discussed in Chapter 5, we continue the discussion on leveragin these symbolic representations for the problem of sequential event modeling. In this chapter, we take the spatial-temporal reasoning in RAVEN proposed in Chapter 2 as the central topic of our discussion. Spatial-temporal reasoning is a challenging task due to its demanding but unique nature: a theoretic requirement on *representing* and *reasoning* based on spatial-temporal knowledge in mind, and an applied requirement on a high-level cognitive system capable of *navigating* and *acting* in space and time. Despite the encouraging progress on RPM that achieves human-level performance in terms of accuracy, modern approaches have neither a treatment of human-like reasoning on generalization nor a potential to generate answers. Viewing from the cognitive perspective, psychologists call for weak attribute supervision in RPM. As isolated Amazonians, absent of schooling on primitive attributes, could still correctly solve RPM [DIP06, IPS11], an ideal computational counterpart should be able to learn it *in absent of visual attribute annotations*. This weakly-supervised setting introduces unique challenges: How to jointly learn these visual attributes given only ground-truth images? With uncertainties in perception, how to abduce hidden logic relations from it? How about executing the symbolic logic on inaccurate perception to derive answers?

To support cross-configuration generalization and answer generation, we move a step further towards a neuro-symbolic model with explicit logical reasoning and human-like generative problem-solving while addressing the challenges. Specifically, we propose the *Probabilistic Abduction and Execution (PrAE)* learner; central to it is the process of abduction and execution on the probabilistic scene representation. Inspired by Fodor, Marcus, and neuro-symbolic reasoning [HMG19, MGK19, YGL20a, YWG18], the PrAE learner disentangles the previous monolithic process into two separate modules: a neural visual perception frontend and a symbolic logical reasoning backend. The neural visual frontend operates on object-based representation [HMG19, KSM17, MGK19, YGL20a, YWG18] and predicts conditional probability distributions on its attributes. A scene inference engine then aggregates all object attribute distributions to produce a probabilistic scene representation for the backend. The symbolic logical backend abduces, from the representation, hidden rules that govern the time-ordered sequence via inverse dynamics. An execution engine executes the rules to *generate* an answer representation in a probabilistic planning manner [GNT04, HXZ19, KKL15], instead of directly making a categorical choice among the candidates. The final choice is selected based on the divergence between the generated prediction and the given candidates. The entire system is trained end-to-end with a cross-entropy loss and a curricular auxiliary loss [SHB18, ZGJ19, ZJG19] *without* any visual attribute annotations. Fig. 6.1 compares the proposed PrAE learner with prior methods.

The unique design in PrAE connects perception and reasoning and offers several advantages: (i) With an intermediate probabilistic scene representation, the neural visual perception frontend and the symbolic logical reasoning backend can be *swapped* for different task domains, enabling a greater extent of module reuse and combinatorial *generalization*. (ii) Instead of blending perception and reasoning into one monolithic model without any explicit reasoning, probabilistic abduction offers a more *interpretable* account for reasoning on a logical representation. It also affords a more detailed analysis of both perception and reasoning. (iii) Probabilistic execution permits a *generative* process to be integrated into the system.

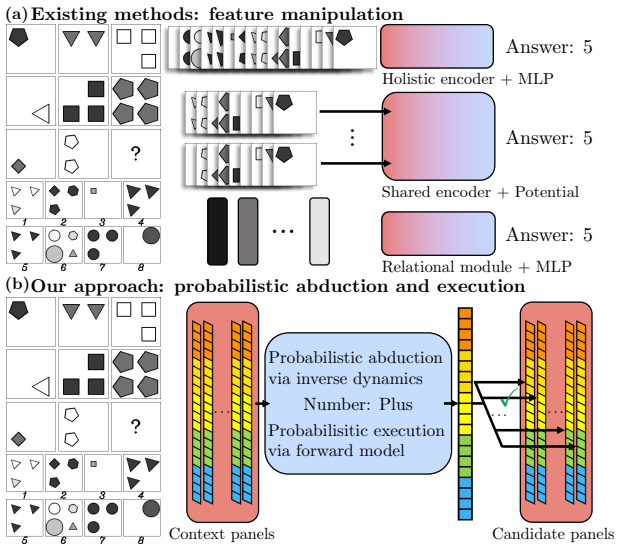


Figure 6.1: Differences between (a) prior methods and (b) the proposed approach.

Symbolic logical constraints can be transformed by the execution engine into a forward model [JR92] and applied in a probabilistic manner to predict the final scene representation, such that the entire system can be trained by analysis-by-synthesis [CHY19, Gre76, HNF19, HQX18, HQZ18, LB14, WTK17, WWX17, XLZ16, XZW19, YK06, ZWM98]. (iv) Instead of making a deterministic decision or drawing limited samples, maintaining probabilistic distributions brings in extra robustness and fault tolerance and allows gradients to be easily propagated.

In this chapter, we provide a detailed review of prior methods for spatial-temporal reasoning in the symbolic domain. We further elaborate on the proposed Probabilistic Abduction and Execution (PrAE) learner, which, unlike previous methods, can disentangle perception and reasoning from a monolithic model with the reasoning process realized by abduction and execution on a probabilistic scene representation. Finally, we provide experimental results to demonstrate that the PrAE learner achieves better generalization results compared to existing methods in the cross-configuration generalization task of RPM.

6.1 Related Work

Neuro-Symbolic Visual Reasoning Neuro-symbolic methods have shown promising potential in tasks involving an interplay between vision and language and vision and causality. Qi *et al*. [QJH20, QJZ18] showed that action recognition could be significantly improved with the help of grammar parsing, and Li *et al*. [LHH20] integrated perception, parsing, and logic into a unified framework. Of particular relevance, Yi *et al*. [YWG18] first demonstrated a prototype of a neuro-symbolic system to solve VQA [AAL15], where the vision system and the language parsing system were separately trained with a final symbolic logic system applying the parsed program to deliver an answer. Mao *et al*. [MGK19] improved such a system by making the symbolic component continuous and end-to-end trainable, despite sacrificing the semantics and interpretability of logic. Han *et al*. [HMG19] built on [MGK19] and studied the metaconcept problem by learning concept embeddings. A recent work investigated temporal and causal relations in collision events [YGL20a] and solved it in a way similar to [YWG18]. The proposed PrAE learner is similar to but has fundamental differences from existing neuro-symbolic methods. Unlike the method proposed by Yi *et al*. [YGL20a, YWG18], our approach is end-to-end trainable and does not require intermediate visual annotations, such as ground-truth attributes. Compared to [MGK19], our approach preserves logic semantics and interpretability by explicit logical reasoning involving probabilistic abduction and execution in a probabilistic planning manner [GNT04, HXZ19, KKL15].

Computational Approaches to RPM Initially proposed as an intelligence quotient test into general intelligence and fluid intelligence [Rav36, RC98], Raven's Progressive Matrices (RPM) has received notable attention from the research community of cognitive science. Psychologists have proposed reasoning systems based on symbolic representations and discrete logics [CJS90, LF17, LFU10, LTF09]. However, such logical systems cannot handle visual uncertainty arising from imperfect perception. Similar issues also pose challenges to methods based on image similarity [LLG12, MG14, MKG14, MSD18, SG18a]. Recent works approach

this problem in a data-driven manner. The first automatic RPM generation method was proposed by Wang and Su [WS15]. Santoro *et al.* [SHB18] extended it using procedural generation and introduced the WReN to solve the problem. Zhang *et al.* [ZGJ19] and Hu *et al.* [HML21] used stochastic image grammar [ZM07] and provided structural annotations to the dataset. Unanimously, existing methods do not explicitly distinguish perception and reasoning; instead, they use one monolithic neural model, sacrificing interpretability in exchange for better performance. The differences in previous methods lie in how features are manipulated: Santoro *et al.* [SHB18] used the relational module to extract final features, Zhang *et al.* [ZGJ19] stacked all panels into the channel dimension and fed them into a residual network, Hill *et al.* [HSB19] prepared the data in a contrasting manner, Zhang *et al.* [ZJG19] composed the context with each candidate and compared their potentials, Wang *et al.* [WJL20] modeled the features by a multiplex graph, and Hu *et al.* [HML21] integrated hierarchical features. Zheng *et al.* [ZZW19] studied a teacher-student setting in RPM, while Steenbrugge *et al.* [SLV18] focused on a generative approach to improve learning. Concurrent to our work, Spratley *et al.* [SEM20] unsupervisedly extracted object embeddings and conducted reasoning via a ResNet. In contrast, PrAE is designed to address cross-configuration generalization and disentangles perception and reasoning from a monolithic model, with symbolic logical reasoning implemented as probabilistic abduction and execution.

6.2 The PrAE Learner

Problem Setup In this section, we explain our approach to tackling the RPM problem. Each RPM instance consists of 16 panels: 8 context panels form an incomplete 3×3 matrix with a 9th missing entry, and 8 candidate panels for one to choose. The goal is to pick one candidate that best completes the matrix to satisfy the latent governing rules. Existing datasets [HML21, SHB18, WS15, ZGJ19] assume fixed sets of object attributes, panel attributes, and rules, with each panel attribute governed by one rule. The value of a

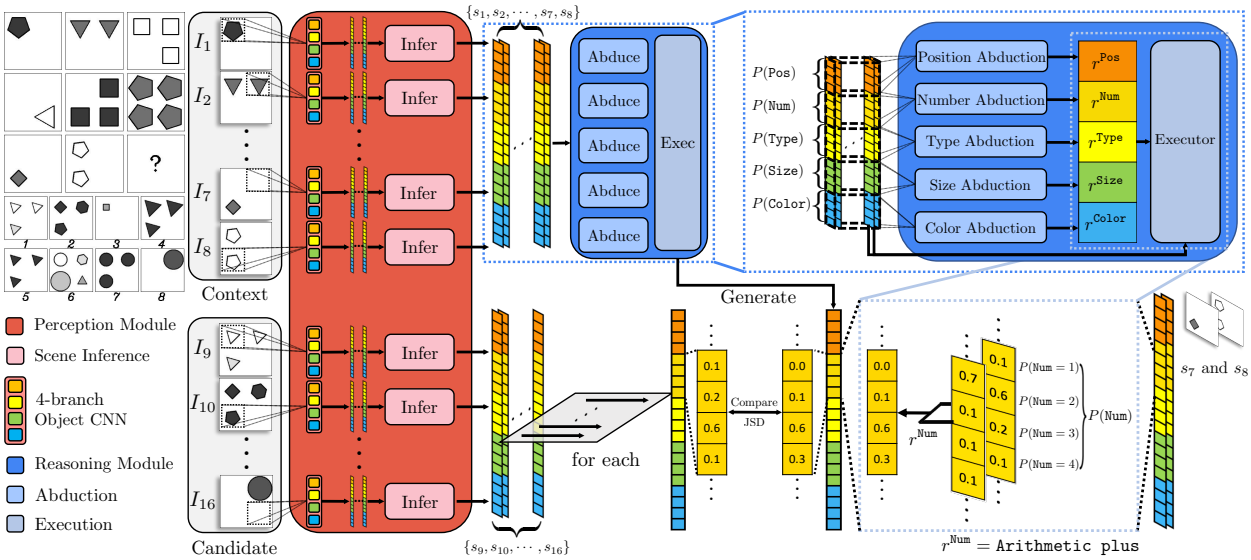


Figure 6.2: An overview of learning and reasoning of the proposed PrAE learner. We color the neural perception front end in red, the scene inference engine in pink, and the symbolic reasoning backend in blue.

panel attribute constrains the value of the corresponding object attribute for each object in it.

Overview The proposed neuro-symbolic PrAE learner disentangles previous monolithic visual reasoning into two modules: the neural visual perception frontend and the symbolic logical reasoning backend. The frontend uses a CNN to extract object attribute distributions, later aggregated by a scene inference engine to produce panel attribute distributions. The set of all panel attribute distributions in a panel is referred to as its *probabilistic scene representation*. The backend retrieves this compact scene representation and performs logical abduction and execution in order to predict the answer representation in a generative manner. A final choice is made based on the divergence between the prediction and each candidate. Using REINFORCE [Wil92], the entire system is trained *without attribute annotations* in a curricular manner; see Fig. 6.2 for an overview of PrAE.

6.2.1 Neural Visual Perception

The neural visual perception frontend operates on each of the 16 panels *independently* to produce probabilistic scene representation. It has two sub-modules: object CNN and scene inference engine.

Object CNN Given an image panel I, a sliding window traverses its spatial domain and feeds each image region into a 4-branch CNN. The 4 CNN branches use the same LeNet-like architecture [LBB98] and produce the probability distributions of object attributes, including objectiveness (whether the image region has an object), type, size, and color. Of note, the distributions of type, size, and color are conditioned on objectiveness being true. Attribute distributions of each image region are kept and sent to the scene inference engine to produce panel attribute distributions.

Scene Inference Engine The scene inference engine takes in the outputs of object CNN and produces panel attribute distributions (over position, number, type, size, and color) by marginalizing over the set of object attribute distributions (over objectiveness, type, size, and color). Take the panel attribute of `Number` as an example: Given N objectiveness probability distributions produced by the object CNN for N image regions, the probability of a panel having k objects can be computed as

$$P(\texttt{Number} = k) = \sum_{\substack{B^o \in \{0,1\}^N \\ |B^o| = k}} \prod_{j=1}^{N} P(b_j^o = B_j^o), \tag{6.1}$$

where B^o is an ordered binary sequence corresponding to objectiveness of the N regions, $|\cdot|$ the number of 1 in the sequence, and $P(b_j^o)$ the objectiveness distribution of the jth region. We assume $k \geqslant 1$ in each RPM panel, leave $P(\texttt{Number} = 0)$ out, and renormalize the probability to have a sum of 1. The panel attribute distributions for position, type, size, and color, can be computed similarly.

We refer to the set of all panel attribute distributions in a panel its *probabilistic scene representation*, denoted as s, with the distribution of panel attribute a denoted as $P(s^a)$.

6.2.2 Symbolic Logical Reasoning

The symbolic logical reasoning backend collects probabilistic scene representation from 8 context panels, abduces the probability distributions over hidden rules on each panel attribute, and executes them on corresponding panels of the context. Based on a prior study [CJS90], we assume a set of symbolic logical constraints describing rules is available. For example, the `Arithmetic plus` rule on `Number` can be represented as: for each row (column), $\forall l, m \geqslant 1$

$$(\texttt{Number}_1 = m) \wedge (\texttt{Number}_2 = l) \wedge (\texttt{Number}_3 = m + l), \tag{6.2}$$

where $\texttt{Number}_i$ denotes the number of objects in the ith panel in a row (column). With access to such constraints, we use inverse dynamics to abduce the rules in an instance. They can also be transformed into a forward model and executed on discrete symbols: For instance, `Arithmetic plus` deterministically adds `Number` in the first two panels to obtain the `Number` of the last panel.

Probabilistic Abduction Given the probabilistic scene representation of 8 context panels, the probabilistic abduction engine calculates the probability of rules for each panel attribute via inverse dynamics. Formally, for each rule r on a panel attribute a,

$$P(r^a \mid I_1, \ldots, I_8) = P(r^a \mid I_1^a, \ldots, I_8^a), \tag{6.3}$$

where I_i denotes the ith context panel, and I_i^a the component of context panel I_i corresponding to a. Note Eq. (6.3) generalizes inverse dynamics [JR92] to 8 states, in contrast to that of a conventional MDP.

To model $P(r^a \mid I_1^a, \ldots, I_8^a)$, we leverage the compact probabilistic scene representation

with respect to attribute a and logical constraints:

$$P(r^a \mid I_1^a, \ldots, I_8^a) \propto \sum_{S^a \in \texttt{valid}(r^a)} \prod_{i=1}^{8} P(s_i^a = S_i^a), \tag{6.4}$$

where $\texttt{valid}(\cdot)$ returns a set of attribute value assignments of the context panels that satisfy the logical constraints of r^a, and i indexes into context panels. By going over all panel attributes, we have the distribution of hidden rules for each of them.

Take `Arithmetic plus` on `Number` as an example. A row-major assignment for context panels can be $[1, 2, 3, 1, 3, 4, 1, 2]$ (as in Fig. 6.2), whose probability is computed as the product of each panel having k objects as in Eq. (6.1). Summing it with other assignment probabilities gives an unnormalized rule probability.

We note that the set of valid states for each r^a is a product space of valid states on each row (column). Therefore, we can perform partial marginalization on each row (column) first and aggregate them later to avoid directly marginalizing over the entire space. This decomposition will help reduce computation and mitigate numerical instability.

Probabilistic Execution For each panel attribute a, the probabilistic execution engine chooses a rule from the abduced rule distribution and executes it on corresponding context panels to predict, in a generative fashion, the panel attribute distribution of an answer. While traditionally, a logical forward model only works on discrete symbols, we follow a generalized notion of probabilistic execution as done in probabilistic planning [HXZ19, KKL15]. The probabilistic execution could be treated as a distribution transformation that redistributes the probability mass based on logical rules. For a binary rule r on a,

$$P(s_3^a = S_3^a) \propto \sum_{\substack{(S_2^a, S_1^a) \in \texttt{pre}(r^a) \\ S_3^a = f(S_2^a, S_1^a; r^a)}} P(s_2^a = S_2^a) P(s_1^a = S_1^a), \tag{6.5}$$

where f is the forward model transformed from logical constraints and $\texttt{pre}(\cdot)$ the rule precondition set. Predicted distributions of panel attributes compose the final probabilistic

scene representation s_f.

As an example of `Arithmetic plus` on `Number`, 4 objects result from the addition of $(1, 3)$, $(2, 2)$, and $(3, 1)$. The probability of an answer having 4 objects is the sum of the instances' probabilities.

During training, the execution engine samples a rule from the abduced probability. During testing, the most probable rule is chosen.

Candidate Selection With a set of predicted panel attribute distributions, we compare it with that of each candidate answer. We use the Jensen–Shannon Divergence (JSD) [Lin91] to quantify the divergence between the prediction and the candidate, *i.e.*,

$$d(s_f, s_i) = \sum_a \mathbb{D}_{\text{JSD}}(P(s_f^a) \mid\mid P(s_i^a)), \tag{6.6}$$

where the summation is over panel attributes and i indexes into the candidate panels. The candidate with minimum divergence will be chosen as the final answer.

Discussion The design of reasoning as probabilistic abduction and execution is a computational and interpretable counterpart to human-like reasoning in RPM [CJS90]. By abduction, one infers the hidden rules from context panels. By executing the abduced rules, one obtains a probabilistic answer representation. Such a probabilistic representation is compared with all candidates available; the most similar one in terms of divergence is picked as the final answer. Note that the probabilistic execution adds the generative flavor into reasoning: Eq. (6.5) depicts the predicted panel attribute distribution, which can be sampled and sent to a rendering engine for panel generation. The entire process resembles bi-directional inference and combines both top-down and bottom-up reasoning missing in prior works. In the meantime, the design addresses the aforementioned challenges by marginalizing over perception and abducing and executing rules probabilistically.

