Learning to See Agents
with Deep Variational Inference

A THESIS PRESENTED

BY

Aneesh Chenna Reddy Muppidi

to

THE DEPARTMENT OF COMPUTER SCIENCE

AND THE FACULTY OF THE COMMITTEE ON DEGREES IN NEUROSCIENCE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE WITH HONORS

OF BACHELOR OF ARTS

Advised by Wilka Carvalho and Samuel Gershman

HARVARD UNIVERSITY
CAMBRIDGE, MASSACHUSETTS
MARCH 2025

Neuroscience Concentration
Division of Life Sciences
Harvard University

The Harvard College Honor Code

Members of the Harvard College community commit themselves to producing academic work of integrity – that is, work that adheres to the scholarly and intellectual standards of accurate attribution of sources, appropriate collection and use of data, and transparent acknowledgement of the contribution of others to their ideas, discoveries, interpretations, and conclusions. Cheating on exams or problem sets, plagiarizing or misrepresenting the ideas or language of someone else as one's own, falsifying data, or any other instance of academic dishonesty violates the standards of our community, as well as the standards of the wider world of learning and affairs.

Digital Signature: Aneesh Muppidi

Learning to See Agents
with Deep Variational Inference

ABSTRACT

Unsupervised agent discovery is the ability to identify and model intentional agents from raw perceptual data without explicit supervision. While neurocognitive theories propose different neural mechanisms for agent perception—including mirror neurons and the superior temporal sulcus (STS), we lack computational algorithms that can fully describe agent perception. Existing computational models of agent perception operate on simplified symbolic inputs rather than the raw perceptual data that biological systems process. We introduce a variational objective ($\mathcal{L}_{\mathbf{VAD}}$) that formulates vision-based agent discovery as structured inference over latent actions. Based on $\mathcal{L}_{\text{VAD}}$, we implement a deep conditional slot-based variational autoencoder called **VAD (Variational Agent Discovery)** model. Our model learns internal agent representations directly from raw pixel-based observations, outperforming baselines on predictive tasks including agent action and goal inference in three video-game settings. VAD's internal representations generalize robustly to novel agents and environmental configurations, demonstrating up to 33% advantage in transfer scenarios. The VAD model exhibits predictive capabilities analogous to those observed in infant cognition studies, correctly predicting that agents will take efficient paths to goals when environmental constraints change. Analysis of learned representations reveals functional decomposition of visual scenes along agent-centric lines, with certain neural features exhibiting human mirror-neuron-like activation patterns across different agents performing the same actions. When incorporated as an auxiliary loss in multi-agent reinforcement learning, our $\mathcal{L}_{\text{VAD}}$ objective improves sample efficiency by 21.8% and final performance by 7.6%.

Code, Data, and Conceptual Animations are available here.

ACKNOWLEDGEMENTS

I am deeply grateful to Professor Samuel Gershman and Dr. Wilka Carvalho for their guidance and mentorship throughout this thesis. They continually pushed me to ask new questions and challenged me when it mattered most.

To my parents, Rami and Rani Muppidi, and my little brother, Akshay—thank you for your unwavering support and belief in me.

To my closest friends: Rohil Dhaliwal and Shahmir Aziz—thank you for always demanding rigor and reminding me to reach higher. To Mani Chadaga, Brandon Pham, and Reza Shamji—my roommates—thank you for listening to me talk about this thesis a thousand times, for your feedback, your help with figures, and your company during the late nights. To Krisha Patel, Amiya Tiwari, and Vishnu Emani—thank you for grounding me and giving me perspective. To Michael Waxman, Neel Joshi, Zaki Lakhani, Alvira Tyagi, Tilly Krishna, Jack Wyss, and Edward Dong—thank you for being there when I needed comfort the most. Finally, To Aidan Doyle and Suhan Suresh-thank you for being home.

I am also grateful to the faculty and staff of the Department of Computer Science and the Committee on Degrees in Neuroscience for cultivating an intellectually stimulating environment that has shaped my academic journey. Their expertise and generosity in sharing knowledge have been instrumental in my education at Harvard.

Finally, this research was funded by Harvard University's Kempner Institute for the Study of Natural and Artificial Intelligence through the KURE and KRANIUM grants, which also provided computational resources, mentorship, and institutional support.

Contents

1 Introduction **8**
- 1.1 Cognitive Neuroscience Motivation . . . 9
- 1.2 Reinforcement Learning Motivation . . . 10
- 1.3 Agent Discovery as Variational Inference . . . 10

2 Related Works **12**
- 2.1 Foundations of Agency Perception in Cognitive Science . . . 12
  - 2.1.1 Classic Experimental Paradigms . . . 13
  - 2.1.2 Theoretical Frameworks for Agency Perception . . . 15
- 2.2 Neuroscientific Basis for Agency Perception . . . 19
  - 2.2.1 Neural Pathways Contributing to Agency Perception . . . 19
  - 2.2.2 Mirror Neurons . . . 20
  - 2.2.3 Developmental Trajectories of Social Perception . . . 22
  - 2.2.4 Clinical Evidence . . . 23
  - 2.2.5 Advanced Methodological Approaches . . . 23
  - 2.2.6 Integrative Perspectives . . . 24
- 2.3 A Primer on Variational Inference . . . 25
  - 2.3.1 The Challenge of Posterior Inference . . . 25
  - 2.3.2 Variational Inference: An Optimization Approach . . . 25
  - 2.3.3 Deriving the Evidence Lower Bound . . . 26
  - 2.3.4 Understanding the ELBO Components . . . 27
  - 2.3.5 Deep Variational Inference: Autoencoders . . . 29
  - 2.3.6 The Reparameterization Trick . . . 29
  - 2.3.7 Practical Applications and Extensions . . . 30
- 2.4 Computational Agent Perception Models . . . 31
  - 2.4.1 Bayesian Models of Goal Attribution . . . 31
  - 2.4.2 Cognitive Architecture for Animacy Detection . . . 31
  - 2.4.3 Generative Models for Active Vision . . . 31
  - 2.4.4 Our Approach: Variational Agent Discovery . . . 32
- 2.5 Object-Centric Representation Learning: Foundations for Agent-Centric Models 33
  - 2.5.1 From Convolutional Neural Networks to Object-Centric Representations 33
  - 2.5.2 Attention Mechanisms and Slot Attention . . . 33
  - 2.5.3 Extending Slot-Based Models to Video . . . 34
- 2.6 Reinforcement Learning . . . 36

2.6.1 Foundations of Reinforcement Learning and World Models 36
2.6.2 Policy Gradient Methods and PPO 36
2.6.3 World Models and Model-Based RL 38
2.6.4 Multi-Agent Reinforcement Learning 39
2.6.5 Agent Modeling and Theory of Mind 40

3 Method **42**
3.1 Background and Problem Statement 42
3.2 Structured Variational Approach to Agent Discovery 43
3.3 Derivation of the Variational Agent Discovery Objective 43
3.4 Implementation Details 47
3.4.1 Architecture Overview 47
3.4.2 Object-Centric Representation Learning 48
3.4.3 Variational Action Inference 48
3.4.4 Spatial Broadcast Decoder 50
3.4.5 Training 51
3.4.6 XLA Acceleration 52

4 Experimental Evaluation **54**
4.1 Agent-centric Slot Representations and Generalization 54
4.1.1 Evaluation Methodology 55
4.1.2 Experimental Environments 55
4.1.3 Representation Quality and Generalization Results 58
4.1.4 Qualitative Analysis of Learned Representations 60
4.1.5 Rational Action Prediction 61
4.2 Improving Multi-Agent Reinforcement Learning 62
4.3 Mirror-Like Neural Representations in Slot Activations 63
4.3.1 Evaluation Methodology 63
4.3.2 Results 64
4.4 Summary of Findings 66

5 Discussion **69**
5.1 Generality of the $\mathcal{L}_{\text{VAD}}$ Objective 69
5.2 Agent-Centric Representations 70
5.2.1 Predictive Power for Agent Properties 70
5.2.2 Functional Entity Decomposition 71
5.2.3 Generalization to Novel Scenarios 71
5.3 Rational Action Understanding 72
5.4 Improving Multi-Agent Reinforcement Learning 72
5.5 Mirror-Like Neural Representations 72
5.6 Limitations and Future Work 73
5.6.1 Action Representation Limitations 73
5.6.2 Generalization to Novel Agent Structures 74
5.6.3 MARL Generalization Evaluation 74
5.6.4 Scaling to More Complex Environments 74

5.6.5 Mirror Neuron Analysis Limitations 75
5.6.6 Comparison with Language-Based Social Reasoning 75
5.6.7 Limited Complexity of Social Understanding 75
5.7 Connections to Cognitive Science and Future Validation 76
5.7.1 Potential Connections to Neural Mechanisms 76
5.7.2 Future Neuroscientific Validation 76

6 Conclusion **78**

To **చెన్న రెడ్డి తాత** *(Chenna Reddy Tata) and* **చంద మామ** *(Chanda Mama), whose quiet yes echoed louder than a thousand nos, and paved the path beneath my feet.*

Chapter 1

Introduction

What constitutes an agent? Look at a sequence of frames from Heider and Simmel's classic 1944 experiment (Figure 1.1). The experiment consists of both moving and static geometric shapes: two triangles, a circle, and a rectangular box. *Which of these geometric objects are agents?*

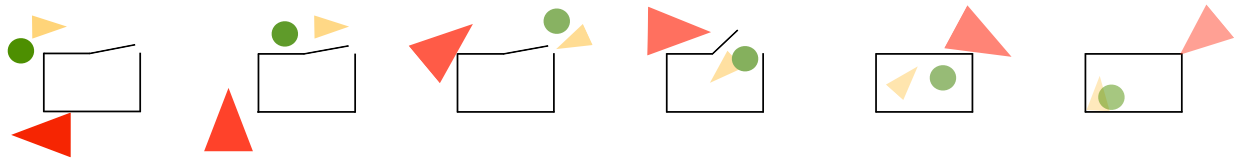

Figure 1.1: Drawing of a sequence of frames from Heider and Simmel (1944)'s experiment.

Even without animation, most observers intuitively identify the triangles and circle as intentional agents while perceiving the rectangular walls as inanimate objects. This phenomenon demonstrates our remarkable ability to distinguish agents from non-agents based on minimal visual information, without requiring biological features like faces or limbs. *Could a deep learning or AI model do the same?*

When we observe moving entities, our brains analyze motion patterns, identifying signatures of agency. Self-propelled movement, sudden changes in direction, and contingent reactivity to other entities trigger neural processing that leads us to perceive intention beyond mere motion (Heider and Simmel, 1944; Michotte, 1964; Scholl and Gao, 2013). As Scholl and Tremoulet (2000) argue, agency detection engages specialized visual processing mechanisms triggered by specific motion cues that violate expected physical dynamics. This perception can also allows us to attribute goals to entities based on their movement patterns Gergely and Csibra (2003).

Our ability to detect and represent agents is fundamental to navigating our social world. From predator avoidance to social cooperation, from interpreting communicative gestures to predicting others' behavior, agent recognition supports numerous aspects of social cognition. Neuroscientific evidence indicates that our brains utilize multiple regions for this task—including the superior temporal sulcus (STS) which processes movement trajectories, and the mirror neuron system, which activates similarly whether we perform an action or observe others performing it (Frith, 2007; Fulvia Castelli, 2000).

Despite this foundational role in human cognition, artificial intelligence models struggle with perceiving agency from visual input. While modern deep learning approaches have successfully modeled physical object interactions, scene representations, and object recognition, these models lack an integrated understanding of agency (Brenden M. Lake, 2017). They may identify people or animals in scenes and label their actions based on visual features, but cannot fundamentally distinguish between an entity acting with intention and one merely following physical dynamics. Even state-of-the-art multi-agent reinforcement learning (MARL) algorithms typically rely on hard-coded agent representations rather than discovering them from sensory data, limiting their generalization capabilities. *How might we formalize the computations that transform visual input into representations of agency?*

This thesis introduces a variational objective $\mathcal{L}_{\text{VAD}}$ for *unsupervised agent discovery* from visual observations. Based on this objective, we develop a deep conditional variational slot-based autoencoder, which we call the **Variational Agent Discovery (VAD) model**. The VAD model operates directly on high-dimensional visual input, processing pixel data to learn agent-centric representations—advancing beyond existing approaches that operate on simplified state representations. By formulating agent discovery as a structured variational inference problem with latent actions, our model decomposes visual scenes into entity-centric representations and infers the latent actions that best explain observed state transitions, distinguishing entities whose behavior follows intentional policies from those governed by environmental dynamics.

Our VAD model learns agent-centric representations across environments of increasing complexity and exhibits properties analogous to human cognitive mechanisms. We discover that individual features within our model's representations activate consistently across different agents performing the same action—similar to mirror neurons in the primate brain. Furthermore, the model demonstrates the ability to predict rational action, correctly anticipating that agents will take direct paths to goals when obstacles are removed—paralleling expectations documented in infant studies of teleological reasoning and agent perception. By formulating agent discovery as a variational inference problem, we not only learn meaningful representations of agents but also provide an auxiliary objective $\mathcal{L}_{\text{VAD}}$ that improves learning efficiency in multi-agent reinforcement learning settings.

1.1 Cognitive Neuroscience Motivation

From a cognitive and computational neuroscience standpoint, we seek to propose a quantitative, falsifiable hypothesis about agent perception in the form of an optimization objective. Recent work by Cao and Yamins (2021a,b) suggests that understanding neural mechanisms requires computational models that both explain how the brain processes information and are understandable to humans—what they term "cognitive manipulability." Their Contravariance Principle suggests that models tackling realistic complex tasks without oversimplification are more likely to discover solutions similar to those used by biological systems.

While several computational models of agency perception exist, including Gao et al. (2019), Baker et al. (2009), and Ullman et al. (2009), these models typically operate on simplified symbolic inputs rather than raw perceptual data. By formulating agent discovery as a variational inference problem that operates directly on high-dimensional visual data, we

propose an optimization objective $\mathcal{L}_{\text{VAD}}$ that addresses the transformation of visual input into agent representations without relying on pre-processed information, creating a testable hypothesis about the computational principles underlying agency perception.

1.2 Reinforcement Learning Motivation

From a machine learning perspective, agent-centric representations can improve multi-agent reinforcement learning algorithms. Current MARL approaches face a limitation: they typically rely on privileged information about other agents. Algorithms often assume direct access to other agents' states or explicitly encode agent representations within the observation space. This approach simplifies training but undermines generalization when the number of agents changes or when agents adopt novel behaviors.

Agents equipped with the ability to autonomously model other agents could adapt more readily to novel multi-agent scenarios. Rather than encoding fixed assumptions about other agents' capabilities or goals, such algorithms would flexibly construct models of previously unseen agents through observation alone. This would enable generalization to environments with different numbers of agents or agents with different capabilities—a crucial requirement for deploying autonomous policies in open-world settings.

Furthermore, agent-centric world models may improve sample efficiency in MARL. By explicitly modeling other agents as entities with goals and policies rather than as part of the environmental dynamics, learning algorithms could make more accurate predictions about future states, enabling more effective planning and exploration.

1.3 Agent Discovery as Variational Inference

In this thesis, we introduce a probabilistic approach for discovering agent representations from visual observations without supervision.

Our key assumption is that agent behavior involves decision-making processes that can be modeled through latent action variables. Unlike physical objects whose transitions follow deterministic laws, agents generate actions from internal policies that can be inferred from observations. By explicitly modeling these latent actions and learning to infer them from state transitions, our approach creates an inductive bias toward learning representations of entities whose behavior exhibits agency.

Mathematically, we formalize this intuition through variational inference. The true transition probability between entity states requires marginalizing over all possible unobserved actions—an intractable computation. We address this through a variational lower bound on the marginal likelihood, introducing an approximate posterior in our VAD model that estimates the most likely actions given observed transitions.

Our experimental results demonstrate the VAD model's capability to learn agent representations across environments of increasing complexity—from single-agent navigation in grid worlds to two-agent cooperative tasks and three-agent competitive scenarios. Moreover, the model learns consistent agent representations that generalize to novel scenarios and novel

agents, and when our $\mathcal{L}_{\text{VAD}}$ objective is incorporated into reinforcement learning policies, it improves their sample efficiency.

Key Contributions

1. A computational objective $\mathcal{L}_{\text{VAD}}$ for agent discovery formulated as a variational inference problem that operates directly on visual input, providing a quantitative, falsifiable optimization objective.
2. A VAD model that successfully learns agent-centric representations across environments of increasing complexity and demonstrates generalization to novel agents, goals, and environmental configurations.
3. Evidence of mirror-neuron-like representational properties, where specific features in the VAD model activate consistently across different agents performing the same actions.
4. Teleological-like reasoning capability, where the VAD model predicts rational actions in novel scenarios similar to expectations in human infant studies.
5. Improved sample efficiency in multi-agent reinforcement learning when our $\mathcal{L}_{\text{VAD}}$ objective is incorporated as an auxiliary task.

In the following chapters, we begin by reviewing foundations of agent perception theories, object-centric representation learning, reinforcement learning, and variational inference. We then derive our variational agent discovery objective $\mathcal{L}_{\text{VAD}}$ and describe the implementation of our VAD model. Finally, we present experimental results demonstrating the model's capabilities across a range of environments and tasks.

This thesis aims to inspire the development of artificial agents who can navigate the social world with flexibility and intuition, perceiving and understanding other agents as naturally as humans do.

Chapter 2

Related Works

2.1 Foundations of Agency Perception in Cognitive Science

What constitutes an agent, and how do we recognize agency in our environment? These questions have intrigued scientists across disciplines for decades. For humans, the ability to identify agents appears effortless and automatic. We readily distinguish between a falling rock (a passive physical entity) and a descending eagle (an intentional agent), despite both exhibiting downward motion trajectories. This perceptual capacity holds immense adaptive value: accurately representing agents in our environment enables us to predict their behavior, avoid threats, and engage in social interaction. The scientific investigation of agency perception has progressed through systematic exploration of when humans perceive agency, what visual cues trigger it, and how this capacity develops.

Key Terminology

In cognitive science literature, agency perception encompasses several related phenomena characterized by distinct processing levels:
Animacy perception refers to the detection that an entity is alive and self-propelled. The perceptual signatures of animacy include self-initiated movement, violations of Newtonian physics (like sudden stops or direction changes), and movement that lacks obvious external causes. This represents the most basic level of agency detection and can be triggered by simple motion cues.
Goal attribution involves inferring that an entity's movements are directed toward achieving specific outcomes. This builds upon animacy perception but goes further, as it requires understanding that movements aren't just self-generated but purposeful. Key cues include efficiency of movement relative to environmental constraints and persistent orientation toward specific targets.
Intentionality extends goal attribution to include the inference that an agent has internal mental states driving its behavior. This involves representing an entity as capable of making decisions among alternative actions based on its goals and beliefs about the environment.
These concepts form a processing hierarchy, where higher levels build upon but go beyond the information provided by lower levels. This thesis focuses primarily on the first two levels, examining how visual processing mechanisms extract agency cues from motion patterns and how these support the attribution of goals and prediction of behavior.

2.1.1 Classic Experimental Paradigms

The scientific investigation of agency perception has progressed through systematic exploration of several fundamental questions. This section reviews the classic experimental paradigms that have shaped our understanding of how humans detect and interpret agency from visual motion.

When do humans perceive agency from visual input?

The foundational work on agency perception was conducted by Heider and Simmel (1944), who presented participants with a simple animation depicting geometric shapes—two triangles and a circle—moving around a rectangular enclosure. Despite the minimalist nature of these stimuli, participants spontaneously described the shapes as animate entities engaged in intentional interactions, attributing to them emotions, goals, and even personality traits. Viewers typically characterized the large triangle as "bullying" or "chasing" the smaller shapes, and described the small triangle and circle as "hiding" or "escaping." This striking demonstration revealed that humans do not require biological features (such as faces or limbs) to perceive agency; motion patterns alone can trigger rich social interpretations. Indeed, it's a captivating animation, despite being so visually minimalistic.

The power of these simple displays to evoke consistent social interpretations has made them an enduring paradigm in psychology. Heider and Simmel's original study demonstrated

that specific motion patterns—such as contingent movement, apparent pursuit, and coordinated actions—are sufficient to trigger robust perceptions of animacy, intentionality, and even complex social relationships. This work established that agency perception can be studied systematically using controlled stimuli, opening the door to more detailed investigations of the specific visual cues that drive this phenomenon.

What specific motion patterns and cues trigger agency perception?

Following Heider and Simmel's pioneering work, researchers have systematically investigated the specific motion characteristics that elicit perceptions of agency. Tremoulet and Feldman (2000) demonstrated that even a single dot moving across a screen can appear animate when it changes direction or speed in certain ways, suggesting that self-propulsion serves as a basic cue for animacy perception.

Research on perceived chasing has been particularly informative about the specific parameters that influence agency detection. Gao et al. (2009) conducted systematic psychophysical studies of chasing perception, showing that detection follows a U-shaped function relative to directness of pursuit. When a "wolf"shape pursues a "sheep" with slight deviations from a perfect direct path, chasing is most easily perceived. Counterintuitively, perfectly direct pursuit and highly variable pursuit both reduced participants' ability to detect chasing, suggesting that agency perception is tuned to specific patterns of motion that characterize natural predator-prey interactions.

Dittrich and Lea (1994) identified several features that contribute to the perception of intentionality in motion, including acceleration, changes in direction related to another object's position, and contingent responsiveness. Similarly, Gao and Scholl (2010) demonstrated the "wolfpack effect," showing that when multiple objects consistently orient toward a target (like a pack of predators), viewers perceive intentionality even when the objects' actual trajectories are entirely random. These findings suggest that orientation cues serve as a powerful trigger for agency perception, independent of actual motion trajectories.

Collectively, these studies have identified several key motion cues that reliably trigger agency perception: self-propulsion, non-Newtonian changes in direction or speed, orientation toward targets, contingent responsiveness to other objects, and patterns of pursuit or evasion. Importantly, these cues appear to operate in a fast, automatic manner that is resistant to conscious override, suggesting they may be processed by specialized perceptual mechanisms.

How universal and developmentally early is agency perception?

A third major question concerns the universality and developmental trajectory of agency perception. Cross-cultural studies have investigated whether the tendency to perceive agency from motion cues varies across human populations. Barrett et al. (2005) found consistent interpretations of intention from motion cues across diverse cultures, including hunter-gatherer societies with minimal exposure to Western media. Similarly, Morris and Peng (1994) demonstrated similar attributions of agency across American and Chinese observers. These findings suggest that the perception of animacy from motion might be a universal human capacity, though cultural factors may influence the specific intentions attributed (Rimé et al., 1985).

Developmental research has shown that agency perception emerges remarkably early

in human development. Simion et al. (2008) demonstrated that even 2-day-old newborns show a preference for biological motion in point-light displays, can discriminate between different motion patterns, and show orientation-specific responses. By 5 months of age, infants develop more sophisticated processing of biological motion, including sensitivity to specific biomechanical properties like joint rigidity in point-light walker displays (Bertenthal et al., 1987). This sensitivity is orientation-specific, suggesting specialized processing of upright human figures. As infants develop further, by 6-12 months, they attribute goals to moving agents based on their patterns of motion (Gergely, 1995; Csibra, 1999) and expect agents to take efficient paths toward goals (Gergely and Csibra, 2003).

Frankenhuis and Barrett (2013) have proposed that early action understanding may be centered on domain-specific action schemas that guide attention toward domain-relevant events. Examining chasing as a case study, they suggest that natural selection may have built "islands of competence" in early action understanding that serve as foundations for future learning and development. This evolutionary developmental perspective suggests that learning mechanisms may have evolved to exploit recurrent properties of fitness-relevant domains, with specialized attention to particular kinds of motion patterns that have been evolutionarily significant.

The early emergence and cross-cultural consistency of agency perception raises important questions about the underlying mechanisms that support this capacity. How do we transform simple motion cues into rich perceptions of agency? What processes enable this transformation? The next section examines theoretical frameworks that attempt to address these questions.

2.1.2 Theoretical Frameworks for Agency Perception

Building on the experimental paradigms reviewed in the previous section, we now examine theoretical frameworks that explain the mechanisms underlying agency perception. These frameworks address how the brain transforms motion cues into representations of agents with goals and intentions.

What mechanisms underlie our perception of agency?

Agency perception can be understood as operating across multiple processing levels, from rapid perceptual detection to more sophisticated cognitive inference. These levels form a processing hierarchy that transforms visual input into increasingly abstract representations:

Perceptual Detection of Agency Cues At the most basic level, specific motion patterns automatically trigger agency detection. Scholl and Gao (2013) argue that this occurs through specialized visual processing mechanisms sensitive to motion signatures such as self-propulsion, acceleration changes, and contingent responsiveness. This processing is rapid, automatic, and operates directly on low-level visual features without requiring conscious inference.

Attribution of Goals and Intentions Building on perceptual detection, we attribute goals and intentions to perceived agents. Dennett (1988) characterized this as adopting the

"intentional stance"—a predictive strategy in which we treat an entity as a rational agent with goals and intentions. According to Dennett, this stance is adopted when it enables more accurate and efficient behavioral predictions than alternative strategies.

Research with infants supports this distinction between lower-level agency detection and higher-level goal attribution. Woodward (1998) demonstrated that 6-month-old infants attribute goals to human hands but not to mechanical claws performing identical movements, suggesting that goal attribution builds upon, but goes beyond, mere motion analysis. Similarly, Johnson et al. (1998) showed that 12-month-olds follow the 'gaze' of a novel amorphous object that has facial features or behaves contingently, but not when it lacks both characteristics. This indicates that infants are sensitive to specific morphological and behavioral cues when attributing agency and intentionality to objects..

