Probabilistic Recurrent Intention Switching Model

A Thesis
submitted to the Department of Computer Science
of the Faculty of Engineering, University of Freiburg
for the acquisition of the degree of
Master of Science
in
Computer Science

Wenyuan Sheng
July 2026

First examiner: Prof. Joschka Boedecker
Second examiner: JProf. Maria Kalweit
Supervisor: Hao Zhu

Probabilistic Recurrent Intention Switching Model

Wenyuan Sheng

Department of Computer Science
University of Freiburg

universität freiburg

University of Freiburg
Freiburg im Breisgau, Baden-Württemberg, Germany

Contents

Declaration

I hereby declare, that I am the sole author and composer of my thesis and that no other sources or learning aids, other than those listed, have been used. Furthermore, I declare that I have acknowledged the work of others by providing detailed references of said work. I hereby also declare, that my Thesis has not been prepared for another examination or assignment, either wholly or excerpts thereof.

Place, Date

Signiture

Abstract

Inverse reinforcement learning (IRL) recovers reward functions from observed behavior, yet traditional methods assume a single stationary reward that cannot capture goal switching within an episode. Recent multi-intention IRL methods address this by segmenting trajectories, but model intention transitions as either a memoryless Markov chain or via manual state augmentation with a fixed history window. We propose the Probabilistic Recurrent Intention Switching Model (PRISM), which replaces both mechanisms with a lightweight recurrent network that maps observation history to a per-step intention distribution. We prove that the resulting EM objective decomposes exactly into independent per-intention reward subproblems, each solvable in closed form, yielding an $\mathcal{O}(nK)$ E-step with no variational approximation. We evaluate PRISM on a non-Markovian gridworld, a mouse labyrinth, and BridgeData V2 robotic manipulation, the first large-scale robotic application of multi-intention IRL. Across all settings PRISM achieves the highest held-out log-likelihood while recovering nameable, temporally coherent intentions from unlabeled demonstrations, suggesting that discrete goal switching is present in both biological and artificial agents.

Chapter 1

Introduction

Animals and robots frequently pursue multiple goals within a single episode. A mouse navigating a labyrinth may alternate between seeking water and returning home; a teleoperated robot arm may switch between reaching, grasping, and placing within one demonstration. Understanding these behaviors requires recovering not only what rewards drive each goal, but also when the agent switches between them and what aspects of the history trigger these switches.

Inverse reinforcement learning (IRL) addresses the first question by inferring a reward function from observed trajectories. Standard IRL assumes a single stationary reward, conflating multiple objectives into one function that cannot distinguish between them. Three lines of work have attempted to lift this stationarity assumption. Dynamic IRL (DIRL) [AJP22] models the reward as a smoothly time-varying combination of spatial goal maps, but its continuity assumption conflicts with evidence that animals switch between *discrete* strategies [ARS+22], and its inference requires T separate Bellman solves, one per timestep. Hierarchical Inverse Q-Learning (HIQL) [ZDLCK+24] replaces smooth variation with a first-order Markov chain over intentions, but the memoryless transition model cannot capture switching driven by cumulative experience such as fatigue or satiation, and its E-step requires the Baum–Welch forward–backward algorithm at $\mathcal{O}(nK^2)$ cost per trajectory. SWIRL [KWW+25] adds state-dependent transition kernels and history-augmented rewards, but represents history via state augmentation: with history length L on $|\mathcal{S}|$ states the augmented state space grows as $|\mathcal{S}|^L$ (e.g., L=2 on 127 states yields 127^2=16,129 augmented states), limiting L to small values in practice. None of these approaches scale naturally beyond small tabular environments.

We propose the *Probabilistic Recurrent Intention Switching Model* (PRISM), a framework that absorbs all temporal complexity into a single lightweight recurrent neural network. At each timestep the network reads an observation from the environment, updates a hidden state that summarizes the trajectory so far, and outputs a soft assignment over a finite set of intentions. This assignment is combined with per-intention Boltzmann policy likelihoods to form a per-step posterior responsibility over intentions. We prove that the resulting expectation-maximization (EM) objective decomposes exactly into independent per-intention reward subproblems,

each solvable in closed form via inverse action-value iteration (IAVI) [KHWB20], requiring no variational approximation. The posterior factorizes into independent per-step terms, yielding an $\mathcal{O}(nK)$ E-step. With approximately 50K trainable parameters in a single-layer RNN, PRISM trains in minutes on a laptop GPU and produces human-interpretable reward maps without manual specification of the temporal horizon.

We evaluate PRISM on three domains of increasing complexity, each testing a different property of the framework. Our central application is the *127-node mouse labyrinth* [RZPM21], where PRISM recovers three intentions (water-seeking, homing, exploration) aligned with known biological drives, matching the modes identified by both DIRL [AJP22] and SWIRL [KWW+25] on the same dataset while achieving higher held-out log-likelihood. A *frustration gridworld* with a hidden counter provides controlled validation that PRISM captures provably non-Markovian switching, which is essential for modeling history-dependent behavioral states such as satiation and fatigue. The *BridgeData V2 robotic manipulation dataset* [WBZ+23] tests whether the method generalizes beyond neuroscience: without any supervision, PRISM discovers four temporally coherent manipulation phases (approach, grasp, carry, idle) from human-teleoperated demonstrations. Together, these experiments suggest that discrete intention switching is present in both biological and artificial agents, and that the reward maps PRISM recovers provide a useful lens for interpreting the latent goals behind complex sequential behavior.

In summary, our contributions are: (i) the PRISM framework with a proven EM decomposition and closed-form reward recovery; (ii) an $\mathcal{O}(nK)$ E-step via posterior factorization, compared to the $\mathcal{O}(nK^2)$ forward–backward pass required by Markov-chain-based alternatives; (iii) to our knowledge, the first application of multi-intention IRL to a large-scale robotic manipulation dataset; (iv) recovery of nameable, interpretable intentions across three domains without supervision.

Chapter 2

Related Work

Inverse reinforcement learning. IRL recovers a reward function from expert demonstrations by assuming that the observed agent acts so as to maximise long-term discounted return. The problem was formalised by [NR+00], who showed it is inherently ill-posed: infinitely many reward functions are consistent with any finite set of demonstrations. Abbeel and Ng [AN04] proposed apprenticeship learning, which sidesteps the uniqueness issue by seeking a reward whose feature expectations match those of the expert rather than recovering the reward itself. Maximum Entropy IRL [ZMB+08] and its causal variant [ZBD10] take a different route, resolving ambiguity by selecting the reward that induces a maximum-entropy distribution over trajectories consistent with the expert's feature counts. [KHWB20] showed that casting IRL as a maximum-likelihood estimation problem under a Boltzmann policy yields a closed-form reward solution via inverse action-value iteration (IAVI), which we adopt as the inner-loop solver inside our EM framework; the same work also derives a model-free variant, inverse Q-learning, for the case where the transition model is unknown. A line of adversarial methods initiated by GAIL [HE16] frames imitation as a two-player game between a policy and a discriminator, recovering reward implicitly through the discriminator's signal; while effective, these methods require on-policy rollouts and do not produce interpretable reward maps. For a comprehensive survey of the field, see [AD21].

Multi-intention and dynamic IRL. The single-reward assumption is violated whenever an agent pursues different goals at different moments within the same episode—a situation that is the norm rather than the exception in both biological and artificial behavior. Empirical evidence that animals switch between discrete strategies rather than smoothly interpolating among them [ARS+22] motivates models that treat the reward as a discrete latent variable rather than a continuous mixture. Early work by [BMSL11] clusters entire trajectories by intention, but forbids within-episode switching, which is precisely the phenomenon we seek to model. Bayesian nonparametric approaches [DR11, SS14] avoid committing to a fixed number of intentions through Dirichlet process priors, but their inference procedures do not scale to the trajectory lengths and state-space sizes encountered in the present work. Ramponi et al. [RLM+20] introduced model-based multi-task IRL with Lipschitz reward con-

straints, demonstrating that sharing structure across intentions can improve sample efficiency when the transition dynamics are shared—a structural assumption that PRISM inherits. DIRL [AJP22] models the reward as a continuously time-varying combination of spatial goal maps, fitting a separate Bellman solve per timestep; HIQL [ZDLCK+24] replaces continuous variation with a first-order Markov chain over discrete intentions, cutting the per-timestep cost but limiting switching to memoryless dynamics; SWIRL [KWW+25] adds state-dependent transition kernels and fixed-window history augmentation, capturing some memory at the cost of an exponentially growing augmented state space. PRISM subsumes the strengths of all three predecessors: it uses discrete intentions like HIQL but learns the switching dynamics end-to-end from a recurrent network rather than imposing a Markov prior; it is history-dependent like SWIRL but avoids state augmentation entirely; and it is data-driven like DIRL but recovers discrete, nameable intentions with closed-form reward recovery.

Switching state-space models and structured latent dynamics. PRISM is structurally related to a broader family of models that combine continuous dynamics with discrete mode switches. Switching linear dynamical systems (SLDS) [GH96] assign each time step a discrete mode that governs a local linear emission and transition model, with inference via approximate EM. Recurrent SLDS (rSLDS) [LJM+17] extends this by making mode transitions depend on the continuous latent state, yielding a nonparametric generalisation that has been applied to neural population recordings and animal movement data to discover interpretable dynamical regimes. The key distinction from PRISM is in objective and output: SLDS-family methods perform unsupervised state-space identification—they recover latent dynamics but not reward functions—whereas PRISM recovers per-intention reward functions that explain *why* an agent acts as it does rather than merely *how* its hidden state evolves. Hidden Markov Models (HMMs) can be seen as the discrete-emission limit of this family; they have been applied directly to behavioural data to discover action motifs [WJI+15], but the resulting motifs carry no reward semantics. PRISM can thus be understood as lifting kinematic segmentation methods into the reward space: where an HMM or rSLDS assigns each timestep a latent mode whose meaning must be inferred post hoc from the kinematics, PRISM assigns each timestep a latent *intention* whose meaning is grounded in a recovered reward function over the state–action space.