6.2.3 Learning Objective

During training, we transform the divergence in Eq. (6.6) into a probability distribution by

$$P(\text{Answer} = i) \propto \exp(-d(s_f, s_i)) \tag{6.7}$$

and minimize the cross-entropy loss. Note that the learning procedure follows a general paradigm of analysis-by-synthesis [CHY19, Gre76, HNF19, HQX18, HQZ18, LB14, WTK17, WWX17, XLZ16, XZW19, YK06, ZWM98]: The learner synthesizes a result and measures difference analytically.

As the reasoning process involves rule selection, we use REINFORCE [Wil92] to optimize:

$$\min_{\theta} \mathbb{E}_{P(r)}[\ell(P(\text{Answer}; r), y)], \tag{6.8}$$

where θ denotes the trainable parameters in the object CNN, $P(r)$ packs the rule distributions over all panel attributes, ℓ is the cross-entropy loss, and y is the ground-truth answer. Note that here we make explicit the dependency of the answer distribution on rules, as the predicted probabilistic scene representation s_f is dependent on the rules chosen.

In practice, the PrAE learner experiences difficulty in convergence with cross-entropy loss only, as the object CNN fails to produce meaningful object attribute predictions at the early stage of training. To resolve this issue, we jointly train the PrAE learner to optimize the auxiliary loss, as discussed in recent literature [SHB18, ZGJ19, ZJG19]. The auxiliary loss regularizes the perception module such that the learner produces the correct rule prediction. The final objective is

$$\min_{\theta} \mathbb{E}_{P(r)}[\ell(P(\text{Answer}; r), y)] + \sum_{a} \lambda^a \ell(P(r^a), y^a), \tag{6.9}$$

where λ^a is the weight coefficient, $P(r^a)$ the distribution of the abduced rule on a, and y^a

Method	Acc	Center	2x2Grid	3x3Grid	L-R	U-D	O-IC	O-IG
WReN	9.86/14.87	8.65/14.25	29.60/20.50	9.75/15.70	4.40/13.75	5.00/13.50	5.70/14.15	5.90/12.25
LSTM	12.81/12.52	12.70/12.55	13.80/13.50	12.90/11.35	12.40/14.30	12.10/11.35	12.45/11.55	13.30/13.05
LEN	12.29/13.60	11.85/14.85	41.40/18.20	12.95/13.35	3.95/12.55	3.95/12.75	5.55/11.15	6.35/12.35
CNN	14.78/12.69	13.80/11.30	18.25/14.60	14.55/11.95	13.35/13.00	15.40/13.30	14.35/11.80	13.75/12.85
MXGNet	20.78/13.07	12.95/13.65	37.05/13.95	24.80/12.50	17.45/12.50	16.80/12.05	18.05/12.95	18.35/13.90
ResNet	24.79/13.19	24.30/14.50	25.05/14.30	25.80/12.95	23.80/12.35	27.40/13.55	25.05/13.40	22.15/11.30
ResNet+DRT	31.56/13.26	31.65/13.20	39.55/14.30	35.55/13.25	25.65/12.15	32.05/13.10	31.40/13.70	25.05/13.15
SRAN	15.56/29.06	18.35/37.55	38.80/38.30	17.40/29.30	9.45/29.55	11.35/28.65	5.50/21.15	8.05/18.95
CoPINet	52.96/22.84	49.45/24.50	61.55/31.10	**52.15**/25.35	68.10/20.60	65.40/19.85	39.55/19.00	34.55/19.45
PrAE Learner	**65.03/77.02**	**76.50/90.45**	**78.60/85.35**	28.55/**45.60**	**90.05/96.25**	**90.85/97.35**	**48.05/63.45**	**42.60/60.70**
Human	84.41	95.45	81.82	79.55	86.36	81.81	86.36	81.81

Table 6.1: Model performance (%) on RAVEN / I-RAVEN. All models are trained on 2x2Grid only.

the ground-truth rule. In reinforcement learning terminology, one can treat the cross-entropy loss as the negative reward and the auxiliary loss as behavior cloning [SB98].

6.2.4 Curriculum Learning

In preliminary experiments, we notice that accurate objectiveness prediction at the early stage is essential to the success of the learner while learning without auxiliary will reinforce the perception system to produce more accurate object attribute predictions in the later stage when all branches of the object CNN are already warm-started. This observation is consistent with human learning: One learns object attributes only after one can correctly distinguish objects from the scene, and their perception will be enhanced with positive signals from the task.

Based on this observation, we train our PrAE learner in a 3-stage curriculum [BLC09]. In the first stage, only parameters corresponding to objectiveness are trained. In the second stage, objectiveness parameters are frozen while weights responsible for type, size, and color prediction are learned. In the third stage, we perform joint fine-tuning for the entire model via REINFORCE [Wil92].

Object Attribute	Acc	Center	2x2Grid	3x3Grid	L-R	U-D	O-IC	O-IG
Objectiveness	93.81/95.41	96.13/96.07	99.79/99.99	99.71/97.98	99.56/95.00	99.86/94.84	71.73/88.05	82.07/95.97
Type	86.29/89.24	89.89/89.33	99.95/95.93	83.49/85.96	99.92/92.90	99.85/97.84	91.55/91.86	66.68/70.85
Size	64.72/66.63	68.45/69.11	71.26/73.20	71.42/62.02	73.00/85.08	73.41/73.45	53.54/62.63	44.36/40.95
Color	75.26/79.45	75.15/75.65	85.15/87.81	62.69/69.94	85.27/83.24	84.45/81.38	84.91/75.32	78.48/82.84

Table 6.2: Accuracy (%) of the object CNN on each attribute, reported as RAVEN / I-RAVEN. The CNN module is trained with the PrAE learner on 2x2Grid only without any visual attribute annotations.

Panel Attribute	Acc	Center	2x2Grid	3x3Grid	L-R	U-D	O-IC	O-IG
Pos/Num	90.53/91.67	-	90.55/90.05	92.80/94.10	-	-	-	88.25/90.85
Type	94.17/92.15	100.00/95.00	99.75/95.30	63.95/68.40	100.00/99.90	100.00/100.00	100.00/100.00	86.08/77.60
Size	90.06/88.33	98.95/99.00	90.45/89.90	65.30/70.45	98.15/96.78	99.45/92.45	93.08/96.13	77.35/70.78
Color	87.38/87.25	97.60/93.75	88.10/85.35	37.45/45.65	98.90/92.38	99.40/98.43	92.90/97.23	73.75/79.48

Table 6.3: Accuracy (%) of the probabilistic abduction engine on each attribute, reported as RAVEN / I-RAVEN. The PrAE learner is trained on 2x2Grid only.

6.3 Experiments

We demonstrate the efficacy of the proposed PrAE learner in RPM. In particular, we show that the PrAE learner achieves the best performance among all baselines in the cross-configuration generalization task of RPM. In addition, the modularized perception and reasoning process allows us to probe into how each module performs in the RPM task and analyze the PrAE learner's strengths and weaknesses. Furthermore, we show that probabilistic scene representation learned by the PrAE learner can be used to generate an answer when equipped with a rendering engine.

6.3.1 Experimental Setup

We evaluate the proposed PrAE learner on RAVEN [ZGJ19] and I-RAVEN [HML21]. Both datasets consist of 7 distinct RPM configurations, each of which contains $10,000$ samples, equally divided into 6 folds for training, 2 folds for validation, and 2 folds for testing. We compare our PrAE learner with simple baselines of LSTM, CNN, and ResNet, and strong baselines of WReN [SHB18], ResNet+DRT [ZGJ19], LEN [ZZW19], CoPINet [ZJG19], MXGNet [WJL20], and SRAN [HML21]. To measure cross-configuration generalization, we

train all models using the 2x2Grid configuration due to its proper complexity for probability marginalization and a sufficient number of rules on each panel attribute. We test the models on all *other* configurations. All models are implemented in PyTorch [PGC17] and optimized using ADAM [KB14] on an Nvidia Titan Xp GPU. For numerical stability, we use log probability in PrAE.

6.3.2 Cross-Configuration Generalization

Table 6.1 shows the cross-configuration generalization performance of different models. While advanced models like WReN, LEN, MXGNet, and SRAN have fairly good fitting performance on the training regime, these models fail to learn transferable representation for other configurations, which suggests that they do not learn logic or any forms of abstraction but visual appearance only. Simpler baselines like LSTM, CNNs, ResNet, and ResNet+DRT show less severe overfitting, but neither do they demonstrate satisfactory performance. This effect indicates that using only deep models in abstract visual reasoning makes it very difficult to acquire the generalization capability required in situations with similar inner mechanisms but distinctive appearances. By leveraging the notion of contrast, CoPINet improves generalization performance by a notable margin.

Equipped with symbolic reasoning and neural perception, not only does the PrAE learner achieve the best performance among all models, but it also shows performance better than humans on three configurations. Compared to baselines trained on the full dataset (see supplementary material), the PrAE learner surpasses all other models on the 2x2Grid domain, despite other models seeing 6 times more data. The PrAE learner does not exhibit strong overfitting either, achieving comparable and sometimes better performance on Center, L-R, and U-D. However, limitations of the PrAE learner do exist. In cases with overlap (O-IC and O-IG), the performance decreases, and a devastating result is observed on 3x3Grid. The first failure is due to the domain shift in the region appearance that neural models cannot handle, and the second could be attributed to marginalization over probability distributions

of multiple objects in 3x3Grid, where uncertainties from all objects accumulate, leading to inaccurate abduced rule distributions. These observations are echoed in our analysis shown next.

6.3.3 Analysis on Perception and Reasoning

RAVEN and I-RAVEN provide multiple levels of annotations for us to analyze our modularized PrAE learner. Specifically, we use the region-based attribute annotations to evaluate our object CNN in perception. Note that the object CNN is not trained using any attribute annotations. We also use the ground-truth rule annotations to evaluate the accuracy of the probabilistic abduction engine.

Table 6.2 details the analysis of perception using the object CNN: It achieves reasonable performance on object attribute prediction, though not trained with any visual attribute annotations. The model shows a relatively accurate prediction of objectiveness in order to solve an RPM instance. Compared to the size prediction accuracy, the object CNN is better at predicting texture-related attributes of type and color. The object CNN has similar results on 2x2Grid, L-R, and U-D. However, referencing Table 6.1, we notice that 2x2Grid requires marginalization over more objects, resulting in an inferior performance. Accuracy further drops on configurations with overlap, leading to unsatisfactory results on O-IC and O-IG. For 3x3Grid, more accurate predictions are necessary as uncertainties accumulate from probabilities over multiple objects.

Table 6.3 details the analysis on reasoning, showing how the probabilistic abduction engine performs on rule prediction for each attribute across different configurations. Since rules on position and number are exclusive, we merge their performance as Pos/Num. As Center, L-R, U-D, and O-IC do not involve rules on Pos/Num, we do not measure the abduction performance on them. We note that, in general, the abduction engine shows good performance on all panel attributes, with a perfect prediction on type in certain configurations. However, the design of abduction as probability marginalization is a double-edged sword. While the

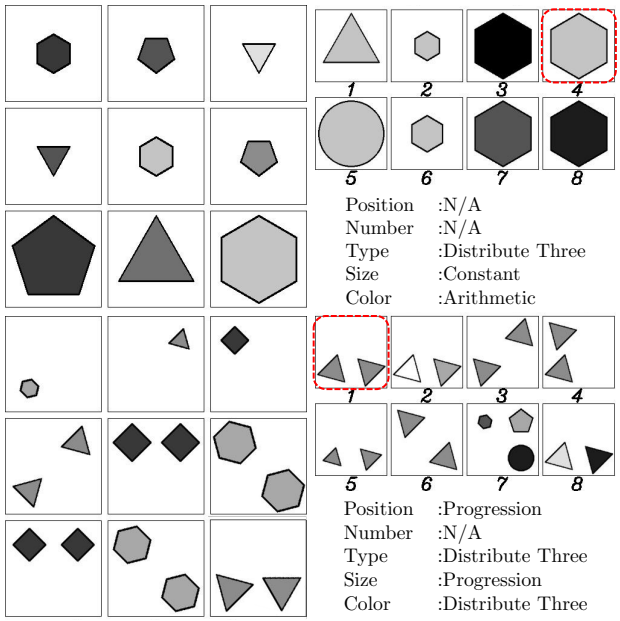


Figure 6.3: Two RPM instances with the final 9th panels filled by our generation results.

object CNN's performance on size prediction is only marginally different on 2x2Grid and 3x3Grid in RAVEN, their abduction accuracies drastically vary. The difference occurs because uncertainties on object attributes accumulate during marginalization as the number of objects increases, eventually leading to poor performance on rule prediction and answer selection. However, in configurations with fewer objects, unsatisfactory object attribute predictions can still produce accurate rule predictions. Note there is no guarantee that a correct rule will necessarily lead to a correct final choice, as the selected rule still operates on panel attribute distributions inferred from object attribute distributions.

6.3.4 Generation Ability

One unique property of the proposed PrAE learner is its ability to directly generate a panel from the predicted representation when a rendering engine is given. The ability resembles the bi-directional top-down and bottom-up reasoning, adding a generative flavor commonly ignored in prior discriminative-only approaches [HSB19, HML21, SHB18, WJL20, ZGJ19, ZJG19, ZZW19]. As the PrAE learner predicts final panel attribute distributions and is

trained in an analysis-by-synthesis manner, we can sample panel attribute values from the predicted distributions and render the final answer using a rendering engine. Here, we use the rendering program released with RAVEN [ZGJ19] to show the generation ability of the PrAE learner. Fig. 6.3 shows examples of the generation results. Note that one of our generations is slightly different from the ground-truth answer due to random sampling of rotations during rendering. However, it still follows the rules of the problem and should be considered a correct answer.

6.4 Conclusion

We propose the *Probabilistic Abduction and Execution (PrAE)* learner for spatial-temporal reasoning in Raven's Progressive Matrices (RPM) that decomposes the problem-solving process into neural perception and logical reasoning. The proposed PrAE learner is a hybrid of generative models and discriminative models, closing the loop in a human-like, top-down bottom-up bi-directional reasoning process. In the experiments, we show that the PrAE learner achieves the best performance on the cross-configuration generalization task on RAVEN and I-RAVEN. The modularized design of the PrAE learner also permits us to probe into how perception and reasoning work independently during problem-solving. Finally, we show the unique generative property of the PrAE learner by filling in the missing panel with an image produced by the values sampled from the probabilistic scene representation.

While we answer questions about generalization and generation in RPM, one crucial question remains to be addressed: How perception learned from other domains can be transferred and used to solve this abstract reasoning task. Unlike humans that arguably apply knowledge learned from elsewhere to solve RPM, current systems still need training on the same task to acquire the capability. While feature transfer is still challenging for computer vision, we anticipate that progress in answering transferability in RPM will help address similar questions [ZJE21, ZZZ20, ZGF20] and further advance the field.

CHAPTER 7

A Generalized Earley Parser for Human Activity Parsing and Prediction

In this chapter, we move forward from the spatial-temporal reasoning in RPMs as discussed in Chapter 6. With our ultimate goal of modeling real-world human activities, we propose an algorithm to tackle the task of understanding complex human activities from (partially observed) videos from two important aspects: activity recognition and prediction. To find a joint solution of activity recognition and prediction, we again consider two questions for real-world human activities: 1) what is a good representation for the structure of human activities/tasks, and 2) what is a good inference algorithm to cope with such a representation. A popular family of representations for events is the Markov models (*e.g.*, hidden Markov Model). However, Markov models are not expressive enough since human tasks often exhibit non-Markovian and compositional properties. Hence we argue that 1) a representation should reflect the hierarchical/compositional task structure of long-term human activities, and 2) an inference algorithm should recover the hierarchical structure given the past observations, and be able to predict the future.

We refer to the Chomsky hierarchy to choose a model to capture the hierarchical structure of the entire history. The Chomsky hierarchy is a containment hierarchy of classes of formal grammar in the formal languages of computer science and linguistics. The reason is that activities are analogous to languages: actions are like words and activities are like languages. The Chomsky hierarchy categorizes language models into four levels: 1) Turing machines, 2) context-sensitive grammar, 3) context-free grammar, and 4) regular grammar. Higher-level

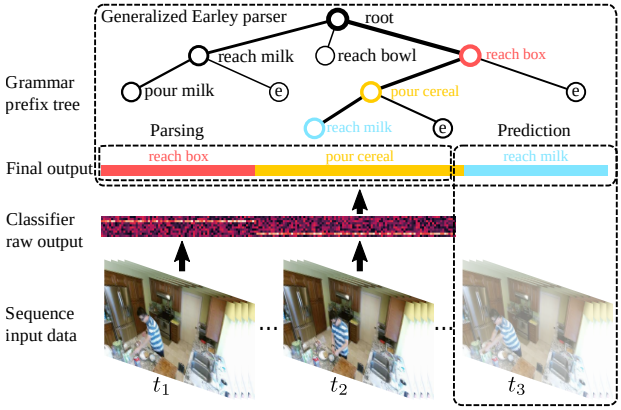


Figure 7.1: The generalized Earley parser segments and labels the sequence data into a label sentence in the language of a given grammar.

models contain lower-level models, and Markov models belong to the lowest level (regular grammar). In this paper, we propose to use *context-free grammar* to parse and predict human activities. In the definition of formal language theory, grammar is a set of production rules for sentences in a formal language. In our case, the rules describe how to form sentences (activities) from the language's alphabet (actions) that are valid. These grammar serves a similar role with rules in described Chapter 6 where information about the world model dynamics, or causal chains, is embedded.

However, it has not been possible to directly use symbolic grammar to parse and label sequence data (*e.g.*, videos). Traditional grammar parsers take symbolic sentences as inputs instead of noisy sequence data. The data has to be i) segmented and ii) labeled to be parsed by existing grammar parsers. One naive solution is to first segment and label the data using a detector and thus generate a label sentence. Then grammar parsers can be applied on top of it for parsing prediction. But this is apparently non-optimal since the grammar rules are not considered in the detection/classification process. It may not even be possible to parse this label sentence, because the output from detectors is very often grammatically incorrect.

In this chapter, we design a grammar-based parsing algorithm that directly operates on input sequence data, which goes beyond the scope of symbolic string inputs for classic parsing algorithms. Specifically, we propose a generalized Earley parser to take *probabilistic sequence*

inputs instead of deterministic symbolic inputs, based on the classic Earley parser [Ear70]. The algorithm finds the optimal segmentation and label sentence according to both a symbolic grammar and a classifier output of probabilities of labels for each frame as shown in Fig. 7.1. Optimality here means maximizing the joint probability of the label sentence according to the grammar prior and classifier output while being *grammatically correct*. In the following sections, we discuss in detail about related works on event parsing and prediction. Next, we elaborate on the algorithm design of the generalized Earley parser, especially on how we incorporate world model knowledge into activity parsing and prediction with grammar-guided inference. Finally, we provide experimental results and analysis on real-world activity understanding datasets.

