CONSISTENT ZERO-SHOT IMITATION WITH CONTRASTIVE GOAL INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

In the same way that generative models today conduct most of their training in a self-supervised fashion, how can agentic models conduct their training in a self-supervised fashion, interactively exploring, learning, and preparing to quickly adapt to new tasks? A prerequisite for embodied agents deployed in real world interactions ought to be training with interaction, yet today's most successful AI models (e.g., VLMs, LLMs) are trained without an explicit notion of action. The problem of reward-free exploration is well studied in the unsupervised reinforcement learning (URL) literature but fails to prepare agents for rapid adaptation to new demos. Today's language and vision models are trained on data provided by humans, which provides a strong inductive bias for the sorts of tasks that the model will have to solve. However, when prompted to imitate a new task, some methods perform distribution matching against the demonstration data without properly accounting for the difficulty of various tasks. The key contribution of our paper is a method for pre-training interactive agents in a self-supervised fashion, so that they can instantly mimic expert demonstrations. Our method treats goals (i.e., observations) as the atomic construct. During training, our method automatically proposes goals and practices reaching them, building off prior work in reinforcement learning exploration. During evaluation, our method solves an (amortized) inverse reinforcement learning problem to explain demonstrations as optimal goal-reaching behavior. Experiments on standard benchmarks (not designed for goal-reaching) show that our approach outperforms prior methods for zero-shot imitation.

1 INTRODUCTION

Today's AI agents, whether in language or robotics, are trained primarily by mimicking human demonstrations. But, in the same way that children conduct a large degree of learning in an unsupervised (adult-free) fashion (Gweon & Schulz, 2019; Gopnik, 2020; Stahl & Feigenson, 2015; Poli et al., 2025; Bonawitz et al., 2011), how might AI agents develop a foundation of knowledge through exploration and play, rather than through mimicry? In this paper, we study the setting where agent pre-training is done with no demonstrations, no internet-scale data, and no rewards, but rather through self-supervised interaction (Agarwal et al., 2024; Ma et al., 2022; Wu et al., 2018; Eysenbach et al., 2018; Pathak et al., 2017; Mendonca et al., 2021). The agent proposes goals, attempts to reach them, and learns from these self-collected data. After training, this agent is assessed by its ability to imitate: given a demonstration, the agent uses a (learned) inverse RL module to infer the demonstrator's goal, and then uses the (learned) goal-conditioned policies to reach that goal. Our problem setting is thus *zero-shot imitation learning (IL)*, where we would like to infer behaviors

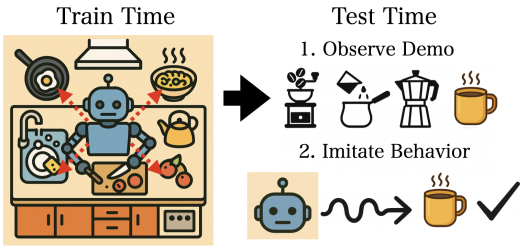

Figure 1: **Zero shot imitation learning.** Assuming access to a multi-task environment, our generalist agent must imagine and practice its own tasks to effectively imitate unknown task demonstrations at test time.

from a single demonstration without additional gradient updates (Pirotta et al., 2024; Pathak et al., 2018; Jang et al., 2021).

It is unclear whether today's recipe for building generative AI foundation models will be directly applicable to *interactive* settings. Though the premise of agents is online exploration or action, generative models are primarily built by optimizing self-supervised objectives on input data (Bommasani et al., 2021) collected offline by humans. In robotics, policies are typically constructed by either mimicking human demonstrations (Chi et al., 2023; 2024; Octo Model Team et al., 2024; Reed et al., 2022) or maximizing human-specified rewards (Silver et al., 2016; Wurman et al., 2022). These approaches do have an explicit notion of action, but agents typically practice on a limited set of tasks and are not required to infer a demonstrator's intention. The key idea in our paper is that self-supervised pretraining for agentic systems should involve interactive exploration and inverse RL that accounts for the relative difficulty of different tasks (Eysenbach et al., 2020; Ziebart et al., 2008; Ng & Russell, 2000).