Representation of Mental States While not the focus of this thesis, it's worth noting that agency perception provides the foundation for more complex social-cognitive processes like theory of mind—the representation of others' beliefs, desires, and other mental states (Baron-Cohen et al., 1985; Wimmer and Perner, 1983). While early theories suggested theory of mind emerges suddenly around age four (Wellman et al., 2001), recent work indicates that more implicit forms may exist earlier in development (Onishi and Baillargeon, 2005; Southgate et al., 2007), suggesting a more continuous developmental trajectory from basic agency perception to sophisticated mental state reasoning.

These processing levels interact and mutually constrain each other—perceptual detection influences higher-level attributions, while conceptual knowledge can shape what motion patterns are detected as agent-like.

What evidence suggests agency detection operates at a perceptual level?

A substantial body of research supports the view that the initial detection of agency involves specialized perceptual processing rather than solely higher-level reasoning. Scholl and Gao (2013) marshal five lines of evidence supporting this "social vision" interpretation:

Phenomenology of Visual Experience Agency perception has an immediate, compelling quality similar to other visual percepts. Just as we directly "see" color or depth rather than inferring them, we directly "see" animacy and intentionality in appropriately structured motion displays. This phenomenology persists even when observers know the displays consist only of simple geometric shapes, suggesting a level of cognitive impenetrability characteristic of perceptual processes.

Sensitivity to Subtle Visual Parameters Agency perception exhibits sensitivity to fine-grained visual parameters in ways that would be difficult to explain through deliberate reasoning. For example, Gao et al. (2009) demonstrated that chasing detection follows a U-shaped function relative to pursuit directness, with optimal detection occurring with slight deviations rather than perfect pursuit. This nuanced psychophysical function suggests perceptual tuning to specific motion signatures rather than categorical judgments.

Implicit Influences on Visual Performance Agency perception automatically influences visual performance even when irrelevant or detrimental to the task at hand. Gao and Scholl (2010) demonstrated the "wolfpack effect" where performance was significantly impaired when darts pointed toward a user-controlled object compared to identical motion trajectories with different orientations. This occurred despite explicit instructions to ignore orientation and even when participants were fully informed about the effect beforehand. Additional studies by Van Buren et al. (2016) and Pratt et al. (2010) have shown that perceived animacy automatically captures visual attention, further demonstrating its automatic influence on visual processing.

Activation of Visual Brain Areas Neuroimaging studies have identified selective activation of visual processing regions during agency perception. Gao et al. (2012) demonstrated that the motion-selective region MT+ shows differential activation in response to displays that evoke percepts of animacy, even when controlling for lower-level motion parameters. This suggests that agency detection involves specialized visual processing rather than only higher-level reasoning areas (more on this later in Ch. 2.2). Schultz et al. (2005) similarly found that posterior STS activation correlates with parametric manipulations of perceived animacy, further supporting the involvement of specialized visual processing.

Interaction with Other Visual Processes Agency perception interacts with other visual processes in ways characteristic of perceptual rather than cognitive phenomena. New et al. (2010) demonstrated that animacy perception operates even in individuals with autism spectrum disorders who show impairments in higher-level social cognition, suggesting its operation at a more basic perceptual level. Meyerhoff et al. (2014) showed that perceived chasing influences multiple object tracking performance even under high cognitive load, consistent with automatic perceptual processing.

These lines of evidence collectively suggest that the detection of agency from visual cues involves specialized perceptual processing that is fast, automatic, and sensitive to specific motion parameters.

Is goal attribution perceptual or cognitive?

An intriguing question that extends beyond basic agency detection is whether goal attribution itself might operate partly at a perceptual level rather than being entirely a higher-level cognitive process. This question challenges the traditional distinction between perception (detecting agents) and attribution (inferring goals).

Gao et al. (2009) found that the perception of chasing is automatic and difficult to suppress, suggesting that not just agency detection but also the specific attribution of pursuit goals might operate at a perceptual level. Similarly, Scholl and Tremoulet (2000) showed that even complex social percepts like "stalking" versus "following" might be processed through specialized perceptual mechanisms.

More recently, Rolfs et al. (2013) demonstrated adaptation effects for causal perception that are retinotopically specific, suggesting that even seemingly high-level properties like causality may be encoded at a visual processing level. If causality can be perceptual, it raises the possibility that goal attribution might similarly have perceptual components.

The "perceptual animacy" account proposed by Scholl and Gao (2013) suggests that the distinction between perception and attribution may be more fluid than traditionally assumed. Certain forms of attribution—particularly those related to basic goals like chasing, avoiding, or helping—may operate through specialized perceptual mechanisms rather than requiring explicit reasoning.

How do perception and attribution interact in predicting agent behavior?

The relationship between agency perception and goal attribution enables us to predict agent behavior based on attributed goals and environmental constraints. The teleological stance theory, proposed by Gergely and Csibra (2003), provides a framework for understanding this predictive capacity.

According to this theory, observers interpret actions through a three-part representational structure connecting: (1) the observed action, (2) the future goal state, and (3) the situational constraints. This structure enables both goal attribution (inferring goals from observed actions and constraints) and action prediction (anticipating actions based on known goals and constraints). Central to this framework is the principle of rational action—the assumption that agents take the most efficient path to achieve their goals given environmental constraints.

Csibra (1999) demonstrated that infants attribute goals to agents based on the rationality of their actions rather than their perceptual features. When observing an agent taking an unusual path to reach a target, infants interpreted this as rational if an obstacle was present but showed surprise if the same path was taken with no obstacle. This suggests that rationality evaluation guides goal attribution, illustrating how attribution builds upon but goes beyond mere perceptual detection.

In 4.1.5, we closely model this experiment, and show that our model trained with our variational objective accurately predicts that agents will take direct paths to goals when obstacles are removed—mirroring the expectations documented in infant studies.

2.2 Neuroscientific Basis for Agency Perception

Neuroscience research has identified neural systems that process social and intentional information, with many of these systems also contributing to non-social cognitive functions. This section explores the key neural substrates and clinical evidence informing our understanding of agency perception.

2.2.1 Neural Pathways Contributing to Agency Perception

Multiple neural systems support different aspects of agent perception, from detecting biological motion to inferring goals and mental states, though many of these systems also serve broader cognitive functions.

The Social Brain Network

The **Superior Temporal Sulcus (STS)** plays a particularly critical role in processing biological motion and intentional action. This region, located along the lateral surface of the temporal lobe, activates selectively when observing goal-directed movements and biological motion patterns (Frith, 2007). Allison et al. (2000) demonstrated that the **posterior STS (pSTS)** responds preferentially to biological motion compared to non-biological motion, even when the stimuli are highly simplified point-light displays (Johansson, 1973). Similarly, Saxe et al. (2004) found that the pSTS is specifically activated when observing intentional actions. Importantly, Gao et al. (2012) found that the right pSTS shows dissociable patterns of activity for animacy versus intentionality, suggesting functional specialization within this region for different aspects of agent perception.

Blakemore et al. (2003) demonstrated a key neural dissociation between perceiving physical and social causality. Using stimuli similar to Michotte (1964)'s launching displays, they found distinct patterns of brain activation when observers perceived social interaction (contingent motion) versus physical causation (launching). Social perception recruited regions including the STS, while physical causality activated areas associated with visual motion processing. This dissociation suggests that the brain processes social and physical events through partially separate pathways, although these pathways may rely on similar computational principles operating on different types of input.

The STS does not operate in isolation but functions as part of a broader network that Frith (2007) termed the "social brain," building on earlier work by Brothers (2002). As Frith emphasizes, many components of this network are not exclusively dedicated to social cognition but rather perform computations that are particularly useful for social interaction while also serving other cognitive functions.

The **Temporal Parietal Junction (TPJ)** is activated during perspective-taking tasks and when inferring others' mental states or false beliefs (Saxe and Kanwisher, 2003; Frith, 2007). However, the TPJ is also implicated in non-social attentional processes and has been characterized as playing a domain-general role in reorienting attention (Corbetta et al., 2008). The **Medial Prefrontal Cortex (MPFC)** constitutes another critical component, playing a key role in representing others' mental states and intentions (Mitchell et al., 2006; Amodio and Frith, 2006; Frith, 2007). Frith notes that the anterior rostral MPFC may have

a specialized role in handling communicative intentions and second-order representations of mental states (representing someone else's representation of our mental state). Recent work by Schurz et al. (2014) suggests that MPFC involvement in mentalizing tasks reflects domain-general processes related to scene construction and self-projection rather than social cognition specifically.

An important but often overlooked component of the social brain is the **Temporal Poles**. As Frith describes, these regions function as "convergence zones" where information from different modalities comes together to define unique individuals and situations (Frith, 2007). The temporal poles store our rich social knowledge—facts about specific people, appropriate behaviors for different situations, and how emotions affect behavior in various contexts. This stored knowledge allows us to apply our past social experiences to new situations. However, this integrative function applies to both social and non-social semantic knowledge (Patterson et al., 2007).

The Amygdala's Role in Social and Emotional Processing

The **Amygdala** also plays a crucial role in social perception, though as Frith explicitly emphasizes, its function is not exclusively social. The amygdala is involved in attaching emotional value to stimuli through conditioning processes (Frith, 2007; Adolphs, 2010). It responds to faces rated as untrustworthy and is implicated in the automatic, implicit aspects of prejudice. The amygdala's role in recognizing expressions of fear likely stems from its more general function in associating stimuli with potential threats. This perspective suggests that while the amygdala contributes significantly to social cognition, it does so through domain-general mechanisms that apply equally to social and non-social stimuli.

2.2.2 Mirror Neurons

A particularly intriguing neural system implicated in agency perception is the **mirror neuron system**. First discovered in macaque monkeys (Rizzolatti et al., 1996), mirror neurons fire both when an animal performs an action and when it observes another agent performing the same action. In humans, homologous mirror systems have been identified in the premotor cortex and inferior parietal lobule (Molenberghs et al., 2012). This system may provide a neural mechanism for action understanding through motor simulation: by activating our own motor representations when observing others' actions, we can understand those actions "from the inside" (Rizzolatti and Craighero, 2004).

Cross-Species and Cross-Domain Generalization The mirror neuron system demonstrates remarkable flexibility across different agent types. Gallese et al. (1996) first showed that monkey mirror neurons respond when monkeys observe humans performing grasping actions, indicating cross-species mirroring. Building on this, Buccino et al. (2004) found that the human mirror system activates when observing actions performed by non-human species, though with varying intensity depending on the similarity to human actions. For actions within the human motor repertoire (like a monkey's lip-smacking), activation was stronger than for actions outside it (like a dog's barking), suggesting that mirror responses are tuned to actions that map onto the observer's own motor capabilities.

Intriguingly, Gazzola et al. (2007) demonstrated that the human mirror system can even generalize to robotic actions. When participants observed a robotic arm performing goal-directed actions, their premotor and parietal mirror areas showed activation comparable to when observing human actions. This suggests that the goal or intentional structure of an action may be more critical for mirror system engagement than the biological nature of the agent performing it.

White et al. (2014) further supported this finding using transcranial magnetic stimulation, showing motor facilitation effects when humans observed actions performed by non-human animals (rats and elephants) and robotic arms. In some cases, observing non-human animals actually produced stronger motor resonance than observing humans, indicating complex tuning that may be sensitive to the distinctive mechanics of different agents' actions.

Action-Specific Encoding An important aspect of mirror neurons is their action specificity. Rather than responding to all observed movements, mirror neurons show selective activation for particular actions. For example, different populations of mirror neurons in macaque F5 respond specifically to grasping, holding, or tearing actions. This action-specific encoding allows for precise mapping between observed and executed actions, facilitating detailed understanding of others' behaviors.

Connection to Our Computational Model

Theoretical work by Kilner et al. (2007b,a), Frith (2007), and Friston et al. (2011) has proposed that the mirror neuron system can be understood within a predictive coding framework based on empirical Bayesian inference. According to this view, the mirror system helps us infer the intentions behind observed actions by minimizing prediction errors across the cortical hierarchy. This addresses a fundamental problem in action understanding: identical movements can serve different intentions, making the mapping from observation to intention ambiguous. Through predictive coding, the brain can infer the most likely cause of an observed movement by generating predictions at multiple levels and updating these predictions based on error signals.

Our computational VAD model (described in Ch 3.4) addresses a similar inference challenge—determining the latent causes (actions) that explain observed entity state transitions. Both our variational approach and predictive coding frameworks minimize prediction errors, though through different mathematical formalisms. Our $\mathcal{L}_{\text{VAD}}$ objective contains a reconstruction term that encourages accurate prediction of state transitions given inferred actions, which conceptually aligns with the error-minimization principle in predictive coding.

Interestingly, our VAD model's learned representations exhibit mirror neuron-like properties analogous to those observed in biological systems. As demonstrated in Section 4.3, when analyzing slot vector activations across different agents performing the same actions, we found specific feature dimensions (e.g., F57 for rightward movement, F107 for upward movement) that consistently activate across different agent representations. This suggests our model develops a shared neural code for actions that generalizes across entities, similar to biological mirror neurons. Furthermore, like the cross-species generalization capabilities observed by Gazzola et al. (2007) and White et al. (2014), our VAD model successfully generalizes its action understanding to novel agents not seen during training, maintaining high prediction accuracy for both familiar and novel entities.

2.2.3 Developmental Trajectories of Social Perception

The developmental trajectory of social perception systems offers insights into how these neural networks emerge. Grossman et al. (2000) found that sensitivity to biological motion in the STS emerges early but continues to refine with experience. This aligns with evidence that basic biological motion detection appears in early infancy (Simion et al., 2008) but undergoes substantial development through childhood. Similar developmental patterns have been observed for mirror neuron responses, suggesting parallel maturation of interconnected systems for social perception. The question of how much of this neural architecture is innately specified versus shaped by experience remains debated and has implications for computational modeling approaches that aim to recapitulate the development of these systems.

2.2.4 Clinical Evidence

Neuropsychological evidence from clinical populations offers a complementary perspective on the neural basis of agency perception. In particular, studies of individuals with autism spectrum disorder (ASD) and amygdala damage have provided insights into how different neural systems contribute to social perception.

Autism Spectrum Disorders

Individuals with **Autism Spectrum Disorder (ASD)** often show altered patterns of social perception, though the specifics vary considerably across the spectrum. Abell et al. (2000) found that children with autism attributed fewer mental states to animated geometric shapes in Heider-Simmel-type displays compared to typically developing children. Similarly, Klin (2000) developed the Social Attribution Task, which revealed that individuals with high-functioning autism used significantly fewer social attributions when describing such animations. Rutherford et al. (2006) demonstrated that children with autism showed reduced perception of animacy from motion cues, suggesting difficulties with the perceptual foundations of agency detection.

Amygdala Lesion Studies

Studies of individuals with **amygdala damage** provide further evidence for the role of this structure in social perception. The amygdala, an almond-shaped structure in the medial temporal lobe, plays a key role in emotional processing and social evaluation. Heberlein and Adolphs (2004) found that a patient with bilateral amygdala damage showed severely impaired spontaneous anthropomorphizing when viewing geometric shapes in motion, despite preserved general intelligence and basic motion perception. This suggests that the amygdala may be critical for the social interpretation of motion cues. Similarly, Adolphs et al. (1998) demonstrated the amygdala's importance in social judgment, showing that patients with amygdala damage had difficulty evaluating the trustworthiness of faces, an observation also highlighted by Frith in his discussion of the amygdala's role in social cognition.

2.2.5 Advanced Methodological Approaches

Approaches using **multivariate pattern analysis (MVPA)** in functional neuroimaging have further refined our understanding of how social information is represented in the brain (Brooks and Freeman, 2017). Rather than simply localizing social perception to specific brain regions, MVPA examines the patterns of activity within and across regions, revealing how social categories, identities, and emotions are encoded. This research has shown that the neural representations of social agents are structured along meaningful psychological dimensions (e.g., warmth-competence) and are influenced by both bottom-up perceptual features and top-down conceptual knowledge.

2.2.6 Integrative Perspectives

An important insight from Frith's analysis is that many components of the so-called "social brain" are not specifically social in function. The amygdala's role in conditioning, the temporal poles' function as convergence zones, and the TPJ's involvement in perspective-taking all have applications beyond social cognition. What makes these systems crucial for social understanding is not their exclusive dedication to social processing, but rather their recruitment and coordination in service of the particularly complex demands of social interaction. As Frith suggests, it may be that social complexity has driven these cognitive functions to higher levels of sophistication. This domain-general perspective has gained further support from recent meta-analyses showing substantial overlap between brain regions activated during social and non-social tasks (Van Overwalle, 2009; Spunt and Adolphs, 2015).

2.3 A Primer on Variational Inference

Variational inference provides a powerful mathematical framework for approximating complex probability distributions, serving as the theoretical foundation for many approaches to probabilistic modeling with latent variables. This section introduces the key concepts underlying variational methods, with a focus on their application to learning structured latent variable models from observations.

2.3.1 The Challenge of Posterior Inference

Let's begin with a fundamental problem in probabilistic modeling: given observed data x, *how can we infer the underlying latent variables z that might have generated it?* In Bayesian statistics, we are interested in computing the posterior distribution $p(z|x)$, which tells us how likely different values of z are, given our observation x.

Using Bayes' rule, we can write this posterior as:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} \tag{2.1}$$

Here, $p(x|z)$ is the likelihood model describing how latent variables generate observations, $p(z)$ is our prior belief about the latent variables before seeing any data, and $p(x)$ is the marginal likelihood (or evidence) of observing x under our model.

The denominator $p(x) = \int p(x|z)p(z)dz$ requires integrating over all possible configurations of latent variables—a computation that quickly becomes intractable as the dimensionality of z increases or when the likelihood model is complex. This integration challenge is the central problem that variational methods address.

2.3.2 Variational Inference: An Optimization Approach

Rather than computing the posterior exactly, variational inference reformulates inference as an optimization problem. The core idea is to approximate the true posterior $p(z|x)$ with a simpler distribution $q(z|x)$ from a tractable family, then find the member of this family that most closely resembles the true posterior.

The quality of this approximation is measured using the Kullback-Leibler divergence:

$$D_{KL}(q(z|x)\|p(z|x)) = \mathbb{E}_{q(z|x)}\left[\log \frac{q(z|x)}{p(z|x)}\right] \tag{2.2}$$

This divergence quantifies the information lost when using $q(z|x)$ to approximate $p(z|x)$. Our goal is to find the $q(z|x)$ that minimizes this divergence.

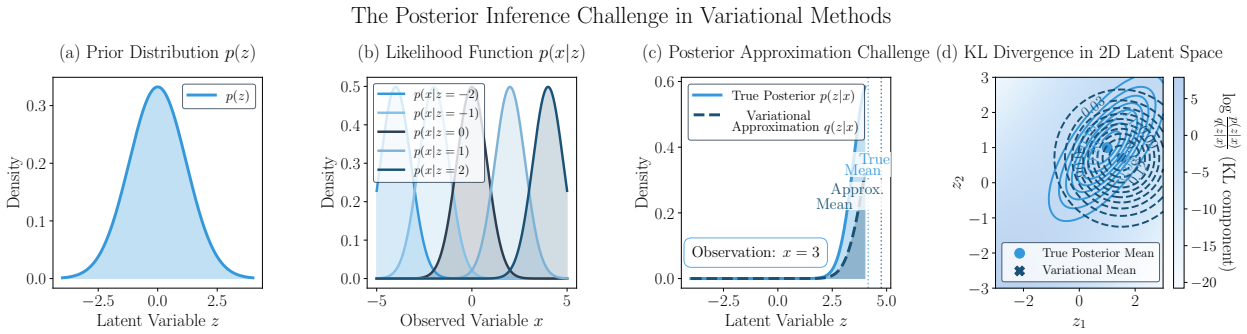

Figure 2.1: The posterior inference challenge in variational methods. (a) The prior distribution $p(z)$ represents our belief about latent variables before seeing data. (b) The likelihood function $p(x|z)$ defines how latent variables generate observations for different values of z. (c) The true posterior distribution $p(z|x)$ (solid blue line) arises from combining the prior and likelihood, but is often intractable to compute exactly. Variational inference approximates this with a simpler distribution $q(z|x)$ (dashed dark blue line). (d) In higher dimensions, this approximation challenge involves minimizing the KL divergence between the true and approximate posteriors, visualized here with contours and a heatmap in a 2D latent space.

Key Insight: From Sampling to Optimization

Variational inference transforms a difficult sampling problem into an optimization problem:

- Instead of sampling from the intractable $p(z|x)$
- We find a simpler distribution $q(z|x)$ that approximates it
- The approximation is optimized to minimize the KL divergence
- This allows us to leverage efficient optimization techniques

2.3.3 Deriving the Evidence Lower Bound

While we would like to minimize $D_{KL}(q(z|x)\|p(z|x))$ directly, we cannot compute it without knowing the true posterior. Let's expand this expression to see if we can find a more tractable objective:

$$D_{KL}(q(z|x)\|p(z|x)) = \mathbb{E}_{q(z|x)}\left[\log \frac{q(z|x)}{p(z|x)}\right] \tag{2.3}$$

$$= \mathbb{E}_{q(z|x)}\left[\log q(z|x) - \log p(z|x)\right] \tag{2.4}$$

$$= \mathbb{E}_{q(z|x)}\left[\log q(z|x) - \log \frac{p(x|z)p(z)}{p(x)}\right] \tag{2.5}$$

$$= \mathbb{E}_{q(z|x)}\left[\log q(z|x) - \log p(x|z) - \log p(z) + \log p(x)\right] \tag{2.6}$$

Since $p(x)$ does not depend on z, we can take it out of the expectation:

$$D_{KL}(q(z|x)\|p(z|x)) = \mathbb{E}_{q(z|x)}\left[\log q(z|x) - \log p(x|z) - \log p(z)\right] + \log p(x) \tag{2.7}$$

Rearranging, we get:

$$\log p(x) - D_{KL}(q(z|x)\|p(z|x)) = \mathbb{E}_{q(z|x)}\left[\log p(x|z) + \log p(z) - \log q(z|x)\right] \tag{2.8}$$

$$= \mathbb{E}_{q(z|x)}\left[\log p(x|z)\right] - \mathbb{E}_{q(z|x)}\left[\log \frac{q(z|x)}{p(z)}\right] \tag{2.9}$$

$$= \mathbb{E}_{q(z|x)}\left[\log p(x|z)\right] - D_{KL}(q(z|x)\|p(z)) \tag{2.10}$$

This final expression is the Evidence Lower Bound (ELBO), which we denote as $\mathcal{L}(q)$:

$$\mathcal{L}(q) = \mathbb{E}_{q(z|x)}\left[\log p(x|z)\right] - D_{KL}(q(z|x)\|p(z)) \tag{2.11}$$

From our derivation, we can see that:

$$\log p(x) = \mathcal{L}(q) + D_{KL}(q(z|x)\|p(z|x)) \tag{2.12}$$

Since the KL divergence is always non-negative, $\mathcal{L}(q)$ provides a lower bound on $\log p(x)$, hence the name "Evidence Lower Bound." Moreover, this bound becomes tight (equal to $\log p(x)$) when $q(z|x) = p(z|x)$, i.e., when our approximation perfectly matches the true posterior.

2.3.4 Understanding the ELBO Components

The ELBO consists of two distinct terms, each with an intuitive interpretation:

The Two Faces of the ELBO

$$\mathcal{L}(q) = \underbrace{\mathbb{E}_{q(z|x)}\left[\log p(x|z)\right]}_{\text{Reconstruction term}} - \underbrace{D_{KL}(q(z|x)\|p(z))}_{\text{Regularization term}} \tag{2.13}$$

Reconstruction Term	Regularization Term
• Encourages finding latent variables that explain the data well	• Prevents overfitting to specific latent configurations
• Maximized when samples from $q(z\|x)$ lead to accurate reconstructions	• Keeps the approximate posterior close to the prior
• Focuses on modeling the data	• Controls the complexity of the model

These terms create a natural trade-off: the reconstruction term pushes $q(z|x)$ to concentrate on values of z that explain the data well, while the regularization term pulls $q(z|x)$ toward the simpler prior distribution. This balance is crucial for learning meaningful latent representations.