Hierarchical imitation learning, option discovery, and demonstration segmentation. The options framework [SPS99] provides a formal language for temporally extended sub-policies, motivating a large literature on unsupervised skill and option discovery from demonstrations. CompILE [KLD+19] segments demonstrations into composable latent skills via a differentiable segmentation mechanism; play-data approaches [LKX+20] discover reusable motor primitives from unstructured teleoperation by training a latent plan encoder alongside a goal-conditioned policy. The Option-Critic architecture [BHP17] learns options and their termination conditions end-to-end from reward, while DDCO [FKSG17] recovers options from demonstrations by maximising the likelihood of observed action sequences under a

mixture-of-options model. All of these methods recover *policies* or *skills*, which are tied to the specific transition dynamics under which they were acquired. PRISM instead recovers per-intention *reward functions*, which are more compact representations, remain valid under changes in dynamics, and admit direct interpretation without a separate post-hoc labelling step. In this sense PRISM can be viewed as performing "inverse option discovery": rather than finding the options that a reward function would induce, it finds the reward functions that the observed options are optimising.

Reward learning from human feedback and preference-based RL. A parallel line of work learns reward functions not from state–action trajectories but from human preference signals. Christiano et al. [CLB+17] showed that pairwise comparisons between trajectory segments are sufficient to learn reward functions that guide policy optimisation in continuous control tasks; this preference-based paradigm now underpins reinforcement learning from human feedback (RLHF), which is central to aligning large language models with human intentions [OWJ+22]. T-REX [BGNN19] demonstrated that reward learning from ranked demonstrations can *extrapolate* beyond the quality of the provided demonstrations, recovering rewards that support policies superior to the expert. PRISM occupies a complementary niche: it requires no preference annotations or reward signal at all, recovering multiple reward functions simultaneously from unlabelled demonstration trajectories. The conceptual connection to RLHF is noteworthy—both frameworks ultimately seek to answer the question of *what* an agent is trying to achieve from observations of its behaviour, but RLHF does so with explicit human judgements while PRISM does so through unsupervised latent-variable inference. As multi-intention alignment becomes an increasingly pressing concern for deployed language models (where a single model is expected to serve as an assistant, a coder, a creative writer, and many other roles simultaneously), the PRISM framework offers a complementary analytical lens for studying how intention-switching emerges from and can be recovered from sequential interaction data.

Representation learning for reward recovery. Classical IRL operates on explicitly enumerated state and action spaces, a requirement that has historically limited its application to low-dimensional, tabular environments. Recent work on state abstraction for RL formalises the conditions under which learned representations preserve the information required for correct policy recovery, *e.g.*, bisimulation metrics [FPP04] characterise behavioural equivalence between states, and DBC [ZMC+20] shows that training an encoder to respect bisimulation distances yields representations that support better policy transfer than reconstruction-based alternatives. In the contrastive RL framework [EZLS22], goal-reaching policies are learned via a contrastive objective over state embeddings, with stabilisation techniques proposed in [ZEW+24] to make offline training reliable. These abstraction-aware approaches motivate our choice of self-supervised visual encoders for the BridgeData V2 experiment. DINOv2 [ODM+23] produces patch-level features that preserve fine-grained visual structure such as gripper pose and object layout, while SigLIP2 [TGW+25] aligns representations with natural language descriptions, col-

lapsing visually distinct frames that share a semantic label. Our ablation shows that for IRL the former property, *i.e.*, action-level distinguishability of states, dominates over semantic compactness, consistent with the theoretical prediction that bisimulation-preserving representations are preferable for reward recovery. For action discretization, learned codebook methods such as VQ-BeT [LWE+24] and SAQ [LDW+23] adapt cluster boundaries to maximise a task-relevant objective rather than minimising reconstruction error, and represent a promising direction for improving the discrete-MDP approximation that our tabular IAVI solver requires.

Imitation learning at scale. Behavioural cloning [Pom88] directly regresses actions onto states, and GAIL [HE16] improves upon it via adversarial reward shaping; both learn a policy without recovering any reward structure. DAgger [RGB11] provides a principled online correction mechanism for behavioural cloning, reducing the compounding errors that plague open-loop policies at test time. Recent generalist robot policies such as RT-1 [BBC+22] and $\pi_{0.5}$ [PFD+25] scale this paradigm to transformer architectures and diverse task suites, leveraging large pretrained vision-language backbones. Behaviour Transformers [SCAP22] address the multi-modal structure of demonstration data by discretising actions and modelling their distribution with a transformer, recovering multiple action modes implicitly through the categorical output. PRISM is complementary to this entire family: it does not aim to produce a deployable policy, but rather to decompose demonstrations into semantically coherent segments with associated reward maps, which could serve as a pre-processing step for skill-conditioned or intention-conditioned imitation learning.

Interpretable reward and decision models. Making the internal objectives of learned agents transparent is a recognised challenge in the RL literature. Reward decomposition [JKF+19] attributes a scalar reward signal to a set of named sub-objectives, enabling post-hoc explanation of decisions in terms of competing incentives. Programmatic RL [VMS+18] goes further by constraining the policy to be expressible in a human-readable domain-specific language, trading representational capacity for complete symbolic transparency. World-model approaches such as DreamerV3 [HPBL25] and IRIS [MAF22] produce compact latent representations of environment dynamics that support planning and introspection, but their latent states carry no explicit reward semantics. PRISM contributes to the interpretability agenda from the IRL side: by recovering a small set of discrete, nameable intentions, each with an associated spatial reward map and a greedy policy, it provides a structured characterisation of the latent goals that drive observed behavior without requiring any reward signal, supervision, or architectural constraint on the policy being analysed.

Positioning. PRISM is a tabular, model-based, offline multi-intention IRL method. It requires a known or estimated transition model P, operates on fully observed demonstration trajectories, and alternates between closed-form reward recovery via IAVI and gradient-based updates to the recurrent intention network, without end-to-end differentiation through the Bellman equation. Table 2.1 summarises how

PRISM compares to the most closely related methods along the axes that matter most for the experimental settings considered in this thesis.

Method	Multiple intentions	In-episode switching	History-dependent	Closed-form reward	E-step cost
MaxEnt IRL	×	×	×	×	—
IAVI	×	×	×	✓	—
DIRL	✓	✓	×	×	$\mathcal{O}(nK)$
HIQL	✓	✓	×	✓	$\mathcal{O}(nK^2)$
SWIRL	✓	✓	limited	✓	$\mathcal{O}(nK^2)$
PRISM (ours)	✓	✓	✓	✓	$\mathcal{O}(nK)$

Table 2.1 Comparison of IRL methods along key axes. "History-dependent" indicates whether switching dynamics can condition on an unbounded trajectory history. "Limited" for SWIRL refers to the fixed window of length L.

Chapter 3

Background

3.1 Markov Decision Process

An MDP can be denoted by a tuple $\langle \mathcal{S}, \mathcal{A}, P_a, r, \gamma \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state- and action-space, respectively; The function $P_a \colon \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$ is the state transition function with $P_a(s,a,s') = \mathbf{P}(s' \mid s,a)$ and $\mathbf{1}^\top P_a(s,a,\cdot) = 1$; The function $r \colon \mathcal{S} \times \mathcal{A} \to \mathbf{R}$ defines the reward function, and $\gamma \in [0,1)$ denotes the discount factor. Additionally, the function $\pi \colon \mathcal{S} \times \mathcal{A} \to [0,1]$ with $\pi(s,a) = \mathbf{P}(a \mid s)$ and $\mathbf{1}^\top \pi(s,\cdot) = 1$ is used to represent the policy according to which actions are selected in the MDP.

3.2 Value Functions and the Boltzmann Policy

Two central quantities in an MDP are the state-value function and the action-value function, both of which measure the expected long-run return an agent can achieve from a given starting configuration. Given a policy π, the *state-value function* $V^\pi \colon \mathcal{S} \to \mathbf{R}$ is defined as the expected discounted sum of rewards obtained by following π from state s:

$$V^\pi(s) = \mathbf{E}\left(\sum_{t=0}^{\infty} \gamma^t \, r(s_t, a_t) \;\middle|\; s_0 = s,\ a_t \sim \pi(s_t, \cdot) \right). \tag{3.1}$$

The *action-value function* (or Q-function) $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbf{R}$ extends this to condition on both a starting state and a starting action:

$$Q^\pi(s,a) = \mathbf{E}\left(\sum_{t=0}^{\infty} \gamma^t \, r(s_t, a_t) \;\middle|\; s_0 = s,\ a_0 = a,\ a_{t>0} \sim \pi(s_t, \cdot) \right). \tag{3.2}$$

Both functions satisfy recursive *Bellman equations*. For the optimal policy π^* the optimal Q-function Q^* obeys

$$Q^*(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P_a(s,a,s') \max_{a' \in \mathcal{A}} Q^*(s',a'), \quad \text{for all } s \in \mathcal{S},\ a \in \mathcal{A}, \tag{3.3}$$

which is identical to the Q-function constraint that appears in the IRL optimisation problem (3.7). The optimal policy is then given by $\pi^*(s) = \operatorname{argmax}_a Q^*(s, a)$.