7.1 Related Work

This paper is an extension of previous ICCV and ICML papers [QHW17, QJZ18]. The extension includes two major aspects. 1) For the method, we have extended the algorithm to incorporate a non-trivial grammar prior to the generalized Earley parser. This makes the algorithm applicable to not only context-free grammar (CFGs) but also probabilistic context-free grammar (PCFGs). 2) In the experiments, we tested the model on more datasets with more comparisons and in-depth analyses.

Activity parsing refers to the recognition and segmentation of long-term and complicated activities from videos, whereas action recognition corresponds to short-term actions. The mainstream of work on activity recognition is to extend mid-level representations to high-level representations. These extensions are designed in several different ways to model complex activity structures. A number of method have been proposed to model the high-level temporal structure of low-level features extracted from video [LLK07, LMS08, NCF10, GHS11, TFK12, JGR13]. Some other approaches represent complex activities as collections of attributes [LKS11, SC12, RRA12, FHX12]. Another important type of methods builds

compositional/hierarchical models on actions [GSS09, WM10, SC12, SMD13, ZWY13, LZR15, HZG16]. Koppula *et al*. [KGS13] proposed a model incorporating object affordances for detecting and predicting human activities. Wei *et al*. [WZZ17] proposed a 4D human-object interaction model for event recognition. In some recent works, structural models are implicitly learned by neural networks [WFG16, CZ17, IM18, CSG18, ZTS19].

Grammar models fall into the category of compositional models for temporal structures. Ivanov *et al*. [IB00] proposed to first generate a discrete symbol stream from continuous low-level detectors, and then applied stochastic context-free parsing to incorporate prior knowledge of the temporal structure. Pei *et al*. [PJZ11] detected atomic actions and used a stochastic context sensitive grammar for video parsing and intent prediction. Similar to the generalized Earley parser, it parses the video in an online fashion and enables prediction. However, the algorithm uses manually defined thresholds to detect action transitions. Kuehne *et al*. [KAS14] modeled action units by hidden Markov models (HMMs), and models the higher-level action sequence by context-free grammar. Pirsiavash *et al*. [PR14] proposed segmental grammar for video parsing, which extends regular grammar to allow non-terminals to generate a segment of terminals of certain lengths. Vo *et al*. [VB14] generated a Bayes network, termed Sequential Interval Network (SIN), where the variable nodes correspond to the start and end times of component actions. This network then makes inference about start and end times for detected action primitives. Qi *et al*. [QHW17] proposed to integrate spatial-temporal attributes to terminal nodes of a context-free grammar. Based on Earley parser, an activity parsing and prediction algorithm is proposed. Overall, grammar-based methods have shown effectiveness on tasks that have compositional structures.

Future activity prediction is a relatively new domain in computer vision. [ZRG09, YT10, Ryo11, KZB12, KKS12, WDA12, PSY13, WGH14, VOL14, LF14, WZZ17, HZG16, AGR16, XST18, RK17, MHL17, QJZ18] predict human trajectories/actions in various settings including complex indoor/outdoor scenes and crowded spaces. Li *et al*. [LF14] built a probabilistic suffix tree to model the Markov dependencies between action units and thus

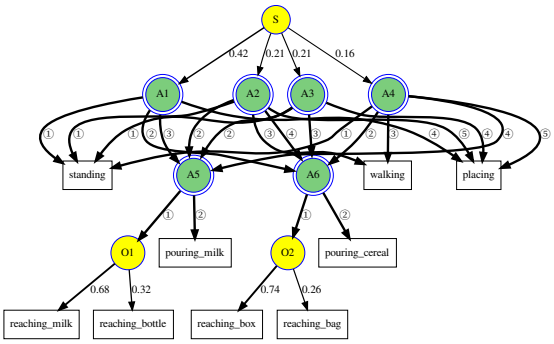


Figure 7.2: An example of a temporal grammar representing the activity “making cereal". The green and yellow nodes are And-nodes and Or-nodes, respectively.

predict future events using a compositional model. Walker *et al*. [WGH14] predicted not only the future motions in the scene but also the visual appearances. In some recent work, Koppula *et al*. [KS16] used an anticipatory temporal conditional random field to model the spatial-temporal relations through object affordances. Jain *et al*. [JZS16] proposed structural-RNN as a generic method to combine high-level spatial-temporal graphs and recurrent neural networks, which is a typical example that takes advantage of both graphical models and deep learning. Qi *et al*. [QHW17] proposed a spatial-temporal And-Or graph (ST-AOG) for activity prediction.

7.2 Preliminaries

7.2.1 Probabilistic Context-Free grammar

We model complex activities by grammar, where low-level actions are terminal symbols, *i.e*., like words in a language. In formal language theory, a *context-free grammar* (CFG) is a type of formal grammar, which contains a set of production rules that describe all possible sentences in a given formal language. In Chomsky Normal Form, a context-free grammar G is defined by a 4-tuple $G = (V, \Sigma, R, \Gamma)$ where

- V is a finite set of non-terminal symbols that can be expanded to a sequence of symbols.
- Σ is a finite set of terminal symbols that represent words in a language.
- R is a finite set of production rules describing the replacement of symbols, typically of the form $A \rightarrow BC$ or $A \rightarrow \alpha$ for $A, B, C \in V$ and $\alpha \in \Sigma$. A production rule replaces the left-hand side non-terminal symbol with the right-hand side expression. For example, $A \rightarrow BC|\alpha$ means that A can be replaced by either BC or α.
- $\Gamma \in V$ is the start symbol (root of the grammar).

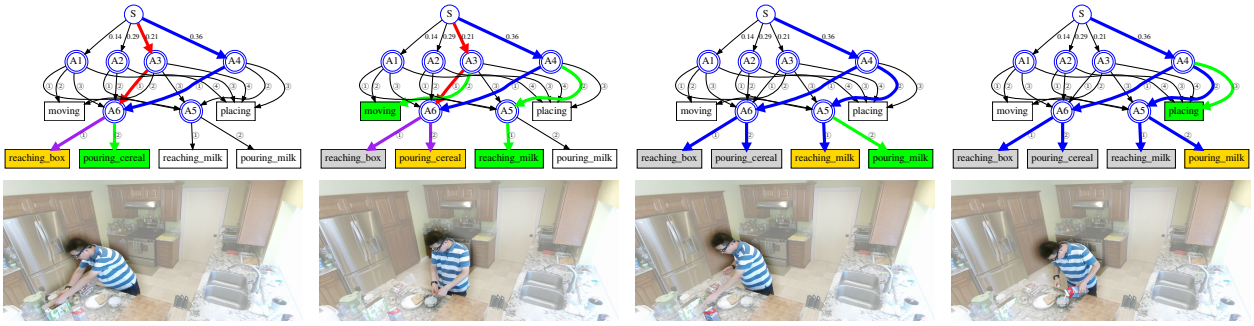


Figure 7.3: An example illustrating the symbolic parsing and prediction process based on the Earley parser and detected actions. We use red edges and blue edges to indicate different parse graphs for the past observations, purple edges for the overlap of the two possible explanations, and green edges for the possible future steps.

Probabilistic Context-Free grammar (PCFGs) augment CFGs by associating each production rule with a probability. Formally, it is defined by a 5-tuple $G = (V, \Sigma, R, \Gamma, P)$, where P is the set of probabilities on production rules. Fig. 7.2 shows an example probabilistic temporal grammar of the activity “making cereal".

7.2.2 Earley Parser

Earley parser [Ear70] is a classic grammar parsing algorithm with useful concepts that will be extended in the generalized Earley parser. Earley parser is an algorithm for parsing sentences of a given context-free language. In the following descriptions, α, β, and γ represent any string of terminals/nonterminals (including the empty string ϵ), A and B represent single

nonterminals, and a represents a terminal symbol. We adopt Earley's dot notation: for production rule of form $A \to \alpha\beta$, the notation $A \to \alpha \cdot \beta$ means α has been parsed and β is expected.

Input position n is defined as the position after accepting the nth token, and input position 0 is the position prior to input. At each input position m, the parser generates a state set $S(m)$. Each state is a tuple $(A \to \alpha \cdot \beta, i)$, consisting of

- The production currently being matched $(A \to \alpha\beta)$.
- The dot: the current position in that production.
- The position i in the input where the matching of this production began.

Seeded with $S(0)$ containing only the top-level rule, the parser then repeatedly executes three operations: prediction, scanning, and completion:

- **Prediction**: for every state in $S(m)$, $(A \to \alpha \cdot B\beta, i)$, where i is the origin position as above, add $(B \to \cdot\gamma, m)$ to $S(m)$ for every production in the grammar with B on the left-hand side (*i.e.*, $B \to \gamma$).
- **Scanning**: if a is the next symbol in the input stream, for every state in $S(m)$, $(A \to \alpha \cdot a\beta, i)$, add $(A \to \alpha a \cdot \beta, i)$ to $S(m+1)$.
- **Completion**: for every state in $S(m)$, $(A \to \gamma\cdot, j)$, find states in $S(j)$ of the form $(B \to \alpha \cdot A\beta, i)$ and add $(B \to \alpha A \cdot \beta, i)$ to $S(m)$.

In this process, duplicate states are not added to the state set. These three operations are repeated until no new states can be added to the set.

7.3 Generalized Earley Parser

In this section, we introduce the proposed generalized Earley parser. Instead of taking symbolic sentences as input, we aim to design an algorithm that can parse raw sequence data

$\boldsymbol{x}$ of length T into a sentence l of labels of length $|l| \leqslant T$, where each label $k \in \{0, 1, \cdots, K\}$ corresponds to a segment of a sequence.

To achieve that, a classifier (*e.g.*, a neural network) is first applied to each sequence $\boldsymbol{x}$ to get a $T \times K$ probability matrix $\boldsymbol{y}$ (*e.g.*, softmax activations of the neural network), with y_t^k representing the probability of frame t being labeled as k. The proposed generalized Earley parser takes $\boldsymbol{y}$ as input and outputs the sentence l^* that best explains the data according to a grammar G of Chomsky normal form. The best solution is found by performing a heuristic search in the prefix tree according to the grammar, where the heuristic is computed based on the probability matrix given by the classifier. A prefix tree is composed of three types of nodes. 1) The root node of the "empty" symbol ϵ represents the start of a sentence. 2) The non-leaf nodes (except the root node) correspond to terminal symbols in the grammar. A path from the root node to any non-leaf node represents a partial sentence (prefix). 3) The leaf nodes e are terminations that represent ends of sentences. To find the best label sentence for a probability matrix, we perform a heuristic search in the prefix expanded according to the grammar: each node in the tree is associated with a probability, and the probabilities prioritize the nodes to be expanded in the prefix tree. The parser finds the best solution when it expands a termination node in the tree. It then returns the current prefix string as the best solution.

We compute two different heuristic probabilities for non-leaf nodes and leaf nodes. For non-leaf nodes, the heuristic is a prefix probability $p(l_{\cdots}|x_{0:T})$: the probability that the current path is the prefix for the label sentence. In other words, it measures the probability that $\exists t \in [0, T]$, the current path l is the label for frame $x_{0:t}$. For leaf nodes e, the heuristic $p(l|x_{0:T})$ is a parsing probability: the probability that the current path l is the label sentence for $x_{0:T}$. The computation for $p(l|x_{0:T})$ and $p(l_{\cdots}|x_{0:T})$ are based on the input probability matrix $\boldsymbol{y}$. The formulation is derived in detail in Section 7.3.2.

This heuristic search generalizes the Earley parser to parse the probability matrix. Specifically, the scan operation in the Earley parser essentially expands a new node in the grammar

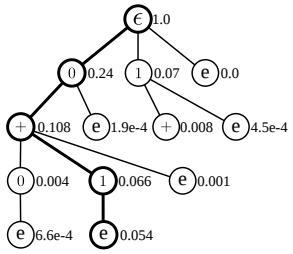
$\Gamma \to R$	1.0
$R \to N$	0.4
$R \to N$ " + " N	0.6
$N \to$ "0"	0.3
$N \to$ "1"	0.7

frame	"0"	"1"	"+"
0	0.8	0.1	0.1
1	0.8	0.1	0.1
2	0.1	0.1	0.8
3	0.1	0.8	0.1
4	0.1	0.8	0.1

(a) Left: input grammar. Right: input probability matrix.

Frame	ϵ	0	1	0 +	1 +	0 + 0	0 + 1
0	0.000	0.240	0.070	0.000	0.000	0.000	0.000
1	0.000	0.192	0.007	0.014	0.004	0.000	0.000
2	0.000	0.019	7.0e-04	0.104	0.007	4.3e-04	0.001
3	0.000	0.002	5.6e-04	0.012	7.1e-04	0.003	0.059
4	0.000	1.9e-04	4.5e-04	0.001	1.1e-04	6.6e-04	0.054
prefix	1.000	0.240	0.070	0.108	0.008	0.004	0.066

(b) Cached probabilities



(c) Prefix tree

state #	rule	μ	ν	prefix	comment
$S(0,0): l =$ "ϵ", $p(l\|G) = 1.000, p(l\|x,G) = 0.000, p(l_{\dots}\|x,G) = 1.000$					
(0)	$\Gamma \to \cdot R$	1.000	1.000	"ϵ"	start rule
(1)	$R \to \cdot N$	0.400	0.400	"ϵ"	predict: (0)
(2)	$R \to \cdot N + N$	0.600	0.600	"ϵ"	predict: (0)
(3)	$N \to \cdot 0$	0.300	0.300	"ϵ"	predict: (1),(2)
(4)	$N \to \cdot 1$	0.700	0.700	"ϵ"	predict: (1),(2)
$S(1,0): l =$ "0", $p(l\|G) = 0.300, p(l\|x,G) = 1.9e-04, p(l_{\dots}\|x,G) = 0.240$					
(0)	$N \to 0\cdot$	0.300	0.300	"0"	scan: S(0, 0)(3)
(1)	$R \to N\cdot$	0.120	0.120	"0"	complete: (0) and S(0, 0)(1)
(2)	$R \to N \cdot + N$	0.180	0.180	"0"	complete: (0) and S(0, 0)(2)
(3)	$\Gamma \to R\cdot$	0.120	0.120	"0"	complete: (1) and S(0, 0)(0)
$S(1,1): l =$ "1", $p(l\|G) = 0.700, p(l\|x,G) = 4.5e-04, p(l_{\dots}\|x,G) = 0.070$					
(0)	$N \to 1\cdot$	0.700	0.700	"1"	scan: S(0, 0)(4)
(1)	$R \to N\cdot$	0.280	0.280	"1"	complete: (0) and S(0, 0)(1)
(2)	$R \to N \cdot + N$	0.420	0.420	"1"	complete: (0) and S(0, 0)(2)
(3)	$\Gamma \to R\cdot$	0.280	0.280	"1"	complete: (1) and S(0, 0)(0)
$S(2,0): l =$ "0 + ", $p(l\|G) = 0.180, p(l\|x,G) = 0.001, p(l_{\dots}\|x,G) = 0.108$					
(0)	$R \to N + \cdot N$	0.180	0.180	"0+"	scan: S(1, 0)(2)
(1)	$N \to \cdot 0$	0.054	0.300	"0+"	predict: (0)
(2)	$N \to \cdot 1$	0.126	0.700	"0+"	predict: (0)
$S(2,1): l =$ "1 + ", $p(l\|G) = 0.420, p(l\|x,G) = 1.1e-04, p(l_{\dots}\|x,G) = 0.008$					
(0)	$R \to N + \cdot N$	0.420	0.420	"1+"	scan: S(1, 1)(2)
$S(3,0): l =$ "0 + 0", $p(l\|G) = 0.054, p(l\|x,G) = 6.6e-04, p(l_{\dots}\|x,G) = 0.004$					
(0)	$N \to 0\cdot$	0.054	0.300	"0 + 0"	scan: S(2, 0)(1)
$S(3,1): l =$ "0 + 1", $p(l\|G) = 0.126, \mathbf{p(l\|x,G) = 0.054}, p(l_{\dots}\|x,G) = 0.066$					
(0)	$N \to 1\cdot$	0.126	0.700	"0 + 1"	scan: S(2, 0)(2)
(1)	$R \to N + N\cdot$	0.126	0.126	"0 + 1"	complete: (0) and S(2, 0)(0)
(2)	$\Gamma \to R\cdot$	0.126	0.126	"0 + 1"	complete: (1) and S(0, 0)(0)
Final output: l^* = "0 + 1" with probability 0.054					

(d) A run-through of the algorithm

Table 7.1: An example of the generalized Earley parser. A classifier is applied to a 5-frame signal and outputs a probability matrix (a) as the input and our algorithm expands a grammar prefix tree (c), where e represents termination. It finally outputs the best label "0 + 1" with probability 0.054.

prefix tree. We organize the states into state sets by the partial sentence (prefix) each state represents. Instead of matching the sentence to the symbolic input, we now process state sets according to their prefix probabilities.

7.3.1 Parsing Operations

We now describe the details of the parsing operations. Each scan operation will create a new state set $S(m, n) \in S(m)$, where m is the length of the scanned string, n is the total number of the terminals that have been scanned at position m. This can be thought of as creating a new node in the prefix tree, and $S(m)$ is the set of all created nodes at level m. A priority queue q is kept for state sets for prefix search. Scan operations will push the newly created

set into the queue with priority $p(l_{\dots})$, where l is the parsed string of the state being scanned. For brevity, we use $p(l_{\dots})$ as a shorthand for $p(l_{\dots}|x_{0:t})$ when describing the algorithm.

Each state is a tuple $(A \to \alpha \cdot \beta, i, j, l, p(l_{\dots}))$ augmented from the original Earley parser by adding $j, l, p(l_{\dots})$. Here l is the parsed string of the state, and i, j are the indices of the set that this rule originated. The parser then repeatedly executes three operations: prediction, scanning, and completion modified from Earley parser:

- **Prediction**: for each state in $S(m, n)$, $(A \to \alpha \cdot B\beta, i, j, l, p(l_{\dots}))$, add $(B \to \cdot\Gamma, m, n, l, p(l_{\dots}))$ to $S(m, n)$ for every production in the grammar with B on the left-hand side.
- **Scanning**: for each state in $S(m, n)$, $(A \to \alpha \cdot a\beta, i, j, l, p(l_{\dots}))$, append the new terminal a to l and compute the probability $p((l + a)_{\dots})$. Create a new set $S(m + 1, n')$ where n' is the current size of $S(m + 1)$. Add $(A \to \alpha a \cdot \beta, i, j, l + a, p((l + a)_{\dots}))$ to $S(m + 1, n')$. Push $S(m + 1, n')$ into q with priority $p((l + a)_{\dots})$.
- **Completion**: for each state in $S(m, n)$, $(A \to \Gamma\cdot, i, j, l, p(l_{\dots}))$, find states in $S(i, j)$, $(B \to \alpha \cdot A\beta, i', j', l', p(l'_{\dots}))$, and add $(B \to \alpha A \cdot \beta, i', j', l, p(l_{\dots}))$ to $S(m, n)$.