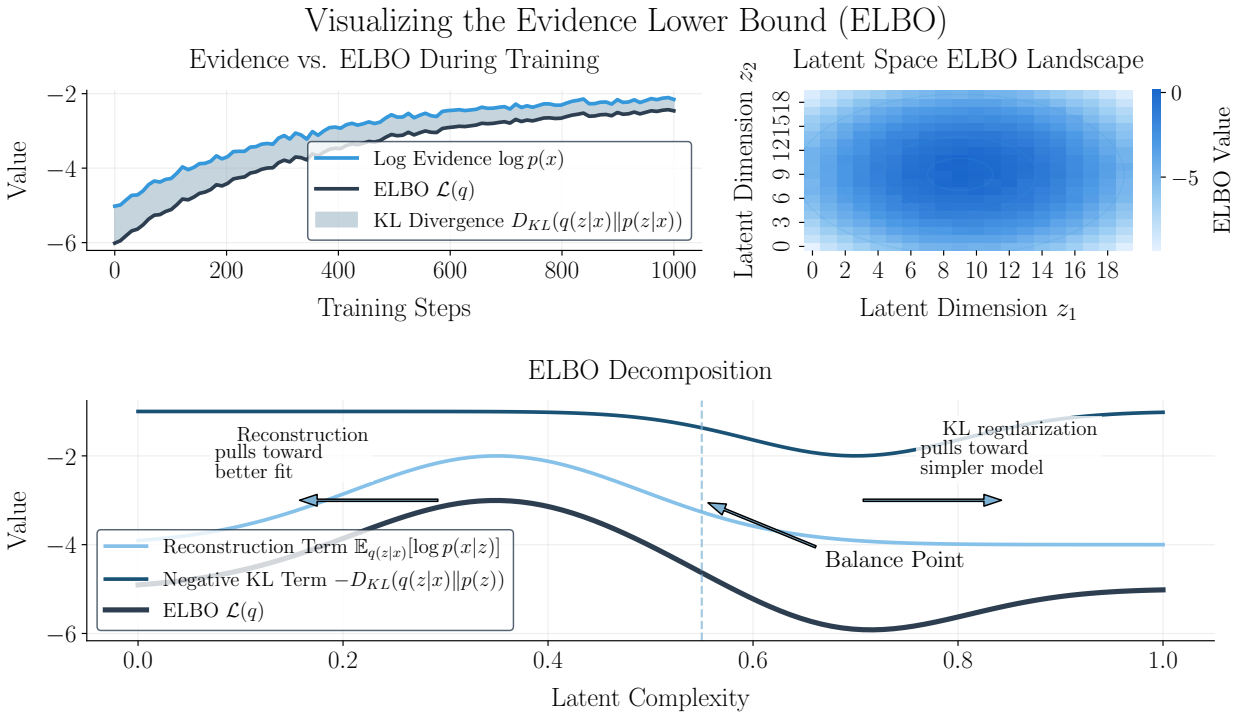

Figure 2.2: Visualizing the Evidence Lower Bound (ELBO). Top left: During training, the ELBO (dark blue) increases toward the log evidence (light blue), with their gap representing the KL divergence. Top right: The ELBO creates a landscape in latent space with higher values (lighter blue) in regions that better explain the data. Bottom: The ELBO decomposition shows how it balances the reconstruction term (light blue) against the negative KL term (dark blue). The combined ELBO (darkest blue line) represents a compromise between these competing objectives, with arrows indicating how each term pulls the optimization in different directions.

2.3.5 Deep Variational Inference: Autoencoders

Deep Variational Autoencoders (VAEs) (Kingma and Welling, 2014; Rezende et al., 2014) apply the variational inference framework to settings where both the generative model $p(x|z)$ and the inference model $q(z|x)$ are parameterized by neural networks. In the standard VAE, we assume a simple prior on the latent variables, typically a standard Gaussian $p(z) = \mathcal{N}(0, I)$.

The approximate posterior is usually modeled as a Gaussian with parameters produced by an encoder network:

$$q_\phi(z|x) = \mathcal{N}(z|\mu_\phi(x), \Sigma_\phi(x)) \tag{2.14}$$

where ϕ represents the parameters of the encoder. The generative model (decoder) with parameters θ then maps latent variables back to the data space:

$$p_\theta(x|z) = f(x; g_\theta(z)) \tag{2.15}$$

where f is an appropriate probability distribution (e.g., Gaussian for continuous data or Bernoulli for binary data) and g_θ is a neural network transforming z into the parameters of this distribution.

The ELBO for a VAE can thus be written as (Higgins et al., 2017):

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x|z)\right] - D_{KL}(q_\phi(z|x)\|p(z)) \tag{2.16}$$

With Gaussian assumptions, the KL divergence term has a closed-form expression (Doersch, 2016):

$$D_{KL}(\mathcal{N}(\mu, \Sigma)\|\mathcal{N}(0, I)) = \frac{1}{2}\left(\text{tr}(\Sigma) + \mu^T\mu - k - \log\det(\Sigma)\right) \tag{2.17}$$

where k is the dimensionality of z.

2.3.6 The Reparameterization Trick

To optimize the ELBO using gradient-based methods, we need to compute gradients of the expectation $\mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x|z)\right]$ with respect to ϕ. This presents a challenge because the distribution $q_\phi(z|x)$ itself depends on ϕ.

The reparameterization trick addresses this by reformulating sampling from $q_\phi(z|x)$ as a deterministic transformation of a fixed noise distribution. For a Gaussian $q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \Sigma_\phi(x))$, we can write:

$$z = \mu_\phi(x) + \Sigma_\phi^{1/2}(x) \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \tag{2.18}$$

This allows us to rewrite the expectation as:

$$\mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x|z)\right] = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)} \left[\log p_\theta(x|\mu_\phi(x) + \Sigma_\phi^{1/2}(x) \cdot \epsilon)\right] \tag{2.19}$$

Now the expectation is taken with respect to a fixed distribution that doesn't depend on ϕ, making it possible to pass gradients through the sampling operation. This enables end-to-end training of both the encoder and decoder networks using stochastic gradient descent. Later we will explore other reparameterization tricks, such as categorical reparameterization with the Gumbel-Softmax (Fig. 3.2) trick.

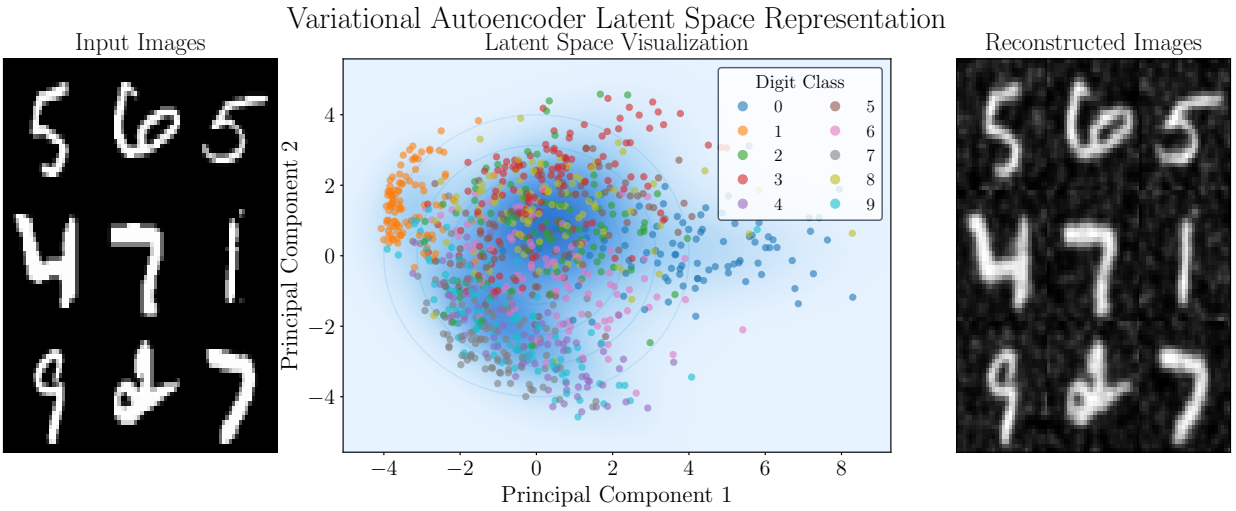

Figure 2.3: Variational Autoencoder Latent Space Representation. Left: Original input images from the MNIST dataset. Middle: The 2D projection of the latent space shows how different digit classes form clusters, with concentric circles indicating the standard normal prior distribution. The color-coded points represent different digit classes encoded in the latent space. Right: Reconstructed images show some loss of detail compared to the inputs, illustrating the information bottleneck created by the low-dimensional latent representation. The central density in the latent space represents the encoded distribution of digits.

2.3.7 Practical Applications and Extensions

The variational inference framework we've described extends naturally to a variety of more complex models and applications. These include conditional variants where we model $p(y|x)$ rather than just $p(x)$, hierarchical models with multiple layers of latent variables, and temporal models that capture dynamics over time.

The mathematical principles remain the same: we define a model that captures our beliefs about how latent variables generate observations, then use variational methods to approximate the posterior distribution over these latent variables. The ELBO provides a tractable objective that balances reconstruction accuracy against the complexity of the latent representation.

This framework will be particularly valuable as we move toward modeling the behavior of agents, where the latent variables represent unobserved actions or intentions that drive observable state changes. By applying the tools of variational inference to these agent models, we can discover latent structure in behavior while maintaining computational tractability.

2.4 Computational Agent Perception Models

Computational approaches to agent perception have evolved from symbolic models of goal attribution to integrated architectures for animacy detection. We review this progression with emphasis on optimization objectives and inference frameworks.

2.4.1 Bayesian Models of Goal Attribution

Baker et al. (2009) pioneered a Bayesian inverse planning framework for action understanding that formalized how observers infer agents' goals by inverting a model of rational planning. Given observed actions and states, their model computes posterior distributions over goals by assuming agents act approximately rationally to maximize expected utility. Through maze-world experiments, they demonstrated that humans employ similar inverse reasoning when attributing goals to moving entities. This approach operates on symbolic state and action representations, whereas our method processes raw visual observations.

Building upon this foundation, Ullman et al. (Ullman et al., 2009) extended the inverse planning framework to multi-agent settings, addressing how people infer social goals like helping or hindering. Their model introduces a formalism where social agents' reward functions depend on other agents' utilities—positive for helping and negative for hindering. In controlled experiments with simple 2D animations, their Bayesian model accurately predicted human judgments of social intentions, outperforming alternatives based on perceptual cues. While this work shares our intuition that agency is defined by internal decision processes, our setting focuses on unsupervised discovery rather than classifying predefined social relationships.

2.4.2 Cognitive Architecture for Animacy Detection

Moving beyond symbolic models, Gao et al. (2019) proposed a cognitive architecture integrating bottom-up and top-down processes for detecting animacy in visual scenes. Their model specifically addresses chasing perception through parallel pre-attentive detection of agent-like motion followed by capacity-limited Bayesian inference over candidate hypotheses. This hybrid approach successfully reproduced human performance in chasing detection tasks, capturing sensitivity to both stimulus complexity and cognitive constraints. Our VAD model adopts a similar hybrid strategy by combining object-centric representation learning with structured dynamics modeling, but focuses on unsupervised learning rather than implementing fixed detection mechanisms.

2.4.3 Generative Models for Active Vision

Parr et al. (2021) presents a generative modeling framework for active vision that explains how agents sample their environment through eye movements. Their approach formulates vision as inverting a generative process that predicts retinal input given scene contents and viewpoint. By treating perception and action as jointly optimizing a variational free energy objective, they unify various visual phenomena under a single computational principle. This

emphasis on generative modeling conceptually informs our approach, though we focus on discovering agents in visual scenes rather than modeling visual sampling behavior.

2.4.4 Our Approach: Variational Agent Discovery

Our method synthesizes insights from these prior frameworks while addressing their limitations. Unlike Baker et al. (2009) and Ullman et al. (2009), we operate directly on visual observations without requiring symbolic representations. In contrast to Gao et al. (2019)'s fixed detection strategy, our approach learns to discover agents through unsupervised training on image sequences.

The key innovation in our approach is framing agent discovery as structured variational inference in a factorized latent space. While previous models assumed known agent identities, our VAD model learns agent representations by explicitly modeling latent actions driving state transitions. Our $\mathcal{L}_{\text{VAD}}$ objective creates an inductive pressure to model entities whose dynamics are better explained by internal policies. We formalize this through an evidence lower bound (ELBO) derived and discussed in detail in Section 3.3.

In summary, our work connects Bayesian models of goal inference with modern representation learning techniques, resulting in a differentiable, end-to-end trainable VAD architecture that discovers agents directly from raw visual input—a capability not present in prior computational frameworks.

2.5 Object-Centric Representation Learning: Foundations for Agent-Centric Models

The ability to decompose visual scenes into separate objects and track them across time is fundamental to how humans perceive their environment. This section explores how object-centric learning provides the technical foundation for our agent-centric representation model.

2.5.1 From Convolutional Neural Networks to Object-Centric Representations

While traditional computer vision models process images holistically, object-centric learning explicitly decomposes scenes into separate object representations. Standard CNNs struggle with tasks requiring reasoning about individual objects and their interactions (Johnson et al., 2017; Yi et al., 2024). By structuring representations around discrete entities, object-centric models enable more systematic generalization and may align more closely with human cognitive abilities to parse scenes into meaningful components (Spelke, 1990; Kahneman et al., 1992).

In fact, object-centric decomposition may strongly align with the object-file theory proposed by Kahneman et al. (1992), where the visual system maintains distinct "files" for each significant object in a scene. Just as human perception operates through object-based attentional selection (Scholl, 2009; Alvarez and Cavanagh, 2005), computational object-centric models aim to replicate this capacity for discrete entity representation.

2.5.2 Attention Mechanisms and Slot Attention

The Transformer architecture (Vaswani et al., 2017) introduced self-attention mechanisms that compute pairwise interactions between elements through scaled dot-product attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{2.20}$$

where Q, K, and V represent queries, keys, and values respectively, and d_k is the dimensionality of the key vectors.

Building on this development, Locatello et al. (2020) introduced **Slot Attention**, a specialized attention mechanism designed specifically for unsupervised object-centric learning. Slot Attention provides an inductive bias for grouping perceptual features into object-centric representations called "slots"(think latent variables or factors), without requiring explicit supervision about object identities.

The Slot Attention module operates by iteratively refining a set of slot representations through attention-based competition over input features. As illustrated in Figure 2.4, given input features $\boldsymbol{I} \in \mathbb{R}^{N \times D_I}$ (where N is the number of input elements and D_I is the feature dimension) and a set of slot representations $\tilde{\boldsymbol{s}} \in \mathbb{R}^{K \times D_{\tilde{s}}}$ (where K is the number of slots and $D_{\tilde{s}}$ is the slot dimension), Slot Attention performs the following operations:

Linear projections to obtain queries, keys, and values:

$$\boldsymbol{Q} = \tilde{\boldsymbol{s}}\boldsymbol{W}_Q \tag{2.21}$$
$$\boldsymbol{K} = \boldsymbol{I}\boldsymbol{W}_K \tag{2.22}$$
$$\boldsymbol{V} = \boldsymbol{I}\boldsymbol{W}_V \tag{2.23}$$

where $\boldsymbol{W}_Q$, $\boldsymbol{W}_K$, and $\boldsymbol{W}_V$ are learnable projection matrices.

Computation of attention weights with a softmax normalization over slots:

$$\boldsymbol{A} = \frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}} \tag{2.24}$$
$$\bar{\boldsymbol{A}}_{ij} = \frac{\exp(\boldsymbol{A}_{ij})}{\sum_{j'} \exp(\boldsymbol{A}_{ij'})} \tag{2.25}$$

The normalization ensures that input features compete for assignment to different slots, which encourages specialization among the slots.

Weighted aggregation of values based on attention weights:

$$\boldsymbol{U} = \bar{\boldsymbol{A}}\boldsymbol{V} \tag{2.26}$$

Slot update using a GRU cell or a simple weighted residual connection:

$$\tilde{\boldsymbol{s}}^{(t+1)} = \mathrm{GRU}(\boldsymbol{U}, \tilde{\boldsymbol{s}}^{(t)}) \tag{2.27}$$

where $\tilde{\boldsymbol{s}}^{(t)}$ represents the slot representations at iteration t. The Gated Recurrent Unit (GRU) provides a mechanism for selectively updating the slot information while preserving important prior content, enabling the slots to refine their representations over multiple iterations.

A key distinction of Slot Attention from standard attention mechanisms is its normalization scheme. While traditional attention normalizes over the input elements (keys), Slot Attention normalizes over the slots (queries), ensuring that each input feature contributes primarily to a single slot. This creates a form of competition that drives slots to specialize in representing distinct objects.

The competitive binding process in Slot Attention parallels visual selective attention mechanisms described by Choi and Scholl (2004) and Most et al. (2005), where attention selects which perceptual elements are bound together into coherent object representations. Just as human attention may modulate the perception of animacy, Slot Attention's competitive mechanism ensures that perceptual features are selectively bound to the most relevant slot representations.

2.5.3 Extending Slot-Based Models to Video

Understanding agency requires reasoning about dynamics and temporal consistency. Kipf et al. (2021) extended Slot Attention to the temporal domain with Slot Attention for Video (SAVi), which maintains object identity across frames by propagating slot representations temporally:

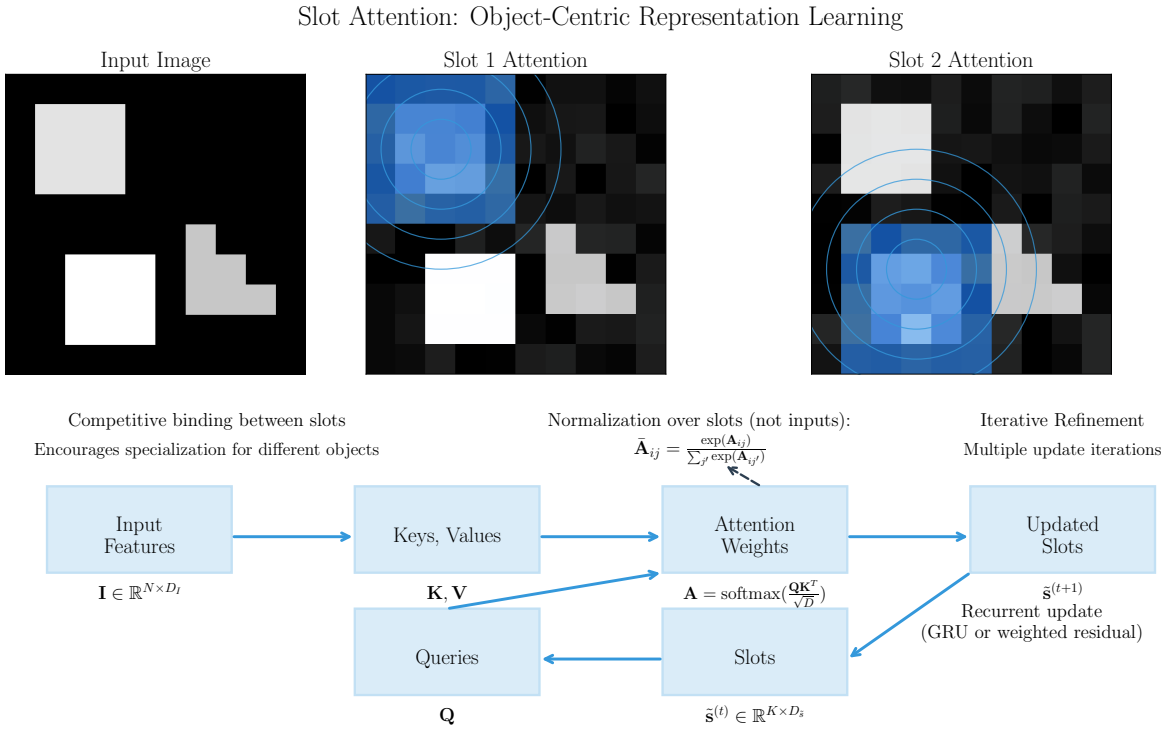

Figure 2.4: Slot Attention mechanism for object-centric representation learning. Top row: Input image (left) is processed to create object-centric attention maps (center, right) where each slot focuses on a different object. Bottom row: The mechanism works by projecting slots into queries and input features into keys and values, computing attention weights across slots (not inputs), and iteratively refining slot representations. This competitive binding encourages slots to specialize in representing distinct objects.

$$\tilde{\boldsymbol{s}}_t = f_{\text{update}}(\tilde{\boldsymbol{s}}_{t-1}, \boldsymbol{I}_t) \tag{2.28}$$

where $\tilde{\boldsymbol{s}}_t$ represents the slot representations at time t, $\boldsymbol{I}_t$ is the current frame, and f_{update} is an update function that incorporates Slot Attention.

SAVi demonstrated the ability to discover and track objects in video sequences without explicit supervision about object identities. By maintaining temporal consistency in slot assignments, SAVi enables reasoning about object dynamics and interactions over time.

This temporal consistency in object tracking may relate to the object-file theory, where Kahneman et al. (1992) and Scholl (2009) describe how the visual system maintains the continuity of object representations across time and space.

2.6 Reinforcement Learning

How do agents learn to make decisions that maximize their future rewards? This question has driven decades of research in reinforcement learning (RL), a framework for modeling sequential decision-making processes (Sutton and Barto, 1998; Kaelbling et al., 1996; Bertsekas and Tsitsiklis, 1996). While the primary contribution of this thesis is the unsupervised discovery of agent representations, reinforcement learning provides insights into how these agents might select actions once identified. This section examines the key concepts from reinforcement learning and multi-agent environments that inform our approach to agent-centric representation learning.

2.6.1 Foundations of Reinforcement Learning and World Models

Reinforcement learning problems are typically formalized as Markov Decision Processes (MDPs), defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$ (Puterman, 1994; Bellman, 1957). Here, $\mathcal{S}$ represents the state space, $\mathcal{A}$ is the action space, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ defines the transition dynamics where $P(s'|s, a)$ is the probability of transitioning to state s' after taking action a in state s[1], $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is a discount factor balancing immediate versus future rewards.

An agent's behavior is characterized by a policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, which defines a probability distribution over actions for each state. The goal in reinforcement learning is to find an optimal policy π^* that maximizes the expected discounted cumulative reward:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right] \tag{2.29}$$

where $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \ldots)$ denotes a trajectory sampled according to the policy π and the environment dynamics P.

There are two primary approaches to solving reinforcement learning problems: value-based methods and policy-based methods. Value-based methods, exemplified by Q-learning (Watkins and Dayan, 1992; Mnih et al., 2015) and DQN (Mnih et al., 2013, 2015), estimate the expected return for state-action pairs and derive a policy by selecting actions that maximize this value. While these methods excel in discrete action spaces, they often face challenges in continuous action domains (Lillicrap et al., 2016; Schulman et al., 2015). It's important to note that modern algorithms often combine elements from both approaches to leverage their complementary strengths.

2.6.2 Policy Gradient Methods and PPO

Policy gradient methods address limitations of pure value-based approaches by directly optimizing a parameterized policy π_θ (Williams, 1992; Sutton et al., 1999; Peters and Schaal,

[1]Note that, s here represents the complete environment state in the context of RL, which differs from our use of $\tilde{s}$ in Sections 2.5.2 and 3.1 where it represents slot-based object-centric representations.

2008). The policy gradient theorem (Sutton et al., 1999) provides the mathematical foundation for these approaches, expressing the gradient of the expected return with respect to policy parameters:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t|s_t) \cdot G_t\right] \tag{2.30}$$

where $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ is the return from time step t. This formulation enables gradient ascent directly on the policy parameters. Various policy gradient algorithms have been developed, including REINFORCE (Williams, 1992), Natural Policy Gradient (Kakade, 2002), and Trust Region Policy Optimization (TRPO) (Schulman et al., 2015). While the basic policy gradient formulation uses only trajectory returns, modern implementations typically incorporate value function estimates to reduce variance and improve learning efficiency, leading to actor-critic approaches that combine policy-based and value-based learning.

For continuous action spaces, policies often output the parameters of a Gaussian distribution (Schulman et al., 2015, 2017; Haarnoja et al., 2018):

$$\pi_\theta(a|s) = \mathcal{N}(a|\mu_\theta(s), \sigma_\theta(s)) \tag{2.31}$$

where $\mu_\theta(s)$ and $\sigma_\theta(s)$ are the state-dependent mean and standard deviation produced by the policy network. This parameterization enables sampling of continuous actions while maintaining differentiability for gradient-based optimization (Silver et al., 2014).

While vanilla policy gradient methods can be effective, they suffer from high variance and sensitivity to step sizes (Kakade, 2002; Schulman et al., 2015). Proximal Policy Optimization (PPO) (Schulman et al., 2017) addresses these issues by constraining policy updates to remain within a trust region around the current policy. Building on ideas from Trust Region Policy Optimization (TRPO) (Schulman et al., 2015), PPO uses a simpler clipped objective function that prevents excessively large policy changes:

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t\right)\right] \tag{2.32}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio between the new and old policies, A_t is the advantage estimate (Schulman et al., 2016), and ϵ is a hyperparameter (typically 0.1 or 0.2) that constrains the policy update. Importantly, PPO employs an actor-critic architecture where a value function is learned alongside the policy to estimate advantages, thereby reducing variance in the policy gradient estimates. This integration of value-based and policy-based approaches has contributed to PPO becoming one of the most widely used RL algorithms due to its simplicity, effectiveness, and robustness across a variety of tasks (Andrychowicz et al., 2020).

Figure 2.5 illustrates key concepts in policy gradient methods and PPO. The top row shows a toy policy gradient objective landscape (left), demonstrating how gradient ascent navigates the parameter space to find an optimal policy; the PPO clipping objective (middle), showing how PPO constrains updates to stay within a trust region; and learning curves (right), highlighting PPO's stability compared to vanilla policy gradients. The bottom row visualizes how policy updates differ between vanilla policy gradients (left) and PPO (mid-

dle), with PPO producing more conservative changes, and a trust region comparison (right) showing how PPO constrains updates to prevent destructive policy changes.