The Boltzmann policy. A hard argmax policy is deterministic and therefore assigns probability zero to all non-optimal actions, which makes the likelihood of any demonstration that deviates from perfection exactly zero. To obtain a well-defined probabilistic model of expert behaviour, one common approach is to assume that the expert follows a *Boltzmann* (or *softmax*) policy [ZMB+08]:

$$\pi_r(s, a) = \frac{\exp\big(Q(s, a)\big)}{\sum_{a' \in \mathcal{A}} \exp\big(Q(s, a')\big)} = \exp\left(Q(s, a) - \log \sum_{a' \in \mathcal{A}} \exp Q(s, a')\right), \tag{3.4}$$

where Q is the optimal Q-function induced by r via (3.3). This policy assigns higher probability to actions with higher Q-values while remaining strictly positive everywhere, so every observed action has nonzero likelihood regardless of how suboptimal it appears. From an information-theoretic perspective, the Boltzmann policy is the unique policy that maximises the entropy of the action distribution subject to achieving a fixed expected Q-value [ZBD10]; it is therefore a natural default in maximum-entropy IRL. Equation (3.4) is precisely the constraint on π_r that appears in the IRL problem (3.7), and the same form recurs in the per-intention policy likelihood in PRISM's EM objective (4.1).

3.3 Hidden-intention Markov Decision Process (HI-MDP)

A Hidden-Intention MDP (HI-MDP) extends the standard MDP with a latent intention space $\mathcal{Z}$ and is denoted $\langle \mathcal{Z}, \mathcal{S}, \mathcal{A}, P_a, \{r_z\}_{z \in \mathcal{Z}}, \gamma \rangle$, where the state transition function P_a and policy π are defined as in the standard MDP. At each timestep t, the expert operates under a latent intention $z_t \in \mathcal{Z}$ and receives reward according to $r_{z_t} : \mathcal{S} \times \mathcal{A} \to \mathbf{R}$. The expert also perceives an observation $\varphi_t \in \mathbf{R}^m$ from the environment; the observation need not be in one-to-one correspondence with the state and may encode any information available at step t, including the state, the action, or features derived from either. For each trajectory $\xi = \{(s_1, a_1), \ldots, (s_n, a_n)\}$ there is a corresponding observation sequence $\psi = \{\varphi_1, \ldots, \varphi_n\}$, and we denote the set of all such sequences by $\mathcal{O}$. The mechanism by which observations influence intention assignments is left unspecified in the HI-MDP; PRISM instantiates this mechanism via a learned gating function f_θ (§4.1).

3.4 The Options Framework

Many sequential tasks exhibit structure at multiple temporal scales: an agent commits to a high-level sub-goal for an extended period before switching to a different one, rather than reconsidering its choice at every primitive time step. The *options framework* [SPS99] formalises this idea within the MDP setting by introducing an *option* $\omega = \langle \mathcal{I}_\omega, \pi_\omega, \beta_\omega \rangle$, where $\mathcal{I}_\omega \subseteq \mathcal{S}$ is the set of states in which ω may be initiated, π_ω is the intra-option policy executed while ω is active, and $\beta_\omega : \mathcal{S} \to [0, 1]$ is

the termination probability in each state. The resulting semi-Markov process admits the same value-function analysis as a standard MDP, with Bellman equations defined over options rather than primitive actions.

Option discovery methods typically start from a reward signal and seek options that make planning more efficient. PRISM reverses this direction: starting from unlabelled demonstrations, it recovers the *reward functions* that explain the observed sub-behaviours, a process one might call *inverse option discovery*. Each latent intention $z \in \mathcal{Z}$ in the HI-MDP (§3.3) plays a role analogous to an option, but its switching dynamics are governed by a learned gating network f_θ that conditions on the full observation history rather than a Markovian termination function over states alone.

3.5 Expectation-Maximization

Many statistical learning problems involve latent variables that are not directly observed. A natural approach in such settings is the *Expectation-Maximization* (EM) algorithm [DLR77], which provides a principled and computationally convenient procedure for finding maximum-likelihood parameter estimates when part of the data is missing or hidden.

Let $\mathcal{D}$ denote the observed data, η the latent variables, and Θ the parameters to be estimated. Direct maximisation of the marginal log-likelihood $\log \mathbf{P}(\mathcal{D} \mid \Theta) = \log \sum_\eta \mathbf{P}(\mathcal{D}, \eta \mid \Theta)$ is in general intractable because the sum over all latent configurations does not factor. EM circumvents this by introducing an *auxiliary function*

$$J(\Theta^+ \mid \Theta) = \mathop{\mathbf{E}}_{\eta \mid \mathcal{D},\, \Theta} \log \mathbf{P}(\mathcal{D}, \eta \mid \Theta^+), \tag{3.5}$$

which is the expected complete-data log-likelihood under the posterior over latent variables computed at the *current* parameters Θ. Each iteration of EM consists of two steps. The *E-step* computes the posterior $\mathbf{P}(\eta \mid \mathcal{D}, \Theta)$ and evaluates $J(\Theta^+ \mid \Theta)$ as a function of the new parameters Θ^+. The *M-step* maximises $J(\Theta^+ \mid \Theta)$ with respect to Θ^+, producing an updated parameter estimate. A fundamental property of the algorithm is that every EM iteration is guaranteed not to decrease the marginal log-likelihood:

$$\log \mathbf{P}(\mathcal{D} \mid \Theta^+) \;\geq\; \log \mathbf{P}(\mathcal{D} \mid \Theta), \tag{3.6}$$

which follows from Jensen's inequality applied to the concavity of the logarithm function [DLR77]. Convergence to a local maximum of the marginal likelihood is therefore guaranteed under mild regularity conditions, even when the M-step is only carried out approximately (the *generalised EM* variant).

In PRISM, the observed data are the demonstration trajectories $\mathcal{D}$ together with their observation sequences $\mathcal{O}$, and the latent variables are the per-step intention indices $\eta = \{z_1, \ldots, z_n\}$. The E-step computes the posterior responsibility $\mathbf{P}(z_i = k \mid \xi, \psi, \Theta)$ for each step and intention, which in PRISM factorises across time steps and requires only $\mathcal{O}(nK)$ evaluations per trajectory (§4.1). The M-step decomposes into K independent reward recovery problems, each solved in closed form via IAVI, and a single gradient update to the recurrent intention network.

3.6 Inverse Reinforcement Learning

Given the set of expert demonstrations $\mathcal{D}$ in an MDP, where each trajectory $\xi \in \mathcal{D}$ is denoted with a sequence of state-action pairs: $\xi = \{(s_1, a_1), \ldots, (s_n, a_n)\}$, the IRL problem consists of determining a reward function r that best explains the observed expert behavior in the form of demonstrated trajectories. Unfortunately, IRL is generally ill-posed because infinitely many reward functions are consistent with the expert's observed behavior. To resolve this issue, various approaches have been proposed (see [AD21] for a detailed review). Particularly, [KHWB20] formulated the IRL problem as a *maximum likelihood estimation* (MLE) problem:

$$\begin{array}{ll} \text{maximize} & \mathbf{E}_{\xi \sim \mathcal{D}} \log \mathbf{P}\left(\xi \mid \pi_r\right) \\ \text{subject to} & \pi_r(s,a) = \exp\left(Q(s,a) - \log \sum \exp Q(s,\cdot)\right) \\ & Q(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s,a,s') \max_{a' \in \mathcal{A}} Q(s',a') \\ & s \in \mathcal{S},\ a \in \mathcal{A}, \end{array} \tag{3.7}$$

where r is the optimization variable and $\mathcal{D}$ is the problem data. By assuming that the expert policy satisfies a Boltzmann distribution, if the environment model P is known, problem (3.7) can be solved in closed-form via least squares [KHWB20], leading to the model-based *inverse action-value iteration* (IAVI) algorithm. On the other hand, when P is unknown, the problem can be addressed via stochastic approximation, giving rise to the model-free *inverse Q-learning* (IQL) algorithm [KHWB20]. Compared to the popular Maximum Entropy IRL algorithm from [ZMB+08] and some of its variants, the class of IQL algorithms excels at both learning performance and time-efficiency.

Chapter 4

Methods

4.1 Probabilistic Recurrent Intention Switching Model

We formulate the multi-intention IRL problem under three assumptions.

Assumption 4.1. Each expert demonstration step is generated according to a Boltzmann-optimal policy under one of the reward functions in a K-dimensional finite set $\mathcal{R} = \{r_1, \ldots, r_K\}$.

Assumption 4.2. Each trajectory ξ in the demonstration set $\mathcal{D}$ is accompanied by an observation sequence $\psi = \{\varphi_1, \ldots, \varphi_n\} \in \mathcal{O}$, as defined in §3.3. The observation φ_i may encode any information available at step i, including the state, the action, or features derived from either.

Assumption 4.3. The soft intention assignment at each demonstration step is produced by a parametric function $f_\theta \colon \mathbf{R}^m \to \Delta^K$, where $f_\theta(\varphi_i)_k$ denotes the weight assigned to intention k given observation φ_i, for $k = 1, \ldots, K$. The function f_θ may maintain internal state across steps within a trajectory, as realized by recurrent neural networks.

Under these assumptions the expert's decision process is fully specified by $\Theta = \{\theta, \mathcal{R}\}$. We refer to the resulting framework as the *Probabilistic Recurrent Intention Switching Model* (PRISM). The PRISM inference problem consists of determining (1) a set of reward functions and (2) the intention index for each demonstration step that best jointly explain the observed expert behavior. An expectation-maximization (EM) algorithm can be devised to iteratively learn Θ. For convenience, we introduce $\eta = \{z_1, \ldots, z_n\}$ as the predicted sequence of intention indices for trajectory ξ and corresponding observations ψ. Then each iteration of the EM process maximizes the auxiliary function:

$$\text{maximize} \quad J(\Theta^+ \mid \Theta) = \mathbf{E}_{(\xi,\psi)\sim(\mathcal{D},\mathcal{O}),\,\eta} \log \mathbf{P}(\xi, \eta \mid \psi, \Theta^+), \tag{4.1}$$

where Θ^+ is the optimization variable and $(\mathcal{D}, \mathcal{O}), \Theta$ are the problem data.