This parsing process is efficient since we do not need to search through the entire tree. As shown in Table 7.1 and Algorithm 3, the best label sentence l is returned when the probability of termination is larger than any other prefix probabilities. As long as the parsing and prefix probabilities are computed correctly, it is guaranteed to return the best solution.

7.3.2 Parsing & Prefix Probability Formulation

Table 7.2 summarizes the notations we use in this section. The parsing probability $p(l|x_{0:T})$ is computed in a dynamic programming fashion. Let k be the last label in l. For $t = 0$, the probability is initialized by:

$$p(l|x_0) = \begin{cases} y_0^k & l \text{ contains only } k, \\ 0 & \text{otherwise.} \end{cases} \tag{7.1}$$

Algorithm 3: Generalized Earley Parser

Input : Grammar G, probability matrix y
Output : Best label string l^*

```
/* For brevity, we denote p(·; x_{0:t}) as p(·)                    */
/* Initialization                                                  */
```
$S(0,0) = \{(\Gamma \rightarrow \cdot R, 0, 0, \epsilon, 1.0)\}$
$q = priorityQueue()$
$q.push(1.0, (0, 0, \epsilon, S(0,0)))$
while $(m, n, l^-, currentSet) = q.pop()$ **do**
  for $s = (r, i, j, l, p(l_{\dots})) \in currentSet$ **do**
    if $p(l) > p(l^*)$: $l^* = l$ **then** $l^* = l$
    if r ***is*** $(A \rightarrow \alpha \cdot B\beta)$ **then** `// predict`
      for *each* $(B \rightarrow \Gamma)$ ***in*** G **do**
        $r' = (B \rightarrow \cdot\Gamma)$
        $s' = (r', m, n, l, p(l_{\dots}))$
        $S(m,n).add(s')$
    else if r ***is*** $(A \rightarrow \alpha \cdot a\beta)$ **then** `// scan`
      $r' = (A \rightarrow \alpha a \cdot \beta)$
      $m' = m + 1, n' = |S(m+1)|$
      $s' = (r', i, j, l + a, p((l+a)_{\dots}))$
      $S(m', n').add(s')$
      $q.push(p((l+a)_{\dots}), (m', n', S(m', n')))$
    else if r ***is*** $(B \rightarrow \Gamma\cdot)$ **then** `// complete`
      for *each* $((A \rightarrow \alpha \cdot B\beta), i', j')$ ***in*** $S(i,j)$ **do**
        $r' = (A \rightarrow \alpha B \cdot \beta)$
        $s' = (r', i', j', l, p(l_{\dots}))$
        $S(m,n).add(s')$
    if $p(l^-) > p(l)$, $\forall$ *un-expanded* l **then return** l^*
return l^*

Let l^- be the label sentence obtained by removing the last label k from the label sentence l. For $t > 0$, the last frame t must be classified as k. The previous frames can be labeled as either l or l^-. Then we have:

$$p(l|x_{0:t}) = y_t^k[p(l|x_{0:t-1}) + p(l^-|x_{0:t-1})], \tag{7.2}$$

where $p(l|x_{0:t-1})$ corresponds to the possibility that frame $t-1$ is also labelled as k, and $p(l^-|x_{0:t-1})$ accounts for the possibility that label k starts from frame t. It is worth men-

Table 7.2: Summary of notations used for parsing & prefix probability formulation.

$x_{0:t}$	input frames from time 0 to t
l	a label sentence
k	the last label in l
l^-	the label sentence obtained by removing the last label k from the label sentence l
y_t^k	the probability for frame t to be labelled as k
$p(l\|x_{0:t})$	parsing probability of l for $x_{0:t}$
$p(l_{\cdots}\|x_{0:t})$	prefix probability of l for $x_{0:t}$

tioning that when y_t^k is wrongly given as 0, the dynamic programming process will have trouble correcting the mistake. Even if $p(l^-|x_{0:t-1})$ is high, the probability $p(l|x_{0:t})$ will be 0. Fortunately, since the softmax function is usually adopted to compute y, y_t^k will not be 0 and the solution will be kept for further consideration.

Then we compute the prefix probability $p(l_{\cdots}|x_{0:T})$ based on $p(l^-|x_{0:t})$. For l to be the prefix, the transition from l^- to l can happen at any frame $t \in \{0, \cdots, T\}$. Once the label k is observed (the transition happens), l becomes the prefix and the rest frames can be labeled arbitrarily. Hence the probability of l being the prefix is:

$$p(l_{\cdots}|x_{0:T}) = p(l|x_0) + \sum_{t=1}^{T} y_t^k p(l^-|x_{0:t-1}). \tag{7.3}$$

In practice, the probability $p(l|x_{0:t})$ decreases exponentially as t increases and will soon lead to numeric underflow. To avoid this, the probabilities need to be computed in log space:

$$\log\, p(l|x_{0:t}) = \log(y_t^k) + d + \log\{\exp[\log p(l|x_{0:t-1}) - d] + \exp[\log p(l^-|x_{0:t-1}) - d]\}, \tag{7.4}$$

where d is a constant number and is usually set to be $\max\{\log(y_t^k), \log p(l|x_{0:t-1}), \log p(l^-|x_{0:t-1})\}$. The time complexity of computing the probabilities is $O(T)$ for each sentence l because $p(l^-|x_{0:t})$ are cached. The worst case complexity of the entire parsing is $O(T|G|)$.

7.3.3 Incorporating Grammar Prior

For PCFGs, we can integrate the grammar prior of the sentence l into the above formulation to obtain a posterior parsing probability. The basic idea is that we can compute a "transition probability" of appending a new symbol to the current sentence. This probability will be multiplied to the parsing probability when we append a new symbol.

To compute a transition probability $p(k|l^-, G)$, we can first compute the prefix probabilities $p(l^-_{...}|G)$ and $p(l_{...}|G)$ according to the grammar. Then the transition probability is given by:

$$p(k|l^-, G) = \frac{p(l_{...}|G)}{p(l^-_{...}|G)}. \tag{7.5}$$

The derivation of the grammar prefix probability with Earley parser [Sto95] can be achieved by augmenting the Earley parsing states with additional variables for forward probability and inner probability. We provide a run-through example of the generalized Earley parser with grammar prior in Table 7.1 and Fig. 7.5. There are two important remarks to make here. 1) This prior prefix probability is different from the prefix probability based on the likelihood. The prior is the probability that a string is the prefix of a sentence in the language defined by the grammar, without seeing any data; the likelihood is the probability that a string is the prefix of a video's label. 2) This grammar-based transition probability is non-Markovian, since the new symbol is conditioned on the entire history string that has a variable length.

Now, incorporating the grammar transition probability, for $t = 0$, the probability is initialized by:

$$p(l|x_0, G) \propto \begin{cases} p(k|\epsilon, G)\, y_0^k & l \text{ contains only } k, \\ 0 & \text{otherwise}, \end{cases} \tag{7.6}$$

where $p(k|\epsilon, G)$ is the probability of appending k to the empty string ϵ, which is equivalent to $p(k_{...}|G)$ or $p(l_{...}|G)$. Notice that the equal sign is replaced by $\propto$ since the right hand side

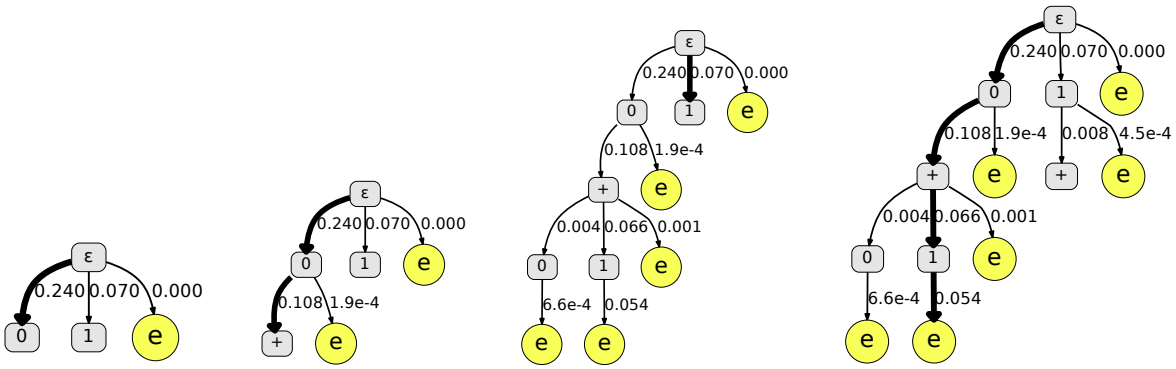


Figure 7.5: An illustration of the parsing process of the example in Table 7.1.

should be normalized by the prior $p(x_0)$ to get the correct posterior.

Whenever we append a new symbol to our sentence, we multiply the probability by the transition probability. Hence for $t > 0$ we have:

$$p(l|x_{0:t}, G) \propto y_t^k [p(l|x_{0:t-1}, G) + p(k|l^-, G)p(l^-|x_{0:t-1}, G)]. \tag{7.7}$$

Compared to Eq. (7.2), we multiply the second term by $p(k|l^-, G)$ to account for the transition to symbol k.

Finally the posterior probability of l being the prefix of the label sentence for data x is:

$$p(l_{\dots}|x_{0:T}, G) = p(l|x_0, G) + \sum_{t=1}^{T} p(k|l^-, G) y_t^k p(l^-|x_{0:t-1}, G). \tag{7.8}$$

7.3.4 Segmentation and Labeling

The generalized Earley parser gives us the best grammatically correct label sentence l to explain the sequence data, which takes all possible segmentations into consideration. Therefore the probability $p(l|x_{0:T})$ is the summation of probabilities of all possible segmentations. Let $p(l|y_{0:e})$ be the probability of the best segmentation based on the classifier output y for sentence

l. We perform a maximization over different segmentations by dynamic programming to find the best segmentation:

$$p(l|y_{0:e}) = \max_{b<e} p(l^-|y_{0:b}) \prod_{t=b}^{e} y_t^k, \tag{7.9}$$

where e is the time frame that l ends and b is the time frame that l^- ends. The best segmentation can be obtained by backtracing the above probability. Similar to the previous probabilities, this probability needs to be computed in log space as well. The time complexity of the segmentation and labeling is $O(T^2)$.

7.3.5 Future Label Prediction

We consider two types of future label predictions: 1) segment-wise prediction that predicts the next segment label at each time t, and 2) frame-wise prediction that predicts the labels for the future δt frames.

Segment-wise Prediction Given the parsing result l, we can make grammar-based top-down predictions for the next label z to be observed. The predictions are naturally obtained by the predict operation in the generalized Earley parser, and it is inherently an online prediction algorithm. To predict the next possible symbols at current position (m, n), we search through the states $S(m, n)$ of the form $(X \rightarrow \alpha \cdot z\beta, i, j, l, p(l_{\ldots}))$, where the first symbol z after the current position is a terminal node. The predictions Σ are then given by the set of all possible z:

$$\Sigma = \{z : \exists s \in S(m, n), s = (X \rightarrow \alpha \cdot z\beta, i, j, l, p(l_{\ldots}))\}. \tag{7.10}$$

The probability of each prediction is then given by the parsing likelihood of the sentence constructed by appending the predicted label z to the current sentence l. Assuming that the best prediction corresponds to the best parsing result, the goal is to find the best prediction

z^* that maximizes the following conditional probability as parsing likelihood:

$$z^* = \arg\max_{z \in \Sigma} p(z, l|G). \tag{7.11}$$

For a grammatically complete sentence u, the parsing likelihood is simply the Viterbi likelihood [Vit67] given by the probabilistic context-free grammar. For an incomplete sentence l of length $|l|$, the parsing likelihood is given by the grammar prefix probability.

Frame-wise Prediction Frame-wise future label prediction is rather straightforward using the generalized Earley parser. We first run activity detection on the input videos, and we sample the duration of the current action. Based on the segment-wise prediction, we can further sample the duration for future segments, thus obtaining frame-wise future predictions according to the prediction range.

7.3.5.1 Maximum Likelihood Estimation for Prediction

We are interested in finding the best grammar and classifier that give us the most accurate segment-wise predictions based on the generalized Earley parser. Let G be the grammar, f be the classifier, and D be the set of training examples. The training set consists of pairs of complete or partial data sequence $\boldsymbol{x}$ and the corresponding label sequence for all the frames in $\boldsymbol{x}$. By merging consecutive labels that are the same, we can obtain partial label sentences l and predicted labels z. Hence we have $D = \{(\boldsymbol{x}, l, z)\}$. The best grammar G^* and the best classifier f^* together minimizes the prediction loss:

$$G^*, f^* = \arg\min_{G,f} \mathcal{L}_{pred}(G, f), \tag{7.12}$$

where the prediction loss is given by the negative log likelihood of the predictions over the entire training set:

$$\mathcal{L}_{pred}(G, f) = -\sum_{(\boldsymbol{x},l,z)\in D} \log p(z|\boldsymbol{x}) = -\sum_{(\boldsymbol{x},l,z)\in D} \{\underbrace{\log p(z|l, G)}_{\text{grammar}} + \underbrace{\log p(l|\boldsymbol{x})}_{\text{classifier}}\}. \tag{7.13}$$

Given the intermediate variable l, the loss is decomposed into two parts that correspond to the induced grammar and the trained classifier, respectively. Let $u \in \{l\}$ be the complete label sentences in the training set (*i.e.*, the label sentence for a complete sequence $\boldsymbol{x}$). The best grammar maximizes the following probability:

$$\prod_{(z,l)\in D} p(z|l, G) = \prod_{(z,l)\in D} \frac{p(z, l|G)}{p(l|G)} = \prod_{u\in D} p(u|G), \tag{7.14}$$

where denominators $p(l|G)$ are canceled by the previous numerator $p(z, l^-|G)$, and only the likelihood of the complete sentences remain. Therefore inducing the best grammar that gives us the most accurate future prediction is equivalent to the maximum likelihood estimation (MLE) of the grammar for complete sentences in the dataset. This finding lets us turn the problem (induce the grammar that gives the best future prediction) into a standard grammar induction problem, which can be solved by existing algorithms, *e.g.*, [SHR05] and [TPZ13].

The best classifier minimizes the second term of Eq. (7.13):

$$f^* = \arg\min_f -\sum_{(\boldsymbol{x},l,z)\in D} \log p(l|\boldsymbol{x}) \approx \arg\min_f -\sum_{(\boldsymbol{x},\boldsymbol{y})\in D} \sum_k y_k \log(\hat{y_k}), \tag{7.15}$$

where $p(l|\boldsymbol{x})$ can be maximized by the CTC loss [GFG06]. In practice, it can be substituted by the commonly adopted cross entropy loss for efficiency. Therefore we can directly apply generalized Earley parser to outputs of general detectors/classifiers for parsing and prediction.

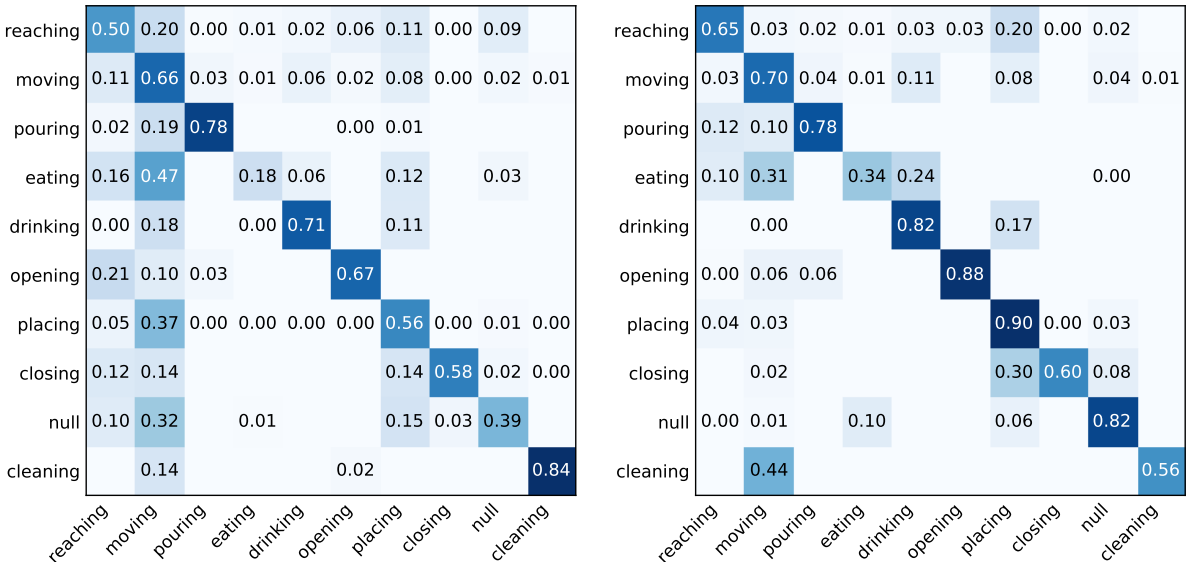
7.4 Experiments

We evaluate our method on the task of human activity detection and prediction. We present and discuss our experimental results on three datasets, CAD-120 [KGS13], Watch-n-Patch [WZS15], and Breakfast [KAS14], for comparisons with state-of-the-art methods and evaluation of the robustness of our approach. CAD-120 is the dataset that most existing prediction algorithms are evaluated on. It contains videos of daily activities that are long sequences of sub-activities. Watch-n-Patch is a daily activity dataset that features forgotten actions. Breakfast is a dataset that contains long videos of daily cooking activities. Results show that our method performs well on both activity detection and activity prediction.

Grammar Induction. In both experiments, we used a modified version of the ADIOS (automatic distillation of structure) [SHR05] grammar induction algorithm to learn the event grammar. The algorithm learns the production rules by generating significant patterns and equivalent classes. The significant patterns are selected according to a context-sensitive criterion defined regarding local flow quantities in the graph: two probabilities are defined over a search path. One is the right-moving ratio of fan-through (through-going flux of path) to fan-in (incoming flux of paths). The other one, similarly, is the left-going ratio of fan-through to fan-in. The criterion is described in detail in [SHR05].

Datasets. We consider three datasets: (i) the CAD-120 dataset [KGS13], a standard dataset for human activity prediction with 120 RGB-D videos of four different subjects performing 10 high-level activities; (ii) Watch-n-Patch [WZS15], an RGB-D dataset that features forgotten actions with 21 types of fully annotated actions (10 in the office, 11 in the kitchen) interacted with 23 types of objects; and (iii) Breakfast [KAS14], a dataset of daily cooking activities that include 52 unique participants, each conducting 10 distinct cooking activities captured in 18 different kitchens.