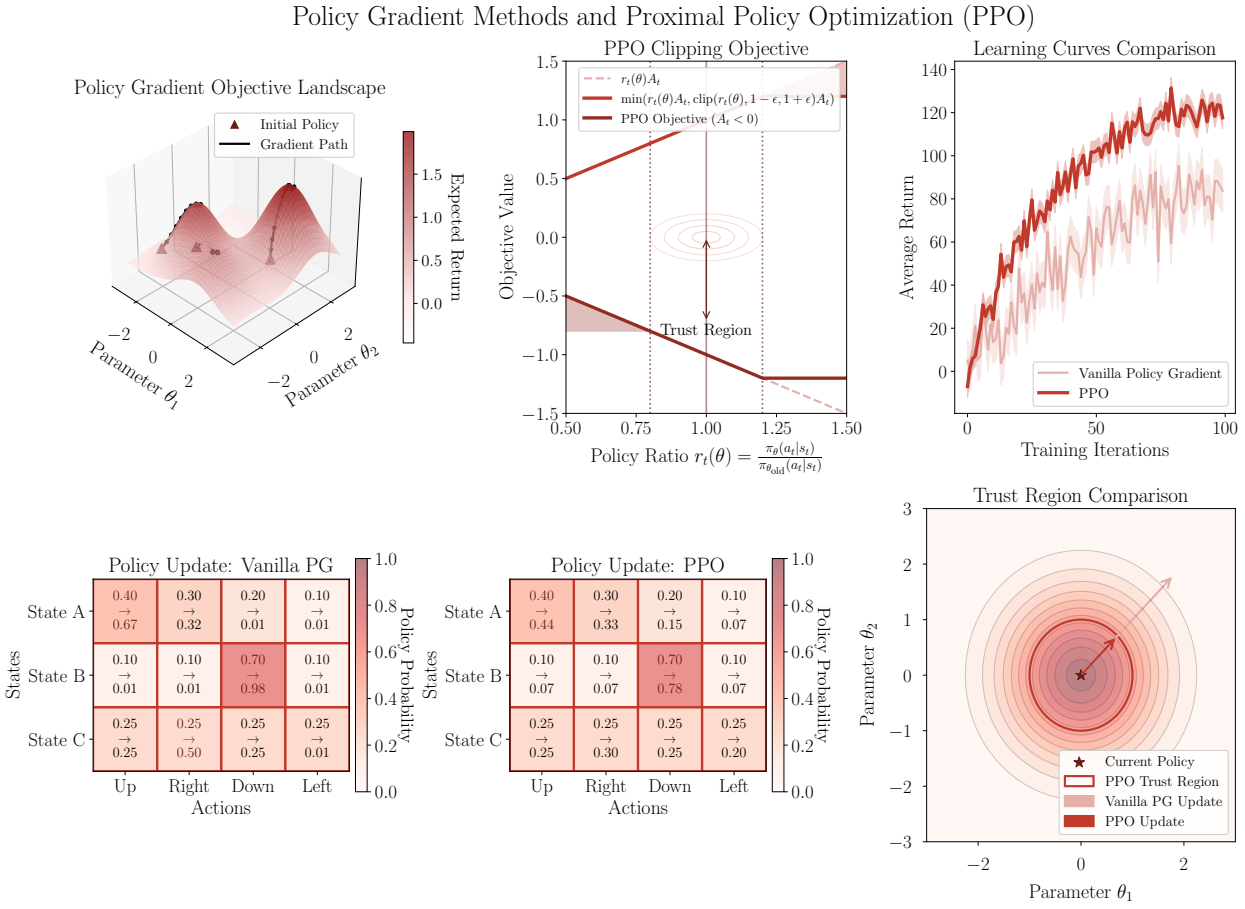

Figure 2.5: Visualization of policy gradient methods and Proximal Policy Optimization (PPO). **Top row:** (Left) Policy gradient objective landscape showing gradient paths toward optimal policies; (Middle) PPO clipping objective illustrating how the algorithm constrains policy updates for both positive and negative advantages; (Right) Learning curves demonstrating PPO's lower variance and improved performance compared to vanilla policy gradients. **Bottom row:** (Left) Vanilla policy gradient update showing potentially large probability shifts; (Middle) PPO policy update demonstrating more conservative probability changes; (Right) Trust region comparison visualizing how PPO constrains updates to remain within a safe region around the current policy, preventing potentially destructive updates.

2.6.3 World Models and Model-Based RL

A particularly relevant area for our agent-centric approach is model-based RL (Moerland et al., 2023; Sutton, 1991; Deisenroth and Rasmussen, 2011), especially the concept of world models. World models are internal representations of environment dynamics that allow agents to predict the consequences of their actions (Ha and Schmidhuber, 2018; Hafner

et al., 2019, 2020). Formally, a world model learns the transition dynamics $P(s_{t+1}|s_t, a_t)$ and possibly the reward function $R(s_t, a_t, s_{t+1})$. With a learned model, an agent can plan ahead by simulating outcomes without environmental interaction (Silver et al., 2017; Wang et al., 2019), learn more efficiently through synthetic experience generation (Sutton, 1990; Ha and Schmidhuber, 2018), and adapt to environmental changes by updating its internal model (Nagabandi et al., 2018; Janner et al., 2019). Recent advances include Dreamer (Hafner et al., 2019, 2020), which learns a latent dynamics model and plans in latent space, and MuZero (Schrittwieser et al., 2020), which combines model-based planning with model-free learning without requiring an explicit dynamics model. This concept of modeling the environment is similar to our task of modeling agents in multi-agent environments. By learning representations of other agents, an agent could effectively incorporate them into its world model, enabling more effective planning and interaction (more on this in 5.4).

2.6.4 Multi-Agent Reinforcement Learning

When multiple agents interact in a shared environment, the single-agent MDP framework extends to multi-agent settings. In a multi-agent MDP with n agents, each agent i has its own action space $\mathcal{A}_i$ and typically observes a local state o_i derived from the global state s. The joint action $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ influences the next state according to the transition function $P(s'|s, \mathbf{a})$.

The learning objective becomes significantly more complex in multi-agent settings because the environment appears non-stationary from each agent's perspective as other agents learn and change their policies. Additionally, coordinated behaviors may require different learning approaches, and partial observability often becomes a challenge as each agent only has access to limited information about the environment state. Several algorithms have been developed to address these challenges:

Multi-Agent PPO (MAPPO) MAPPO (Yu et al., 2022) extends PPO to multi-agent settings by using a centralized value function with decentralized policies. This centralized training with decentralized execution (CTDE) paradigm allows value functions to condition on global information during training while maintaining decentralized execution. MAPPO has demonstrated strong performance across various multi-agent tasks while maintaining the sample efficiency and stability benefits of PPO.

Multi-Agent Deep Deterministic Policy Gradient (MADDPG) MADDPG (Lowe et al., 2017) adapts DDPG for multi-agent environments by training a centralized critic for each agent that has access to all agents' observations and actions, while each agent's policy (actor) uses only its local observations. This approach effectively transforms a non-stationary environment into a stationary one from the perspective of each critic, enabling more stable learning. MADDPG explicitly incorporates information about other agents' policies into each agent's learning process, making it particularly suitable for mixed cooperative-competitive environments.

Multi-Agent Advantage Actor-Critic (MA A2C) MA A2C (Iqbal and Sha, 2019) extends the A2C algorithm to multi-agent settings, often using attention mechanisms to selectively focus on relevant agents. This approach enables agents to adapt their behavior based on the importance of other agents' states and actions in the current context, improving coordination in dynamic team compositions.

A more straightforward approach to multi-agent reinforcement learning is Independent PPO (IPPO), in which each agent independently runs its own instance of PPO without explicit coordination mechanisms. Despite this architectural simplicity, IPPO consistently serves as a strong baseline in the field and is the approach we adopt in our experiments in Section 4.2. While IPPO does not explicitly model other agents' behaviors, it can achieve competitive performance when agents implicitly adapt to one another through environmental feedback. It is worth noting that in our experiments in Section 4.2, the IPPO/PPO state-based agents have access to all agents' states. This highlights a key trade-off in multi-agent algorithm design: balancing the use of explicit agent state information against learning indirectly through observations. Algorithms such as MADDPG and MAPPO formally incorporate information about other agents during training, while even seemingly independent approaches like IPPO may have access to other agents' state information and policies depending on the specific environment configuration.

2.6.5 Agent Modeling and Theory of Mind

Beyond RL, a critical component for multi-agent learning is the ability to model other agents. Agent modeling—constructing representations of other agents' policies, goals, and beliefs—is crucial for effective decision-making in social contexts. While initially developed for competitive scenarios (hence "opponent modeling"), these techniques apply broadly to any interactive agent.

Learning with Opponent Learning Awareness (LOLA) (Foerster et al., 2017) represents a significant advance in this area. LOLA agents explicitly account for the fact that their opponents are also learning, differentiating through the opponent's learning process when computing policy updates:

$$\nabla_{\theta_i} V_i(\theta_i, \theta_{-i}) + \nabla_{\theta_{-i}} V_i(\theta_i, \theta_{-i}) \cdot \nabla_{\theta_i} \theta_{-i} \tag{2.33}$$

where θ_i represents the parameters of agent i's policy, θ_{-i} represents the parameters of other agents' policies, and $\nabla_{\theta_i} \theta_{-i}$ captures how other agents' policy parameters change in response to agent i's policy. This second-order optimization enables more stable learning dynamics, particularly in cases where naive learning might lead to suboptimal outcomes.

Theory of Mind has also inspired computational approaches to agent modeling. Rabinowitz et al. (2018)'s Machine Theory of Mind (ToMnet) uses meta-learning to build models of other agents. Given past trajectories τ_{past} and recent observations o_{recent}, ToMnet approximates:

$$e_{\text{char}} = f_{\text{char}}(\tau_{\text{past}}) \tag{2.34}$$

$$e_{\text{mental}} = f_{\text{mental}}(o_{\text{recent}}, e_{\text{char}}) \tag{2.35}$$

$$\hat{a} = f_{\text{pred}}(s, e_{\text{mental}}) \tag{2.36}$$

where e_{char} is a "character embedding" representing the agent's policy or type, e_{mental} is the mental state embedding, and $\hat{a}$ is the predicted action for a given state s. ToMnet demonstrated the ability to model diverse agent types, including agents with different reward functions, memory capacities, and planning horizons.

While these approaches to agent modeling and Theory of Mind provide frameworks for reasoning about other agents' mental states, they operate on symbolic or highly abstracted state representations (like the models we discussed in 2.4) rather than raw perceptual inputs.

Chapter 3

Method

3.1 Background and Problem Statement

Building on the foundations of agent perception from cognitive science and object-centric representation learning from machine learning (Ch. 2), we now formalize the problem of unsupervised agent discovery from visual observations.

We situate our problem setting as a Partially Observable Multi-Agent Markov Decision Processes (POMDPs), defined as $(S, \{A_i\}_{i=1}^N, T, \{R_i\}_{i=1}^N, \{\mathbf{X}_i\}_{i=1}^N, \gamma)$[1].

Within this formalism, we operate under observational constraints: we access only a sequence of rendered visual observations $\mathbf{X} = \{x_0, x_1, ..., x_T\}$, where each $x_t \in \mathbb{R}^{H \times W \times C}$ represents an image frame containing multiple entities. The agents' actions, rewards, and individual observations remain unobserved—all information must be inferred from the pixel-level visual data.

Object-centric representation learning, particularly slot attention mechanisms, provides our computational foundation for decomposing these complex scenes. As detailed in Section 2.5, these approaches factor visual observations into a set of slot representations $\{\tilde{s}_t^1, \tilde{s}_t^2, ..., \tilde{s}_t^K\}$[2], where each $\tilde{s}_t^k$ can be assumed to encode some structural entity. When extended to video sequences as in SAVi (Kipf et al., 2021), these models can be optimized using maximum likelihood to predict the next frame:

$$\max_\theta \log p(x_{t+1} | \{\tilde{s}_t^1, \tilde{s}_t^2, ..., \tilde{s}_t^K\}) \approx -\|x_{t+1} - \text{Decoder}(f_{\text{dyn}}(\{\tilde{s}_t^1, \tilde{s}_t^2, ..., \tilde{s}_t^K\}))\|^2 \tag{3.1}$$

This approximation maps the probabilistic objective to a deterministic reconstruction loss, where the negative squared error term $-\|x_{t+1} - \text{Decoder}(f_{\text{dyn}}(\{\tilde{s}_t^1, \tilde{s}_t^2, ..., \tilde{s}_t^K\}))\|^2$ corresponds to the log-likelihood under a Gaussian observation model with fixed variance. While effective for general scene decomposition, SAVi treats all entities uniformly. We introduce a variational inference perspective that explicitly models actions as latent variables, creating an inductive bias toward identifying entities whose transitions reflect agency rather than merely physical/environmental dynamics.

[1] We use $\mathbf{X}$ to denote observations, which are commonly represented as O in standard POMDP notation.

[2] Similar to 2.5.2, $\tilde{s}$ denotes a slot-based representation and not the RL state notation s in 2.6.1.

3.2 Structured Variational Approach to Agent Discovery

Our approach formalizes agent discovery as a structured probabilistic inference problem with latent variables. Unlike standard object-centric models that directly predict state transitions, we model these transitions as consequences of agent decisions—actions sampled from internal policies.

We make the following assumptions about our problem: First, agents follow coherent policies that can be modeled as conditional distributions $p(a_i|\tilde{s}_i)$ over actions given states. Second, slot attention mechanisms provide sufficiently structured representations for entity-centric modeling. Third, observed transitions between states are generated by unobservable actions that must be inferred from dynamics. Fourth, these dynamics arise from agents executing actions according to their policies, treating observed data as samples from a process where agent decisions drive environmental dynamics.

In our slot-based representation framework, the state of slot i at time t is denoted as $\tilde{s}_t^i$. We explicitly model the transition probability $p(\tilde{s}_{t+1}^i|\tilde{s}_t^1, \tilde{s}_t^2, ..., \tilde{s}_t^K)$ while accounting for a latent action variable a_i that potentially drives this transition. The true transition probability involves marginalizing over all possible actions:

$$p(\tilde{s}_{t+1}^i|\tilde{\mathbf{s}}_t) = \int p(\tilde{s}_{t+1}^i, a_i|\tilde{\mathbf{s}}_t)\, da_i \tag{3.2}$$

where $\tilde{\mathbf{s}}_t = \{\tilde{s}_t^1, \tilde{s}_t^2, ..., \tilde{s}_t^K\}$ represents the complete set of slots at time t. Similar to the challenge described in Section 2.3.1, this marginalization becomes computationally intractable, especially since in our case we do not have direct access to the joint distribution $p(\tilde{s}_{t+1}^i, a_i|\tilde{\mathbf{s}}_t)$. By applying Bayes' rule, we can decompose this joint probability into more familiar quantities, and then employ variational inference techniques as outlined in Section 2.3.2 to approximate the marginalization efficiently.

3.3 Derivation of the Variational Agent Discovery Objective

To formalize our approach mathematically, we begin with the log marginal likelihood for a single slot transition and derive a tractable evidence lower bound (ELBO), following the approach introduced in Section 2.3.3. We introduce a variational posterior $q(a_i|\tilde{s}_t^i, \tilde{s}_{t+1}^i)$ to approximate the true posterior distribution over actions—an inverse model that infers the actions most likely to have generated observed transitions.

Starting with the log marginal likelihood:

$$\log p(\tilde{s}_{t+1}^i|\tilde{\mathbf{s}}_t) = \log \int p(\tilde{s}_{t+1}^i, a_i|\tilde{\mathbf{s}}_t)\, da_i \tag{3.3}$$

We apply the standard variational inference trick (see equation 2.8):

$$\log p(\tilde{s}^i_{t+1}|\tilde{\mathbf{s}}_t) = \log \int q(a_i|\tilde{s}^i_t, \tilde{s}^i_{t+1}) \frac{p(\tilde{s}^i_{t+1}, a_i|\tilde{\mathbf{s}}_t)}{q(a_i|\tilde{s}^i_t, \tilde{s}^i_{t+1})} \, da_i \tag{3.4}$$

Applying Jensen's inequality and expanding the joint probability using the chain rule, assuming that an agent's action depends primarily on its own state:

$$p(\tilde{s}^i_{t+1}, a_i|\tilde{\mathbf{s}}_t) = p(\tilde{s}^i_{t+1}|a_i, \tilde{\mathbf{s}}_t) p(a_i|\tilde{s}^i_t) \tag{3.5}$$

We derive the ELBO for each slot, which closely parallels the structure presented in equation 2.11:

$$\begin{aligned} \mathcal{L}_i(\tilde{\mathbf{s}}_t, \tilde{s}^i_{t+1}; \theta, \phi) = \mathbb{E}_{q_\phi(a_i|\tilde{s}^i_t, \tilde{s}^i_{t+1})} \left[\log p_\theta(\tilde{s}^i_{t+1}|a_i, \tilde{\mathbf{s}}_t)\right] \\ - D_{KL}(q_\phi(a_i|\tilde{s}^i_t, \tilde{s}^i_{t+1})||p_\theta(a_i|\tilde{s}^i_t)) \end{aligned} \tag{3.6}$$

This ELBO decomposes into the following conceptual components and terms:

Variational Objective Components

$$\mathcal{L}_i(\tilde{\mathbf{s}}_t, \tilde{s}^i_{t+1}; \theta, \phi) = \mathbb{E}_{q_\phi(a_i|\tilde{s}^i_t, \tilde{s}^i_{t+1})}\left[\log p_\theta(\tilde{s}^i_{t+1}|a_i, \tilde{\mathbf{s}}_t)\right] - D_{KL}(q_\phi(a_i|\tilde{s}^i_t, \tilde{s}^i_{t+1})||p_\theta(a_i|\tilde{s}^i_t)) \tag{3.7}$$

Component	Function
$p_\theta(\tilde{s}^i_{t+1}\|a_i, \tilde{\mathbf{s}}_t)$	**Forward dynamics model** predicts the next state given current state's slots and inferred action. It forms the core of our predictive model.
$q_\phi(a_i\|\tilde{s}^i_t, \tilde{s}^i_{t+1})$	**Inverse action model** infers the action that most likely caused the observed transition between $\tilde{s}^i_t$ and $\tilde{s}^i_{t+1}$.
$p_\theta(a_i\|\tilde{s}^i_t)$	**Agent policy** represents a conditional distribution over latent action variables given the current entity slot $\tilde{s}^i_t$.
Loss terms:	
$\mathbb{E}_{q_\phi(a_i\|\tilde{s}^i_t, \tilde{s}^i_{t+1})}\left[\log p_\theta(\tilde{s}^i_{t+1}\|a_i, \tilde{\mathbf{s}}_t)\right]$	**Reconstruction term** encourages accurate prediction of the next state given the inferred action, similar to the reconstruction term in equation 2.13.
$D_{KL}(q_\phi(a_i\|\tilde{s}^i_t, \tilde{s}^i_{t+1})\|\|p_\theta(a_i\|\tilde{s}^i_t))$	**Policy regularization** keeps inferred actions close to the agent's learned policy, analogous to the regularization term described in the box 2.3.4.

These components create a computational pressure that drives the emergence of agent-specific representations (4.4, 5.1). Entities whose transitions exhibit agency will develop structured policies.

To apply this objective to scenes with multiple entities, we sum the ELBO across all slots and combine it with a reconstruction loss to form our full Variational Agent Discovery objective $\mathcal{L}_{\text{VAD}}$:

$$\mathcal{L}_{\text{VAD}} = \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \lambda_{\text{ELBO}} \sum_{i \in \text{slots}} \mathcal{L}_i \tag{3.8}$$

where $\mathcal{L}_{\text{recon}}$ is a reconstruction loss that encourages accurate prediction of future frames, and λ_{recon} and λ_{ELBO} are hyperparameters that balance the importance of reconstruction versus structured latent action modeling.

This $\mathcal{L}_{\text{VAD}}$ objective (Figure 3.1) serves as the foundation for our **Variational Agent Discovery (VAD) model** (which we introduce next 3.4) and can also be used as an auxiliary loss to improve sample efficiency in reinforcement learning, as we demonstrate in Section 4.2.

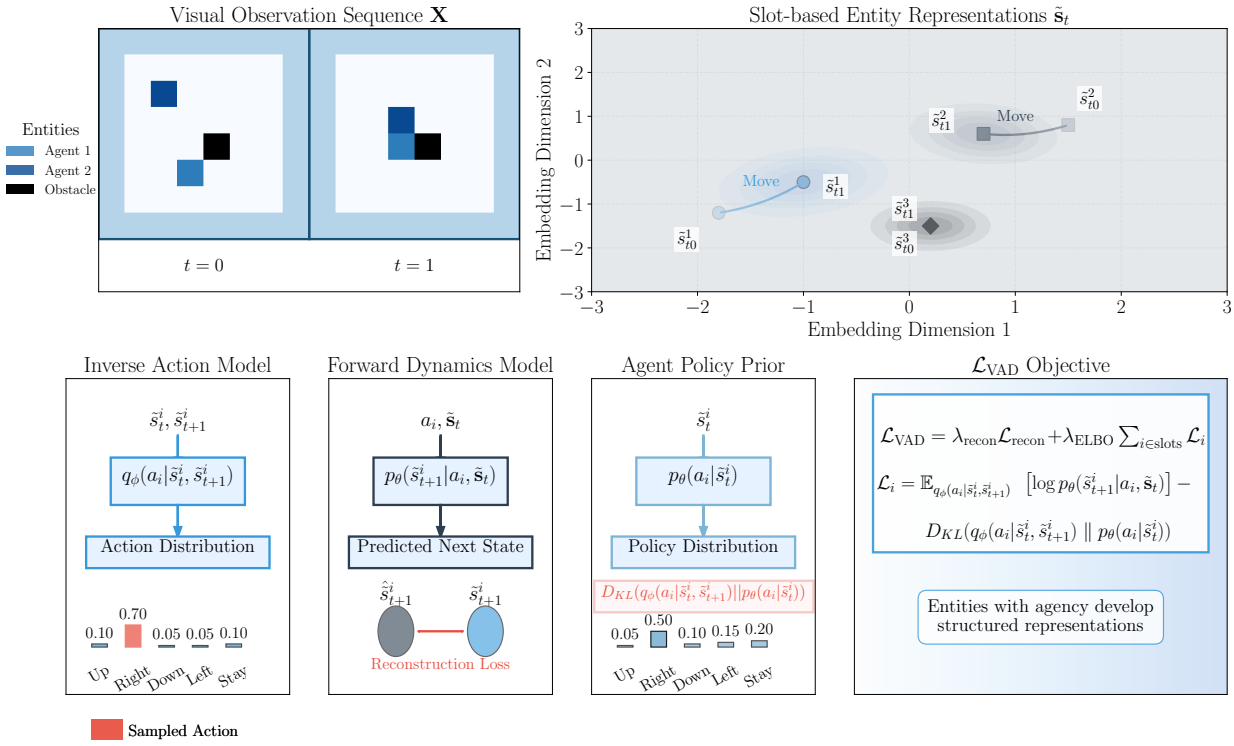

Figure 3.1: **Conceptual illustration of the Variational Agent Discovery ($\mathcal{L}_{\text{VAD}}$) objective.** *Top left:* Visual observation sequence showing frames with two agents (blue and navy) and a static obstacle (black). *Top right:* Slot-based entity representations in embedding space, showing how agents exhibit movement between timesteps while the obstacle remains static. *Bottom row:* The three key components of our VAD framework: (1) An inverse action model q_ϕ that infers the most likely action given the observed state transition, (2) A forward dynamics model p_θ that predicts the next state given the current state and inferred action, with reconstruction loss measuring prediction accuracy, and (3) An agent policy prior p_θ that learns a conditional distribution over actions given the current state, regularized by KL divergence. *Bottom right:* The complete $\mathcal{L}_{\text{VAD}}$ objective combines these components to create an inductive bias that distinguishes entities with agency from those that follow environmental dynamics.

3.4 Implementation Details

This section presents the detailed implementation of our **Variational Agent Discovery (VAD) model**, which operationalizes the theoretical objective $\mathcal{L}_{\text{VAD}}$ formulated in Section 3.3. Our architecture is a deep conditional slot-based variational autoencoder that implements each component of the variational objective through specific neural network modules designed to work together within the probabilistic framework.

3.4.1 Architecture Overview

Our implementation consists of: (i) a slot-based object-centric encoder that decomposes visual scenes into entity-specific representations; (ii) an inverse dynamics model that infers latent actions from observed state transitions; (iii) a forward dynamics model that predicts future states conditioned on actions; and (iv) a reconstruction decoder that projects slot representations back to pixel space. Together, the inverse and forward dynamics models function as a conditional variational autoencoder (CVAE), with actions serving as the latent variables conditioned on states. Figure 3.3 provides a schematic representation of this architecture and the information flow between components.