Theorem 4.4. *Solving problem (4.1) is equivalent to solving a sequence of independent optimization problems. First, maximize over the intention network parameters θ^+:*

$$\bar{J}_\theta(\theta^+) = \max_{\theta^+} \mathop{\mathbf{E}}_{(\xi,\psi)\sim(\mathcal{D},\mathcal{O})} \left(\sum_{k=1}^{K} \sum_{i=1}^{n} \mathbf{P}(z_i{=}k \mid \xi, \psi, \Theta) \log f_{\theta^+}(\varphi_i)_k \right) \tag{4.2}$$

and then maximize independently over each reward $r_k^+ \in \mathcal{R}^+$:

$$\begin{aligned} \bar{J}_{r_k}(r_k^+) = \max_{r_k^+} \mathop{\mathbf{E}}_{(\xi,\psi)\sim(\mathcal{D},\mathcal{O})} & \left(\sum_{i=1}^{n} \mathbf{P}(z_i{=}k \mid \xi, \psi, \Theta) \log \pi_{r_k^+}(a_i \mid s_i) \right) \\ & \text{subject to} \quad \pi_{r_k^+}(s,a) = \exp\Big(Q(s,a) - \log \textstyle\sum_{a' \in \mathcal{A}} \exp Q(s,a')\Big), \\ & Q(s,a) = r_k^+(s,a) + \gamma \textstyle\sum_{s'} P(s' \mid s,a) \max_{a' \in \mathcal{A}} Q(s',a'), \\ & s \in \mathcal{S}, \quad a \in \mathcal{A}. \end{aligned} \tag{4.3}$$

This decomposition is exact, requiring no variational approximation. Each reward subproblem (4.3) reduces to a weighted IAVI problem solvable in closed form. Detailed proof is given in §A.1.

Posterior factorization and complexity. A key structural consequence is that, conditioned on (ξ, ψ, Θ), the intention variables $\{z_1, \ldots, z_n\}$ are mutually independent. The posterior decomposes as $\mathbf{P}(\eta \mid \xi, \psi, \Theta) = \prod_{i=1}^{n} \mathbf{P}(z_i \mid \xi, \psi, \Theta)$, where each per-step responsibility is the normalized product of the gating output and the per-intention policy:

$$P(z_i{=}k \mid \xi, \psi, \Theta) = \frac{f_\theta(\varphi_i)_k \, \pi_{r_k}(a_i \mid s_i)}{\sum_{j=1}^{K} f_\theta(\varphi_i)_j \, \pi_{r_j}(a_i \mid s_i)}, \tag{4.4}$$

where f_θ assigns soft responsibility over intentions given the observed context φ_i, and π_{r_k} scores how well the observed action a_i is explained under intention k. Since the intention variables decouple across time steps, the E-step requires only $\mathcal{O}(nK)$ evaluations per trajectory. This stands in contrast to HIQL [ZDLCK+24], where a Markov transition matrix couples consecutive intentions (z_{t-1}, z_t) and the E-step requires the Baum–Welch forward–backward algorithm at $\mathcal{O}(nK^2)$ cost per trajectory. The recurrent network thus serves a dual role: it captures arbitrarily long-range history dependence that a first-order Markov chain cannot represent, while simultaneously simplifying inference.

4.2 Training Objective

The training objective for f_θ combines the negative log-likelihood from the M-step with temporal smoothness penalties. Writing $w_{i,k}$ for the per-step responsibility in

(4.4):

$$\mathcal{L} \;=\; \mathcal{L}_{\mathrm{NLL}} \;+\; \lambda_{\ell_1}\,\mathcal{L}_{\ell_1} \;+\; \lambda_{\mathrm{kl}}\,\mathcal{L}_{\mathrm{kl}}, \tag{4.5}$$

where

$$\begin{aligned}
\mathcal{L}_{\mathrm{NLL}} &= -\mathop{\mathbf{E}}_{(\xi,\psi)\sim(\mathcal{D},\mathcal{O})} \sum_{i=1}^{n}\sum_{k=1}^{K} w_{i,k}\log f_\theta(\varphi_i)_k,\\
\mathcal{L}_{\ell_1} &= \mathop{\mathbf{E}}_{(\xi,\psi)\sim(\mathcal{D},\mathcal{O})} \sum_{i=2}^{n}\sum_{k=1}^{K} w_{i,k}\big|f_\theta(\varphi_i)_k - f_\theta(\varphi_{i-1})_k\big|,\\
\mathcal{L}_{\mathrm{kl}} &= \mathop{\mathbf{E}}_{(\xi,\psi)\sim(\mathcal{D},\mathcal{O})} \sum_{i=2}^{n} \mathrm{D}_{\mathrm{KL}}\big(f_\theta(\varphi_{i-1})\,\|\,f_\theta(\varphi_i)\big).
\end{aligned}$$

The smoothness terms are computed over consecutive time steps within each trajectory. The ℓ_1-penalty directly suppresses absolute changes in intention probabilities between consecutive steps, providing strong compression of rapid switches; however, used alone it can over-smooth genuine transitions and reduce likelihood. The KL-divergence term penalizes distributional shift more gently, preserving the shape of the intention distribution but offering weaker suppression on its own. The two are complementary: ℓ_1 provides sparse switching while KL preserves smooth transitions. Both weights $\lambda_{\ell_1}, \lambda_{\mathrm{kl}} \geq 0$ are set in §5.2. We note that these penalties act as discriminative regularizers rather than a probabilistic prior over intention sequences. The RNN hidden state provides implicit temporal modeling, while the regularizers encourage the output distribution to vary smoothly.

4.3 Intention Network Architecture

The intention network maps a sequence of observations $(\varphi_1, \ldots, \varphi_n)$ to a per-step distribution over K intentions. Each observation φ_t is embedded into a d-dimensional vector; the sequence of embeddings is processed by a recurrent encoder (RNN, LSTM, or Transformer); and each encoder output is projected to K logits followed by a softmax to produce $f_\theta(\varphi_t) \in \Delta^K$. In the tabular experiments we set $\varphi_t = (s_t, a_t)$ and use learned state and action embedding matrices; in visual domains φ_t may be the output of a pretrained encoder. A single-layer IntentionRNN with d=128 and K=4 has approximately 50K trainable parameters.

Chapter 5

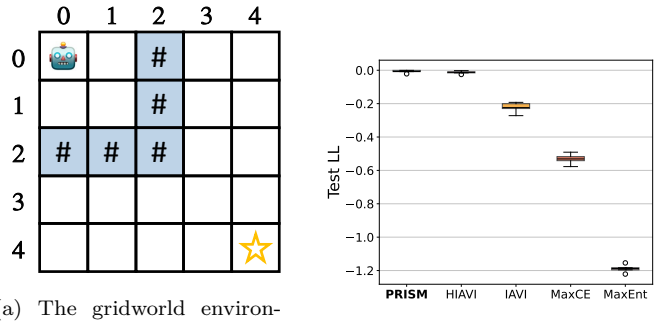
Experiments

5.1 Gridworld

5.1.1 Environment setup

The simulated gridworld environment features a 5×5 map, where the agent chooses from five actions at each step: up, down, left, right, or remain stationary. The agent starts from $(0, 0)$ and operates under stochastic dynamics such that intended actions succeed with 90% probability, with the remaining 10% causing random movement. It has two possible policies: π_{goal} prefers going to the destination $(4, 4)$, and π_{abandon} prefers returning to origin $(0, 0)$. Each episode starts with the π_{goal} policy.

The key feature of this environment is a hidden frustration mechanism that makes intention switching non-Markovian. The expert maintains a counter c, initially zero, that increments each time it enters a barrier state #. At each barrier encounter, the probability of switching to the other policy is $\min\{0.15c, 0.9\}$. Early encounters rarely trigger a switch, but repeated encounters make switching increasingly likely. The counter resets upon switching. Each episode ends when the agent reaches either the origin or the destination.



(a) The gridworld environment.

(b) Log-likelihood on test dataset.

Figure 5.1 Frustration gridworld.

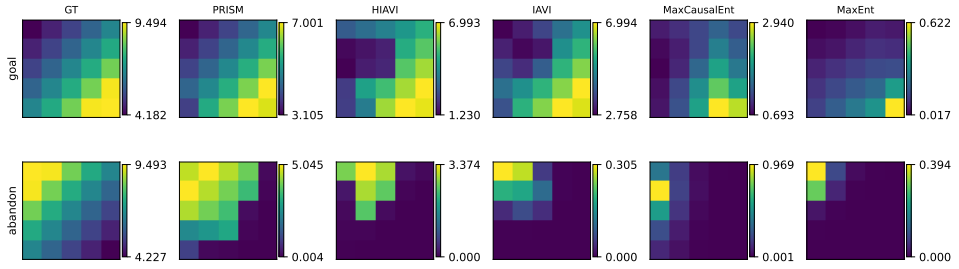


Figure 5.2 State-value heatmaps on the frustration gridworld.

5.1.2 Results

We compared PRISM against the following four baselines: HIQL [ZDLCK+24] ($K = 2$), its single-intention counterpart IAVI [KHWB20], maximum causal entropy IRL [ZBD10], and maximum entropy IRL [ZMB+08]. The dataset consisted of 1024 trajectories evaluated with 5-fold cross-validation.

Quantitative results. PRISM achieves the highest test log-likelihood with the smallest variance across folds (Figure 5.1b), and the lowest EVD under both intentions (Table A.1). HIQL follows closely in log-likelihood but shows higher EVD under the abandon intention, attributable to its Markov model being unable to capture the cumulative counter. The single-intention baselines perform substantially worse on both metrics, confirming that modeling multiple intentions is necessary in this environment.

Qualitative results. The state-value heatmaps (Figure 5.2) tell a consistent story. PRISM's recovered values closely match the ground truth under both intentions: the goal-value gradient increases toward $(4, 4)$ while attenuating in the barrier-dense region, and the abandon-value concentrates near the origin. HIQL recovers a reasonable goal heatmap but a notably weaker abandon heatmap. The single-intention baselines produce spatially smooth maps that fail to distinguish the two modes. The recovered reward maps and greedy actions (Figure 5.3) show the same pattern: PRISM's per-state arrows align with the expert's strategies under both intentions, whereas HIQL shows misaligned actions under the abandon policy and the single-intention baselines produce diffuse or conflicting directional signals.