Evaluation Metrics. We use the following metrics to evaluate and compare the algorithms. 1) Frame-wise detection accuracy of sub-activity labels for all frames. 2) Frame-wise (future



(a) Future 3s (b) Next segment

Figure 7.6: Confusion matrices for predictions on CAD-120.

3s) online prediction accuracy. We compute the frame-wise accuracy of prediction of the sub-activity labels of the future 3s (using the frame rate of 14Hz as reported in [KGS13]). The predictions are made online at each frame t, *i.e.*, the algorithms only see frame 0 to t and predicts the labels of frame $t+1$ to $t+\delta t$. 3) Segment-wise online prediction accuracy. At each frame t, the algorithm predicts the sub-activity label of the next video segment. We consider the overall micro accuracy (P/R), macro precision, macro recall and macro F1 score for all evaluation metrics. Micro accuracy is the percentage of correctly classified labels. Macro precision and recall are the average of precision and recall respectively for all classes.

7.4.1 Experiment on CAD-120 Dataset

Comparative methods. We compare the results for the following methods: (1) KGS [KGS13], a Markov random field model. Future frames are predicted based on the transition probabilities given the inferred label of the last frame; (2) Anticipatory temporal CRF (ATCRF) [KS16], an anticipatory temporal conditional random field that models the spatial-temporal relations through object affordances. Future frames are predicting by sampling a spatial-temporal

Table 7.3: Detection results on CAD-120.

Method	Micro	Macro		
	P/R	Prec.	Recall	F1-score
KGS [KGS13]	68.2	71.1	62.2	66.4
ATCRF [KS16]	70.3	74.8	66.2	70.2
ST-AOG + Earley [QHW17]	76.5	77.0	75.2	76.1
MLP	67.2	58.7	51.5	51.1
MLP + GEP	73.8	72.8	61.1	61.0
Bi-LSTM	76.2	78.5	74.5	74.9
Bi-LSTM + GEP	**79.4**	**87.4**	**77.0**	**79.7**

Table 7.4: Future 3s prediction results on CAD-120.

Method	Micro	Macro		
	P/R	Prec.	Recall	F1-score
KGS [KGS13]	28.6	–	–	11.1
ATCRF [KS16]	49.6	–	–	40.6
ST-AOG + Earley [QHW17]	55.2	**56.5**	**56.6**	**56.6**
LSTM	49.4	40.9	37.3	37.8
LSTM + GEP	**57.1**	52.3	54.1	52.3

Table 7.5: Segment prediction results on CAD-120.

Method	Micro	Macro		
	P/R	Prec.	Recall	F1-score
ST-AOG + Earley [QHW17]	54.3	61.4	39.2	45.4
LSTM	52.8	52.5	52.8	47.6
LSTM + GEP	**70.6**	**72.1**	**70.6**	**70.1**

graph; (3) ST-AOG [QHW17], a spatial-temporal And-Or graph (ST-AOG) that uses a symbolic context-free grammar to model activity sequences; (4) Multilayer Perceptron (MLP); (5) MLP + GEP, the proposed generalized Earley parser (GEP) applied to the classifier output generated by a multilayer perceptron for detection; (6) Bidirectional LSTM (Bi-LSTM), a simple frame-wise detection classifier based on LSTM. It outputs a sub-activity label for every input frame feature. (7) LSTM, a simple prediction classifier; and Bi-LSTM/LSTM + GEP, the proposed generalized Earley parser (GEP) applied to the classifier output for detection and prediction.

Implementation details. We use the same Bi-LSTM as the base classifier for detection task and LSTM as the base classifier for prediction tasks on all three datasets. For both models, we used a 2 layer LSTM backbone with hidden size 256 and added bidirectional propagation for the Bi-LSTM model. For training, we use a Adam optimizer with learning rate 1×10^{-3} and set weight decay as 0.8 for every 20 epochs. All methods in the experiment use the same publicly available features from KGS [KGS13].

Experiment results. We follow the convention in KGS [KGS13] to train on three subjects and test on a new subject with a 4-fold validation. The results for the three evaluation metrics are summarized in Table 7.3, Table 7.4 and Table 7.5, respectively. Fig. 7.6 shows the confusion matrices for the two prediction tasks.

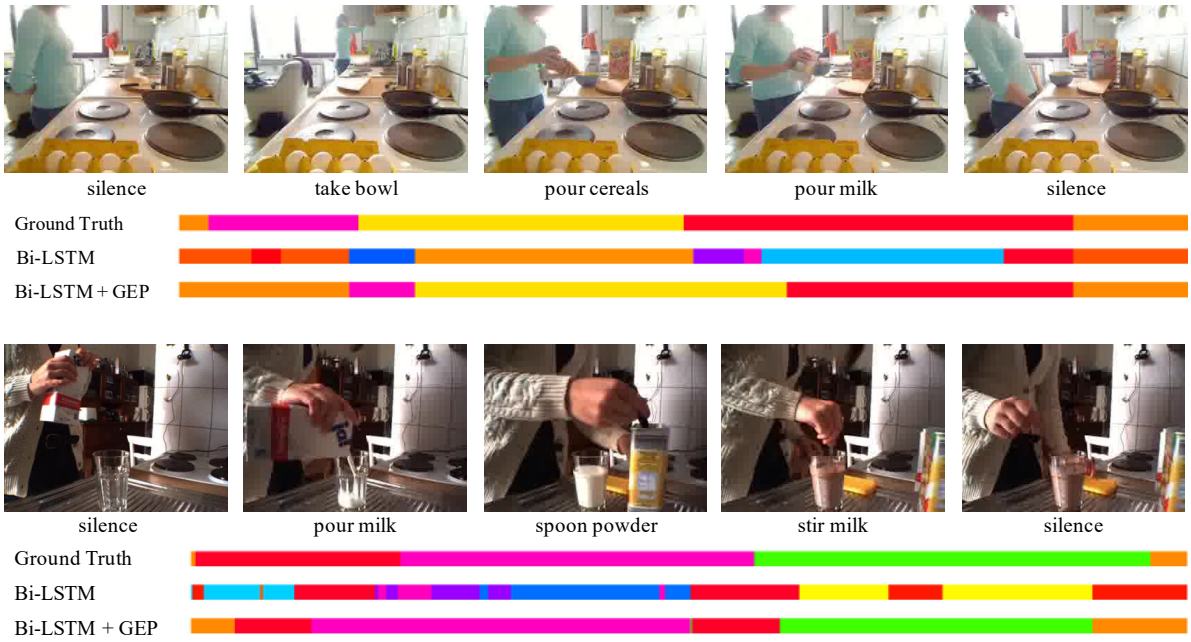


Figure 7.7: Qualitative results on the Breakfast dataset. The top row pictures show the typical frames and labels of the groud-truth segments. The bottom rows show the ground-truth segmentation, Bi-LSTM, and Bi-LSTM + GEP results.

Our method outperforms the comparative methods on all three tasks. Specifically, the generalized Earley parser on top of a Bi-LSTM performs better than ST-AOG, while ST-AOG outperforms the Bi-LSTM. More discussions are highlighted in Section 7.4.4.

7.4.2 Experiment on Watch-n-Patch Dataset

Implementation details We follow the model implementation details provided in Section 7.4.1 and extract the same features as described in [WZS15] for all methods. Similar to the previous experiment, the features are composed of skeleton features and human-object interaction features extracted from RGB-D images.

Experiment results. We use the same evaluation metrics as the previous experiment and compare our method to ST-AOG [QHW17] and Bi-LSTM. For detection, we also use the base classifier Multilayer Perceptron (MLP) and MLP with our generalized Earley parser (GEP), i.e MLP + GEP. We use the train/test split in [WZS15]. The results for the three evaluation

Table 7.6: Detection results on Watch-n-Patch.

Method	Micro	Macro		
	P/R	Prec.	Recall	F1-score
ST-AOG + Earley [QHW17]	79.3	71.5	73.5	71.9
MLP	55.6	48.7	46.7	46.4
MLP + GEP	78.1	73.0	68.4	69.7
Bi-LSTM	84.0	79.7	82.2	80.3
Bi-LSTM + GEP	**84.8**	**80.7**	**83.4**	**81.5**

Table 7.7: Future 3s prediction results on Watch-n-Patch.

Method	Micro	Macro		
	P/R	Prec.	Recall	F1-score
ST-AOG + Earley [QHW17]	48.9	43.1	39.3	39.3
LSTM	43.9	28.3	26.6	24.9
LSTM + GEP	**58.7**	**50.5**	**49.9**	**49.4**

Table 7.8: Segment prediction results on Watch-n-Patch.

Method	Micro	Macro		
	P/R	Prec.	Recall	F1-score
ST-AOG + Earley [QHW17]	29.4	28.5	18.9	19.9
LSTM	44.6	43.6	44.6	40.4
LSTM + GEP	**49.5**	**50.1**	**49.4**	**45.5**

metrics are summarized in Table 7.6, Table 7.7 and Table 7.8, respectively. Our method slightly improves the detection results over the Bi-LSTM outputs, and outperforms the other methods on both prediction tasks. In general, the algorithms make better predictions on CAD-120, since Watch-n-Patch features forgotten actions and the behaviors are more unpredictable. Fig. 7.8 shows some qualitative results, and more details are discussed in Section 7.4.4.

7.4.3 Experiment on Breakfast Dataset

Comparative methods. Besides Bi-LSTM, we compare Bi-LSTM + generalized Earley parser (GEP) with state-of-art methods for activity detection on the Breakfast dataset. The base classifier Multilayer Perceptron (MLP) and MLP + GEP are also tested. The other comparative methods include (1) HOGHOF+HTK [KAS14], a grammar-based hidden Markov model (HMM) for modeling individual action units in the sequence recognition problem; (2) ED-TCN [LFV17], an end-to-end method tackling the action classification problem; (3) TCFPN [DX18], one of the end-to-end state-of-the-art methods with temporal convolutional feature pyramid; and (4) Fisher+HTK [KGS16], the other grammar-based state-of-the-art method that leverages feature (Fisher kernels [JH99]) in HMM-based action recognition.

Table 7.9: Detection results on Breakfast.

Method	Micro	Macro		
	P/R	Prec.	Recall	F1-score
HOGHOF+HTK [KAS14]	28.8	–	–	–
ED-TCN [LFV17]*	43.3	–	–	–
TCFPN [DX18]	52.0	–	–	–
Fisher+HTK [KGS16]	56.3	38.1	–	–
MLP	15.4	7.2	7.5	5.9
MLP + GEP	32.5	35.9	15.6	18.5
Bi-LSTM	45.6	29.2	25.4	25.6
Bi-LSTM + GEP	**59.7**	**45.8**	**36.3**	**38.5**

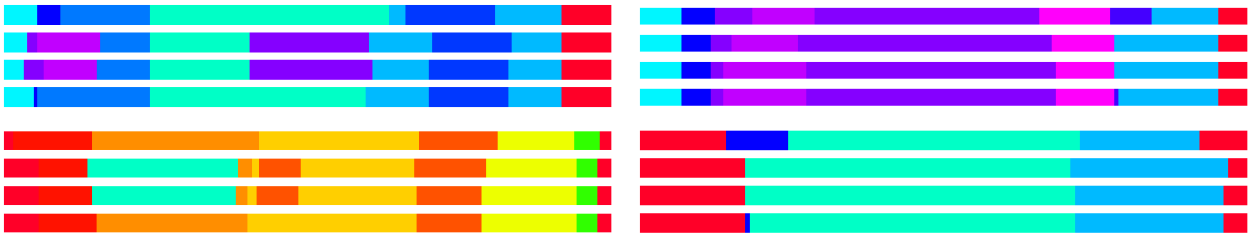
*The results for [LFV17] is obtained from [DX18].

Experiment results. To eliminate the factors of feature extraction for a fair comparison, we use the pre-computed feature provided by [KGS16] to train the underlying Bi-LSTM classifier. Fig. 7.7 shows the qualitative results of activity detection on the Breakfast dataset. The quantitative results (Table 7.9) show that a simple Bi-LSTM is far from state-of-the-art methods (an absolute difference of 10.7%). Our full algorithm Bi-LSTM + Genearlized Earley improves the absolute performance by 14.1%, and outperforms the state-of-the-art by 3.6%. This shows that our explicit grammar regularization is effective in correcting the mistakes of the underlying classifier. Although the underlying classifier is simple, it is able to perform well in the activity detection task well.

7.4.4 Discussion

How different are the classifier outputs and the final outputs for detection? Fig. 7.8 shows some qualitative examples of the ground truth segmentations and results given by different methods. The segmentation results show that the refined outputs are overall similar to the classifier outputs since the confidence given by the classifiers are often very high, but some segments are modified to ensure the grammatical correctness.

How does the generalized Earley parser refine the classifier detection outputs? When the classifier outputs violate the grammar, two types of refinements occur: i) correction and deletion of wrong labels as shown in Fig. 7.8a; ii) insertion of new labels as shown in



(a) Correction (b) Insertion

Figure 7.8: Qualitative results of segmentation results on Watch-and-Patch. The rows from the top to the bottom show the results of: 1) ground-truth, 2) ST-AOG + Earley, 3) Bi-LSTM, and 4) Bi-LSTM + generalized Earley parser.

Fig. 7.8b. The inserted segments are usually very short to accommodate both the grammar and the classifier outputs. Most boundaries of the refined results are well aligned with the classifier outputs.

How useful is the grammar for activity modeling? From Table 7.4, Table 7.5, Table 7.7 and Table 7.8 we can see that both ST-AOG and generalized Earley parser outperforms Bi-LSTM for prediction. Prediction algorithms need to give different outputs for similar inputs based on the observation history. Hence the non-Markovian property of grammar is useful for activity modeling, especially for future prediction.

7.5 Conclusion

We proposed a generalized Earley parser for parsing sequence data according to symbolic grammar. Detections and predictions are made efficiently by the parser given the probabilistic outputs from a general base classifier. Experiments show that the generalized Earley parser improves the performance of a base classifier for both detection and prediction tasks in general. We are optimistic about and interested in further applications of the generalized Earley parser. In general, we believe this is a step towards the goal of integrating the connectionist and symbolic approaches.

CHAPTER 8

Conclusion

In this dissertation, we introduced our contributions to the task of event parsing, prediction, and reasoning from both the data and modeling perspective. From the data perspective, we identified critical challenges in event understanding and reasoning. Overall, the key findings of our continual efforts reflects the need for better representations and models that we addressed in Section 1.3. With the natural complexity of events, we need proper ways to represent knowledge and use them for understanding and reasoning.

For the representation problem, we proprosed BO-QSA in Chapter 5 for emerging concepts from static images without supervision. However, as we found in the experiments, there exists a strong correlation between the powerfulness of encoder-decoder architectures and model performance. In contrast to supervised learning, more powerful encoders/decoders do not guarantee superior performance. This suggests the potential limitation of the proposed methods for more complex data. Gaining insights from how contrastive learning methods have shown the effect of concept emergence with large-scale pretraining, we should consider incorporating representations learned by self-supervised learning into object-centric learning to unite the best of both worlds. Additionally, the current learned slot initialization vectors do not explicitly bind towards concepts and is an important next-step to be extended given the significance of semantically meaningful representations in the majority of AI tasks. To achieve this goal, we believe one potential direction to be explored is to combine unsupervised object-centric learning with semantic alignments from language for concept grounding. This opens future research directions on learning finer-level organization of object concepts under

more complex scenarios (*e.g.* hierarchical grouping) with weak supervision of correspondance.

For the world model problem, we discussed two ways for incorporating world model knowledge into event parsing, prediction, and reasoning. In the synthetic domain of RAVEN as described in Chapter 2, we treated world model knowledge in the form of abduction rules in solving spatial-temporal reasoning in PrAE as discussed in Chapter 6. However, the probabilistic abduction procedure depends heavily on the clear definition of states and rules. When the number of objects increases, uncertainties over multiple objects will accumulate, making the entire process sensitive to perception performance, thus PrAE is still limited by the scalability and efficiency of probabilistic inference over knowledge bases. We further show that in real-world human activity understanding scenarios, we need to incorporate simpler and clear world model knowledge in the form of grammar. Nonetheless, the GEP is shares the same limitation as PrAE on scalability and efficiency in inference. For a comprehensive evaluation on real-world activities like the LEMMA and EgoTaskQA introduced in Chapters 3 to 4, we believe one potential way to address the critical issues in modeling is the "new" neuro-symbolic paradigm given the recent culmination of large-scale pretrained models (both language models and multimodal models). In this new paradigm, we can levereage natural language as generic symbols, large-scale pre-trained models (both visual and language) as implicit knowledge base, and prompting as the intermediate tool for bridging the two channels. However, our experiments on EgoTaskQA suggest that adopting such models directly to a specific domain is non-trivial. Compared to their capabilities in commonsense reasoning, how to enable pre-trained models with the ability to fastly adapt to complex reasoning tasks still remains an interesting problem to be solved.

REFERENCES

[AAL15] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. "Vqa: Visual question answering." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[ABA16] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. "Unsupervised learning from narrated instruction videos." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[AGR16] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. "Social lstm: Human trajectory prediction in crowded spaces." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[AK17] Brandon Amos and J Zico Kolter. "Optnet: Differentiable optimization as a layer in neural networks." In *Proceedings of International Conference on Machine Learning (ICML)*, pp. 136–145, 2017.

[ALS17] Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Simon Lacoste-Julien. "Joint discovery of object states and manipulation actions." In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2127–2136, 2017.

[AYB18] Somak Aditya, Yezhou Yang, and Chitta Baral. "Explicit reasoning over end-to-end neural architectures for visual question answering." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[BB01] Dare A Baldwin and Jodie A Baird. "Discerning intentions in dynamic human action." *Trends in Cognitive Sciences*, **5**(4):171–178, 2001.

[BBC19] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. "Dota 2 with Large Scale Deep Reinforcement Learning." *arXiv preprint arXiv:1912.06680*, 2019.

[BBS01] Dare A Baldwin, Jodie A Baird, Megan M Saylor, and M Angela Clark. "Infants parse dynamic action." *Child development*, **72**(3):708–717, 2001.

[BFM20] Daniel Bear, Chaofei Fan, Damian Mrowca, Yunzhu Li, Seth Alter, Aran Nayebi, Jeremy Schwartz, Li F Fei-Fei, Jiajun Wu, Josh Tenenbaum, et al. "Learning physical graph representations from visual scenes." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[BHS18] David Barrett, Felix Hill, Adam Santoro, Ari Morcos, and Timothy Lillicrap. "Measuring abstract reasoning in neural networks." In *Proceedings of International Conference on Machine Learning (ICML)*. PMLR, 2018.