Related work in inferring intentions projects a demonstration onto a hypothesis space of reward functions and then trains a general-purpose zero-shot RL policy to this space of rewards (Touati & Ollivier, 2021; Agarwal et al., 2024; Wu et al., 2018). We make the additional key observation that many tasks can be described in terms of goals, such as navigation or manipulation tasks (Brockman et al., 2016). In these settings, goals are described by the agent's state, and we can imagine natural extensions of goals to more complex behaviors expanding the state space. Tasks where the necessary actions are more complex or hierarchical, such as cooking a recipe in a kitchen, could also be described by a high-dimensional observational state, natural language, or multiple subgoals. In addition, maintaining a prior that tasks can be described via goals allows us to define reward functions probabilistically in terms of whether we will reach the goal state in the future and apply state-of-art goal conditioned reinforcement learning methods (GCRL) to an even further reduced hypothesis space of reward functional forms (Kaelbling, 1993; Schaul et al., 2015; Andrychowicz et al., 2017). Although this heuristic limits the reward functions we can infer, we show experimentally that by projecting behavior on the restricted space of goal-conditioned reward functions, we can more efficiently summarize and imitate a range of tasks from important benchmarks. Therefore, we re-imagine solving the zero-shot imitation learning task by first inferring the expert's goal and then commanding a zero-shot goal-conditioned RL policy to this inferred goal. We start by assessing our method on goal-reaching tasks, and then evaluate on reward-maximization tasks not tied to particular goal states. Our main contributions can be summarized as follows:

- We propose a contrastive inverse reinforcement learning algorithm (CIRL) for self-supervised pretraining of interactive agents that extends contrastive reinforcement learning (CRL) methods to the MaxEnt RL setting and includes automatic goal sampling during pretraining. Training involves exploration and learning via trial and error, yet requires no demonstrations, no rewards, and no preferences.
- Unlike some structurally similar methods, we prove that our method is consistent: it correctly infers the user's goal using inverse RL, accounting for the relative difficulty of reaching different goals.
- Empirically, we show that our method performs effective autonomous exploration and rapid adaptation in the standard URLB benchmark (Laskin et al., 2021), outperforming prior zero-shot imitation and zero-shot RL methods.

2 Related Work

We turn to GCRL benchmarks to test our hypotheses for goal-conditioned zero-shot IL. Several state-of-art methods on goal-reaching RL use variants of temporally contrastive objectives to learn representations and policies, and extend successor feature-based methods to high dimensional environments (Wang et al., 2023; Eysenbach et al., 2022; Myers et al., 2024). However, prior methods are limited in their assumption of access to the test-time distribution of goals, focus on the offline setting, or a hand-designed exploration policy (Pathak et al., 2018; Eysenbach et al., 2022). Given the strength of these methods in RL settings, we naturally ask whether their representations would be useful for imitation, and whether we can extend them to also learn to command their own goals. We build off the JaxGCRL benchmark to test our ideas with the Contrastive Reinforcement Learning (CRL) algorithm on a well-designed suite of tasks (Bortkiewicz et al., 2025).

Our work builds on a rich literature on URL, which use reward-free data to improve performance and generalization of RL algorithms. Some methods focus on extracting reusable representations (Agarwal et al., 2024; Wu et al., 2018; Ghosh et al., 2023; Ma et al., 2022; Blier et al., 2021; Sikchi et al., 2024). However, some of these methods that offer strategies for inferring rewards from demonstrations contain a faulty assumption that matching the expert distribution is sufficient for inferring the demonstrator's policy without accounting for the partition function over tasks (Touati & Ollivier, 2021; Sikchi et al., 2024). Other works focus on unsupervised discovery of diverse skills (Gregor et al., 2016; Machado et al., 2017; Eysenbach et al., 2018; Sharma et al., 2019; Eysenbach et al., 2021; Klissarov & Machado, 2023; Zahavy et al., 2022; Park et al., 2023a;b; Zheng et al., 2024; Wang et al., 2024). While some of these exhibit zero-shot policy inference capabilities from rewards or goals, they are not designed to perform zero-shot imitation from demonstrations and do not necessarily discover all possible skills (Agarwal et al., 2024; Zheng et al., 2024; Eysenbach et al., 2021). Methods of online unsupervised exploration for pretraining policies tackle problems under similar assumptions as our work, but do not handle zero-shot inference given trajectories (Pathak et al., 2017; 2019; Mendonca et al., 2021; Rajeswar et al., 2022). There has also been significant development of offline unsupervised pre-training methods, but these could suffer under poor exploration and do not focus on the interaction between exploration and policy learning in unfamiliar environments. Our method is highly connected to prior work on learning universal, high dimensional successor representations, with applications to both online and offline settings (Dayan, 1993; Barreto et al., 2016; Borsa et al., 2018; Ma et al., 2018; Touati & Ollivier, 2021; Touati et al., 2022; Pirotta et al., 2024). Like these works, our method contains an inductive bias influencing which tasks we focus on, namely those that are goal-reaching. We show that this restriction becomes particularly useful for inverse RL, even for arbitrary reward functions.