5.2 Mouse Labyrinth Navigation

5.2.1 Dataset and benchmarks

We evaluate PRISM on the freely moving mouse navigation dataset from [RZPM21], the same dataset used in [KWW+25] and [ZDLCK+24]. Ten water-restricted mice explored a 127-node labyrinth for 7 hours in darkness; a water reward was available at a designated end node but could be collected at most once every 90 seconds.

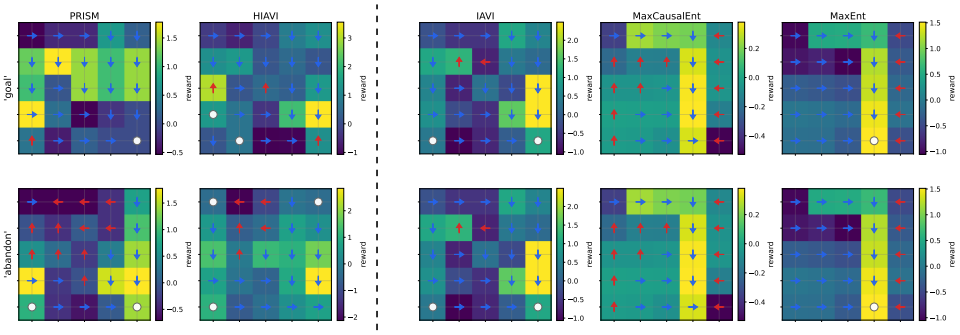


Figure 5.3 Recovered per-state reward maps and greedy actions on the frustration gridworld.

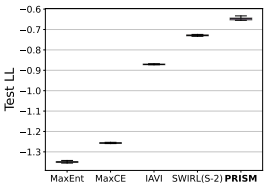
This 90-second constraint forces mice to leave the port after drinking, creating a non-Markovian reward structure in which the value of returning to the water node depends on elapsed time rather than the current state alone.

Following [KWW+25], we segment the raw node visit sequences into 238 trajectories of 500 time steps each, retaining the full behavioral detail without stereotyped-path clustering. Models are trained on 80% of trajectories and evaluated on the held-out 20% using 5-fold cross-validation, with held-out negative log-likelihood (NLL) as the primary metric. We compare PRISM ($K = 3$, IntentionRNN) against three baselines: IAVI (single intention), SWIRL (S-2) from [KWW+25] which incorporates state-dependent transition kernels and history-dependent rewards, and maximum causal entropy IRL [ZBD10]. For this experiment we activate both smoothness penalties ($\lambda_{\ell_1} = 2.22$, $\lambda_{\mathrm{kl}} = 1.48$); in the gridworld and Bridge V2 experiments both are set to zero. The full list of hyperparameters used for the labyrinth and Bridge V2 experiments is given in Table A.2 (Appendix).

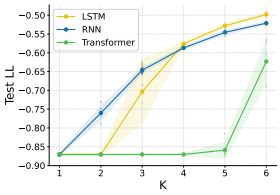
5.2.2 Behavior prediction performance

Figure 5.4a shows the distribution of test NLL across folds. PRISM achieves the highest test NLL (-0.65), outperforming SWIRL S-2 (-0.73), the single-intention baseline (IAVI), and maximum causal entropy IRL. The improvement over SWIRL is notable because both methods model history-dependent intention dynamics on the same data: PRISM obtains a better fit while learning the relevant history horizon end-to-end, without the manual state augmentation that SWIRL requires.

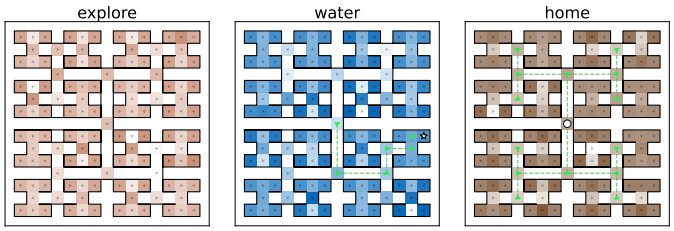
Figure 5.4b shows test log-likelihood as a function of the number of latent intentions K for three intention network architectures. All three improve monotonically with K, confirming the presence of multiple behavioral modes; we adopt the vanilla RNN as default for its simpler dynamics and stable performance. We select K=3 because the labyrinth task contains only two explicit behavioral events (water-seeking and homing), and the plateau in test log-likelihood beginning at K=4 (Figure 5.4b) indicates that three intentions already capture the dominant behavioral structure, with the third corresponding to exploration. This setting also allows direct comparison with SWIRL [KWW+25], which reports its primary results at K=3. The



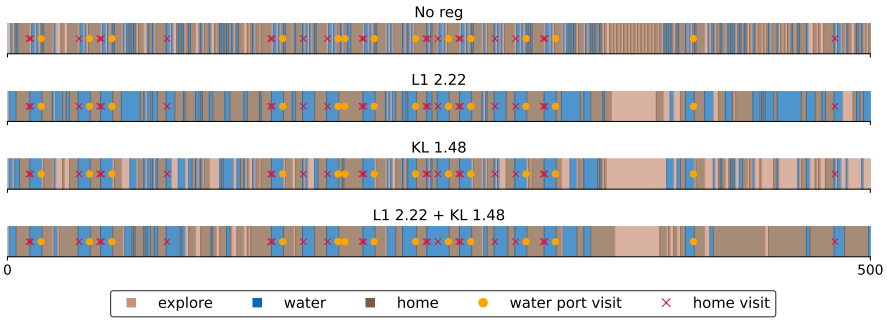
(a) Test log-likelihood across baselines.



(b) Test log-likelihood for three intention temporal models.



(c) Recovered reward maps and greedy policy flow for the three inferred intentions (star: water port; circle: entrance).



(d)

Figure 5.4 Labyrinth results (PRISM, K=3, IntentionRNN, hybrid regularization, 238 mouse trajectories). **(a)** Test log-likelihood: PRISM outperforms all baselines. **(b)** Test log-likelihood vs K for three architectures. **(c)** Recovered reward maps and greedy policy flow for the three inferred intentions (star: water port; circle: entrance). **(d)** Temporal intention segmentation under four regularization configurations. Orange dots: water-port visits; red crosses: home visits.

RNN's simpler hidden dynamics also enable the interpretability analysis in §C; all main conclusions hold under LSTM as well. Exact numerical values are provided in Table B.1 (Appendix).

5.2.3 Regularization and intention segmentation

All results in Figure 5.4a use the hybrid regularization described above. We now examine how different regularization settings affect the recovered reward maps and intention segments.

Figure 5.4c shows the greedy policy and per-state action confidence for each intention, where color depth indicates how decisively the greedy action dominates over alternatives. Under *water*, the greedy policy traces a clear path from the entrance toward the water port, with high confidence along the main corridor. Under *home*, the major branching nodes consistently point toward the entrance, forming convergent flow inward. Under *explore*, actions are distributed across peripheral nodes with no dominant target and lower overall confidence, reflecting the diffuse nature of exploratory behavior. These three modes are consistent with those identified by SWIRL [KWW+25].

Figure 5.4d shows per-step intention assignments for a representative held-out trajectory under four regularization configurations, spanning the regularization spectrum from unconstrained to fully penalized. Without regularization, the model produces rapid switching among all three intentions. Both ℓ_1-regularizer and KL-divergence regularizer reduce switching compared to the unregularized case: The ℓ_1-regularizer produces the sparsest segments due to its constant-magnitude gradient penalty, while KL-divergence regularizer suppresses large distributional jumps but tolerates gradual shifts, resulting in more frequent low-amplitude transitions. The hybrid settings inherit the temporal sparsity induced by the ℓ_1-norm while preserving the smooth transition behavior from the KL-divergence, and their intention boundaries align well with behaviorally meaningful events — water-port visits (orange dots) and home visits (red crosses). The full-weight hybrid ($\lambda_{\ell_1} = 2.22$, $\lambda_{\text{kl}} = 1.48$) is used as the default setting for results reported in Figure 5.4a.

Table 5.1 reports wall-clock timing. PRISM converges in ∼140 EM iterations on the labyrinth (∼5 min total) on a NVIDIA RTX 3060 mobile GPU, and ∼80 iterations on Bridge V2 (∼12 min), on a NVIDIA RTX 5060Ti GPU.

Dataset	**E-step**	**M-step (per latent)**		**Total / iter**	**Conv. iters**
	(posterior)	IAVI	Intention net		
Labyrinth	0.03 s	0.60 s	0.09 s	1.92 s	∼140
Bridge V2	4.10 s	1.53 s	2.90 s	8.54 s	∼80

Table 5.1 Wall-clock time per EM iteration.

5.3 BridgeData V2: Robotic Manipulation

In our third experiment, we push PRISM to a qualitatively new regime with high-dimensional visual observations and no prior knowledge of intentions, which constitutes, to our knowledge, the first application of multi-intention IRL to a large-scale robotic manipulation dataset. We apply PRISM to the BridgeData V2 robot arm dataset from [WBZ+23], which presents high-dimensional continuous observations (256×256 RGB) and actions (7D end-effector motion plus gripper) that must be discretized before tabular IRL can be applied.

This experiment serves two purposes. First, it tests whether PRISM scales to substantially larger state and action spaces than the previous settings. Second, it investigates whether intention structure can be recovered from robot demonstrations without explicit intention labels. In the labyrinth experiment, the recovered intentions (water-seeking, home-seeking, exploration) align with known biological drives of water-deprived mice. BridgeData V2 contains no such prior knowledge: each trajectory is a human-teleoperated manipulation sequence annotated only with a prompt description of the task. If PRISM can nonetheless decompose these trajectories into temporally coherent segments that correspond to recognizable manipulation phases, it provides evidence that intention structure is a general property of goal-directed sequential behavior, not specific to biological agents.