[BKK19] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. "Deep equilibrium models." *Advances in Neural Information Processing Systems*, **32**, 2019.

[BKM20] Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. "Emergent tool use from multi-agent autocurricula." In *Proceedings of International Conference on Learning Representations (ICLR)*, 2020.

[BLB14] Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. "Weakly supervised action labeling in videos under ordering constraints." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2014.

[BLC09] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. "Curriculum learning." In *Proceedings of International Conference on Machine Learning (ICML)*, 2009.

[BLC13] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. "Estimating or propagating gradients through stochastic neurons for conditional computation." *arXiv preprint arXiv:1308.3432*, 2013.

[BLC15] Sven Bambach, Stefan Lee, David J Crandall, and Chen Yu. "Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[BMW19] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. "Monet: Unsupervised scene decomposition and representation." *arXiv preprint arXiv:1901.11390*, 2019.

[BSC18] Yonatan Bisk, Kevin J Shih, Yejin Choi, and Daniel Marcu. "Learning Interpretable Spatial Operations in a Rich 3D Blocks World." *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[BSS07] Marcel Brass, Ruth M Schmitt, Stephanie Spengler, and György Gergely. "Investigating action understanding: inferential processes versus action simulation." *Current biology*, 2007.

[BST09] Chris L Baker, Rebecca Saxe, and Joshua B Tenenbaum. "Action understanding as inverse planning." *Cognition*, 2009.

[BW20] Yaniv Benny and Lior Wolf. "Onegan: Simultaneous unsupervised learning of conditional image generation, foreground segmentation, and fine-grained clustering." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[CAD19] Mickaël Chen, Thierry Artières, and Ludovic Denoyer. "Unsupervised object segmentation by redrawing." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[CEG15] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. "Activitynet: A large-scale video benchmark for human activity understanding." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[CFG20] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. "Improved baselines with momentum contrastive learning." *arXiv preprint arXiv:2003.04297*, 2020.

[CG98] Gergely Csibra and György Gergely. "The teleological origins of mentalistic action explanations: A developmental hypothesis." *Developmental Science*, 1998.

[CG07] Gergely Csibra and György Gergely. "'Obsessed with goals': Functions and mechanisms of teleological interpretation of actions in humans." *Acta Psychologica*, 2007.

[CGL22] Michael Chang, Thomas L Griffiths, and Sergey Levine. "Object Representations as Fixed Points: Training Iterative Refinement Algorithms with Implicit Differentiation." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[CHX20] Chien-Yi Chang, De-An Huang, Danfei Xu, Ehsan Adeli, Li Fei-Fei, and Juan Carlos Niebles. "Procedure planning in instructional videos." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[CHY19] Yixin Chen, Siyuan Huang, Tao Yuan, Yixin Zhu, Siyuan Qi, and Song-Chun Zhu. "Holistic++ Scene Understanding: Single-view 3D Holistic Scene Parsing and Human Pose Estimation with Human-Object Interaction and Physical Commonsense." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019.

[CJS90] Patricia A Carpenter, Marcel A Just, and Peter Shell. "What one intelligence test measures: a theoretical account of the processing in the Raven Progressive Matrices Test." *Psychological Review*, **97**(3):404, 1990.

[CKS16] Minjie Cai, Kris M Kitani, and Yoichi Sato. "Understanding Hand-Object Manipulation with Grasp Types and Object Attributes." In *Proceedings of Robotics: Science and Systems (RSS)*, 2016.

[CLL18] Qingxing Cao, Xiaodan Liang, Bailing Li, Guanbin Li, and Liang Lin. "Visual question reasoning on general dependency tree." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[CLR21] Subhabrata Choudhury, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. "Unsupervised part discovery from contrastive reconstruction." *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[CM09] Kate Crookes and Elinor McKone. "Early maturity of face recognition: No childhood development of holistic processing, novel face encoding, or face-space." *Cognition*, **111**(2):219–247, 2009.

[CMM20] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. "Unsupervised learning of visual features by contrasting cluster assignments." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[CMS20] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. "End-to-end object detection with transformers." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[CP19] Eric Crawford and Joelle Pineau. "Spatially invariant unsupervised object detection with convolutional neural networks." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[CR68] Fergus W Campbell and JG Robson. "Application of Fourier analysis to the visibility of gratings." *The Journal of physiology*, **197**(3):551–566, 1968.

[CSB17] Berk Calli, Arjun Singh, James Bruce, Aaron Walsman, Kurt Konolige, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. "Yale-CMU-Berkeley dataset for robotic manipulation research." *International Journal of Robotics Research (IJRR)*, 2017.

[CSG18] Anoop Cherian, Suvrit Sra, Stephen Gould, and Richard Hartley. "Non-linear temporal subspace representations for activity recognition." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[Csi03] Gergely Csibra. "Teleological and referential understanding of action in infancy." *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 2003.

[CSS09] Wongun Choi, Khuram Shahid, and Silvio Savarese. "What are they doing?: Collective activity classification using spatio-temporal relationship among people." In *International Conference on Computer Vision Workshops (ICCV Workshops)*, 2009.

[CTM21] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. "Emerging properties in self-supervised vision transformers." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021.

[CVG14] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." In *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[CZ17] Joao Carreira and Andrew Zisserman. "Quo vadis, action recognition? a new model and the kinetics dataset." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[DCL18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805*, 2018.

[DCS17] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. "Scannet: Richly-annotated 3d reconstructions of indoor scenes." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[DDF22] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. "Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100." *International Journal of Computer Vision (IJCV)*, **130**(1):33–55, 2022.

[DDM18] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. "Scaling egocentric vision: The epic-kitchens dataset." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.

[DIP06] Stanislas Dehaene, Véronique Izard, Pierre Pica, and Elizabeth Spelke. "Core knowledge of geometry in an Amazonian indigene group." *Science*, **311**(5759):381–384, 2006.

[DLM16] Dima Damen, Teesid Leelasawassuk, and Walterio Mayol-Cuevas. "You-Do, I-Learn: Egocentric unsupervised discovery of objects and their modes of interaction towards video-based guidance." *Computer Vision and Image Understanding (CVIU)*, **149**:98–112, 2016.

[DLS21] Yilun Du, Shuang Li, Yash Sharma, Josh Tenenbaum, and Igor Mordatch. "Unsupervised learning of compositional energy concepts." *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[DPC12] Kun Duan, Devi Parikh, David Crandall, and Kristen Grauman. "Discovering localized attributes for fine-grained recognition." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[DX18] Li Ding and Chenliang Xu. "Weakly-supervised action segmentation with iterative soft boundary assignment." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[Ear70] Jay Earley. "An efficient context-free parsing algorithm." *Communications of the ACM*, 1970.

[EHW16] SM Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Geoffrey E Hinton, et al. "Attend, infer, repeat: Fast scene understanding with generative models." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[EKJ20] Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner. "Genesis: Generative scene inference and sampling with object-centric latent representations." In *Proceedings of International Conference on Learning Representations (ICLR)*, 2020.

[EKM84] R E Snow, Patrick Kyllonen, and B Marshalek. "The topography of ability and learning correlations." *Advances in the psychology of human intelligence*, pp. 47–103, 1984.

[EMS22] Gamaleldin F Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael C Mozer, and Thomas Kipf. "Savi++: Towards end-to-end object-centric learning from real-world videos." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[EPP21] Martin Engelcke, Oiwi Parker Jones, and Ingmar Posner. "Genesis-v2: Inferring unordered object representations without iterative refinement." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[Eva62] TG Evans. *A Heuristic Program to Solve Geometric Analogy Problems*. PhD thesis, MIT, 1962.

[Eva64] Thomas G Evans. "A heuristic program to solve geometric-analogy problems." In *Proceedings of the April 21-23, 1964, spring joint computer conference*, 1964.

[EWS21] Dave Epstein, Jiajun Wu, Cordelia Schmid, and Chen Sun. "Learning temporal dynamics from cycles in narrated video." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021.

[FAL17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." In *Proceedings of International Conference on Machine Learning (ICML)*, 2017.

[Fan19] Chenyou Fan. "EgoVQA: An Egocentric Video Question Answering Benchmark Dataset." In *Proceedings of International Conference on Computer Vision Workshops (ICCVW)*, 2019.

[FF20] Antonino Furnari and Giovanni Maria Farinella. "Rolling-unrolling lstms for action anticipation from first-person video." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, **43**(11):4021–4036, 2020.

[FFM19] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. "Slowfast networks for video recognition." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[FHR12] Alircza Fathi, Jessica K Hodgins, and James M Rehg. "Social interactions: A first-person perspective." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[FHX12] Yanwei Fu, Timothy M Hospedales, Tao Xiang, and Shaogang Gong. "Attribute learning for understanding unstructured social activity." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2012.

[FKE18] David F Fouhey, Wei-cheng Kuo, Alexei A Efros, and Jitendra Malik. "From lifestyle vlogs to everyday interactions." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[FN71] Richard E Fikes and Nils J Nilsson. "STRIPS: A new approach to the application of theorem proving to problem solving." *Artificial intelligence*, **2**(3-4):189–208, 1971.

[FQZ21] Lifeng Fan, Shuwen Qiu, Zilong Zheng, Tao Gao, Song-Chun Zhu, and Yixin Zhu. "Learning triadic belief dynamics in nonverbal communication from videos." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[FR13] Alireza Fathi and James M Rehg. "Modeling actions through state changes." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2579–2586, 2013.

[Fu74] King Sun Fu. *Syntactic methods in pattern recognition*, volume 112. Elsevier, 1974.

[FZ15] Amy Fire and Song-Chun Zhu. "Learning perceptual causality from video." *ACM Transactions on Intelligent Systems and Technology (TIST)*, **7**(2):1–22, 2015.

[FZZ19] Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. "Heterogeneous memory enhanced multimodal attention model for video question answering." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[GBK02] György Gergely, Harold Bekkering, and Ildikó Király. "Rational imitation in preverbal infants." *Nature*, **415**(6873):755–755, 2002.

[GDG15] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. "Draw: A recurrent neural network for image generation." In *Proceedings of International Conference on Machine Learning (ICML)*, 2015.

[GDG17] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. "Accurate, large minibatch sgd: Training imagenet in 1 hour." *arXiv preprint arXiv:1706.02677*, 2017.

[GEM07] Vittorio Gallese, Morris N Eagle, and Paolo Migone. "Intentional attunement: Mirror neurons and the neural underpinnings of interpersonal relations." *Journal of the American Psychoanalytic Association*, 2007.

[GF16] Noah D Goodman and Michael C Frank. "Pragmatic language interpretation as probabilistic inference." *Trends in cognitive sciences*, **20**(11):818–829, 2016.

[GFG06] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks." In *Proceedings of International Conference on Machine Learning (ICML)*, 2006.

[GFP18] Oliver Groth, Fabian B Fuchs, Ingmar Posner, and Andrea Vedaldi. "Shapestacks: Learning vision-based physical intuition for generalised object stacking." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.

[GG98] Vittorio Gallese and Alvin Goldman. "Mirror neurons and the simulation theory of mind-reading." *Trends in Cognitive Sciences*, 1998.

[GG21] Rohit Girdhar and Kristen Grauman. "Anticipative video transformer." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021.

[GHS11] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. "Actom sequence models for efficient action detection." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[GJF18] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. "Social gan: Socially acceptable trajectories with generative adversarial networks." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[GKA21] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. "AGQA: A Benchmark for Compositional Spatio-Temporal Reasoning." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[GKK19] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. "Multi-object representation learning with iterative variational inference." In *Proceedings of International Conference on Machine Learning (ICML)*, 2019.

[GKL22] Akash Gokul, Konstantinos Kallidromitis, Shufan Li, Yusuke Kato, Kazuki Kozuka, Trevor Darrell, and Colorado J Reed. "Refine and Represent: Region-to-Object Representation Learning." *arXiv preprint arXiv:2208.11821*, 2022.

[GKM17] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. "The "Something Something" Video Database for Learning and Evaluating Visual Common Sense." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017.

[GKS17] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. "Making the v in vqa matter: Elevating the role of image understanding in visual question answering." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[GLH21] Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. "Recurrent independent mechanisms." In *Proceedings of International Conference on Learning Representations (ICLR)*, 2021.

[GNC95] György Gergely, Zoltán Nádasdy, Gergely Csibra, and Szilvia Bíró. "Taking the intentional stance at 12 months of age." *Cognition*, 1995.

[GNT04] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: theory and practice*. Elsevier, 2004.

[GR20] Rohit Girdhar and Deva Ramanan. "CATER: A diagnostic dataset for Compositional Actions and TEmporal Reasoning." *Proceedings of International Conference on Learning Representations (ICLR)*, 2020.

[GRB16] Klaus Greff, Antti Rasmus, Mathias Berglund, Tele Hao, Harri Valpola, and Jürgen Schmidhuber. "Tagger: Deep unsupervised perceptual grouping." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[Gre76] Ulf Grenander. "Lectures in pattern theory I, II and III: Pattern analysis, pattern synthesis and regular structures.", 1976.

[Gri75] Herbert P Grice. "Logic and conversation." In *Speech acts*, pp. 41–58. Brill, 1975.

[GSA20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. "Bootstrap your own latent-a new approach to self-supervised learning." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[GSR18] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. "Ava: A video dataset of spatio-temporally localized atomic visual actions." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[GSR22] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. "Omnivore: A Single Model for Many Visual Modalities." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[GSS09] Abhinav Gupta, Praveen Srinivasan, Jianbo Shi, and Larry S Davis. "Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[GVS17] Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. "Neural expectation maximization." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[GVS20] Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. "On the binding problem in artificial neural networks." *arXiv preprint arXiv:2012.05208*, 2020.

[GWB21] Difei Gao, Ruiping Wang, Ziyi Bai, and Xilin Chen. "Env-QA: A Video Question Answering Benchmark for Comprehensive Understanding of Dynamic Environments." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021.

[GWB22] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. "Ego4d: Around the world in 3,000 hours of egocentric video." *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[GYB18] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. "First-person hand action benchmark with rgb-d videos and 3d hand pose annotations." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[GZB21] Zhengyang Geng, Xin-Yu Zhang, Shaojie Bai, Yisen Wang, and Zhouchen Lin. "On training implicit models." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[GZW07] Cheng-en Guo, Song-Chun Zhu, and Ying Nian Wu. "Primal sketch: Integrating structure and texture." *Computer Vision and Image Understanding (CVIU)*, **106**(1):5–19, 2007.

[HA85] Lawrence Hubert and Phipps Arabie. "Comparing partitions." *Journal of classification*, **2**(1):193–218, 1985.

[HAR17] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. "Learning to Reason: End-to-End Module Networks for Visual Question Answering." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017.

[HGD17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. "Mask r-cnn." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017.

[HHT96] Keith J Holyoak, Keith James Holyoak, and Paul Thagard. *Mental leaps: Analogy in creative thought*. MIT press, 1996.

[HKS22] Olivier J Hénaff, Skanda Koppula, Evan Shelhamer, Daniel Zoran, Andrew Jaegle, Andrew Zisserman, João Carreira, and Relja Arandjelović. "Object discovery and representation networks." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2022.

[HM18] Drew A Hudson and Christopher D Manning. "Compositional attention networks for machine reasoning." In *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.

[HM19] Drew A Hudson and Christopher D Manning. "Gqa: A new dataset for real-world visual reasoning and compositional question answering." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[HMA01] Bernhard Hommel, Jochen Müsseler, Gisa Aschersleben, and Wolfgang Prinz. "The theory of event coding (TEC): A framework for perception and action planning." *Behavioral and Brain Sciences*, 2001.

[HMG19] Chi Han, Jiayuan Mao, Chuang Gan, Josh Tenenbaum, and Jiajun Wu. "Visual Concept-Metaconcept Learning." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[HML21] Sheng Hu, Yuqing Ma, Xianglong Liu, Yanlu Wei, and Shihao Bai. "Stratified Rule-Aware Network for Abstract Visual Reasoning." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[HNF19] Tian Han, Erik Nijkamp, Xiaolin Fang, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. "Divergence Triangle for Joint Training of Generator Model, Energy-based Model, and Inferential Model." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[HQX18] Siyuan Huang, Siyuan Qi, Yinxue Xiao, Yixin Zhu, Ying Nian Wu, and Song-Chun Zhu. "Cooperative holistic scene understanding: Unifying 3d object, layout, and camera pose estimation." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[HQZ18] Siyuan Huang, Siyuan Qi, Yixin Zhu, Yinxue Xiao, Yuanlu Xu, and Song-Chun Zhu. "Holistic 3D scene parsing and reconstruction from a single RGB image." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory." *Neural computation*, 1997.

[HS18] David Ha and Jürgen Schmidhuber. "Recurrent world models facilitate policy evolution." *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[HSB19] Felix Hill, Adam Santoro, David GT Barrett, Ari S Morcos, and Timothy Lillicrap. "Learning to Make Analogies by Contrasting Abstract Relational Structure." In *Proceedings of International Conference on Learning Representations (ICLR)*, 2019.

[HW17] Dokhyam Hoshen and Michael Werman. "IQ of Neural Networks." *arXiv preprint arXiv:1710.01692*, 2017.

[HXZ19] De-An Huang, Danfei Xu, Yuke Zhu, Animesh Garg, Silvio Savarese, Li Fei-Fei, and Juan Carlos Niebles. "Continuous Relaxation of Symbolic Planner for One-Shot Imitation Learning." In *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[HZG16] Steven Holtzen, Yibiao Zhao, Tao Gao, Joshua B Tenenbaum, and Song-Chun Zhu. "Inferring human intent from video by sampling hierarchical plans." In *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[HZR16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[IB00] Yuri A Ivanov and Aaron F Bobick. "Recognition of visual activities and interactions by stochastic parsing." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2000.

[ILA15] Phillip Isola, Joseph J Lim, and Edward H Adelson. "Discovering states and transformations in image collections." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[IM18] Mostafa S Ibrahim and Greg Mori. "Hierarchical relational networks for group activity recognition and retrieval." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.

[IMD16] Mostafa S Ibrahim, Srikanth Muralidharan, Zhiwei Deng, Arash Vahdat, and Greg Mori. "A hierarchical deep temporal model for group activity recognition." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[IPO13] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. "Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, **36**(7):1325–1339, 2013.

[IPS11] Véronique Izard, Pierre Pica, Elizabeth S Spelke, and Stanislas Dehaene. "Flexible intuitions of Euclidean geometry in an Amazonian indigene group." *Proceedings of the National Academy of Sciences (PNAS)*, **108**(24):9782–9787, 2011.

[IS15] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." In *Proceedings of International Conference on Machine Learning (ICML)*, 2015.

[JBA21] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. "Perceiver io: A general architecture for structured inputs & outputs." *arXiv preprint arXiv:2107.14795*, 2021.