Algorithm 1 Variational Agent Discovery Algorithm

Require: Video sequence $\mathbf{X} = \{x_0, x_1, \ldots, x_T\}$, number of slots K, number of actions A
Ensure: Slot representations $\tilde{\mathbf{s}}_t$, inferred actions a_t^i

$\mathbf{f}_t \leftarrow \text{CNN}(x_t)$ for all $t \in \{0, \ldots, T\}$ {Extract features}
Initialize $\tilde{\mathbf{s}}_0$ randomly
for $t \in \{1, \ldots, T\}$ **do**
  $\tilde{\mathbf{s}}_t \leftarrow \text{SlotAttention}(\tilde{\mathbf{s}}_{t-1}, \mathbf{f}_t)$ {Update object slots}
end for
for $t \in \{0, \ldots, T-1\}$ **do**
  for $i \in \{1, \ldots, K\}$ **do**
    $\boldsymbol{\lambda}_t^i \leftarrow \text{InverseDynamicsModel}(\tilde{s}_t^i, \tilde{s}_{t+1}^i)$ {Action logits}
    $a_t^i \sim \text{GumbelSoftmax}(\boldsymbol{\lambda}_t^i, \tau)$ {Sample action}
    $\hat{\tilde{s}}_{t+1}^i \leftarrow \text{ForwardDynamicsModel}(a_t^i, \tilde{\mathbf{s}}_t)$ {Predict next state}
    $\mathcal{L}_{\text{ELBO}}^{t,i} \leftarrow \log p_\theta(\hat{\tilde{s}}_{t+1}^i | \tilde{s}_{t+1}^i) - D_{\text{KL}}(q_\phi(a_t^i | \tilde{s}_t^i, \tilde{s}_{t+1}^i) \| p_\theta(a_t^i | \tilde{s}_t^i))$ {Compute ELBO}
  end for
  $\hat{\tilde{\mathbf{s}}}_{t+1} \leftarrow \{\hat{\tilde{s}}_{t+1}^1, \hat{\tilde{s}}_{t+1}^2, \ldots, \hat{\tilde{s}}_{t+1}^K\}$ {Collect predicted slots}
  $\hat{x}_{t+1} \leftarrow \text{SpatialBroadcastDecoder}(\hat{\tilde{\mathbf{s}}}_{t+1})$ {Decode to pixels}
  $\mathcal{L}_{\text{recon}}^t \leftarrow \|x_{t+1} - \hat{x}_{t+1}\|^2$ {Reconstruction loss}
end for
Compute total objective using $\mathcal{L}_{\text{VAD}}$ from Equation 3.8

return $\tilde{\mathbf{s}}_t, a_t^i$

3.4.2 Object-Centric Representation Learning

We build our agent discovery model on top of the Slot Attention for Video (SAVi) architecture (Kipf et al., 2021). As detailed in Section 2.5, SAVi extends the original Slot Attention mechanism to handle temporal data by propagating slot representations across frames.

Our implementation processes sequences of frames $\mathbf{X} = \{x_0, x_1, \ldots, x_T\}$ through a convolutional neural network to extract features $\mathbf{f}_t \in \mathbb{R}^{N \times D}$, where $N = H' \times W'$ is the flattened spatial dimension and D is the feature dimension:

$$\mathbf{f}_t = \text{CNN}(x_t) \tag{3.9}$$

We adopt the 'implicit differentiation' approach for Slot Attention as explored in Wu et al. (2023):

Slot Attention with Implicit Differentiation

During training, we propagate gradients only through the final iteration of Slot Attention. Specifically, we detach gradients for iterations $l = 0$ to $L - 2$ by applying stop-gradient to each $\tilde{\mathbf{s}}_t^{(l+1)} = \text{SlotAttentionStep}(\mathbf{f}_t, \tilde{\mathbf{s}}_t^{(l)})$, and only allow gradient flow for the final update $\tilde{\mathbf{s}}_t^{(L)} = \text{SlotAttentionStep}(\mathbf{f}_t, \tilde{\mathbf{s}}_t^{(L-1)})$.

This approach effectively treats earlier iterations as a fixed-point optimization process, only backpropagating through the final iteration. As observed by Wu et al. (2023) and our own exploration, this technique can improve performance by stabilizing training dynamics and allowing better scaling with increased iterations, especially on complex datasets. The implicit differentiation approach helps prevent optimization instabilities that may arise when backpropagating through multiple recurrent iterations.

3.4.3 Variational Action Inference

Inverse Dynamics Model

The inverse dynamics model $q_\phi(a_i | \tilde{s}_t^i, \tilde{s}_{t+1}^i)$ functions as the encoder component of our conditional VAE, inferring the latent action that most likely caused the transition from $\tilde{s}_t^i$ to $\tilde{s}_{t+1}^i$, as introduced in Section 3.3. We implement this using a neural network that outputs parameters of a categorical distribution over discrete actions:

$$\mathbf{h} = \text{Concat}(\tilde{s}_t^i, \tilde{s}_{t+1}^i) \tag{3.10}$$
$$\mathbf{h}_1 = \text{GeLU}(\mathbf{W}_1 \mathbf{h} + \mathbf{b}_1) \tag{3.11}$$
$$\mathbf{h}_2 = \text{GeLU}(\mathbf{W}_2 \mathbf{h}_1 + \mathbf{b}_2) \tag{3.12}$$
$$\boldsymbol{\lambda} = \mathbf{W}_3 \mathbf{h}_2 + \mathbf{b}_3 \tag{3.13}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^A$ represents logits for a categorical distribution over A possible actions.

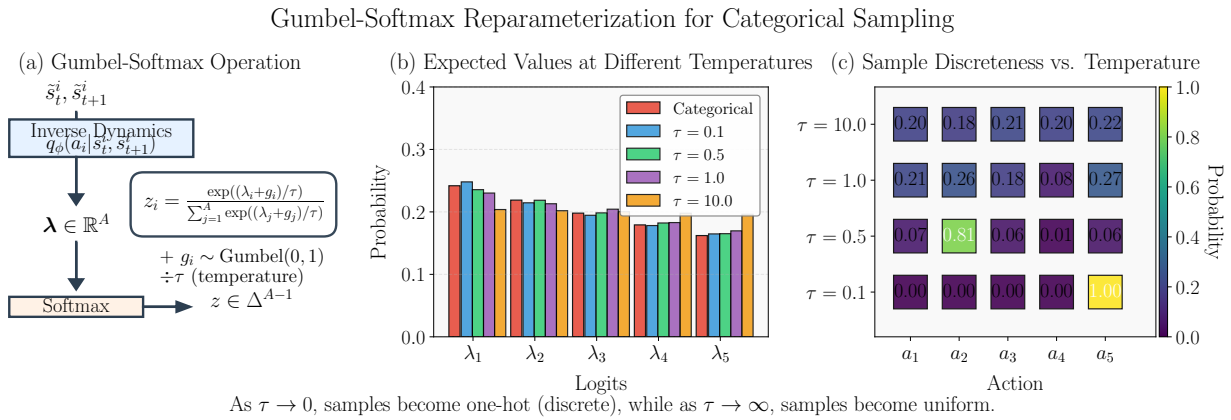

Figure 3.2: Visualization of the Gumbel-Softmax reparameterization trick used for differentiable discrete action sampling. (a) The Gumbel-Softmax operation transforms logits from the inverse dynamics model into differentiable samples. (b) Expected values of Gumbel-Softmax samples at different temperatures, showing convergence to the categorical distribution as $\tau \to 0$. (c) Sample discreteness illustration showing how temperature controls the sharpness of the distribution—lower temperatures produce more one-hot-like vectors while higher temperatures yield more uniform distributions.

To enable end-to-end differentiability (addressing the challenge described in Section 2.3.6), we use the Gumbel-Softmax reparameterization trick to sample from this categorical distribution (illustrated in Figure 3.2):

$$a_i = \frac{\exp((\lambda_i + g_i)/\tau)}{\sum_{j=1}^{A} \exp((\lambda_j + g_j)/\tau)} \tag{3.14}$$

where $g_i \sim \text{Gumbel}(0, 1)$ are i.i.d. samples from the Gumbel distribution and τ is a temperature parameter that controls the discreteness of the distribution.

Forward Dynamics Model

The forward dynamics model $p_\theta(\tilde{s}_{t+1}^i | a_i, \tilde{\mathbf{s}}_t)$ serves as the decoder in our conditional VAE framework, predicting the next state given the current state and inferred action, corresponding to the component described in the box 3.3. Rather than using a simple MLP, we implement this function using a Transformer architecture that can model complex dependencies between the action and state:

$$\mathbf{h}_{\text{cond}} = \text{Concat}(a_i, \tilde{s}_t^i) \tag{3.15}$$

$$\hat{\tilde{s}}_{t+1}^i = \text{Transformer}(\mathbf{h}_{\text{cond}}) \tag{3.16}$$

Our Transformer implementation uses pre-normalization with a residual connection structure that has been shown to improve training stability. The multi-head attention mechanism allows the model to attend to different aspects of the state conditioned on the action.

Policy Prior

The policy model $p_\theta(a_i|\tilde{s}_t^i)$ represents the prior distribution over actions given the current state, as introduced in the ELBO formulation in equation 3.6. Rather than using a fixed prior, we implement a learned prior through a simple parameterization:

$$p_\theta(a_i|\tilde{s}_t^i) = \text{Categorical}(\boldsymbol{\pi}) \tag{3.17}$$

where $\boldsymbol{\pi} \in \mathbb{R}^A$ are learnable parameters representing logits for each action. This learned prior adapts to the structure of the data, encouraging the model to discover coherent policies for each entity rather than arbitrary action assignments.

Implementing the ELBO Component

To implement the ELBO component of the $\mathcal{L}_{\text{VAD}}$ objective derived in Section 3.3, we compute:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_\phi(a_i|\tilde{s}_t^i,\tilde{s}_{t+1}^i)}\left[\log p_\theta(\tilde{s}_{t+1}^i|a_i,\tilde{\mathbf{s}}_t)\right] - D_{\text{KL}}(q_\phi(a_i|\tilde{s}_t^i,\tilde{s}_{t+1}^i)\|p_\theta(a_i|\tilde{s}_t^i)) \tag{3.18}$$

This computation directly corresponds to the individual slot-wise terms in the $\mathcal{L}_{\text{VAD}}$ objective from Equation 3.8.

3.4.4 Spatial Broadcast Decoder

We implement a spatial broadcast decoder that projects slot representations back to pixel space. This approach, introduced by Watters et al. (2019), aligns with our objective to maintain spatial consistency in object representations.

The key of spatial broadcast decoding is to first broadcast each slot representation spatially and then process it with convolutional layers, rather than using transposed convolutions. This avoids checkerboard artifacts and maintains spatial consistency.

For each slot $\tilde{s}_{t+1}^i$, the spatial broadcast process creates a feature map where the slot representation is repeated at each spatial location and augmented with positional embeddings:

$$\mathbf{h}_{\text{spatial}}^i = \text{SpatialBroadcast}(\tilde{s}_{t+1}^i, H, W) + \text{PositionalEmbedding}(H, W) \tag{3.19}$$

This feature map is then processed with a CNN to generate both a reconstruction and an alpha mask:

$$\mathbf{x}_{t+1}^i, \alpha_{t+1}^i = \text{DecoderCNN}(\mathbf{h}_{\text{spatial}}^i) \tag{3.20}$$

The final reconstruction combines these per-slot outputs using a weighted sum:

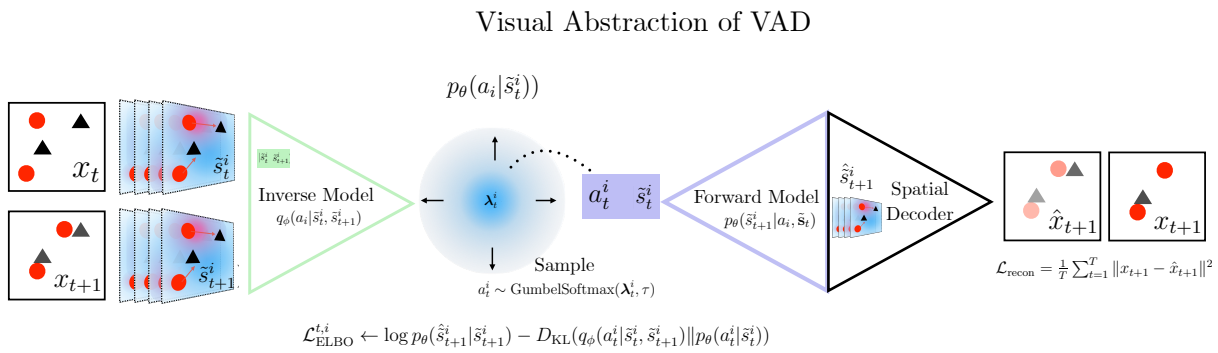

Figure 3.3: Visual abstraction of the VAD conditional variational autoencoder architecture. The process flows from left to right: consecutive frames x_t and x_{t+1} are encoded into slot representations $\tilde{\mathbf{s}}_t$ and $\tilde{\mathbf{s}}_{t+1}$. These slot representations feed into the inverse dynamics model (encoder), which produces logits $\boldsymbol{\lambda}$ that are used to sample latent actions a via the Gumbel-Softmax trick. The sampled action a, together with slot representation $\tilde{\mathbf{s}}_t$, is processed by the forward dynamics model (decoder) to predict the next slot state $\hat{\tilde{\mathbf{s}}}_{t+1}$. Finally, the predicted slot representation is passed through the spatial broadcast decoder to generate the reconstructed frame $\hat{x}_{t+1}$. This end-to-end architecture optimizes the $\mathcal{L}_{\text{VAD}}$ objective by balancing reconstruction quality with the consistency between inferred actions and learned policies.

$$\hat{\alpha}_{t+1}^i = \frac{\exp(\alpha_{t+1}^i)}{\sum_{j=1}^{K} \exp(\alpha_{t+1}^j)} \tag{3.21}$$

$$\hat{x}_{t+1} = \sum_{i=1}^{K} \hat{\alpha}_{t+1}^i \odot \mathbf{x}_{t+1}^i \tag{3.22}$$

3.4.5 Training

Training Stabilization Techniques

Challenge	Solution
Slot representation optimization	Implicit differentiation in Slot Attention (gradient flow only through final iteration)
Discrete action sampling	Gumbel-Softmax with temperature annealing
Mode collapse in action space	Entropy regularization on policy prior

The temperature parameter τ in the Gumbel-Softmax is annealed during training from an initial value of 1.0 to a final value of 0.1, gradually making the action sampling more discrete. This annealing schedule is crucial for balancing exploration and exploitation in the action space.

For the reconstruction loss component of $\mathcal{L}_{\text{VAD}}$, we use a mean squared error between the original frames and their next time-step reconstructions:

$$\mathcal{L}_{\text{recon}} = \frac{1}{T} \sum_{t=1}^{T} \|x_{t+1} - \hat{x}_{t+1}\|^2 \tag{3.23}$$

As shown in Section 4.2, the same $\mathcal{L}_{\text{VAD}}$ objective can also serve as an effective auxiliary loss for reinforcement learning, improving sample efficiency across various multi-agent tasks.

3.4.6 XLA Acceleration

The implementation leverages JAX (Bradbury et al., 2018) and Flax (Heek et al., 2024) for efficient computation. JAX's functional programming model enables seamless differentiation through complex computational graphs, including the Gumbel-Softmax reparameterization. Flax provides a modular framework for defining neural network architectures with clear separation of parameters and computation.

The advantage of our implementation is the reduction in training time achieved through XLA (Accelerated Linear Algebra) compilation. Our preliminary experiments with PyTorch implementations of similar variational agent discovery models required weeks of training on high-end GPUs. By reimplementing the entire pipeline in JAX/Flax with just-in-time (JIT) compilation, we reduced training time from weeks to hours—a speedup of more than 50×.

This acceleration stems from several JAX-specific optimizations:

JAX Performance Optimizations

Optimization	Impact
Just-in-time compilation	Converts Python functions to optimized machine code
Automatic vectorization	Parallelizes operations across batch dimensions
Fused operations	Combines multiple operations into single GPU kernels
Static graph optimization	Eliminates Python overhead from the computation loop
Functional programming model	Enables efficient computation caching and reuse

The performance gains are particularly significant for our model because of the complex nested computation graph involving slot attention, transformer-based forward models, and

gumbel-softmax operations. In PyTorch, these operations would be executed sequentially with significant Python overhead, whereas JAX's JIT compilation creates optimized kernels that execute entirely on the accelerator.

Another critical optimization was rendering game frames directly in JAX rather than using Python-based game engines. By implementing environment renders as pure JAX functions, we eliminated costly CPU-GPU transfers and enabled end-to-end differentiation through the entire pipeline, including environment interactions.

The JIT compilation and functional programming model of JAX enables us to leverage hardware acceleration more efficiently, making our approach computationally feasible for complex multi-agent environments with high-dimensional observations.

Chapter 4

Experimental Evaluation

This chapter evaluates our VAD model across multiple environments of increasing agent complexity, and tests the efficacy of our $\mathcal{L}_{\text{VAD}}$ objective as an auxiliary loss for reinforcement learning. We examine the model's capacity to infer latent actions from visual observations and generalize to novel scenarios, as well as how our variational objective can improve sample efficiency in multi-agent settings. Our experiments address four fundamental questions:

Research Questions

1. Do learned slot representations encode agent-centric information about policies, goals, and agent structures? (Addressed in Tables 4.4, 4.5, 4.1, 4.2, 4.3, and Figure 4.4)
2. Does our VAD model generalize to novel agents, goals, environmental configurations, and capture principles of rational agency? (Addressed in Tables 4.4, 4.5, 4.1, 4.2, 4.3, Figures 4.4 and 4.5)
3. Does $\mathcal{L}_{\text{VAD}}$ from 3.3 improve sample efficiency in multi-agent reinforcement learning when used as an auxiliary loss to PPO? (Addressed in Figure 4.6 and Table 5.1)
4. Do learned slot representations exhibit mirror-neuron-like properties for agent action perception? (Addressed in Figures 4.7–4.11 and Section 4.3)

4.1 Agent-centric Slot Representations and Generalization

In this section, we investigate the first two research questions by examining whether our VAD model learns meaningful agent-centric slot representations and whether these representations enable generalization to novel scenarios.

4.1.1 Evaluation Methodology

To evaluate the representations of our learned slot embeddings, we employ a linear probing methodology. Linear probes assess whether learned representations encode specific information without extensive fine-tuning (Alain and Bengio, 2018).

For each environment, we collect ground truth data on agents' actions a_t^i and goals g^i (where applicable). We train simple single-layer MLP classifiers on the learned slot representations $\tilde{s}_t^i$ to predict these ground truth labels:

$$\hat{a}_t^i = \text{softmax}(W_a \tilde{s}_t^i + b_a) \tag{4.1}$$

$$\hat{g}^i = \text{softmax}(W_g \tilde{s}_t^i + b_g) \tag{4.2}$$

where W_a, b_a, W_g, b_g are the learned parameters of the linear probes. The classifiers are trained using standard cross-entropy loss.

At each time step, we compute the prediction accuracy for each of the K slots, rank them from highest to lowest, and then average these ranked accuracies across all time steps and episodes. This provides a clear picture of how well the best slots capture agent-centric information. We quantify generalization capabilities using four complementary metrics. The **Performance Drop** measures the percentage (change) decrease from familiar to novel conditions. The **Novel Agent Gap** measures the percentage drop from the average performance on familiar agents to the novel agent. The **Generalization Advantage** quantifies the absolute percentage point improvement of our method over the baseline in novel conditions. The **Relative Improvement** shows how many times better our model handles generalization compared to the baseline.

All experimental results (Tables 4.4, 4.5, 4.1, 4.2, 4.3) are averaged over 15 random seeds.

4.1.2 Experimental Environments

We evaluate our model across three distinct environments of increasing complexity, systematically progressing from single-agent to multi-agent scenarios. Across all environments, several experimental parameters remain consistent: all environments are rendered as $64 \times 64 \times 3$ RGB images, agents follow optimal policies designed to maximize task performance, and we conduct evaluations in both familiar scenarios (similar to training) and novel scenarios (to test generalization).

Minigrid: Single-Agent Goal-Directed Behavior

The Minigrid environment (Chevalier-Boisvert et al., 2018) provides a discrete grid-world setting where a single agent navigates to pursue particular goal objects.

The environment consists of a 10×10 grid where a single agent (red triangle) navigates around obstacles to reach goal objects. The agent has full observability of the grid and operates with four discrete actions: forward, left, right, and pickup object. We configure the model with $K = 4$ slots to account for the agent, goal object, and potential distractor goals. During training, the model observes the agent pursuing a green box, while a red key is reserved for generalization testing.

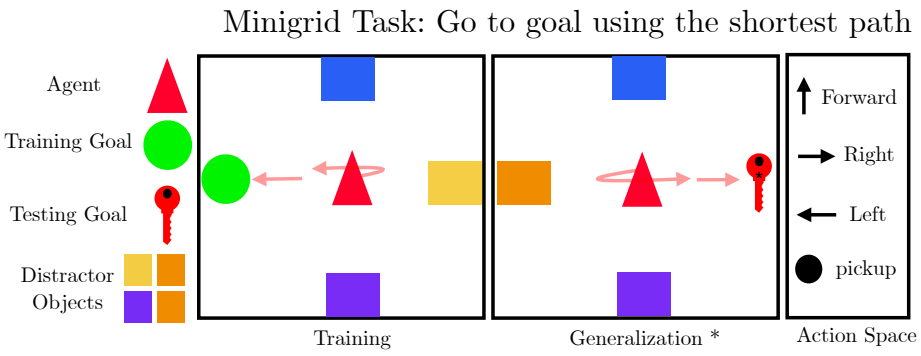

Figure 4.1: Minigrid environment configurations. Left: Training setup with red triangle agent navigating to green box goal using optimal (shortest) path. Right: Testing setup where agent must navigate to novel red key ($\star$) goal. The environment contains distractor goals that are irrelevant to the task.

The agent follows an optimal policy that navigates toward the goal object using the shortest feasible path while avoiding obstacles. For evaluation, we assess both action prediction (predicting the agent's next action given its slot representation) and goal prediction (identifying which of four possible goals the agent is pursuing) (Fig 4.1).

Overcooked: Two-Agent Cooperative Interaction

The Overcooked environment (Carroll et al., 2020) is a cooperative game where two agents must coordinate to prepare and serve dishes in a virtual kitchen.

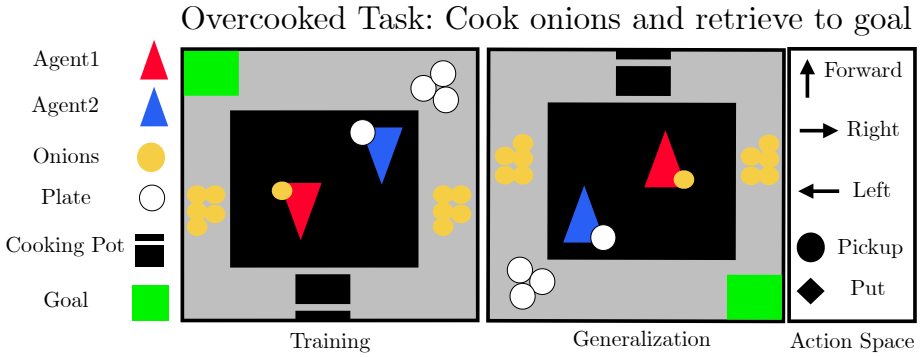

Figure 4.2: Overcooked environment configurations. Left: Training setup with red and blue triangle agents coordinating to collect yellow onions, cook them in black pots, place them on white plates, and deliver to green goal. Right: Testing setup with vertically flipped layout, requiring agents to adapt coordination strategies to novel spatial configurations.

The environment consists of a grid-based kitchen with various stations: grey counters,

black cooking pots, yellow onion ingredients, white plates, and green serving goals. Two agents (red and blue triangles) navigate this space to complete a cooking workflow. The action space consists of five discrete actions: forward, left, right, pickup, and put down. Agents can pick up raw ingredients from counters, put them in cooking pots, retrieve cooked ingredients with plates, and deliver completed dishes to goal locations. As shown in Figure 4.2, we evaluate both the training configuration and a test scenario where the kitchen layout is vertically flipped, altering the positions of all agents, stations, and goals to assess spatial generalization capabilities. We configure the model with $K = 8$ slots to account for both agents and multiple objects in the scene.

The agents follow optimal cooperative policies that maximize the number of dishes served in the fewest steps. These policies require sophisticated coordination: typically one agent specializes in collecting ingredients while the other focuses on cooking and serving. For evaluation, we focus on action prediction for both agents, assessing how well the model can identify and predict the complementary roles each agent adopts.

Multi-Agent Particle Environment (MPE): Novel Three Agent Discovery

The Multi-Agent Particle Environment (Lowe et al., 2020) provides a controlled testbed for multi-agent coordination and generalization to novel agents.

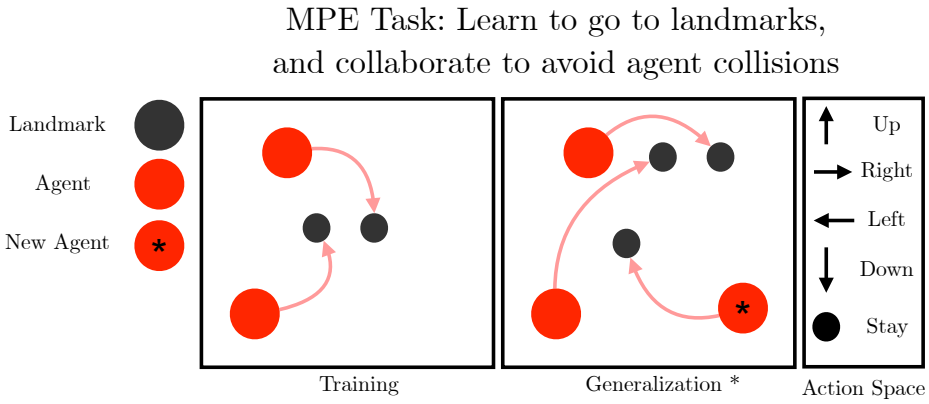

Figure 4.3: MPE environment configurations. Left: Training setup with two red circle agents (connected by light red trajectory lines) navigating toward dark grey landmark goals while avoiding collisions. Right: Testing setup with an additional third agent (marked with $\star$) and a third goal, requiring agents to generalize their coordination strategy.