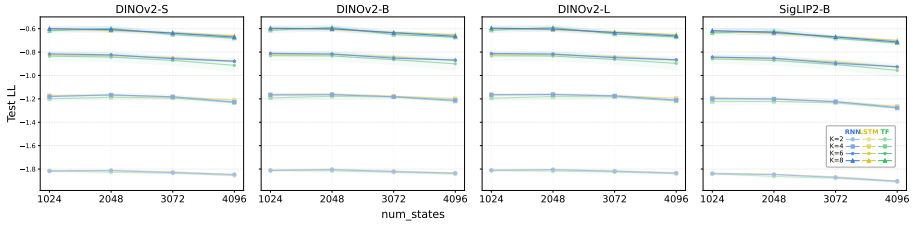


Figure 5.5 BridgeData V2: test log-likelihood by encoder, intention network, and number of latents K.

5.3.1 Continuous-to-discrete pipeline

PRISM operates on tabular MDPs with discrete states and actions. To bridge the gap between the raw continuous modalities in BridgeData V2 and this requirement, we construct a two-stage discretization pipeline. In the first stage, each 256×256 RGB frame is forwarded through a frozen pretrained visual encoder to obtain a fixed-dimensional embedding. For actions, each 7-dimensional continuous vector (6D delta end-effector pose plus gripper command) is retained in its raw form. In the second stage, k-means clustering is applied independently to the pooled state embeddings and action vectors across the full dataset, producing $|\mathcal{S}|$ discrete state tokens and $|\mathcal{A}|$ discrete action tokens. Each trajectory is then re-encoded as a sequence of integer state–action pairs, yielding a discrete MDP that is directly amenable to PRISM.

The quality of the discrete state representation depends on the visual encoder.

Figure 5.6 Per-timestep intention assignments on BridgeData V2 trajectories. Frame borders are color-coded by predicted intention: orange (CARRY), blue (IDLE), green (GRASP), magenta (APPROACH/DEPART).

We compare two families of pretrained encoders that differ in their training objective. The first family is DINOv2 [ODM+23], a self-supervised vision encoder trained to produce general-purpose visual features without any language or task supervision. We test three model scales: ViT-S/14, ViT-B/14, and ViT-L/14. The second is SigLIP2 [TGW+25], a vision-language encoder trained with a sigmoid pairwise contrastive loss to align image representations with natural language descriptions. We test its ViT-B scale. The two training objectives lead to different discretization behaviors. DINOv2 preserves fine-grained visual distinctions (gripper pose, object orientation, spatial layout), producing clusters that are visually homogeneous but spread across a larger portion of the codebook. SigLIP2 aligns its representations with semantic content, collapsing visually distinct frames that share the same language description into fewer, more heavily visited clusters. Which property matters more for IRL is not obvious: denser clusters provide more reliable transition estimates for IAVI, but overly coarse clusters may alias states that require different actions, degrading the recovered policy. We quantify this trade-off in Table 5.2 and evaluate its downstream effect on intention recovery in the results below.

5.3.2 Discretization granularity

The resolution of the k-means codebook controls a trade-off between state aliasing and transition sparsity: a larger codebook $|\mathcal{S}|$ assigns finer-grained cluster identities but reduces the number of revisits per state, degrading the empirical transition probability estimates that IAVI relies on. We ablate over $|\mathcal{S}| \in \{1024, 2048, 3072, 4096\}$ with a fixed action vocabulary $|\mathcal{A}| = 32$ for all four encoder variants (Table 5.2).

Despite its sparser transition tables, DINOv2 achieves higher held-out test log-likelihood than SigLIP2 (Figure 5.5). This result reveals that for IRL, the action-level distinguishability of states matters more than the density of transition esti-

Table 5.2 Effect of k-means state granularity on trajectory discretization statistics (mean±std across trajectories). **Coverage**: fraction of all clusters visited per trajectory (lower = finer discretization). **Avg. revisits**: mean number of time steps each visited state is revisited per trajectory (higher = denser coverage). **Singleton**: fraction of visited states seen only once per trajectory (higher = more "passing-through", fewer revisits).

Encoder	Metric	Number of states $\lvert\mathcal{S}\rvert$			
		1024	2048	3072	4096
DINOv2-S	Coverage (%)	0.64±0.40	0.37±0.23	0.26±0.17	0.21±0.13
	Avg. revisits	8.0±6.1	7.1±5.7	6.8±5.6	6.5±5.6
	Singleton (%)	21.7±19.1	25.5±19.9	27.6±20.5	28.9±20.9
DINOv2-B	Coverage (%)	0.69±0.42	0.40±0.25	0.28±0.18	0.22±0.14
	Avg. revisits	7.6±5.9	6.6±5.4	6.3±5.3	6.0±5.1
	Singleton (%)	23.4±19.3	27.4±20.2	29.6±20.7	30.9±21.0
DINOv2-L	Coverage (%)	0.69±0.42	0.40±0.25	0.29±0.18	0.23±0.14
	Avg. revisits	7.4±5.9	6.5±5.4	6.2±5.3	5.9±5.2
	Singleton (%)	23.4±19.1	27.1±20.0	29.2±20.4	30.8±20.7
SigLIP2-B	Coverage (%)	0.38±0.25	0.22±0.15	0.16±0.11	0.13±0.09
	Avg. revisits	13.3±9.6	11.8±8.9	11.0±8.4	10.5±8.1
	Singleton (%)	14.7±18.5	17.5±19.2	19.0±19.6	20.0±19.9

mates. SigLIP2's semantic alignment collapses frames that look different but share a language description into the same cluster. When those frames in fact require different actions (e.g., approaching an object from the left versus the right), the resulting state aliases degrade the per-state policy that IAVI recovers. DINOv2 retains these visual distinctions at the cost of sparser tables, but the finer-grained states better support accurate reward and policy recovery.

We adopt $|\mathcal{S}| = 2048$ with DINOv2-S as the default configuration: discretization quality is governed by encoder architecture rather than scale, with DINOv2-S, -B, and -L producing nearly identical statistics, so we adopt the smallest variant to minimize compute. The critical gap is between DINOv2 and SigLIP2: at $|\mathcal{S}|$=2048, SigLIP2-B collapses visually distinct frames into shared tokens (11.8 average revisits vs 7.1 for DINOv2-S), reducing the action-level state discrimination that IRL requires. At our default granularity ($|\mathcal{S}|$=2048, DINOv2-S), approximately 75% of visited states receive multiple observations, providing adequate support for transition estimation. This encoder advantage is reflected in test log-likelihood: DINOv2 consistently outperforms SigLIP2 across all configurations (Figure 5.5).

5.3.3 Results

Figure 5.5 reports held-out test log-likelihood as a function of $|\mathcal{S}|$, encoder, number of latent intentions K, and intention network architecture. Test log-likelihood improves monotonically with K (Figure 5.5): from -1.81 at $K{=}2$ to -0.82 at $K{=}6$ with no saturation, unlike the labyrinth where gains plateau by $K{=}4$. IntentionRNN and IntentionLSTM perform similarly; IntentionTransformer lags at small K but narrows the gap as K increases, reproducing the same ranking observed on the labyrinth and supporting IntentionRNN as a robust default. We report visualizations at $K{=}4$ because Bridge V2 trajectories are relatively short; at $K{=}5$ and beyond, segments become excessively brief and lose interpretability, with new classes capturing transient gripper adjustments rather than distinct behavioral modes.

Figure 5.6 visualizes per-timestep intention assignments ($K{=}4$, DINOv2-S). Without any supervision, PRISM recovers four classes: APPROACH/DEPART (arm moving toward or away from target), GRASP (gripper closing), CARRY (transporting object), and IDLE (minor adjustments). The boundaries are sharp and consistent across trajectories with different objects and environments, confirming temporally coherent segmentation, though we note that no ground-truth intention labels exist for formal validation.

Chapter 6

Conclusion

We presented PRISM, an EM-based framework for multi-intention IRL that parameterizes intention dynamics through a lightweight recurrent neural network. By replacing the Markov chain of HIQL and the fixed-window augmentation of SWIRL with an end-to-end learned recurrent mapping, PRISM adapts to the temporal structure of each dataset without manual design choices, while retaining closed-form reward recovery. The proven EM decomposition separates the reward and gating subproblems exactly, and the $\mathcal{O}(nK)$ E-step scales gracefully. Experiments on three progressively complex domains, from an abstract gridworld validating non-Markovian theory, through real mouse behavior confirming biological interpretability, to large-scale robotic manipulation demonstrating cross-domain generality, show that PRISM consistently achieves the highest held-out likelihood while recovering nameable intentions from unlabeled data. The recovered reward maps provide a structured characterization of the latent goals behind observed behavior, revealing not just *what* an agent did but *which objective* it was pursuing at each moment.

Chapter 7

Discussion and Outlook

(a) Tabular assumption. PRISM currently requires a discrete MDP. Our k-means pipeline bridges this gap for visual domains but introduces quantization error. The k-means codebook optimizes within-cluster Euclidean variance, which has no guaranteed relationship to the behavioral equivalence required for correct reward recovery in IRL. Learned codebooks such as VQ-BeT [LWE+24] or SAQ [LDW+23] could adapt cluster boundaries to maximize reward-recovery quality. However, discretization has fundamental limits: even cutting-edge vision-language-action models such as $\pi_{0.5}$ [PFD+25] employ action tokenizers like FAST for pre-training their language-model backbone, yet such tokenization serves the model architecture rather than enabling fine-grained dexterous control directly. The continuous structure that dexterous manipulation demands is inherently lost through any discretization, which motivates moving beyond tabular methods entirely.