[JBJ08] Susanne M Jaeggi, Martin Buschkuehl, John Jonides, and Walter J Perrig. "Improving fluid intelligence with training on working memory." *Proceedings of the National Academy of Sciences (PNAS)*, **105**(19):6829–6833, 2008.

[JCH20] Baoxiong Jia, Yixin Chen, Siyuan Huang, Yixin Zhu, and Song-chun Zhu. "Lemma: A multi-view dataset for learning multi-agent multi-task activities." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[JGB21] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. "Perceiver: General perception with iterative attention." In *Proceedings of International Conference on Machine Learning (ICML)*, 2021.

[JGR13] Arpit Jain, Abhinav Gupta, Mikel Rodriguez, and Larry S Davis. "Representing videos using mid-level discriminative patches." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[JH99] Tommi Jaakkola and David Haussler. "Exploiting generative models in discriminative classifiers." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 1999.

[JH20] Pin Jiang and Yahong Han. "Reasoning with heterogeneous graph alignment for video question answering." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[JHM17] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross B Girshick. "Inferring and Executing Programs for Visual Reasoning." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017.

[JHV17] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[JJD19] Jindong Jiang, Sepehr Janghorbani, Gerard De Melo, and Sungjin Ahn. "Scalor: Generative world models with scalable object representations." In *Proceedings of International Conference on Learning Representations (ICLR)*, 2019.

[JKF20] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. "Action genome: Actions as compositions of spatio-temporal scene graphs." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[JLZ22] Baoxiong Jia, Ting Lei, Song-Chun Zhu, and Siyuan Huang. "EgoTaskQA: Understanding Human Tasks in Egocentric Videos." In *Proceedings of Advances in Neural Information Processing Systems Datasets and Benchmarks (NeurIPS Datasets and Benchmarks Track)*, 2022.

[JMR20] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. "In defense of grid features for visual question answering." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[Joh73] Gunnar Johansson. "Visual perception of biological motion and a model for its analysis." *Perception & psychophysics*, **14**(2):201–211, 1973.

[JR92] Michael I Jordan and David E Rumelhart. "Forward models: Supervised learning with a distal teacher." *Cognitive Science*, **16**(3):307–354, 1992.

[JSY17] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. "TGIF-QA: Toward Spatio-Temporal Reasoning in Visual Question Answering." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[JZS16] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. "Structural-RNN: Deep learning on spatio-temporal graphs." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[KAS14] Hilde Kuehne, Ali Arslan, and Thomas Serre. "The language of actions: Recovering the syntax and semantics of goal-directed human activities." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[KB14] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization." In *Proceedings of International Conference on Learning Representations (ICLR)*, 2014.

[KBM19] Rishabh Kabra, Chris Burgess, Loic Matthey, Raphael Lopez Kaufman, Klaus Greff, Malcolm Reynolds, and Alexander Lerchner. "Multi-Object Datasets." https://github.com/deepmind/multi-object-datasets/, 2019.

[KEM22] Thomas Kipf, Gamaleldin F Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. "Conditional object-centric learning from video." In *Proceedings of International Conference on Learning Representations (ICLR)*, 2022.

[KGS13] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. "Learning human activities and object affordances from rgb-d videos." *International Journal of Robotics Research (IJRR)*, **32**(8):951–970, 2013.

[KGS16] Hilde Kuehne, Juergen Gall, and Thomas Serre. "An end-to-end generative framework for video segmentation and recognition." In *Proceedings of Winter Conference on Applications of Computer Vision (WACV)*, 2016.

[KHA16] Max Kleiman-Weiner, Mark K Ho, Joseph L Austerweil, Michael L Littman, and Joshua B Tenenbaum. "Coordinate to cooperate or compete: abstract goals and joint intentions in social interaction." In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci)*, 2016.

[KHC17] Kyung-Min Kim, Min-Oh Heo, Seong-Ho Choi, and Byoung-Tak Zhang. "DeepStory: Video Story QA by Deep Embedded Memory Networks." In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

[KJY11] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. "Novel dataset for fine-grained image categorization: Stanford dogs." In *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*, 2011.

[KK79] Gaetano Kanizsa and Gaetano Kanizsa. *Organization in vision: Essays on Gestalt perception*, volume 49. Praeger New York, 1979.

[KKL15] George Konidaris, Leslie Kaelbling, and Tomas Lozano-Perez. "Symbol acquisition for probabilistic high-level planning." In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[KKS12] Markus Kuderer, Henrik Kretzschmar, Christoph Sprunk, and Wolfram Burgard. "Feature-Based Prediction of Trajectories for Socially Compliant Navigation." In *Proceedings of Robotics: Science and Systems (RSS)*, 2012.

[KLC98] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. "Planning and acting in partially observable stochastic domains." *Artificial intelligence*, 1998.

[KMG13] Maithilee Kunda, Keith McGreggor, and Ashok K Goel. "A computational model for solving problems from the Raven's Progressive Matrices intelligence test using iconic visual representations." *Cognitive Systems Research*, **22**:47–66, 2013.

[KS16] Hema S Koppula and Ashutosh Saxena. "Anticipating human activities using object affordances for reactive robotic response." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2016.

[KSD13] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. "3d object representations for fine-grained categorization." In *Proceedings of International Conference on Computer Vision Workshops (ICCVW)*, 2013.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

[KSM17] Ken Kansky, Tom Silver, David A Mély, Mohamed Eldawy, Miguel Lázaro-Gredilla, Xinghua Lou, Nimrod Dorfman, Szymon Sidor, Scott Phoenix, and Dileep George. "Schema networks: Zero-shot transfer with a generative causal model of intuitive physics." In *Proceedings of International Conference on Machine Learning (ICML)*, 2017.

[KTS14] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. "Large-scale video classification with convolutional neural networks." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[KW16] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks." *arXiv preprint arXiv:1609.02907*, 2016.

[KW18] Mahnaz Koupaee and William Yang Wang. "Wikihow: A large scale text summarization dataset." *arXiv preprint arXiv:1810.09305*, 2018.

[KZB12] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. "Activity forecasting." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2012.

[LB14] Matthew M Loper and Michael J Black. "OpenDR: An approximate differentiable renderer." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2014.

[LBB98] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE*, **86**(11):2278–2324, 1998.

[LCC20] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. "HERO: Hierarchical Encoder for Video+Language Omni-representation Pre-training." In *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[LCL07] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. "Crowds by example." In *Proceedings of Computer Graphics Forum*, 2007.

[LF14] Kang Li and Yun Fu. "Prediction of human activity by discovering temporal sequence patterns." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2014.

[LF17] Andrew Lovett and Kenneth Forbus. "Modeling visual problem solving as analogical reasoning." *Psychological Review*, **124**(1):60, 2017.

[LFU10] Andrew Lovett, Kenneth Forbus, and Jeffrey Usher. "A structure-mapping model of Raven's Progressive Matrices." In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci)*, 2010.

[LFV17] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. "Temporal convolutional networks for action segmentation and detection." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[LG13] Zheng Lu and Kristen Grauman. "Story-driven summarization for egocentric video." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[LGG12] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman. "Discovering important people and objects for egocentric video summarization." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[LHG21] Alex Lamb, Di He, Anirudh Goyal, Guolin Ke, Chien-Feng Liao, Mirco Ravanelli, and Yoshua Bengio. "Transformers with competitive ensembles of independent mechanisms." *arXiv preprint arXiv:2103.00336*, 2021.

[LHH20] Qing Li, Siyuan Huang, Yining Hong, Yixin Chen, Ying Nian Wu, and Song-Chun Zhu. "Closed Loop Neural-Symbolic Learning via Integrating Neural Perception, Grammar Parsing, and Symbolic Reasoning." In *Proceedings of International Conference on Machine Learning (ICML)*, 2020.

[Lin91] Jianhua Lin. "Divergence measures based on the Shannon entropy." *IEEE Transactions on Information theory*, **37**(1):145–151, 1991.

[LKS11] Jingen Liu, Benjamin Kuipers, and Silvio Savarese. "Recognizing human actions by attributes." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[LLG12] Daniel R Little, Stephan Lewandowsky, and Thomas L Griffiths. "A Bayesian model of rule induction in Raven's Progressive Matrices." In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci)*, 2012.

[LLK07] Benjamin Laxton, Jongwoo Lim, and David Kriegman. "Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[LLK19] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. "Set transformer: A framework for attention-based permutation-invariant neural networks." In *Proceedings of International Conference on Machine Learning (ICML)*, 2019.

[LLR18] Yin Li, Miao Liu, and James M Rehg. "In the eye of beholder: Joint learning of gaze and actions in first person video." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.

[LLV20] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. "Hierarchical conditional relation networks for video question answering." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[LLZ21] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. "Less is more: Clipbert for video-and-language learning via sparse sampling." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[LMB14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. "Microsoft coco: Common objects in context." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2014.

[LMM01] Richard Le Grand, Catherine J Mondloch, Daphne Maurer, and Henry P Brent. "Neuroperception: Early visual experience and face processing." *Nature*, **410**(6831):890, 2001.

[LMR99] Michael Land, Neil Mennie, and Jennifer Rusted. "The roles of vision and eye movements in the control of activities of daily living." *Perception*, **28**(11):1311–1328, 1999.

[LMS08] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. "Learning realistic human actions from movies." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[LPB22] Xudong Lin, Fabio Petroni, Gedas Bertasius, Marcus Rohrbach, Shih-Fu Chang, and Lorenzo Torresani. "Learning To Recognize Procedural Activities with Distant Supervision." *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[LSG19] Xiangpeng Li, Jingkuan Song, Lianli Gao, Xianglong Liu, Wenbing Huang, Xiangnan He, and Chuang Gan. "Beyond rnns: Positional self-attention with co-attention for video question answering." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[LTF09] Andrew Lovett, Emmett Tomai, Kenneth Forbus, and Jeffrey Usher. "Solving geometric analogy problems through two-stage analogical mapping." *Cognitive Science*, **33**(7):1192–1231, 2009.

[LWP09] Liang Lin, Tianfu Wu, Jake Porway, and Zijian Xu. "A stochastic graph grammar for compositional object representation and recognition." *Pattern Recognition*, **42**(7):1297–1307, 2009.

[LWP20] Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. "Space: Unsupervised object-oriented scene representation via spatial attention and decomposition." In *Proceedings of International Conference on Learning Representations (ICLR)*, 2020.

[LWU20] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. "Object-centric learning with slot attention." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[LWZ17] Yang Liu, Ping Wei, and Song-Chun Zhu. "Jointly recognizing object fluents and tasks in egocentric videos." In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2924–2932, 2017.

[LYB18] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L. Berg. "TVQA: Localized, Compositional Video Question Answering." In *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[LYB20] Jie Lei, Licheng Yu, Tamara L. Berg, and Mohit Bansal. "TVQA+: Spatio-Temporal Grounding for Video Question Answering." In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

[LYY19] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. "Visualbert: A simple and performant baseline for vision and language." *arXiv preprint arXiv:1908.03557*, 2019.

[LZL10] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. "Action recognition based on a bag of 3d points." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[LZR15] Tian Lan, Yuke Zhu, Amir Roshan Zamir, and Silvio Savarese. "Action recognition by hierarchical mid-level action elements." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[LZZ16] Wei Liang, Yibiao Zhao, Yixin Zhu, and Song-Chun Zhu. "What Is Where: Inferring Containment Relations from Videos." In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3418–3424, 2016.

[MAZ19] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfruend, Carl Vondrick, et al. "Moments in Time Dataset: one million videos for event understanding." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019.

[McC63] John McCarthy. "Situations, actions, and causal laws." Technical report, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 1963.

[MFK16] Minghuang Ma, Haoqi Fan, and Kris M Kitani. "Going deeper into first-person activity recognition." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[MG14] Keith McGreggor and Ashok Goel. "Confident reasoning on Raven's progressive matrices tests." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2014.

[MGH98] Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. "PDDL-the planning domain definition language." 1998.

[MGK19] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision." In *Proceedings of International Conference on Learning Representations (ICLR)*, 2019.

[MH08] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." *Journal of Machine Learning Research (JMLR)*, 2008.

[MHL17] Wei-Chiu Ma, De-An Huang, Namhoon Lee, and Kris M Kitani. "Forecasting interactive dynamics of pedestrians with fictitious play." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[MKG14] Keith McGreggor, Maithilee Kunda, and Ashok Goel. "Fractals and ravens." *Artificial Intelligence*, **215**:1–23, 2014.

[MKG21] Kanika Madan, Nan Rosemary Ke, Anirudh Goyal, Bernhard Schölkopf, and Yoshua Bengio. "Fast and slow learning of recurrent independent mechanisms." In *Proceedings of International Conference on Learning Representations (ICLR)*, 2021.

[Mon03] Stephen Monsell. "Task switching." *Trends in cognitive sciences*, **7**(3):134–140, 2003.

[MRL21] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. "Finding an unsupervised image segmenter in each of your deep generative models." *arXiv preprint arXiv:2105.08127*, 2021.

[MSC13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

[MSD18] Can Serif Mekik, Ron Sun, and David Yun Dai. "Similarity-Based Reasoning, Raven's Matrices, and General Intelligence." In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

[MSJ17] Jonghwan Mun, Paul Hongsuck Seo, Ilchae Jung, and Bohyung Han. "MarioQA: Answering Questions by Watching Gameplay Videos." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017.

[MTS18] David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. "Transparency by design: Closing the gap between performance and interpretability in visual reasoning." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[MVP21] Tom Monnier, Elliot Vincent, Jean Ponce, and Mathieu Aubry. "Unsupervised layered image decomposition into object prototypes." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021.

[MZA19] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. "HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019.

[NCF10] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei. "Modeling temporal structure of decomposable motion segments for activity classification." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2010.

[New73] Allen Newell. "You can't play 20 questions with nature and win: Projective comments on the papers of this symposium." In William G Chase, editor, *Visual Information Processing: Proceedings of the Eighth Annual Carnegie Symposium on Cognition*. Academic Press, 1973.

[NFG19] Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. "Grounded human-object interaction hotspots from video." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019.

[NG18] Tushar Nagarajan and Kristen Grauman. "Attributes as operators: factorizing unseen attribute-object compositions." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.

[NH10] Vinod Nair and Geoffrey E Hinton. "Rectified linear units improve restricted boltzmann machines." In *Proceedings of International Conference on Machine Learning (ICML)*, 2010.

[NLF20] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. "Ego-topo: Environment affordances from egocentric video." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[NRK22] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. "R3M: A Universal Visual Representation for Robot Manipulation." *arXiv preprint arXiv:2203.12601*, 2022.

[NS18] Alex Nichol and John Schulman. "Reptile: a scalable metalearning algorithm." *arXiv preprint arXiv:1803.02999*, 2018.

[NXJ20] Evonne Ng, Donglai Xiang, Hanbyul Joo, and Kristen Grauman. "You2me: Inferring body pose in egocentric video via first and second person interactions." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[NYM20] Weili Nie, Zhiding Yu, Lei Mao, Ankit B Patel, Yuke Zhu, and Anima Anandkumar. "Bongard-logo: A new benchmark for human-level concept learning and reasoning." *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[NZ10] Maria-Elena Nilsback and Andrew Zisserman. "Delving deeper into the whorl of flower segmentation." *Image and Vision Computing*, **28**(6):1049–1062, 2010.

[NZL20] Curtis Northcutt, Shengxin Zha, Steven Lovegrove, and Richard Newcombe. "Egocom: A multi-person multi-modal egocentric communications dataset." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[OGL15] Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. "Action-conditional video prediction using deep networks in atari games." *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[OHP11] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. "A large-scale benchmark dataset for event recognition in surveillance video." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[PBM20] Jae Sung Park, Chandra Bhagavatula, Roozbeh Mottaghi, Ali Farhadi, and Yejin Choi. "Visualcomet: Reasoning about the dynamic context of a still image." In *European Conference on Computer Vision*, pp. 508–524. Springer, 2020.

[PES09] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. "You'll never walk alone: Modeling social behavior for multi-target tracking." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2009.

[PGC17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. "Automatic Differentiation in PyTorch." In *NIPS Autodiff Workshop*, 2017.

[PHM16] David Paulius, Yongqiang Huang, Roger Milton, William D Buchanan, Jeanine Sam, and Yu Sun. "Functional object-oriented network for manipulation learning." In *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[PJZ11] Mingtao Pei, Yunde Jia, and Song-Chun Zhu. "Parsing video events with goal inference and intent prediction." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2011.

[PR12] Hamed Pirsiavash and Deva Ramanan. "Detecting activities of daily living in first-person camera views." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[PR14] Hamed Pirsiavash and Deva Ramanan. "Parsing videos of actions with segmental grammars." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[PSD18] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. "Film: Visual reasoning with a general conditioning layer." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation." In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[PSY13] Mingtao Pei, Zhangzhang Si, B Yao, and Song-Chun Zhu. "Video event parsing and learning with goal and intent prediction." *Computer Vision and Image Understanding (CVIU)*, 2013.

[PTS22] Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. "Teach: Task-driven embodied agents that chat." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

[PZ15] Seyoung Park and Song-Chun Zhu. "Attributed grammars for joint estimation of human attributes, part and pose." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[QHW17] Siyuan Qi, Siyuan Huang, Ping Wei, and Song-Chun Zhu. "Predicting Human Activities Using Stochastic Grammar." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017.

[QJH20] Siyuan Qi, Baoxiong Jia, Siyuan Huang, Ping Wei, and Song-Chun Zhu. "A generalized earley parser for human activity parsing and prediction." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **43**(8):2538–2554, 2020.

[QJZ18] Siyuan Qi, Baoxiong Jia, and Song-Chun Zhu. "Generalized earley parser: Bridging symbolic grammars and sequence data for future prediction." In *Proceedings of International Conference on Machine Learning (ICML)*, 2018.

[QWJ18] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. "Learning human-object interactions by graph parsing neural networks." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.

[RAA12] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. "A database for fine grained activity detection of cooking activities." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[Rav36] James C Raven. *"Mental tests used in genetic studies: The performance of related individuals on tests mainly educative and mainly reproductive."*. Master's thesis, University of London, 1936.

[Rav38] J. C. et al. Raven. "Raven's progressive matrices." *Western Psychological Services*, 1938.

[RC98] John C Raven and John Hugh Court. *Raven's progressive matrices and vocabulary scales*. Oxford pyschologists Press, 1998.

[RCJ21] Nishant Rai, Haofeng Chen, Jingwei Ji, Rishi Desai, Kazuki Kozuka, Shun Ishizaka, Ehsan Adeli, and Juan Carlos Niebles. "Home action genome: Cooperative compositional action understanding." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[Rei91] Raymond Reiter. "The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression." In *Artificial and Mathematical Theory of Computation*, 1991.