We implement the Simple Spread game, where agents must cover all landmarks while avoiding collisions. Each agent is represented as a colored circle (red) that can move in the 2D space by applying forces in cardinal directions. Landmarks are represented as dark grey circles. The action space consists of five discrete actions: up, down, left, right, and stay. Agents are globally rewarded based on how far the closest agent is to each landmark (sum of the minimum distances), while being locally penalized for agent-agent collisions (-1 per

collision). We configure the model with $K = 6$ slots, allowing it to potentially allocate slots to both agents and landmarks.

This cooperative task requires agents to coordinate their movements to cover all landmarks while avoiding collisions. The optimal policy distributes agents efficiently among landmarks, minimizing the sum of distances while preventing collisions. For evaluation, we define both action prediction (predicting the agent's next action given its slot representation) and categorical goal prediction (identifying which landmark each agent is targeting).

4.1.3 Representation Quality and Generalization Results

Minigrid Results

Tables 4.1 and 4.2 present the results for action and goal prediction in the Minigrid environment under both familiar and novel goal conditions.

Table 4.1: Action prediction accuracy (%) in the Minigrid environment

Model	Old Goal	New Goal	Performance (Change) Drop	Generalization Advantage
Random Baseline	25.0	25.0	0.0%	–
Object-Centric (SAVi)	80.0	46.0	42.5%	–
VAD (Ours)	**93.0**	**79.0**	**15.1%**	**+33.0%**

Performance Drop shows percentage decrease from Old Goal to New Goal. Generalization Advantage shows how much better our method performs compared to the baseline on the new goal.

Table 4.2: Goal prediction accuracy (%) in the Minigrid environment

Model	Old Goal	New Goal	Performance (Change) Drop	Generalization Advantage
Random Baseline	25.0	25.0	0.0%	–
Object-Centric (SAVi)	82.0	51.0	37.8%	–
VAD (Ours)	**95.0**	**83.0**	**12.6%**	**+32.0%**

Goal prediction involves identifying which of four possible goals the agent is pursuing. Performance Drop shows percentage decrease from Old Goal to New Goal conditions.

Our VAD model not only achieves higher action prediction accuracy on the familiar goal (green box) but also shows better generalization to the novel goal (red key). As shown in Table 4.1, the performance drop for our model is only 15.1%, compared to a 42.5% drop for the object-centric baseline. Similarly, for goal prediction (Table 4.2), our VAD model maintains accuracy (83.0%) even on the novel goal, while baseline performance drops to 51.0%.

Overcooked Results

Table 4.3 presents the action prediction results for both agents in the Overcooked environment, comparing performance in familiar versus novel kitchen layouts.

Table 4.3: Action prediction accuracy (%) in the Overcooked environment

Model	Agent	Familiar Config	Novel Config	Performance (Change) Drop	Generalization Advantage
Random Baseline	Agent 1	20.0	20.0	0.0%	–
	Agent 2	20.0	20.0	0.0%	
Object-Centric (SAVi)	Agent 1	69.0	50.0	27.5%	–
	Agent 2	68.0	48.0	29.4%	
VAD (Ours)	Agent 1	**81.0**	**72.0**	**11.1%**	**+22.0%**
	Agent 2	**80.0**	**70.0**	**12.5%**	**+22.0%**

Our VAD model demonstrates higher action prediction accuracy for both agents, with better generalization to novel kitchen configurations. The agent-centric model shows a smaller performance drop (11-13%) compared to the object-centric baseline (27-29%) when tested on novel layouts.

Multi-Agent Particle Environment Results

Tables 4.4 and 4.5 present the results for action and goal prediction in the MPE environment. Our VAD model is evaluated against both a random baseline and a standard object-centric model (SAVi).

Table 4.4: Action prediction accuracy (%) for the best slots in the MPE environment

Model	Agent 1	Agent 2	Agent 3⋆	Novel Agent Gap
Random Baseline	20.0	20.0	20.0	0.0%
Object-Centric (SAVi)	64.0	66.0	48.0	-26.2%
VAD (Ours)	**81.0**	**83.0**	**76.0**	**-7.3%**

⋆ indicates the generalization agent not seen during training. Novel Agent Gap shows the percentage drop from the average of Agent 1 & 2 to Agent 3.

Table 4.5: Goal prediction accuracy (%) for the best slots in the MPE environment

Model	Agent 1	Agent 2	Agent 3⋆	Novel Agent Gap
Random Baseline	33.3	33.3	33.3	0.0%
Object-Centric (SAVi)	77.0	72.0	58.0	-22.1%
VAD (Ours)	**85.0**	**88.0**	**84.0**	**-2.9%**

⋆ indicates the generalization agent not seen during training. Novel Agent Gap shows the percentage drop from the average of Agent 1 & 2 to Agent 3.

Our VAD model significantly outperforms the baseline object-centric model (SAVi) on both action and goal prediction tasks. Notably, our model maintains high accuracy even for

the novel agent (Agent 3⋆), with only a modest performance drop compared to the familiar agents. For action prediction, our VAD model shows only a 7.3% drop in performance for the novel agent, compared to a 26.2% drop for the object-centric baseline. Similarly for goal prediction, our model shows a minimal 2.9% drop compared to the baseline's 22.1% drop.

4.1.4 Qualitative Analysis of Learned Representations

Beyond quantitative results, we perform qualitative analyses on the slot representations to gain insights into what the VAD model's slots have learned about agents and their goals.

Figure 4.4 presents a comparative analysis of slot reconstructions generated by our model's spatial broadcast decoder across all three environments. These visualizations reveal how our approach decomposes scenes into meaningful entity-centric representations that maintain consistency even in novel test scenarios.

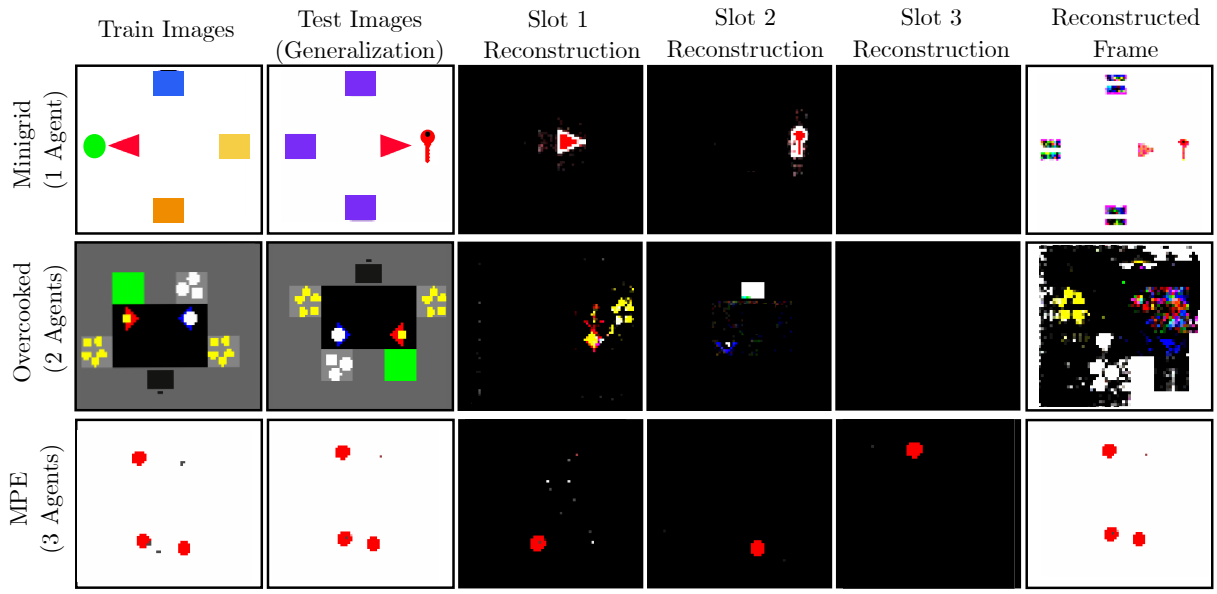

Figure 4.4: Slot-based reconstructions across all three environments demonstrating entity-centric decomposition capabilities. Each row represents one environment (Minigrid, Overcooked, MPE from top to bottom), with columns showing: (1) Training scenario, (2) Testing scenario with novel elements, (3-5) Individual slot reconstructions, (6) Full reconstructed image. In Minigrid, slot 1 isolates the agent and slot 2 captures the goal object, even when switched from green box to red key. In Overcooked, slot 1 reconstructs the red agent and associated onions, while slot 2 captures the blue agent, cooking pot, and progress bar. In MPE, slots 1, 2, and 3 perfectly isolate individual agents, generalizing to the novel third agent in the test scenario.

In the Minigrid environment (top row), our VAD model allocates distinct slots to the agent (slot 1) and the goal object (slot 2), maintaining this separation even when the goal changes from a familiar green box during training to a novel red key during testing. Notably,

slot 3 remains empty (black), demonstrating the model's selectivity in focusing only on behaviorally relevant entities (agents, goals) while ignoring distractor objects.

The Overcooked environment (middle row) showcases more complex agent-object relationships. Slot 1 consistently captures the red agent along with its most frequently interacted objects (onions), while slot 2 reconstructs the blue agent along with the cooking pot and its associated progress bar. This functional grouping suggests the model has learned to associate entities not just by visual appearance but by their behavioral relationships. The consistency of these assignments holds even when the environment is vertically flipped during testing, indicating strong generalization of these entity-centric representations.

The MPE environment (bottom row) demonstrates the VAD model's capacity to scale to scenarios with a novel number of agents. During training with two agents and landmarks, the model assigns slot 1 and slot 2 to each agent. Remarkably, when presented with a novel scenario containing three agents, the model maintains consistent slot assignments for the familiar agents while appropriately allocating slot 3 to the novel agent. This perfect agent isolation across slots provides visual confirmation of our quantitative findings on novel agent generalization.

These qualitative results highlight three important aspects of our VAD model: (1) it consistently assigns specific entities to dedicated slots, (2) it maintains these assignments across novel scenarios, and (3) it appears to organize entities based on behavioral relevance rather than merely visual appearance. This structured agent-centric decomposition provides the foundation for the improved generalization demonstrated in our quantitative evaluations.

4.1.5 Rational Action Prediction

To further evaluate our VAD model's capacity to capture agent intentionality, we designed an experiment inspired by Gergely and Csibra (2003) work on teleological reasoning in infants. In their study, infants who observed an agent jumping over an obstacle to reach a goal expected the agent to take a direct path when the obstacle was removed, suggesting an early-developing capacity to interpret actions in terms of rational means to goals.

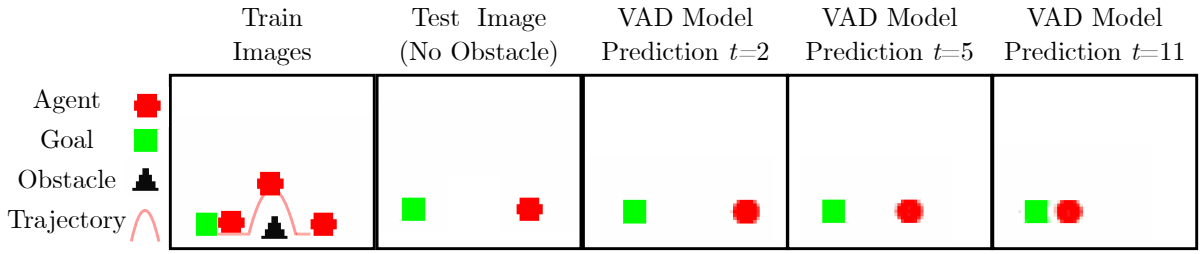

Figure 4.5: Rational action prediction results. From left to right: (1) Training scenario showing red agent jumping over black obstacle to reach green goal, (2) Test scenario with obstacle removed, (3-5) Model-generated predictions (image reconstructions) at timesteps $t = 2$, $t = 5$, and $t = 11$. Despite never observing the agent in an obstacle-free environment during training, the model predicts the agent will take a direct path to the goal rather than maintaining the jumping trajectory, aligning with rational action principles.

We adapted this paradigm to test our model's predictive capabilities. During training, the model only observed scenarios where a red ball agent jumped in an arc trajectory over a black obstacle to reach a green goal box. For the test condition, we removed the obstacle and observed the model's predictions (reconstructed images) of future timesteps without any additional training.

As shown in Figure 4.5, our VAD model correctly predicted that the agent would take a direct, efficient path to the goal when the obstacle was removed, despite never having observed this scenario during training. This behavior mirrors the expectations documented in 12-month-old infants by Gergely and Csibra (2003), who demonstrated that infants show increased looking times when agents maintain unnecessary jumping trajectories after obstacles are removed.

This result suggests that our VAD model captures not just perceptual aspects of agency but also rudimentary principles of rational action—the expectation that agents will take the most efficient path to achieve their goals.

The model appears to have disentangled the agent's goal (reaching the green box) from the specific trajectory necessitated by environmental constraints, enabling it to generate rational predictions when those constraints change.

4.2 Improving Multi-Agent Reinforcement Learning

To address our third research question, we evaluate whether our agent-centric representations lead to improved performance in multi-agent reinforcement learning tasks. We incorporated our $\mathcal{L}_{\text{VAD}}$ loss as an auxiliary loss during MARL training. Figure 4.6 presents performance results in the MPE environment. Our experimental setup consists of three agents collab-

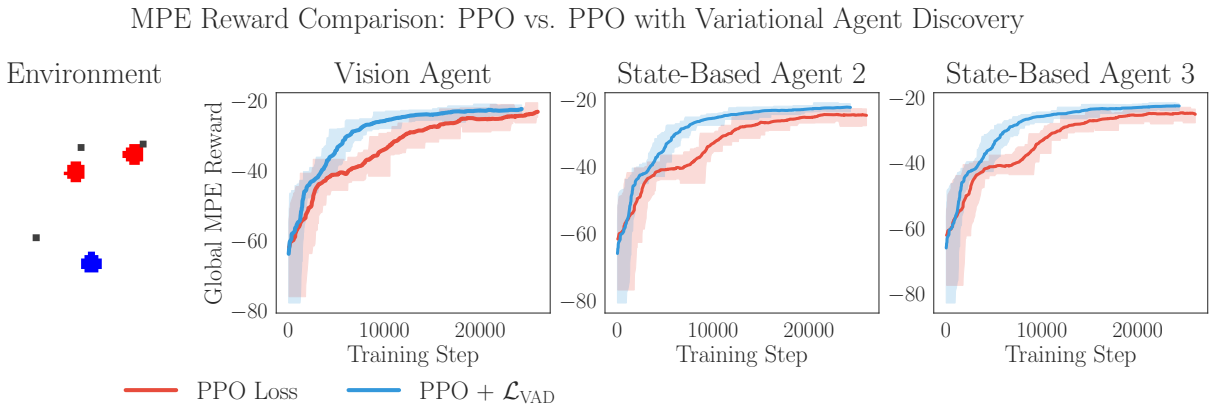

Figure 4.6: Performance comparison between standard Independent PPO (red) and Independent PPO augmented with our $\mathcal{L}_{\text{VAD}}$ loss (blue) in the MPE environment. The leftmost panel shows the environment with three agents: one vision-based agent (dark blue) and two state-based agents (red). The three right panels display reward curves for each agent during training in a generalization scenario with novel agent and landmark configurations. All agents benefit from the additional $\mathcal{L}_{\text{VAD}}$ loss, with the vision agent showing particularly improved sample efficiency.

orating in a parallelized JAX implementation of the MPE Gym environment. The blue agent (leftmost panel) operates solely from visual observations, while the two red agents have full access to environment state information, including the blue agent's position. All agents are trained using Independent PPO but receive rewards influenced by the global team performance, resulting in similar reward trajectories.

Importantly, both the baseline and our model use identical perception architectures: a CNN encoder followed by Slot Attention for scene decomposition. The key difference is that our approach augments this architecture with our $\mathcal{L}_{\text{VAD}}$ loss. This isolates the impact of our variational formulation while controlling for encoder capabilities.

When comparing training with standard PPO loss versus PPO augmented with our $\mathcal{L}_{\text{VAD}}$ loss, we observe consistently improved sample efficiency across all agents. The VAD-augmented model (teal) achieves higher rewards earlier in training and maintains this advantage throughout. While both methods ultimately converge to similar performance levels, our approach reaches convergence with significantly fewer environment interactions.

4.3 Mirror-Like Neural Representations in Slot Activations

This section explores the presence of mirror-like neural representations within the learned slot vectors of our variational agent discovery model, specifically in the MPE environment where multiple agents perform similar actions. We posit that if our model is truly learning agent-centric representations, then the slot vectors corresponding to different agents should exhibit correlated activation patterns when the agents perform the same actions. This behavior would be analogous to biological mirror neurons, which activate both when an animal performs an action and when it observes another agent performing the same action (Rizzolatti and Craighero, 2004).

4.3.1 Evaluation Methodology

To investigate the presence of mirror-like neural patterns, we performed a detailed analysis of the slot vector activations conditioned on agent actions in the MPE environment. We manually mapped three slots to the three agents through visual inspection across 15 episodes, confirming that our model consistently assigns specific slots to individual agents.

Each slot representation $\tilde{\boldsymbol{s}}_t^i$ is a 128-dimensional vector encoding the agent's state. For each agent and corresponding slot, we grouped the slot activation vectors according to the action taken by the agent at that timestep:

$$\tilde{\boldsymbol{s}}_{a,i} = \{\tilde{\boldsymbol{s}}_t^i \mid a_t^i = a\} \tag{4.3}$$

where $\tilde{\boldsymbol{s}}_t^i$ represents the state vector for slot i at timestep t, and a_t^i is the action taken by the agent assigned to slot i. The actions were discretized into five categories: Left, Right, Up, Down, and Stay.

We then computed the Pearson correlation coefficient between the same feature across different agent slots for each action a and feature dimension j:

$$r_{a,j}^{i,k} = \text{Pearson}(\tilde{\boldsymbol{s}}_{a,i}^{j}, \tilde{\boldsymbol{s}}_{a,k}^{j}) \tag{4.4}$$

where $\tilde{\boldsymbol{s}}_{a,i}^{j}$ denotes the j-th component of the slot vectors for agent i when performing action a. In our analysis, we refer to the j-th feature of the 128-dimensional slot vector as Fj (e.g., F57 refers to the 57th feature dimension).

For each action, we computed a mirror score for each feature by averaging the absolute correlation coefficients across all agent pairs:

$$\text{MirrorScore}_{a,j} = \frac{1}{|P|} \sum_{(i,k)\in P} |r_{a,j}^{i,k}| \tag{4.5}$$

where P is the set of all agent pairs. We then identified the top five features with the highest mirror scores for each action.

4.3.2 Results

Our analysis reveals clear evidence of mirror-like neural representations within the learned slot vectors. For each discrete action (Left, Right, Up, Down, Stay), we identified specific features that show strong correlations across different agent slots.

Figure 4.7 shows the mirror neuron analysis for the Right action. The feature activation heatmap (left panel) displays the mean activation values of the top five mirror features across the three agent slots. The middle panel presents a 3D scatter plot where each point represents the values of a specific feature across all three agent slots simultaneously, with points from the same feature sharing the same color. The clustering of points from the same feature indicates consistent activation patterns when different agents perform the same action. The right panel visualizes the averaged slot vector activations with the top mirror features highlighted.

A similar analysis for the Up action (Figure 4.8) identifies feature F107 as having the strongest mirror-like properties, with consistent activation patterns across agent slots when agents move upward.

For the Down action (Figure 4.9), we identified feature F78 as having the strongest mirror properties. This feature shows high positive activation values with strong cross-agent correlations when agents move downward.

The Left action analysis (Figure 4.10) reveals a more distributed pattern of mirroring compared to other directional actions. While there are correlations across agent slots, we don't observe a single dominant feature with strong mirror-like properties. Instead, several features contribute to the representation of leftward movement with varying degrees of correlation.

Finally, the Stay action (Figure 4.11) reveals a more complex encoding pattern where multiple features work in combination rather than a single dominant feature, such as the negative activation of F27 and F43 alongside positive activation of F80.

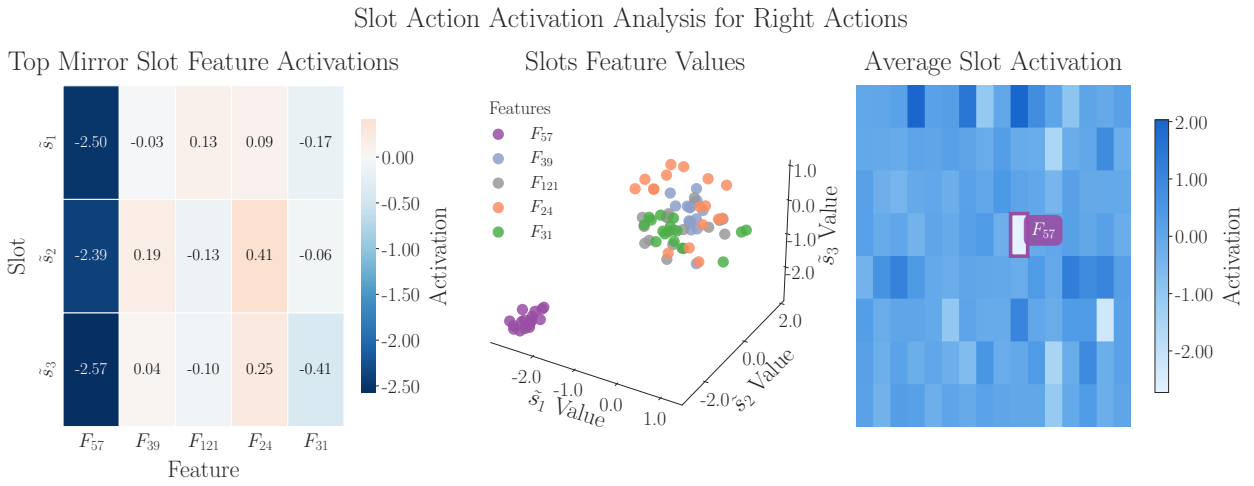

Figure 4.7: Mirror neuron analysis for the Right action. **Left:** Feature activation heatmap showing mean activation values for the top five mirror features (columns) across three agent slots (rows). **Middle:** 3D scatter plot where each point represents a specific feature's values across all three agent slots, with points of the same color representing the same feature. The clustering pattern indicates similar activation profiles across different agents performing the same action. **Right:** Average slot activation pattern across all agents, with feature F57 highlighted (white box) as having the strongest mirror properties for the Right action. Feature dimensions were reshaped to an 8×16 grid for visualization.

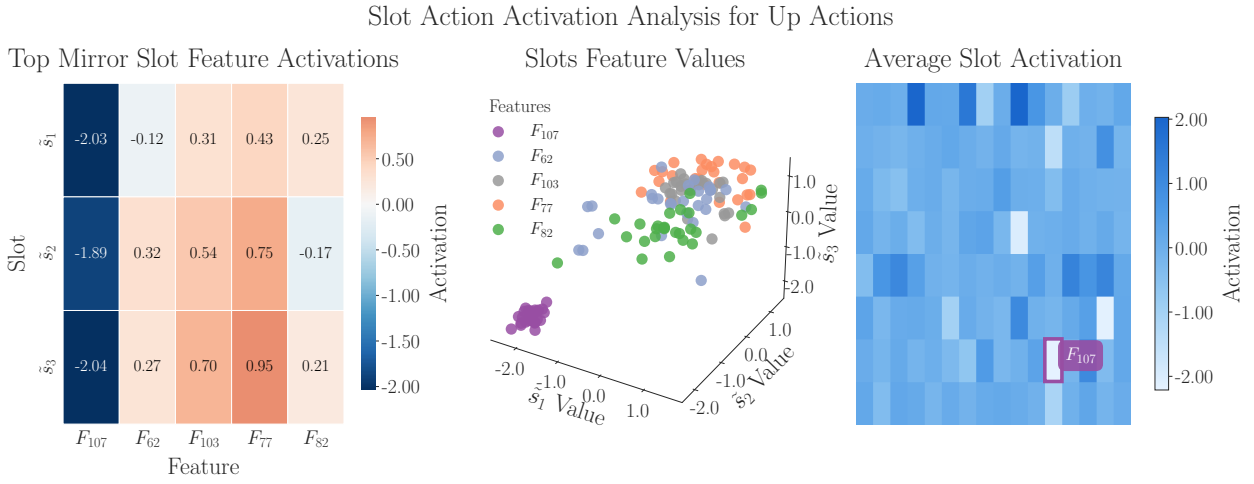

Figure 4.8: Mirror neuron analysis for the Up action. The visualization follows the same structure as Figure 4.7. Feature F107 (highlighted in the right panel) shows the strongest mirror properties for this action, with highly correlated activation patterns across all three agent slots when they perform upward movement.