(b) Model-free extension. The natural path forward is to replace IAVI with model-free inverse Q-learning [KHWB20], which operates directly in continuous state–action spaces via function approximation and does not require an explicit transition matrix. This would remove both the discretization bottleneck of (a) and the model-based dependency, enabling PRISM to scale to dexterous robotic domains where tabular representations are infeasible.

(c) Online extension. PRISM operates offline over fully observed trajectories, which suffices for post-hoc behavioral analysis but precludes real-time intention monitoring. Introducing a probabilistic transition prior $P(z_t \mid z_{t-1}, \varphi_{t-1})$ would enable sequential filtering for online applications. In preliminary experiments, such a Bayesian variant achieved comparable log-likelihood but produced less interpretable reward maps, suggesting that the prior structure requires careful design to preserve reward quality.

(d) Model selection. Test log-likelihood improves monotonically with K, but beyond the true number of behavioral modes the additional intentions fragment trajectories into unnameable segments, particularly when trajectories are short (§5.3). A single-layer RNN with 128 hidden units already saturates performance on both

datasets (Table B.2), suggesting that the switching dynamics in these domains are low-dimensional. Longer, more complex demonstrations would be needed to justify both larger K and more expressive sequence models.

Appendix A

Theoretical and technical details

A.1 Proof of Theorem 4.4

Proof. The objective function $J(\Theta^+ \mid \Theta)$ from problem (4.1) can be written as:

$$
\begin{aligned}
& J(\Theta^+ \mid \Theta) && \text{(A.1)}\\
= \; & \mathop{\mathbf{E}}_{(\xi,\psi)\sim(\mathcal{D},\mathcal{O}),\,\eta} \log \mathbf{P}(\xi,\eta \mid \psi,\Theta^+)\\
= \; & \mathop{\mathbf{E}}_{(\xi,\psi)\sim(\mathcal{D},\mathcal{O})} \left(\sum_{\eta} \mathbf{P}(\eta \mid \xi,\psi,\Theta)\ \log \mathbf{P}(\xi,\eta \mid \psi,\Theta^+) \right)\\
= \; & \mathop{\mathbf{E}}_{(\xi,\psi)\sim(\mathcal{D},\mathcal{O})} \left(\sum_{\eta} \mathbf{P}(\eta \mid \xi,\psi,\Theta)\ \sum_{i=1}^{n} \log \mathbf{P}(z_i^+{=}z_i \mid \varphi_i,\theta^+)\ \mathbf{P}\big((s_i,a_i) \mid r^+_{z_i}\big) \right)\\
= \; & \mathop{\mathbf{E}}_{(\xi,\psi)\sim(\mathcal{D},\mathcal{O})} \left(\sum_{\eta} \mathbf{P}(\eta \mid \xi,\psi,\Theta)\ \sum_{i=1}^{n}\sum_{k=1}^{K} \mathbb{I}_k(z_i)\ \log \mathbf{P}(z_i^+{=}k \mid \varphi_i,\theta^+)\ \mathbf{P}\big((s_i,a_i) \mid r^+_{k}\big) \right)\\
= \; & \mathop{\mathbf{E}}_{(\xi,\psi)\sim(\mathcal{D},\mathcal{O})} \left(\sum_{k=1}^{K}\sum_{i=1}^{n} \underbrace{\sum_{\eta} \mathbf{P}(\eta \mid \xi,\psi,\Theta)\, \mathbb{I}_k(z_i)}_{=\,\mathbf{P}(z_i=k \mid \xi,\psi,\Theta)} \log \mathbf{P}(z_i^+{=}k \mid \varphi_i,\theta^+)\ \mathbf{P}\big((s_i,a_i) \mid r^+_{k}\big) \right)\\
= \; & \underbrace{\mathop{\mathbf{E}}_{(\xi,\psi)\sim(\mathcal{D},\mathcal{O})} \left(\sum_{k=1}^{K}\sum_{i=1}^{n} \mathbf{P}(z_i{=}k \mid \xi,\psi,\Theta)\ \log \mathbf{P}(z_i^+{=}k \mid \varphi_i,\theta^+) \right)}_{\text{(I): depends only on } \theta^+} && \text{(A.2)}\\
& + \underbrace{\mathop{\mathbf{E}}_{(\xi,\psi)\sim(\mathcal{D},\mathcal{O})} \left(\sum_{k=1}^{K}\sum_{i=1}^{n} \mathbf{P}(z_i{=}k \mid \xi,\psi,\Theta)\ \log \mathbf{P}\big((s_i,a_i) \mid r^+_{k}\big) \right)}_{\text{(II): depends only on } \mathcal{R}^+}, && \text{(A.3)}
\end{aligned}
$$

where $\mathbb{I}_k$ is the indicator function with $\mathbb{I}_k(x) = 1$ for $x = k$ and 0 otherwise. Thus maximizing $J(\Theta^+ \mid \Theta)$ over Θ^+ is equivalent to separately maximizing (A.2) over θ^+ and (A.3) over $\mathcal{R}^+ = \{r_1^+, \ldots, r_K^+\}$.

By Assumption 4.3, $f_\theta(\varphi)_k = \mathbf{P}(r_k \mid \varphi)$, so the first optimization problem becomes (4.2). In (A.3), distinct k share no parameters, so the maximization decomposes into K independent subproblems. Each has the same structure as the single-intention IRL problem (3.7), with the i-th demonstration weighted by $\mathbf{P}(z_i = k \mid \xi, \psi, \Theta)$, yielding (4.3). The Boltzmann-policy constraints are introduced to make each subproblem tractable via IAVI. □

A.2 Intention Network Architecture Details

We use a single intention network that can be instantiated as an RNN, LSTM, or Transformer. In all cases, the network takes a sequence of trajectory features and outputs, at each time step, a distribution over K intention classes. Each step (s_t, a_t) is encoded by two independent learned embedding matrices $E_s \in \mathbf{R}^{|\mathcal{S}| \times d}$ and $E_a \in \mathbf{R}^{|\mathcal{A}| \times d}$, producing a d-dimensional input vector $x_t = E_s(s_t) + E_a(a_t)$. Batches of variable-length trajectories are zero-padded to a common length, and a binary mask indicates valid time steps. The recurrent variants (RNN, LSTM) pass the sequence of embeddings through a single recurrent layer with hidden dimension d', using the padding mask to pack and unpack sequences during computation, and project each hidden state to K logits via a linear layer. The Transformer variant additionally applies sinusoidal positional encoding to x_t before passing the sequence through a standard encoder with a src-key-padding mask. In all variants, the K logits are converted to an intention distribution $f_\theta(\varphi_t) \in \Delta^K$ via a softmax, which is then combined with the K agents' policy log-probabilities and normalized according to Equation (4.4) to obtain the per-step posterior over intentions. This posterior serves as the soft assignment in the E-step of the EM algorithm.

A.3 Gridworld

Table A.1 compares PRISM against the baselines on the frustration gridworld, using the expected value difference (EVD) between the expert policy and the policy recovered from the learnt reward. We report EVD as a mean absolute error over all states (EVD (MAE)) and at the initial state (EVD (s_0)), for both the goal and abandon intentions. EVD (MAE) is the mean absolute error between the state-value function under the true reward for the expert policy and that for the Boltzmann-optimal policy w.r.t. the learnt reward, averaged over all states. EVD (s_0) reports the same quantity evaluated at the initial state $(0, 0)$. For multi-intention methods ($K = 2$), each inferred intention is assigned to the best-matching ground-truth intention; for single-intention methods ($K = 1$), the same learnt policy is evaluated under both true reward functions. Values closer to zero are better.

Method	EVD (MAE)		EVD (s_0)	
	goal	abandon	goal	abandon
PRISM ($K = 2$)	$\mathbf{2.40 \pm 0.37}$	$\mathbf{5.60 \pm 0.45}$	$\mathbf{-1.74 \pm 0.59}$	$\mathbf{-6.02 \pm 0.85}$
HIQL ($K = 2$)	2.51 ± 0.13	6.12 ± 0.38	-1.95 ± 0.54	-6.80 ± 1.25
IAVI ($K = 1$)	2.06 ± 0.02	6.74 ± 0.00	-1.41 ± 0.01	-9.20 ± 0.00
MaxCausalEnt ($K = 1$)	5.32 ± 0.00	6.67 ± 0.00	-3.32 ± 0.00	-9.12 ± 0.00
MaxEnt ($K = 1$)	6.61 ± 0.00	6.76 ± 0.00	-4.16 ± 0.00	-9.10 ± 0.00

Table A.1 Expected value difference on the frustration gridworld (mean $\pm$ std, 5-fold cross-validation). Closer to zero is better; see the main text for definitions of EVD (MAE) and EVD (s_0).

A.4 Default hyperparameters

Table A.2 lists the primary configuration used in our main experiments.

Table A.2 Default hyperparameters for the labyrinth and Bridge V2 experiments unless otherwise noted.

Category	Hyperparameter	Symbol	Labyrinth	Bridge V2
MDP	Num. states	$\lvert\mathcal{S}\rvert$	127	2048
	Num. actions	$\lvert\mathcal{A}\rvert$	4	32
	Discount factor	γ	0.97	0.97
EM	Max EM iterations		180	150
	Random seed		42	42
Intention model	Num. latent intentions	K	3	4
	Architecture		IntentionRNN	IntentionRNN
Intention net	Embedding dim.	d	128	128
	RNN / LSTM hidden dim.	d'	128	128
	Num. layers		1	1
	Attn. heads (Transformer)		4	4
Optimiser	Learning rate		10^{-3}	10^{-3}
	RNN / LSTM epochs per M-step		1	1
	Transformer epochs per M-step		8	8
Regularisation	Regularisation type		KL + L1	
	L1 smoothness weight	λ_{ℓ_1}	2.22	0.0
	KL smoothness weight	λ_{kl}	1.48	0.0

Appendix B

Ablation Experiments

This appendix reports the ablation results for the intention network. Table B.1 shows how the log-likelihood changes with the number of latent intentions K, and Table B.2 compares the three temporal architectures (RNN, LSTM, and Transformer) at a fixed K.