[RFK19] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. "Meta-learning with implicit gradients." *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[RHA16] Vignesh Ramanathan, Jonathan Huang, Sami Abu-El-Haija, Alexander Gorban, Kevin Murphy, and Li Fei-Fei. "Detecting events and key actors in multi-person videos." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[RHG15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[RK17] Nicholas Rhinehart and Kris M Kitani. "First-person activity forecasting with online inverse reinforcement learning." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017.

[RKZ15] Mengye Ren, Ryan Kiros, and Richard Zemel. "Exploring models and data for image question answering." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[RME01] Joshua S Rubinstein, David E Meyer, and Jeffrey E Evans. "Executive control of cognitive processes in task switching." *Journal of experimental psychology: human perception and performance*, **27**(4):763, 2001.

[RPG21] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. "Zero-shot text-to-image generation." In *Proceedings of International Conference on Machine Learning (ICML)*, 2021.

[RRA12] Marcus Rohrbach, Michaela Regneri, Mykhaylo Andriluka, Sikandar Amin, Manfred Pinkal, and Bernt Schiele. "Script data for attribute-based recognition of composite activities." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2012.

[RRR16] Marcus Rohrbach, Anna Rohrbach, Michaela Regneri, Sikandar Amin, Mykhaylo Andriluka, Manfred Pinkal, and Bernt Schiele. "Recognizing fine-grained and composite activities using hand-centric features and script data." *International Journal of Computer Vision (IJCV)*, **119**(3):346–373, 2016.

[Ryo11] Michael S Ryoo. "Human activity prediction: Early recognition of ongoing activities from streaming videos." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2011.

[SB98] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*. MIT press Cambridge, 1998.

[SC12] Sreemanananth Sadanand and Jason J Corso. "Action bank: A high-level representation of activity in video." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[SCH16] Manolis Savva, Angel X Chang, Pat Hanrahan, Matthew Fisher, and Matthias Nießner. "PiGraphs: learning interaction snapshots from observations." *ACM Transactions on Graphics (TOG)*, **35**(4):1–12, 2016.

[SCH19] Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. "Truncated back-propagation for bilevel optimization." In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

[SCS13] Claes Strannegård, Simone Cirillo, and Victor Ström. "An anthropomorphic method for progressive matrix problems." *Cognitive Systems Research*, **22**:35–46, 2013.

[SDA21] Gautam Singh, Fei Deng, and Sungjin Ahn. "Illiterate dall-e learns to compose." In *Proceedings of International Conference on Learning Representations (ICLR)*, 2021.

[SDM22] Mehdi SM Sajjadi, Daniel Duckworth, Aravindh Mahendran, Sjoerd van Steenkiste, Filip Pavetić, Mario Lučić, Leonidas J Guibas, Klaus Greff, and Thomas Kipf. "Object Scene Representation Transformer." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[SEM20] Steven Spratley, Krista Ehinger, and Tim Miller. "A Closer Look at Generalisation in RAVEN." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[SG18a] Snejana Shegheva and Ashok Goel. "The Structural Affinity Method for Solving the Raven's Progressive Matrices Test for Intelligence." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[SG18b] Snejana Shegheva and Ashok K. Goel. "The Structural Affinity Method for Solving the Raven's Progressive Matrices Test for Intelligence." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[SGS18] Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. "Charades-ego: A large-scale dataset of paired third and first person videos." *arXiv preprint arXiv:1804.09626*, 2018.

[SGT08] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. "The graph neural network model." *IEEE transactions on neural networks*, **20**(1):61–80, 2008.

[SHB18] Adam Santoro, Felix Hill, David Barrett, Ari Morcos, and Timothy Lillicrap. "Measuring abstract reasoning in neural networks." In *Proceedings of International Conference on Machine Learning (ICML)*, 2018.

[SHK14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting." *The Journal of Machine Learning Research*, 2014.

[SHL22] Jiankai Sun, De-An Huang, Bo Lu, Yun-Hui Liu, Bolei Zhou, and Animesh Garg. "PlaTe: Visually-grounded planning with transformers in procedural tasks." *IEEE Robotics and Automation Letters (RAL)*, **7**(2):4924–4930, 2022.

[SHR05] Zach Solan, David Horn, Eytan Ruppin, and Shimon Edelman. "Unsupervised learning of natural languages." *Proceedings of the National Academy of Sciences (PNAS)*, 2005.

[SK07] Elizabeth S Spelke and Katherine D Kinzler. "Core knowledge." *Developmental science*, **10**(1):89–96, 2007.

[SKL19] Michael Shum, Max Kleiman-Weiner, Michael L Littman, and Joshua B Tenenbaum. "Theory of minds: Understanding behavior in groups through inverse planning." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[SLB21] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. "Toward causal representation learning." In *Proceedings of the IEEE*, 2021.

[SLV18] Xander Steenbrugge, Sam Leroux, Tim Verbelen, and Bart Dhoedt. "Improving generalization for abstract reasoning tasks using disentangled feature representations." *arXiv preprint arXiv:1811.04784*, 2018.

[SM13] Sebastian Stein and Stephen J McKenna. "Combining embedded accelerometers with computer vision for recognizing food preparation activities." In *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2013.

[SMD13] Yale Song, Louis-Philippe Morency, and Randall Davis. "Action recognition by hierarchical sequence summarization." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[SMP22] Mehdi SM Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lučić, Daniel Duckworth, Alexey Dosovitskiy, et al. "Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[SRB17] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. "A simple neural network module for relational reasoning." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[SS15] Hyun Soo Park and Jianbo Shi. "Social saliency prediction." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4777–4785, 2015.

[Sto95] Andreas Stolcke. "An efficient probabilistic context-free parsing algorithm that computes prefix probabilities." *Computational linguistics*, 1995.

[SVW16] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. "Hollywood in homes: Crowdsourcing data collection for activity understanding." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[SWA22] Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. "Simple Unsupervised Object-Centric Learning for Complex and Naturalistic Videos." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[SXR15] Tianmin Shu, Dan Xie, Brandon Rothrock, Sinisa Todorovic, and Song Chun Zhu. "Joint inference of groups, events and human roles in aerial videos." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[SZS12] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. "UCF101: A dataset of 101 human actions classes from videos in the wild." *arXiv preprint arXiv:1212.0402*, 2012.

[TCH17] Sam Toyer, Anoop Cherian, Tengda Han, and Stephen Gould. "Human pose forecasting via deep markov models." In *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2017.

[TCS08] Pavan Turaga, Rama Chellappa, Venkatramana S Subrahmanian, and Octavian Udrea. "Machine recognition of human activities: A survey." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, **18**(11):1473–1488, 2008.

[TDR19] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. "COIN: A large-scale dataset for comprehensive instructional video analysis." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[TFK12] Kevin Tang, Li Fei-Fei, and Daphne Koller. "Learning latent temporal structure for complex event detection." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[TML14] Kewei Tu, Meng Meng, Mun Wai Lee, Tae Eun Choe, and Song-Chun Zhu. "Joint video and text parsing for understanding events and answering queries." *IEEE MultiMedia*, **21**(2):42–70, 2014.

[TPZ13] Kewei Tu, Maria Pavlovskaia, and Song-Chun Zhu. "Unsupervised structure learning of stochastic and-or grammars." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

[TSM15] Kai Sheng Tai, Richard Socher, and Christopher D Manning. "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks." In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015.

[TT41] Louis Leon Thurstone and Thelma Gwinn Thurstone. "Factorial studies of intelligence." *Psychometric monographs*, 1941.

[TZS16] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. "Movieqa: Understanding stories in movies through question-answering." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[VB14] Nam N Vo and Aaron F Bobick. "From stochastic grammar to bayes network: Probabilistic parsing of complex activity." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[VBC19] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. "Grandmaster level in StarCraft II using multi-agent reinforcement learning." *Nature*, **575**(7782):350–354, 2019.

[Vit67] Andrew Viterbi. "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm." *IEEE transactions on Information Theory*, 1967.

[VMB20] Andrey Voynov, Stanislav Morozov, and Artem Babenko. "Big gans are watching you: Towards unsupervised object segmentation with off-the-shelf generative models." 2020.

[VOL14] Tuan-Hung Vu, Catherine Olsson, Ivan Laptev, Aude Oliva, and Josef Sivic. "Predicting actions from static scenes." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2014.

[VPR13] Carl Vondrick, Donald Patterson, and Deva Ramanan. "Efficiently scaling up crowdsourced video annotation." *International Journal of Computer Vision (IJCV)*, **101**(1):184–204, 2013.

[VPT16] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. "Generating videos with scene dynamics." *Advances in neural information processing systems*, **29**, 2016.

[VSP17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[VV17] Aaron Van Den Oord, Oriol Vinyals, et al. "Neural discrete representation learning." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[WBM10] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. "Caltech-UCSD birds 200." 2010.

[WDA12] Zhikun Wang, Marc Peter Deisenroth, Heni Ben Amor, David Vogt, Bernhard Schölkopf, and Jan Peters. "Probabilistic modeling of human movements for intention inference." *Proceedings of Robotics: Science and Systems (RSS)*, 2012.

[WFF19] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. "Long-term feature banks for detailed video understanding." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[WFG16] Xiaolong Wang, Ali Farhadi, and Abhinav Gupta. "Actions˜ transformations." In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2658–2667, 2016.

[WGH14] Jacob Walker, Abhinav Gupta, and Martial Hebert. "Patch to the future: Unsupervised visual prediction." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[Wil92] Ronald J Williams. "Simple statistical gradient-following algorithms for connectionist reinforcement learning." *Machine learning*, **8**(3-4):229–256, 1992.

[WJL20] Duo Wang, Mateja Jamnik, and Pietro Lio. "Abstract Diagrammatic Reasoning with Multiplex Graph Networks." In *Proceedings of International Conference on Learning Representations (ICLR)*, 2020.

[WLM22] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. "MeMViT: Memory-Augmented Multiscale Vision Transformer for Efficient Long-Term Video Recognition." *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[WLS18] Ping Wei, Yang Liu, Tianmin Shu, Nanning Zheng, and Song-Chun Zhu. "Where and why are they looking? jointly inferring human attention and intentions in complex tasks." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[WM10] Yang Wang and Greg Mori. "Hidden part models for human action recognition: Probabilistic versus max margin." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2010.

[WMB19] Nicholas Watters, Loic Matthey, Christopher P Burgess, and Alexander Lerchner. "Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes." *arXiv preprint arXiv:1901.07017*, 2019.

[Woo98] Amanda L Woodward. "Infants selectively encode the goal object of an actor's reach." *Cognition*, 1998.

[WS15] Ke Wang and Zhendong Su. "Automatic Generation of Raven's Progressive Matrices." In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[WSH22] Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L Crowley, and Dominique Vaufreydaz. "Self-supervised transformers for unsupervised object discovery using normalized cut." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[WTK17] Jiajun Wu, Joshua B Tenenbaum, and Pushmeet Kohli. "Neural scene derendering." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[WWX17] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Josh Tenenbaum. "Marrnet: 3d shape reconstruction via 2.5 d sketches." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[WXZ07] Tian-Fu Wu, Gui-Song Xia, and Song-Chun Zhu. "Compositional boosting for computing hierarchical image structures." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[WYC21] Bo Wu, Shoubin Yu, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. "Star: A benchmark for situated reasoning in real-world videos." In *Proceedings of Advances in Neural Information Processing Systems Datasets and Benchmarks (NeurIPS Datasets and Benchmarks Track)*, 2021.

[WZS15] Chenxia Wu, Jiemi Zhang, Silvio Savarese, and Ashutosh Saxena. "Watch-n-patch: Unsupervised understanding of actions and relations." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[WZZ17] Ping Wei, Yibiao Zhao, Nanning Zheng, and Song-Chun Zhu. "Modeling 4d human-object interactions for joint event segmentation, recognition, and object localization." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

[XCW15] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[XDL22] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. "GroupViT: Semantic Segmentation Emerges from Text Supervision." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[XLZ16] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. "A theory of generative convnet." In *Proceedings of International Conference on Machine Learning (ICML)*, 2016.

[XST18] Dan Xie, Tianmin Shu, Sinisa Todorovic, and Song-Chun Zhu. "Modeling and Inferring Human Intents and Latent Functional Objects for Trajectory Prediction." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018.

[XSY21] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. "NExT-QA: Next Phase of Question-Answering to Explaining Temporal Actions." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[XZW19] Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. "Learning Energy-based Spatial-Temporal Generative ConvNets for Dynamic Patterns." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019.

[XZX17] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. "Video Question Answering via Gradually Refined Attention over Appearance and Motion." In *Proceedings of ACM International Conference on Multimedia (MM)*, 2017.

[YGL20a] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. "CLEVRER: Collision Events for Video Representation and Reasoning." In *Proceedings of International Conference on Learning Representations (ICLR)*, 2020.

[YGL20b] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. "Clevrer: Collision events for video representation and reasoning." In *Proceedings of International Conference on Learning Representations (ICLR)*, 2020.

[YGW22] Hong-Xing Yu, Leonidas J Guibas, and Jiajun Wu. "Unsupervised discovery of object radiance fields." In *Proceedings of International Conference on Learning Representations (ICLR)*, 2022.

[YK06] Alan Yuille and Daniel Kersten. "Vision as Bayesian inference: analysis by synthesis?" *Trends in cognitive sciences*, 2006.

[YLL21] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. "Self-supervised video object segmentation by motion grouping." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021.

[YMS21] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. "Just Ask: Learning to Answer Questions from Millions of Narrated Videos." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021.

[YMY18] Takuma Yagi, Karttikeya Mangalam, Ryo Yonetani, and Yoichi Sato. "Future person localization in first-person videos." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[YRJ18] Serena Yeung, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei. "Every moment counts: Dense detailed labeling of actions in complex videos." *International Journal of Computer Vision (IJCV)*, **126**(2-4):375–389, 2018.

[YT10] Jenny Yuen and Antonio Torralba. "A data-driven approach for event prediction." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2010.

[YWG18] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. "Neural-symbolic vqa: Disentangling reasoning from vision and language understanding." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[YXM21] Peiyu Yu, Sirui Xie, Xiaojian Ma, Yixin Zhu, Ying Nian Wu, and Song-Chun Zhu. "Unsupervised foreground extraction via deep region competition." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[YXY19] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. "ActivityNet-QA: A Dataset for Understanding Complex Web Videos via Question Answering." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[YY22] Yafei Yang and Bo Yang. "Promising or Elusive? Unsupervised Object Segmentation from Real-world Single Images." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[ZBF19] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. "From recognition to cognition: Visual commonsense reasoning." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[ZCC17] Kuo-Hao Zeng, Tseng-Hung Chen, Ching-Yao Chuang, Yuan-Hong Liao, Juan Carlos Niebles, and Min Sun. "Leveraging Video Descriptions to Learn Video Question Answering." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[ZCL19] Amir Zadeh, Michael Chan, Paul Pu Liang, Edmund Tong, and Louis-Philippe Morency. "Social-IQ: A Question Answering Benchmark for Artificial Social Intelligence." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[ZGB16] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. "Visual7w: Grounded question answering in images." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[ZGF20] Yixin Zhu, Tao Gao, Lifeng Fan, Siyuan Huang, Mark Edmonds, Hangxin Liu, Feng Gao, Chi Zhang, Siyuan Qi, Ying Nian Wu, et al. "Dark, beyond deep: A paradigm shift to cognitive ai with humanlike common sense." *Engineering*, **6**(3):310–345, 2020.

[ZGJ19] Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. "Raven: A dataset for relational and analogical visual reasoning." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[ZGZ21] Chuhan Zhang, Ankush Gupta, and Andrew Zisserman. "Temporal query networks for fine-grained video understanding." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[ZHP19] Yan Zhang, Jonathon Hare, and Adam Prugel-Bennett. "Deep set prediction networks." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[ZHP21] Rowan Zellers, Ari Holtzman, Matthew Peters, Roozbeh Mottaghi, Aniruddha Kembhavi, Ali Farhadi, and Yejin Choi. "PIGLeT: Language grounding through neuro-symbolic interaction in a 3D world." *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

[ZJE21] Chi Zhang, Baoxiong Jia, Mark Edmonds, Song-Chun Zhu, and Yixin Zhu. "Acre: Abstract causal reasoning beyond covariation." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[ZJG19] Chi Zhang, Baoxiong Jia, Feng Gao, Yixin Zhu, Hongjing Lu, and Song-Chun Zhu. "Learning Perceptual Inference by Contrasting." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[ZJZ21] Chi Zhang, Baoxiong Jia, Song-Chun Zhu, and Yixin Zhu. "Abstract spatial-temporal reasoning via probabilistic abduction and execution." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[ZKL21] Daniel Zoran, Rishabh Kabra, Alexander Lerchner, and Danilo J Rezende. "PARTS: Unsupervised segmentation with slots, attention and independence maximization." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021.

[ZKR17] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. "Deep sets." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[ZM07] Song-Chun Zhu and David Mumford. "A stochastic grammar of images." *Foundations and Trends® in Computer Graphics and Vision*, **2**(4):259–362, 2007.

[ZRG09] Brian D Ziebart, Nathan Ratliff, Garratt Gallagher, Christoph Mertz, Kevin Peterson, J Andrew Bagnell, Martial Hebert, Anind K Dey, and Siddhartha Srinivasa. "Planning-based prediction for pedestrians." In *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2009.

[ZTS19] Yubo Zhang, Pavel Tokmakov, Cordelia Schmid, and Martial Hebert. "A Structured Model For Action Detection." *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[ZWM98] Song-Chun Zhu, Yingnian Wu, and David Mumford. "Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling." *International Journal of Computer Vision (IJCV)*, **27**(2):107–126, 1998.

[ZWY13] Jun Zhu, Baoyuan Wang, Xiaokang Yang, Wenjun Zhang, and Zhuowen Tu. "Action recognition with actons." In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[ZWZ16] Jun Zhu, Tianfu Wu, Song-Chun Zhu, Xiaokang Yang, and Wenjun Zhang. "A reconfigurable tangram model for scene representation and categorization." *IEEE Transactions on Image Processing*, **25**(1):150–166, 2016.

[ZXC18] Luowei Zhou, Chenliang Xu, and Jason J Corso. "Towards automatic learning of procedures from web instructional videos." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[ZXJ22] Chi Zhang, Sirui Xie, Baoxiong Jia, Ying Nian Wu, Song-Chun Zhu, and Yixin Zhu. "Learning algebraic representation for systematic generalization in abstract reasoning." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2022.

[ZZH17] Chen Zhu, Yanpeng Zhao, Shuaiyi Huang, Kewei Tu, and Yi Ma. "Structured attentions for visual question answering." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017.

[ZZW19] Kecheng Zheng, Zheng-Jun Zha, and Wei Wei. "Abstract Reasoning with Distracting Features." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[ZZZ20] Wenhe Zhang, Chi Zhang, Yixin Zhu, and Song-Chun Zhu. "Machine number sense: A dataset of visual arithmetic problems for abstract and relational reasoning." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2020.