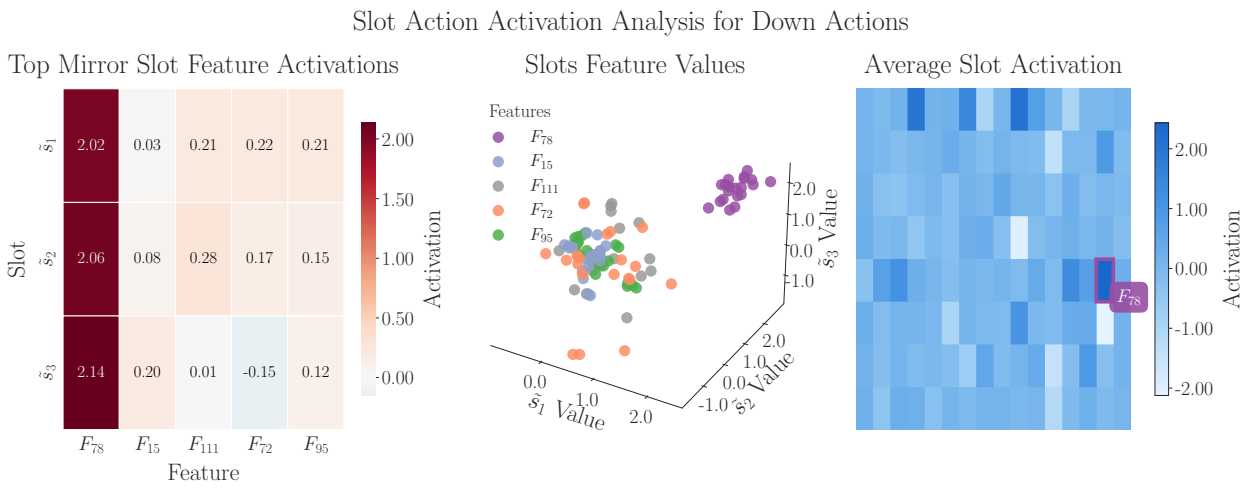

Figure 4.9: Mirror neuron analysis for the Down action. Feature F78 (highlighted in the right panel) exhibits the strongest mirror properties for this action. Note how this feature has consistently positive activation values across all agent slots, in contrast to the mirror features for other actions, which may have negative values for some agents.

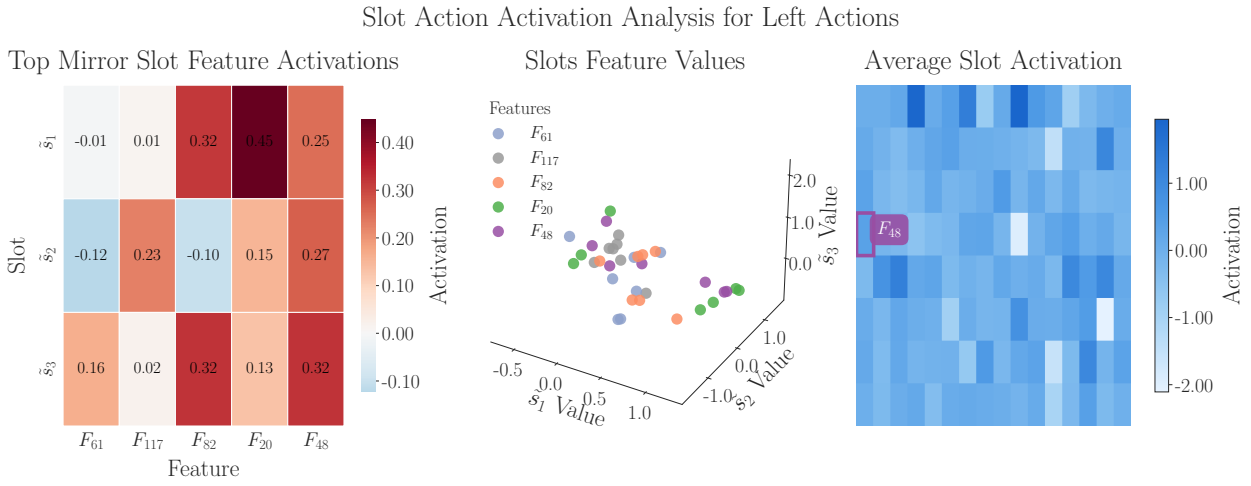

Figure 4.10: Mirror neuron analysis for the Left action. The visualization follows the same structure as previous figures. Unlike Right, Up, and Down actions, leftward movement is represented by multiple features with moderate correlations rather than a single strongly correlated feature across agent slots.

4.4 Summary of Findings

The slot-based representations learned by our VAD model demonstrate consistent entity tracking with functional decomposition of scenes, as evidenced by our qualitative analysis in Figure 4.4. The identification of mirror-like neural patterns in slot activations suggests our VAD model develops a common neural code for agent actions that generalizes across entities—a computational analog to biological mirror neuron systems. The action-specific

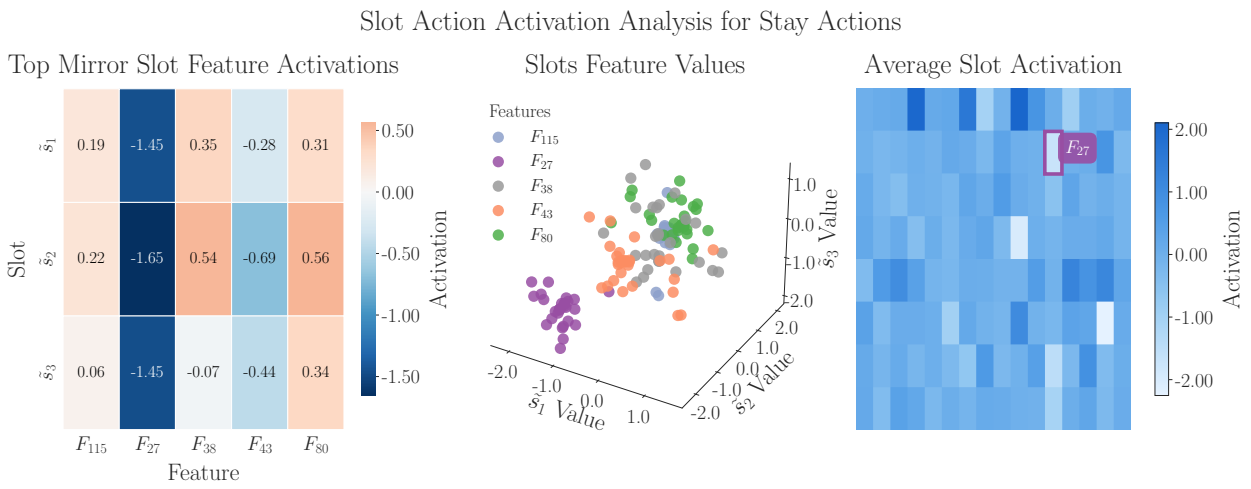

Figure 4.11: Mirror neuron analysis for the Stay action. The stationary state appears to be encoded through a combination of features, including negative activation of F27 and F43 alongside positive activation of F80, suggesting that the absence of movement is represented through a pattern of coordinated feature activations rather than a single dedicated feature.

nature of these patterns indicates specialized feature encodings for different actions.

The VAD model's ability to predict rational actions in modified environments (Figure 4.5) aligns with teleological reasoning capabilities observed in infant cognition studies. This emergent capacity for anticipating efficient means to goals without explicit training suggests deeper abstraction of intentional agency than merely associating visual patterns with actions.

These findings collectively support our hypothesis that modeling latent actions through variational inference using our $\mathcal{L}_{\text{VAD}}$ objective enables learning agent-centric representations that capture meaningful structure in multi-agent environments, improving both representation quality and downstream learning performance. The approach links computational implementation with cognitive science theories of agency perception, offering a quantitative framework for further investigation.

Key Findings

1. **Representation Quality:** Our VAD model consistently outperforms object-centric baselines across all environments (8-17% improvement) as shown in Tables 4.4, 4.5, 4.1, and 4.3, demonstrating more effective encoding of agent-centric information.

2. **Generalization Capability:** The VAD model maintains robust performance in novel scenarios with previously unseen agents, goals, or configurations, with 22-33% advantage over baselines in generalization scenarios, particularly evident in the Novel Agent Gap metrics in Tables 4.4 and 4.5.

3. **Mirror-Like Neural Patterns:** Analysis of slot vector activations in Figures 4.7-4.11 reveals action-specific features that activate consistently across different agent slots, analogous to mirror neuron systems in cognitive neuroscience, with feature F57 strongly correlated with rightward movement and F107 with upward movement.

4. **Rational Action Prediction:** In our adaptation of Gergely & Csibra's (2003) experiment (Figure 4.5), the VAD model correctly predicts efficient trajectories in novel scenarios, suggesting it captures principles of rational agency similar to those observed in infant cognition studies.

5. **Reinforcement Learning Efficiency:** As an auxiliary objective in MARL (Figure 4.6), our $\mathcal{L}_{\text{VAD}}$ improves sample efficiency by 21.8% in early training and maintains a 7.6% final performance advantage, with consistent improvements across both vision-based and state-based agents.

Chapter 5

Discussion

Our experimental evaluation from Ch. 4 demonstrates several key findings, summarized in Table 5.1.

Table 5.1: Summary of performance improvements and generalization capabilities across representational and reinforcement learning tasks

Environment	Task	Accuracy Improvement	Generalization Advantage	Rel. Improvement
MPE	Action Prediction	+15-17%	+28%	3.6×
	Goal Prediction	+8-16%	+26%	7.6×
	MARL Reward	+7.6% (final)	+21.8% (early)	1.14×
Minigrid	Action Prediction	+13%	+33%	2.8×
	Goal Prediction	+13%	+32%	3.0×
Overcooked	Action Prediction	+12-13%	+22%	2.4×

Accuracy Improvement: Performance gain over object-centric baseline on standard test cases. For MARL, final reward improvement. *Generalization Advantage*: Performance gain in novel scenarios. For MARL, early training improvement. *Rel. Improvement*: Ratio of our method's performance to baseline in challenging conditions.

This chapter examines these results in light of our central contribution: the $\mathcal{L}_{\text{VAD}}$ objective for unsupervised agent discovery from visual observations. We begin by discussing the generality of our optimization objective, followed by analysis of our VAD model's agent-centric representations, generalization capabilities, connections to cognitive science, limitations, and future work.

5.1 Generality of the $\mathcal{L}_{\text{VAD}}$ Objective

A key strength of our work lies in the generality of the $\mathcal{L}_{\text{VAD}}$ objective itself. While our implementation uses specific architectural choices—Slot Attention for structured scene representation and Gumbel-Softmax for discrete action modeling—the underlying optimization objective makes minimal assumptions about the specific mechanisms used to realize these components.

The fundamental premise of our approach is that agent discovery can be formulated as a variational inference problem with latent actions. This formulation requires only that we have (1) some form of structured, factored representation of the visual input that distinguishes potential agents from the background environment, and (2) a means of inferring and generating actions that explain transitions between these representations. The specific implementation details are largely independent of this core theoretical framework.

In our current work, we chose Slot Attention (Kipf et al., 2021) as our structured representation mechanism because it has demonstrated strong performance in object-centric learning. However, our $\mathcal{L}_{\text{VAD}}$ objective could readily accommodate other approaches to structured scene decomposition, such as MONet (Burgess et al., 2019), IODINE (Greff et al., 2020), or future advances in object-centric perception. Similarly, while we implemented discrete actions using the Gumbel-Softmax trick, our framework could be adapted to continuous action spaces using traditional reparameterization techniques or more sophisticated density estimation approaches.

This generality suggests that our contribution—the framing of agent discovery as inference over latent actions through the $\mathcal{L}_{\text{VAD}}$ objective—should remain valuable as structured representation methods continue to evolve. Future work might explore how our objective performs when combined with alternative approaches to scene decomposition or when scaled to more complex action spaces. The flexibility of our formulation may allow it to benefit from advances in both object-centric representation learning and variational inference techniques, potentially leading to even stronger agent discovery capabilities.

Furthermore, this generality aligns with cognitive theories suggesting that the perception of agency is a flexible, abstract capability that operates over diverse sensory inputs and behavioral patterns. Just as humans can recognize agency across vastly different perceptual contexts—from simple geometric shapes to complex biological motion—our approach provides a generalizable computational framework that could potentially adapt to many different manifestations of agency.

5.2 Agent-Centric Representations

Our VAD model consistently outperforms the object-centric baseline across all environments and evaluation metrics (see Table 5.1). The VAD model achieves this by structuring its representations around agent-centered policies and latent actions rather than merely tracking visual features or object positions.

5.2.1 Predictive Power for Agent Properties

Beyond the basic capacity to identify and track agents, our VAD model demonstrates strong performance in predicting specific agent properties. As shown in Tables 4.4, 4.5, 4.1, and 4.2, the learned slot representations consistently enable accurate prediction of both agent actions and goals across all environments. This suggests that the VAD model captures not only entities that are agents but also their goals and behavior.

The ability to predict these properties from learned representations could have beneficial implications for artificial social intelligence. A system that can infer actions and goals may

serve as a foundation for more sophisticated agent modeling capabilities. Future work could explore whether these representations also encode other agency-related properties such as beliefs about the environment (essential for modeling false-belief understanding), reward functions, or value estimates (important for predicting long-term strategic behavior).

The current results suggest that our $\mathcal{L}_{\text{VAD}}$ objective provides a starting substrate for such extensions.

5.2.2 Functional Entity Decomposition

The qualitative analysis of slot reconstructions (Figure 4.4) provides insights into how our VAD model decomposes scenes into functionally relevant entities. Unlike traditional object-centric approaches that decompose based solely on visual properties, our VAD model appears to organize representations according to behavioral significance. For instance, in the Overcooked environment, slot 1 consistently captures the red agent along with its frequently manipulated objects (onions), while slot 2 reconstructs the blue agent with its associated cooking pot. This functional grouping suggests the VAD model forms representations based on agent-object relationships and interaction patterns rather than just visual similarity.

The consistency of these assignments across training and testing conditions indicates that the VAD model has learned robust agent-centric representations that transfer to novel scenarios. Particularly notable is the VAD model's performance in the MPE environment, where it not only maintains consistent slot assignments for familiar agents but appropriately allocates a separate slot to a previously unseen third agent. This suggests the VAD model has learned to identify agent-like entities based on their behavioral characteristics rather than memorizing specific visual patterns from training.

5.2.3 Generalization to Novel Scenarios

A key finding across all environments is our VAD model's generalization performance compared to the object-centric baseline. In the Minigrid environment, our approach demonstrated only a 15.1% performance drop in action prediction when generalizing to a novel goal object (Table 4.1), compared to the baseline's 42.5% drop. Similarly, in the Overcooked environment, our VAD model showed approximately 12% performance degradation when tested on spatially reconfigured kitchens (Table 4.3), whereas the baseline exhibited around 28% degradation.

The most compelling evidence for our VAD model's generalization capabilities comes from the MPE environment, where it achieved 76% action prediction accuracy for a novel third agent (Agent 3$\star$ in Table 4.4) - only 7.3% lower than its performance on familiar agents. This contrasts with the object-centric baseline, which showed a 26.2% performance gap.

These generalization results may be attributed to our explicit formulation of agent discovery as a variational inference problem with latent actions. By structuring the $\mathcal{L}_{\text{VAD}}$ objective around inferring policies and actions that explain observed transitions, the VAD model appears to develop more abstract representations of agency that transfer more readily to novel scenarios. Our $\mathcal{L}_{\text{VAD}}$ objective may encourage the model to discover underlying principles of agent behavior (policies) rather than merely associating specific visual patterns with particular transitions.

5.3 Rational Action Understanding

The rational action prediction experiment (Figure 4.5) provides evidence that our VAD model may capture principles of action reasoning similar to those observed in infant cognition studies. When shown an agent jumping over an obstacle to reach a goal during training, and then presented with the same scenario but with the obstacle removed during testing, the VAD model predicted the agent would take a direct path to the goal. This result may suggest the VAD model has disentangled the agent's goal (reaching the goal object) from the specific trajectory necessitated by environmental constraints.

This behavior aligns with Gergely and Csibra (2003)'s teleological stance theory, which proposes that humans interpret actions in terms of rational means to achieve goals given environmental constraints. The VAD model appears to have developed an implicit understanding that agents typically take efficient paths toward goals when possible, despite never being explicitly trained on obstacle-free scenarios. This suggests our approach can capture not just perceptual aspects of agency but also rudimentary principles of rational action that support prediction in novel situations.

The ability to predict rational agent behavior in modified environments indicates that the VAD model has learned to represent goals and constraints separately, allowing it to generate appropriate predictions when constraints change.

5.4 Improving Multi-Agent Reinforcement Learning

The MARL results demonstrate that incorporating our $\mathcal{L}_{\text{VAD}}$ objective as an auxiliary loss during reinforcement learning can improve sample efficiency while maintaining comparable final performance. As shown in Figure 4.6 and quantified in Table 5.1, agents trained with the auxiliary $\mathcal{L}_{\text{VAD}}$ loss consistently achieved higher rewards earlier in training compared to those using standard PPO alone. Both our loss and the baseline loss eventually converge to similar performance levels, but our agent reaches this convergence point with fewer environment interactions, demonstrating better sample efficiency.

This improvement in sample efficiency suggests that the structured agent-centric representations learned through our $\mathcal{L}_{\text{VAD}}$ objective provide valuable inductive biases that accelerate learning in multi-agent settings. By explicitly modeling other agents as entities with goals and policies, agents may plan and explore more effectively during training.

5.5 Mirror-Like Neural Representations

Our analysis of slot vector activations reveals evidence of mirror-like neural representations within the learned VAD model. For specific actions like Right, Up, and Down, we identified individual features (F57, F107, and F78 respectively) that activate consistently across different agent slots when the corresponding actions are performed, as shown in Figures 4.7, 4.8, and 4.9. This may suggest the VAD model has learned neural codes for representing actions that generalize across different agents.

The action-specific nature of these mirror features aligns with findings from neuroscience research on mirror neurons, which show selectivity for specific actions (Rizzolatti and Craighero, 2004). In our VAD model, directional movements appear to be encoded through distinct, specialized features, while the Stay action involves a distributed pattern of activation across multiple features (negative activation of F27 and F43 alongside positive activation of F80), as shown in Figure 4.11.

These mirror-like representations suggest that our VAD model has learned to abstract the concept of actions beyond the specific agent performing them. This emergence of action encodings across different agent representations was not explicitly encouraged in the training objective but appears to arise naturally from the $\mathcal{L}_{\text{VAD}}$ objective. From a computational perspective, such shared representations may facilitate prediction and understanding in multi-agent scenarios by allowing the VAD model to transfer knowledge about one agent's behavior to another.

It is worth noting that the mirror-like patterns for the Left action (Figure 4.10) were less pronounced than for other directional actions. This may be partially explained by the action distribution in our dataset, as illustrated in Figure 5.1. The Left action was sampled substantially less frequently than other actions across all agent slots.

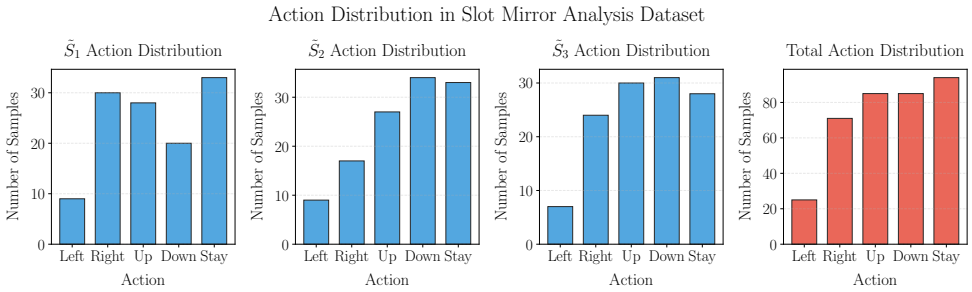

Figure 5.1: Distribution of action samples across the three agent slots (S1, S2, S3) and the cumulative distribution (Total). The horizontal axis shows the five discrete actions (Left, Right, Up, Down, Stay), while the vertical axis shows the sample counts. Note that the Left action consistently has the lowest representation across all slots, potentially explaining the weaker mirror neuron patterns observed for this action.

5.6 Limitations and Future Work

While our VAD model demonstrates significant improvements over object-centric baselines for agent-centric learning, MARL, and other experiments, several limitations suggest directions for future work.

5.6.1 Action Representation Limitations

A key limitation of our experimental design is that we only evaluated environments where agents took discrete actions. While this choice simplified implementation and analysis,

it does not take into account the continuous action space of many other agent environments. Our VAD model architecture is actually quite amenable to continuous action spaces—implementing this would require replacing the Gumbel-Softmax operation with the standard reparameterization (2.3.6) trick commonly used in variational autoencoders with continuous latent variables.

Furthermore, the relatively small action space (5 discrete actions) in our environments provides only a limited test of the VAD model's representational capacity. Real-world scenarios often involve much larger action spaces or continuous action manifolds with high dimensionality. Future work should evaluate the VAD model in environments with significantly larger or continuous action spaces, which would provide a better test of its ability to infer meaningful action distributions and accurately model complex policies.

5.6.2 Generalization to Novel Agent Structures

While our approach demonstrates strong generalization to novel scenarios with familiar agent structures, it may face challenges with agents that have fundamentally different structures or appearances. The Slot Attention mechanism we build upon is known to generalize well to objects with familiar structures but can struggle with entirely novel morphologies (Kipf et al., 2021). Our VAD model inherits this limitation, potentially constraining its application in scenarios where agent appearances differ substantially from those seen during training.

Future work should investigate architectures that can better separate agent function from agent form, perhaps through more explicit disentanglement of structural features in the slot representations. This could enable more robust generalization to agents with novel physical structures but similar behavioral characteristics.

5.6.3 MARL Generalization Evaluation

Our MARL experiments demonstrate improved sample efficiency and final performance when incorporating the $\mathcal{L}_{\text{VAD}}$ objective, but they do not fully explore cross-agent generalization. An important extension would be to evaluate scenarios where the vision-based agent, trained with two red agents, must collaborate with three or more agents at test time.

Additionally, conducting reinforcement learning experiments across all our environments (Minigrid, Overcooked, MPE) would provide a more comprehensive picture of how agent-centric world models affect learning in different contexts. These experiments could reveal whether the benefits of our $\mathcal{L}_{\text{VAD}}$ objective are universal or environment-dependent, and identify which types of multi-agent tasks benefit most from structured agent representations.

5.6.4 Scaling to More Complex Environments

The environments used in our evaluation, while diverse in their agent structures, remain relatively simplified compared to real-world scenarios. The agents follow optimal or near-optimal policies with consistent behaviors, potentially making the task of agent discovery easier than in natural settings where behavior might be more variable or suboptimal.

Future work should test the VAD model in environments with greater visual complexity, more agents, and more varied interaction patterns. Particularly valuable would be evalua-

tions on cognitively inspired visualization datasets like those used in psychological reasoning studies (Shu et al., 2021), which contain more naturalistic agent interactions. Such environments would provide a stronger test of the VAD model's ability to identify agents and infer goals in scenarios closer to human social perception tasks.

Perhaps the most challenging and valid evaluation would be on real-world datasets featuring actual human, animal, or robotic agents. Such datasets would introduce numerous additional complexities absent from simulated environments, including partial occlusions, varying lighting conditions, complex and sometimes irrational behaviors, and significant variations in agent appearances. Evaluating on real-world pedestrian tracking datasets, animal behavior recordings, or sports game footage would provide the ultimate test of whether our approach can scale to the full complexity of naturally occurring agent interactions.

5.6.5 Mirror Neuron Analysis Limitations

The mirror neuron analysis, while suggestive of shared representational structures across agents, has several limitations. First, the correlation-based approach does not account for potential nonlinear relationships between feature activations. Second, the analysis focuses on individual feature dimensions rather than potentially distributed representations across multiple dimensions. More sophisticated techniques such as Canonical Correlation Analysis or methods from interpretable machine learning might reveal additional structure in the learned representations.

Additionally, as shown in Figure 5.1, the uneven distribution of action samples has likely affected the quality of mirror representations for less frequent actions like Left.

5.6.6 Comparison with Language-Based Social Reasoning

Our approach emphasizes vision-based agent detection and social reasoning, which appears to differ meaningfully from language-based Theory of Mind (ToM) capabilities Li et al. (2023); Nguyen (2025). Although not detailed in this thesis, preliminary observations (which we encourage readers to explore) suggest that large vision-language models (VLMs) struggle to identify agents in visual sequences like those used in our experiments. A systematic comparison between vision-based agent discovery models and language-based models on such tasks could illuminate the relationship between different modalities of social cognition and help clarify the unique challenges involved in visual agent perception.

Such benchmarking could strengthen our findings that social reasoning through vision represents a distinct cognitive capacity from language-based ToM reasoning, potentially requiring different computational modules. This comparison would also inform ongoing discussions about embodied versus disembodied social cognition in both artificial intelligence and cognitive science.

5.6.7 Limited Complexity of Social Understanding

While our VAD model shows promising results in agent representation and action prediction, it remains far from capturing the full complexity of human social cognition. The model has not yet been evaluated—and we suspect it may struggle—with representing higher-order

mental states (e.g., beliefs about other agents' beliefs), complex social goals, or deceptive behaviors, all of which are central to human social understanding.

Extending the VAD model to incorporate hierarchical goal structures or nested belief representations would bring it closer to the sophisticated Theory of Mind capabilities observed in human cognition. This might involve augmenting the current architecture with explicit modules for representing mental states of other agents, potentially drawing inspiration from Bayesian models of Theory of Mind.

5.7 Connections to Cognitive Science and Future Validation

5.7.1 Potential Connections to Neural Mechanisms

Our computational approach offers a potential starting point to connect neuroscientific theories of agent perception with computational algorithms. The mirror-like neural patterns observed in our VAD model's representations share some similarities with patterns described in predictive coding frameworks proposed by Kilner et al. (2007b) and Friston et al. (2011) for understanding the mirror neuron system. These frameworks suggest that the mirror system helps humans infer intentions behind observed actions by minimizing prediction errors across the cortical hierarchy.