K	Labyrinth		Bridge V2	
	Train LL	Test LL	Train LL	Test LL
1	-0.86801	-0.87071	-2.45171	-2.52187
2	-0.75432	-0.76024	-1.74644	-1.81282
3	-0.63754	-0.64624	-1.36307	-1.43000
4	-0.58141	-0.58714	-1.09903	-1.16584
5	-0.53726	-0.54573	-0.90255	-0.96912
6	-0.51220	-0.52129	-0.75780	-0.82373

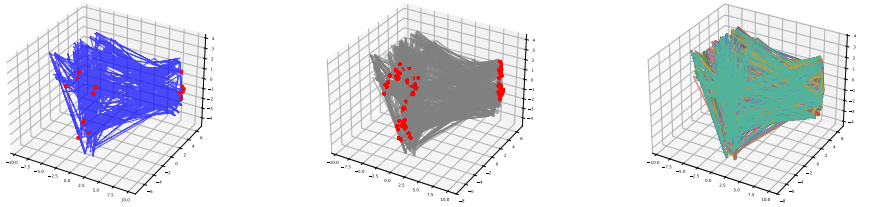
Table B.1 Log-likelihood vs number of latent intentions K (IntentionRNN, 5-fold CV, 3 random seeds).

Model	Labyrinth		Bridge V2	
	Train LL	Test LL	Train LL	Test LL
IntentionRNN	-0.63754	-0.64624	-1.09903	-1.16584
IntentionLSTM	-0.60993	-0.62705	-1.10082	-1.16839
IntentionTransformer	-0.86797	-0.87067	-1.12025	-1.18638

Table B.2 Log-likelihood by intention network architecture (d=128, single layer, 5-fold CV).

Appendix C

Interpretable RNN



(a) Single trajectory (blue) with fixed-point candidates (red).

(b) All fixed-point candidates (red) overlaid on trajectory paths (gray).

(c) Multiple trajectories projected onto the same 3D space.

Figure C.1 Hidden state dynamics of the trained IntentionRNN on the labyrinth dataset, projected onto the first three principal components (95% variance explained). Fixed-point candidates are timesteps where the hidden state step norm $\|h_t - h_{t-1}\|$ falls below the 5th percentile.

A natural question is whether the trained RNN's internal dynamics reflect the discrete intention structure that PRISM recovers. We investigate this by projecting the hidden states onto their principal components and examining the geometry of the resulting trajectories.

We collect 95,000 hidden states across 190 labyrinth training trajectories (128-D each) and project them via PCA. The top three components capture approximately 95% of the total variance (82.5%, 9.0%, 3.3%), indicating that the RNN operates on a roughly three-dimensional manifold that is consistent with the $K = 3$ intention structure, where three degrees of freedom suffice to encode which intention is active and how far it has progressed.

Projected trajectories trace smooth, distinct paths through 3D PCA space (Figures C.1a and C.1c), confirming that the hidden state encodes trajectory-specific context rather than collapsing onto a single template. We identify fixed-point candidates as timesteps where $\|h_t - h_{t-1}\|$ falls below the 5th percentile. These near-stationary states cluster in a small number of spatially separated regions (Figure C.1b), suggestive of attractor-like dynamics during periods of sustained inten-

tion. Whether these clusters align with the three inferred intentions, and what behavioral covariate the dominant first PC encodes, remain open questions for future work.

References

[AD21] Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500, 2021.

[AJP22] Zoe Ashwood, Aditi Jha, and Jonathan W Pillow. Dynamic inverse reinforcement learning for characterizing animal behavior. *Advances in neural information processing systems*, 35:29663–29676, 2022.

[AN04] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.

[ARS+22] Zoe C Ashwood, Nicholas A Roy, Iris R Stone, International Brain Laboratory, Anne E Urai, Anne K Churchland, Alexandre Pouget, and Jonathan W Pillow. Mice alternate between discrete strategies during perceptual decision-making. *Nature Neuroscience*, 25(2):201–212, 2022.

[BBC+22] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[BGNN19] Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International conference on machine learning*, pages 783–792. PMLR, 2019.

[BHP17] Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[BMSL11] Monica Babes, Vukosi Marivate, Kaushik Subramanian, and Michael L Littman. Apprenticeship learning about multiple intentions. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 897–904, 2011.

[CLB+17] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. volume 30, 2017.

[DLR77] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.

[DR11] Christos Dimitrakakis and Constantin A Rothkopf. Bayesian multitask inverse reinforcement learning. In *European workshop on reinforcement learning*, pages 273–284. Springer, 2011.

[EZLS22] Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 35:35603–35620, 2022.

[FKSG17] Roy Fox, Sanjay Krishnan, Ion Stoica, and Ken Goldberg. Multi-level discovery of deep options. 2017.

[FPP04] Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *Uai*, volume 4, pages 162–169, 2004.

[GH96] Zoubin Ghahramani and Geoffrey E Hinton. Switching state-space models. *University of Toronto Technical Report CRG-TR-96-3, Department of Computer Science*, 1996.

[HE16] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

[HPBL25] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

[JKF+19] Zoe Juozapaitis, Anurag Koul, Alan Fern, Martin Erwig, and Finale Doshi-Velez. Explainable reinforcement learning via reward decomposition. In *IJCAI/ECAI Workshop on explainable artificial intelligence*, 2019.

[KHWB20] Gabriel Kalweit, Maria Huegle, Moritz Werling, and Joschka Boedecker. Deep inverse q-learning with constraints. *Advances in neural information processing systems*, 33:14291–14302, 2020.

[KLD+19] Thomas Kipf, Yujia Li, Hanjun Dai, Vinicius Zambaldi, Alvaro Sanchez-Gonzalez, Edward Grefenstette, Pushmeet Kohli, and Peter Battaglia. Compile: Compositional imitation learning and execution. In *International Conference on Machine Learning*, pages 3418–3428. PMLR, 2019.

[KWW+25] Jingyang Ke, Feiyang Wu, Jiyi Wang, Jeffrey Markowitz, and Anqi Wu. Inverse reinforcement learning with switching rewards and history dependency for characterizing animal behaviors. *arXiv preprint arXiv:2501.12633*, 2025.

[LDW+23] Jianlan Luo, Perry Dong, Jeffrey Wu, Aviral Kumar, Xinyang Geng, and Sergey Levine. Action-quantized offline reinforcement learning for robotic skill learning. In *Conference on Robot Learning*, pages 1348–1361. PMLR, 2023.

[LJM+17] Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. Bayesian learning and inference in recurrent switching linear dynamical systems. In *Artificial intelligence and statistics*, pages 914–922. PMLR, 2017.

[LKX+20] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on robot learning*, pages 1113–1132. Pmlr, 2020.

[LWE+24] Seungjae Lee, Yibin Wang, Haritheja Etukuru, H Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior generation with latent actions. *arXiv preprint arXiv:2403.03181*, 2024.

[MAF22] Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

[NR+00] Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

[ODM+23] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[OWJ+22] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[PFD+25] Physical Intelligence, Benjamin Freed, Antoine Dedieu, Clement Gehring, Nikolaos Gkanatsios, Kristian Hartikainen, Nikhil Joshi, Karl Labat, Haotian Li, Jianlan Luo, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[Pom88] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

[RGB11] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[RLM+20] Giorgia Ramponi, Amarildo Likmeta, Alberto Maria Metelli, Andrea Tirinzoni, and Marcello Restelli. Truly batch model-free inverse reinforcement learning about multiple intentions. In *International conference on artificial intelligence and statistics*, pages 2359–2369. PMLR, 2020.

[RZPM21] Matthew Rosenberg, Tony Zhang, Pietro Perona, and Markus Meister. Mice in a labyrinth show rapid learning, sudden insight, and efficient exploration. *Elife*, 10:e66175, 2021.

[SCAP22] Nur Muhammad Shafiullah, Zichen Cui, Ariuntuya Arty Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 35:22955–22968, 2022.

[SPS99] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

[SS14] Amit Surana and Kunal Srivastava. Bayesian nonparametric inverse reinforcement learning for switched markov decision processes. In *2014 13th International Conference on Machine Learning and Applications*, pages 47–54. IEEE, 2014.

[TGW+25] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[VMS+18] Abhinav Verma, Vijayaraghavan Murali, Rishabh Singh, Pushmeet Kohli, and Swarat Chaudhuri. Programmatically interpretable reinforcement learning. In *International conference on machine learning*, pages 5045–5054. PMLR, 2018.

[WBZ+23] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023.

[WJI+15] Alexander B Wiltschko, Matthew J Johnson, Giuliano Iurilli, Ralph E Peterson, Jesse M Katon, Stan L Pashkovski, Victoria E Abraira, Ryan P Adams, and Sandeep Robert Datta. Mapping sub-second structure in mouse behavior. *Neuron*, 88(6):1121–1135, 2015.

[ZBD10] Brian D Ziebart, J Andrew Bagnell, and Anind K Dey. Modeling interaction via the principle of maximum causal entropy. 2010.

[ZDLCK+24] Hao Zhu, Brice De La Crompe, Gabriel Kalweit, Artur Schneider, Maria Kalweit, Ilka Diester, and Joschka Boedecker. Multi-intention inverse q-learning for interpretable behavior representation. *Transactions on Machine Learning Research*, 2024.

[ZEW+24] Chongyi Zheng, Benjamin Eysenbach, Homer Walke, Patrick Yin, Kuan Fang, Ruslan Salakhutdinov, and Sergey Levine. Stabilizing contrastive rl: Techniques for robotic goal reaching from offline data. In *International Conference on Learning Representations*, volume 2024, pages 28236–28264, 2024.

[ZMB+08] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[ZMC+20] Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020.