Similarly, our $\mathcal{L}_{\text{VAD}}$ objective minimizes prediction errors between observed state transitions and those predicted by inferred actions, which may have some analogies to the predictive coding mechanisms hypothesized to underlie biological agent perception. Our optimization objective potentially represents a quantitative, falsifiable hypothesis about mechanisms of agent perception—inspired by Cao and Yamins (2021a)'s Contravariance Principle that models addressing complex computational tasks with minimal simplification might converge on solutions with some shared functional properties with biological systems.

The VAD model's behavior in the rational action prediction experiment (Figure 4.5) suggests that inferring latent actions from transitions may facilitate the emergence of efficiency-based action understanding. This behavior bears some resemblance to Gergely and Csibra (2003)'s teleological stance theory.

By demonstrating these capabilities emerging from the $\mathcal{L}_{\text{VAD}}$ objective operating on visual input, our work suggests the possibility that certain aspects of social cognition might arise from principles of predictive processing, though the extent of overlap between our computational mechanisms and biological implementation remains an open empirical question.

5.7.2 Future Neuroscientific Validation

To move beyond such speculative connections, future work should collect empirical neuroscientific data to test whether similar optimization principles might be implemented in biological systems. Designing human neuroimaging experiments closely mirroring our computational tasks would allow researchers to examine whether neural activity patterns share meaningful similarities with our VAD model's operations.

Such experiments could involve recording neural activity using fMRI or EEG while participants observe agent-based scenarios similar to those in our computational tasks. Of particular interest would be examining neural responses in regions implicated in agent perception, such as the superior temporal sulcus (STS) and temporoparietal junction (TPJ).

For instance, one could examine whether neural representations in the STS contain information about both observed state transitions and potentially inferred actions in a manner that might relate to prediction error minimization. Multivariate pattern analysis (MVPA) techniques could be employed to decode agent-specific information from neural activation patterns and compare the representational geometry with that of our VAD model's learned slot representations.

Additionally, neuroimaging studies could explore whether the mirror-like neural patterns we observed in our VAD model have any corresponding patterns in human mirror neuron systems. This would involve examining whether neural representations for specific actions maintain consistency across observed and executed actions, and whether these patterns generalize across different agents performing the same actions.

Such neuroscientific investigation would help determine whether our computational approach has captured any meaningful aspects of biological agent perception. Testing whether neural data show patterns with any relationship to our variational inference approach could provide valuable insights into the degree of overlap between computational models and biological mechanisms of social perception.

Chapter 6

Conclusion

This thesis began with a fundamental question: *What constitutes an agent?* Inspired by foundational studies in cognitive science—from Heider and Simmel (1944)'s geometric shapes to Gergely and Csibra (2003)'s teleological reasoning experiments—we sought to develop a computational framework that could perceive agency from raw visual data. This goal led us to formulate agent discovery as a structured variational inference problem, resulting in the $\mathcal{L}_{\text{VAD}}$ objective and its implementation through the VAD model.

Throughout the thesis, we posed several core questions that have guided this research. Let us revisit them in light of our findings:

Can we formalize the computations that transform visual input into representations of agency? We addressed this by developing the $\mathcal{L}_{\text{VAD}}$ objective, which provides a mathematically rigorous framework for inferring latent actions from observed state transitions. This objective explicitly models the link between perceptions and actions, creating an inductive bias toward agent-centric representations. The success of this formulation across diverse environments may suggest that a variational approach to action inference captures essential aspects of agent perception.

Can we learn to distinguish agents from non-agents in complex scenes? Our VAD model has demonstrated the ability to decompose visual scenes into meaningful entity-centric representations, with clear separation between agent and non-agent slots as shown in Figure 4.4. The model's slot representations consistently encode agent-specific information like policies, goals, and behavioral patterns, enabling accurate prediction of future actions and goals across environments.

Can agent-centric representations improve multi-agent reinforcement learning? Our experiments with the $\mathcal{L}_{\text{VAD}}$ objective as an auxiliary loss in MARL settings demonstrate improved sample efficiency. These results support that structured agent-centric world models facilitate more efficient learning in multi-agent contexts, potentially by allowing agents to better predict and respond to others' behaviors.

Do computational models of agency share functional properties with human social cognition? Our VAD model exhibits some *weak* parallels with aspects of human social perception. The emergence of mirror-like neural patterns (Figures 4.7–4.11) suggests the model has developed a common neural code for actions across different agents, analogous to mirror neuron systems in primates. Additionally, the model's ability to predict rational actions in novel scenarios (Figure 4.5) aligns with teleological reasoning observed in infant

cognition studies, suggesting deeper action understanding beyond mere pattern recognition.

From a machine learning perspective, our work utilizes object-centric representation learning in multi-agent reinforcement learning, demonstrating how structured world models that explicitly account for agency can improve learning efficiency. The $\mathcal{L}_{\text{VAD}}$ objective offers a flexible loss that could be integrated with various architectures and applied across different domains where agent modeling is essential.

From a cognitive science perspective, our computational approach offers a quantitative, falsifiable hypothesis about mechanisms underlying agent perception. By framing agency detection as variational inference over latent actions, we provide a potential mathematical description of how the brain might transform visual input into structured agent representations.

More broadly, this thesis contributes to an emerging interdisciplinary effort to understand social intelligence through computational modeling. By developing systems that can perceive agency, infer goals, and predict rational actions, we take steps toward artificial intelligence that can navigate social environments with the same intuitive understanding that humans possess. This capability will be essential for AI systems that need to collaborate effectively with humans, interpret social cues, and understand human intentions from observation alone.

In conclusion, this thesis demonstrates that framing agent discovery as variational inference over latent actions provides a principled approach to learning agent-centric representations from visual observations.

Bibliography

Abell, F., Happe, F., and Frith, U. (2000). Do triangles play tricks? attribution of mental states to animated shapes in normal and abnormal development. *Cognitive Development*, 15(1):1–16.

Adolphs, R. (2010). What does the amygdala contribute to social cognition? *Annals of the New York Academy of Sciences*, 1191(1):42–61.

Adolphs, R., Tranel, D., and Damasio, A. R. (1998). The human amygdala in social judgment. *Nature*, 393(6684):470–474.

Alain, G. and Bengio, Y. (2018). Understanding intermediate layers using linear classifier probes.

Allison, T., Puce, A., and McCarthy, G. (2000). Social perception from visual cues: role of the sts region. *Trends in Cognitive Sciences*, 4(7):267–278.

Alvarez, G. A. and Cavanagh, P. (2005). Independent resources for attentional tracking in the left and right visual hemifields. *Psychological Science*, 16(8):637–643.

Amodio, D. M. and Frith, C. D. (2006). Meeting of minds: the medial frontal cortex and social cognition. *Nature Reviews Neuroscience*, 7(4):268–277.

Andrychowicz, M., Raichuk, A., Sta{nczyk, P., Orsini, M., Girgin, S., Marinier, R., Hussenot, L., Geist, M., Pietquin, O., Michalski, M., et al. (2020). What matters for on-policy deep actor-critic methods? a large-scale study. *International Conference on Learning Representations*.

Baker, C. L., Saxe, R., and Tenenbaum, J. B. (2009). Action understanding as inverse planning. *Cognition*, 113(3):329â 349.

Baron-Cohen, S., Leslie, A. M., and Frith, U. (1985). Does the autistic child have a "theory of mind"? *Cognition*, 21(1):37–46.

Barrett, H. C., Todd, P. M., Miller, G. F., and Blythe, P. W. (2005). Accurate judgments of intention from motion cues alone: A cross-cultural study. *Evolution and Human Behavior*, 26(4):313â 331.

Bellman, R. (1957). A markovian decision process. *Journal of Mathematics and Mechanics*, 6(5):679–684.

Bertenthal, B. I., Proffitt, D. R., and Kramer, S. J. (1987). Perception of biomechanical motions by infants: implementation of various processing constraints. *Journal of Experimental Psychology: Human Perception and Performance*, 13(4):577–585.

Bertsekas, D. P. and Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific.

Blakemore, S.-J., Boyer, P., Pachot-Clouard, M., Meltzoff, A., Segebarth, C., and Decety, J. (2003). The detection of contingency and animacy from simple animations in the human brain. *Cerebral Cortex*, 13(8):837–844.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs.

Brenden M. Lake, Tomer D. Ullman, J. B. T. S. J. G. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40.

Brooks, J. A. and Freeman, J. B. (2017). Neuroimaging of person perception: A social-visual interface. *Neuroscience Letters*, 693:127–132.

Brothers, L. A. (2002). The social brain: A project for integrating primate behavior and neurophysiology in a new domain. In Cacioppo, J. T., Berntson, G. G., Adolphs, R., Carter, C. S., Davidson, R. J., McClintock, M., McEwen, B. S., Meaney, M., Schacter, D. L., Sternberg, E. M., Suomi, S., and Taylor, S. E., editors, *Foundations in Social Neuroscience*, pages 367–385. MIT Press.

Burgess, C. P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M. M., and Lerchner, A. (2019). Monet: Unsupervised scene decomposition and representation. *CoRR*, abs/1901.11390.

Cao, R. and Yamins, D. (2021a). Explanatory models in neuroscience: Part 1 – taking mechanistic abstraction seriously.

Cao, R. and Yamins, D. (2021b). Explanatory models in neuroscience: Part 2 – constraint-based intelligibility.

Carroll, M., Shah, R., Ho, M. K., Griffiths, T. L., Seshia, S. A., Abbeel, P., and Dragan, A. (2020). On the utility of learning about humans for human-ai coordination.

Chevalier-Boisvert, M., Willems, L., and Pal, S. (2018). Minimalistic gridworld environment for openai gym.

Choi, H. and Scholl, B. J. (2004). Effects of grouping and attention on the perception of causality. *Perception Psychophysics*, 66(6):926–942.

Corbetta, M., Patel, G., and Shulman, G. L. (2008). The reorienting system of the human brain: from environment to theory of mind. *Neuron*, 58(3):306–324.

Csibra, G. e. a. (1999). Goal attribution without agency cues: the perception of 'pure reason' in infancy. *Cognition*, 72:237–267.

Deisenroth, M. P. and Rasmussen, C. E. (2011). Pilco: A model-based and data-efficient approach to policy search. *Proceedings of the 28th International Conference on Machine Learning*, pages 465–472.

Dennett, D. C. (1988). Precis of the intentional stance. *Behavioral and Brain Sciences*, 11(3):495–505.

Dittrich, W. H. and Lea, S. E. G. (1994). Visual perception of intentional motion. *Perception*, 23(3):253â 268.

Doersch, C. (2016). Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*.

Foerster, J. N., Chen, R. Y., Al-Shedivat, M., Whiteson, S., Abbeel, P., and Mordatch, I. (2017). Learning with opponent-learning awareness. *arXiv preprint arXiv:1709.04326*.

Frankenhuis, W. E. and Barrett, H. C. (2013). Design for learning: The case of chasing. In Rutherford, M. D. and Kuhlmeier, V. A., editors, *Social Perception: Detection and Interpretation of Animacy, Agency, and Intention*, pages 171–195. MIT Press.

Friston, K., Mattout, J., and Kilner, J. (2011). Action understanding and active inference. *Biological Cybernetics*, 104(1-2):137–160.

Frith, C. D. (2007). The social brain? *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1480):671â 678.

Fulvia Castelli, Francesca Happé, U. F. C. F. (2000). Movement and mind: A functional imaging study of perception and interpretation of complex intentional movement patterns. *NeuroImage*, 12(3):314–325.

Gao, T., Baker, C., Tang, N., Xu, H., and Tenenbaum, J. (2019). The cognitive architecture of perceived animacy: Intention, attention, and memory. *Cognitive Science*, 43(8):e12775.

Gao, T., Newman, G. E., and Scholl, B. J. (2009). The psychophysics of chasing: A case study in the perception of animacy. *Cognitive Psychology*, 59(2):154–179.

Gao, T. and Scholl, B. J. (2010). Chasing vs. stalking: Interrupting the perception of animacy. *Journal of Vision*, 10(7):239–239.

Gao, T., Scholl, B. J., and McCarthy, G. (2012). Dissociating the detection of intentionality from animacy in the right posterior superior temporal sulcus. *Journal of Neuroscience*, 32:14276–14280.

Gazzola, V., Rizzolatti, G., Wicker, B., and Keysers, C. (2007). The anthropomorphic brain: The mirror neuron system responds to human and robotic actions. *NeuroImage*, 35(4):1674–1684.

Gergely, G. and Csibra, G. (2003). Teleological reasoning in infancy: the naïve theory of rational action. *Trends in Cognitive Sciences*, 7(7):287–292.

Gergely, G. e. a. (1995). Taking the intentional stance at 12 months of age. *Cognition*, 56:165–193.

Greff, K., Kaufman, R. L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., and Lerchner, A. (2020). Multi-object representation learning with iterative variational inference.

Grossman, E., Donnelly, M., Price, R., Pickens, D., Morgan, V., Neighbor, G., and Blake, R. (2000). Brain areas involved in perception of biological motion. *Journal of Cognitive Neuroscience*, 12(5):711–720.

Ha, D. and Schmidhuber, J. (2018). Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. (2019). Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. (2020). Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*.

Heberlein, A. S. and Adolphs, R. (2004). Impaired spontaneous anthropomorphizing despite intact perception and social knowledge. *Proceedings of the National Academy of Sciences*, 101(19):7487–7491.

Heek, J., Levskaya, A., Oliver, A., Ritter, M., Rondepierre, B., Steiner, A., and van Zee, M. (2024). Flax: A neural network library and ecosystem for JAX.

Heider, F. and Simmel, M. (1944). An experimental study of apparent behavior. *The American Journal of Psychology*, 57(2):243.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations*.

Iqbal, S. and Sha, F. (2019). Actor-attention-critic for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 2961–2970.

Janner, M., Fu, J., Zhang, M., and Levine, S. (2019). When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, pages 12519–12530.

Johansson, G. (1973). Visual perception of biological motion and a model of its analysis. *Perception Psychophysics*, 14:202–211.

Johnson, J. et al. (2017). Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Johnson, S., Slaughter, V., and Carey, S. (1998). Whose gaze will infants follow? the elicitation of gaze-following in 12-month-olds. *Developmental Science*, 1(2):233–238.

Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285.

Kahneman, D., Treisman, A., and Gibbs, B. J. (1992). The reviewing of object files: Object-specific integration of information. *Cognitive Psychology*, 24(2):175–219.

Kakade, S. M. (2002). A natural policy gradient. In *Advances in Neural Information Processing Systems*, volume 14, pages 1531–1538.

Kilner, J. M., Friston, K. J., and Frith, C. D. (2007a). The mirror-neuron system: a bayesian perspective. *Neuroreport*, 18(6):619–623.

Kilner, J. M., Friston, K. J., and Frith, C. D. (2007b). Predictive coding: an account of the mirror neuron system. *Cognitive Processing*, 8:159–166.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kipf, T., Elsayed, G. F., Mahendran, A., Stone, A., Sabour, S., Heigold, G., Jonschkowski, R., Dosovitskiy, A., and Greff, K. (2021). Conditional object-centric learning from video. *arXiv preprint arXiv:2111.12594*.

Klin, A. (2000). Attributing social meaning to ambiguous visual stimuli in higher-functioning autism and asperger syndrome: The social attribution task. *The Journal of Child Psychology and Psychiatry and Allied Disciplines*, 41(7):831–846.

Li, H., Chong, Y., Stepputtis, S., Campbell, J., Hughes, D., Lewis, C., and Sycara, K. (2023). Theory of mind for multi-agent collaboration via large language models. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2016). Continuous control with deep reinforcement learning. *International Conference on Learning Representations*.

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., et al. (2020). Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, volume 33, pages 11525–11538.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, pages 6379–6390.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. (2020). Multi-agent actor-critic for mixed cooperative-competitive environments.

Meyerhoff, H. S., Schwan, S., and Huff, M. (2014). Interobject spacing explains the attentional bias toward interacting objects. *Psychonomic Bulletin & Review*, 21(2):412–417.

Michotte, A. (1964). The perception of causality. by a. michotte, london: Methuen. 1963. pp. 425. price 45s. *British Journal of Psychiatry*, 110(464):142–143.

Mitchell, J. P., Macrae, C. N., and Banaji, M. R. (2006). Dissociable medial prefrontal contributions to judgments of similar and dissimilar others. *Neuron*, 50(4):655–663.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. In *NIPS Deep Learning Workshop*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Moerland, T. M., Broekens, J., Plaat, A., and Jonker, C. M. (2023). Model-based reinforcement learning: A survey. *Foundations and Trends in Machine Learning*, 16(1):1–118.

Molenberghs, P., Cunnington, R., and Mattingley, J. B. (2012). Brain regions with mirror properties: a meta-analysis of 125 human fMRI studies. *Neuroscience & Biobehavioral Reviews*, 36(1):341–349.

Morris, M. W. and Peng, K. (1994). Culture and cause: American and chinese attributions for social and physical events. *Journal of Personality and Social Psychology*, 67:949–971.

Most, S. B., Scholl, B. J., Clifford, E. R., and Simons, D. J. (2005). What you see is what you set: sustained inattentional blindness and the capture of awareness. *Psychological Review*, 112(1):217.

Nagabandi, A., Kahn, G., Fearing, R. S., and Levine, S. (2018). Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7559–7566.

New, J. J., Schultz, R. T., Wolf, J., Niehaus, J. L., Klin, A., German, T. C., and Scholl, B. J. (2010). The scope of social attention deficits in autism: prioritized orienting to people and animals in static natural scenes. *Neuropsychologia*, 48(1):51–59.

Nguyen, H. M. J. (2025). A survey of theory of mind in large language models: Evaluations, representations, and safety risks.

Onishi, K. H. and Baillargeon, R. (2005). Do 15-month-old infants understand false beliefs? *Science*, 308(5719):255–258.

Parr, T., Sajid, N., Da Costa, L., Mirza, M., and Friston, K. (2021). Generative models for active vision. *Frontiers in Neurorobotics*, 15:651432. Edited by: Dimitri Ognibene, University of Milano-Bicocca, Italy. Reviewed by: Emmanuel Dauce, Centrale Marseille, France, Giuseppe Boccignone, University of Milan, Italy. Correspondence: Thomas Parr thomas.parr.12@ucl.ac.uk. Received: 09 January 2021 Accepted: 15 March 2021 Published: 13 April 2021.

Patterson, K., Nestor, P. J., and Rogers, T. T. (2007). Where do you know what you know? the representation of semantic knowledge in the human brain. *Nature Reviews Neuroscience*, 8(12):976–987.

Peters, J. and Schaal, S. (2008). Natural actor-critic. In *Neurocomputing*, volume 71, pages 1180–1190.

Pratt, J., Radulescu, P. V., Guo, R. M., and Abrams, R. A. (2010). Animate motion captures visual attention. *PsycEXTRA Dataset*.

Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons.

Rabinowitz, N., Perbet, F., Song, F., Zhang, C., Eslami, S. A., and Botvinick, M. (2018). Machine theory of mind. In *International Conference on Machine Learning*, pages 4218–4227.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. *International conference on machine learning*, pages 1278–1286.

Rimé, B., Boulanger, B., Laubin, P., Richir, M., and Stroobants, K. (1985). The perception of interpersonal emotions originated by patterns of movement. *Motivation and Emotion*, 9:241–260.

Rizzolatti, G. and Craighero, L. (2004). The mirror-neuron system. *Annual Review of Neuroscience*, 27:169–192.

Rizzolatti, G., Fadiga, L., Gallese, V., and Fogassi, L. (1996). Premotor cortex and the recognition of motor actions. *Brain Research Cogn. Brain Res.*, 3(2):131–141.

Rolfs, M., Dambacher, M., and Cavanagh, P. (2013). Visual adaptation of the perception of causality. *Current Biology*, 23(3):250â 254.

Rutherford, M. D., Pennington, B. F., and Rogers, S. J. (2006). The perception of animacy in young children with autism. *Journal of Autism and Developmental Disorders*, 36:983–992.

Saxe, R. and Kanwisher, N. (2003). People thinking about thinking people. *Neuroimage*, 19:1835–1842.

Saxe, R., Xiao, D. K., Kovacs, G., Perrett, D. I., and Kanwisher, N. (2004). A region of right posterior superior temporal sulcus responds to observed intentional actions. *Neuropsychologia*, 42:1435–1446.

Scholl, B. J. (2009). What have we learned about attention from multiple object tracking (and vice versa). In *Computation, cognition, and Pylyshyn*, pages 49–78.

Scholl, B. J. and Gao, T. (2013). Perceiving animacy and intentionality. In Rutherford, M. D. and Kuhlmeier, V. A., editors, *Social Perception: Detection and Interpretation of Animacy, Agency, and Intention*, pages 197–230. The MIT Press.

Scholl, B. J. and Tremoulet, P. D. (2000). Perceptual causality and animacy. *Trends in Cognitive Sciences*, 4(8):299–309.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al. (2020). Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. (2016). High-dimensional continuous control using generalized advantage estimation. *International Conference on Learning Representations*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. In *arXiv preprint arXiv:1707.06347*.

Schultz, J., Friston, K. J., O'Doherty, J., Wolpert, D. M., and Frith, C. D. (2005). Activation in posterior superior temporal sulcus parallels parameter inducing the percept of animacy. *Neuron*, 45(4):625–635.

Schurz, M., Radua, J., Aichhorn, M., Richlan, F., and Perner, J. (2014). Fractionating theory of mind: a meta-analysis of functional brain imaging studies. *Neuroscience & Biobehavioral Reviews*, 42:9–34.

Shu, T., Bhandwaldar, A., Gan, C., Smith, K., Liu, S., Gutfreund, D., Spelke, E., Tenenbaum, J., and Ullman, T. (2021). Agent: A benchmark for core psychological reasoning. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9614–9625. PMLR.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). Deterministic policy gradient algorithms. *Proceedings of the 31st International Conference on Machine Learning*, pages 387–395.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. In *Nature*, volume 550, pages 354–359.

Simion, F., Regolin, L., and Bulf, H. (2008). A predisposition for biological motion in the newborn baby. *Proceedings of the National Academy of Sciences*, 105(2):809–813.

Southgate, V., Senju, A., and Csibra, G. (2007). Action anticipation through attribution of false belief by 2-year-olds. *Psychological Science*, 18(7):587–592.

Spelke, E. S. (1990). Principles of object perception. *Cognitive Science*, 14(1):29–56.

Spunt, R. P. and Adolphs, R. (2015). Folk explanations of behavior: A specialized use of a domain-general mechanism. *Psychological Science*, 26(6):724–736.

Sutton, R. S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. *Proceedings of the Seventh International Conference on Machine Learning*, pages 216–224.

Sutton, R. S. (1991). Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin*, 2(4):160–163.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, 12:1057–1063.

Tremoulet, P. D. and Feldman, J. (2000). Perception of animacy from the motion of a single object. *Perception*, 29(8):943–951.

Ullman, T., Baker, C., Macindoe, O., Evans, O., Goodman, N., and Tenenbaum, J. (2009). Help or hinder: Bayesian models of social goal inference. *Advances in Neural Information Processing Systems*, 22.

Van Buren, B., Uddenberg, S., and Scholl, B. J. (2016). The automaticity of perceiving animacy: Goal-directed motion in simple shapes influences visuomotor behavior even when task-irrelevant. *Psychonomic Bulletin & Review*, 23(3):797–802.

Van Overwalle, F. (2009). Social cognition and the brain: A meta-analysis. *Human Brain Mapping*, 30(3):829–858.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, ., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Wang, T., Bao, X., Clavera, I., Hoang, J., Wen, Y., Langlois, E., Zhang, S., Zhang, G., Abbeel, P., and Ba, J. (2019). Benchmarking model-based reinforcement learning. In *arXiv preprint arXiv:1907.02057*.

Watkins, C. J. C. H. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3):279–292.

Watters, N., Matthey, L., Burgess, C. P., and Lerchner, A. (2019). Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes.

Wellman, H. M., Cross, D., and Watson, J. (2001). Meta-analysis of theory-of-mind development: the truth about false belief. *Child Development*, 72(3):655–684.

White, N. C., Reid, C., and Welsh, T. N. (2014). Responses of the human motor system to observing actions across species: A transcranial magnetic stimulation study. *Brain and Cognition*, 92:11–18.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer.

Wimmer, H. and Perner, J. (1983). Beliefs about beliefs: representation and constraining function of wrong beliefs in young children's understanding of deception. *Cognition*, 13(1):103–128.

Woodward, A. (1998). Infants selectively encode the goal object of an actorâ s reach. *Cognition*, 69(1):1â 34.

Wu, Y.-F., Greff, K., Elsayed, G. F., Mozer, M. C., Kipf, T., and van Steenkiste, S. (2023). Inverted-attention transformers can learn object representations: Insights from slot attention. In *UniReps: the First Workshop on Unifying Representations in Neural Models*.

Yi, B., Li, Y., and Lee, T. C. M. (2024). Convolutional neural networks: An introduction. In Balakrishnan, N., Colton, T., Everitt, B., Piegorsch, W., Ruggeri, F., and Teugels, J. L., editors, *Wiley StatsRef: Statistics Reference Online*. Wiley.

Yu, C., Velu, A., Vinitsky, E., Wang, Y., Bayen, A., and Wu, Y. (2022). Surprising effectiveness of ppo for multi-agent learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 28867–28881.