Approaches to zero-shot imitation learning combine approaches to inverse RL and exploration/data collection to solve the problem. We'll discuss these individual components first and then discuss key prior methods for zero-shot imitation.

Inverse RL Achieving general, adaptable agents is challenging via reward engineering and may lead to unintended behaviors (Amodei et al., 2016). Thus, we turn to learning from demonstrations (LfD), assuming we have access to limited data from an expert (Finn et al., 2016; Fu et al., 2018; Pirotta et al., 2024; Yu et al., 2019). The main approaches to LfD are behavioral cloning (BC) and inverse reinforcement learning (IRL). BC casts learning an imitation policy as a supervised learning problem. While BC can work well in practice, it suffers from poor performance under distributional shift and can overfit its expert demonstrations (Ross et al., 2011; Pomerleau, 1988; Bojarski et al., 2016). IRL attempts to infer reward functions/corresponding policies from demonstrations (Ng & Russell, 2000). Since the reward inference problem is inherently under-specified, a common modeling choice is the Maximum Entropy assumption, which assumes that expert demonstrations select actions to maximize both the sum of expected discounted rewards and the entropy of the distribution of actions over states (Ziebart et al., 2008). Extensions such as GAIL, AIRL, and GCL were developed to use deep function approximators for single-task IRL (Ho & Ermon, 2016; Fu et al., 2018; Finn et al., 2016). Current multi-task/meta IL algorithms can be categorized as gradient-based or context-based (Chen et al., 2023). Gradient-based approaches, such as (Finn et al., 2017; Yu et al., 2018) combine meta-learning with IL to recover a policy, but at inference time, require a one-shot gradient step to adapt to a new task whereas our method adapts zero-shot. Context-based approaches such as SMILE and PEMIRL learn a latent variable to represent the task contexts and train a context-conditioned policy that can be applied zero-shot to new tasks (Seyed Ghasemipour et al., 2019; Yu et al., 2019). Our approach is similar (encoding goals as a form of context) but takes this one step further by proving that the multi-task IRL problem can actually be reduced to a purely goal-inference problem when we our expert optimizes a goal-conditioned reward function. Therefore, we can use zero-shot RL algorithms to recover policies without loss of performance instead of using less stable adversarial methods.

Exploration While BC and IRL can be performed on offline datasets, we would prefer to enable zero-shot imitation through purely online methods that can be applied out-of-the-box in novel environments. This requires our IL agent to perform its own exploration, which CRL currently does not support (Eysenbach et al., 2022). For our goal-conditioned setting, automatic goal sampling enables us to autonomously generate training objectives. Goal sampling approaches broadly fall into two categories: adversarial methods and distribution-based methods. Adversarial methods such as

ASP and GoalGAN introduce a second policy for sampling goals (OpenAI et al., 2021; Florensa et al., 2018). While effective for simple domains, these methods can struggle with high-dimensional goal spaces and require careful balancing of the adversarial training process. State distribution approximation methods such as Skew-Fit, EDL, VUVC, RIG, MEGA, and DISCERN control the probability of selecting a goal via the empirical state visitation density, usually trying to cover the full state space with exploration (Pong et al., 2020; Campos et al., 2020; Kim et al., 2023; Nair et al., 2018; Pitis et al., 2020; Warde-Farley et al., 2018). Our method, GoalKDE, adopts a simple form of RIG, although more complex methods could also be benchmarked in future work.

Zero-Shot Imitation Learning BC-Zero addresses multi-task zero-shot imitation by scaling diverse, human-in-the-loop data collection and training a single task-conditioned behavior-cloned policy that can execute novel text instructions at test time (Jang et al., 2021). However, unlike our method, BC-Zero gathers task-labelled expert data via teleoperation and requires human interventions in a DAgger-style loop, whereas our method trains purely online and collects its own data using a self-supervised objective and exploration. Zero-Shot Visual Imitation uses goal-conditioned policies to imitate experts trained via a model-based forward consistency loss (Pathak et al., 2018). However, unlike our work, they hand-devised an exploration policy to generate data for model-based training, whereas our data collection is fully self-supervised for model-free training. Forward-Backward (FB) Representations and RLZero enable zero-shot imitation through matching the demonstrator's state visitation distributions (Touati & Ollivier, 2021; Pirotta et al., 2024; Sikchi et al., 2024). However, we prove for the FB representation that without accounting for the partition function, this method leads to systematic misidentification of the demonstrator's true policy.

3 PRELIMINARIES

Definition 1. *The **zero-shot imitation learning** problem assumes we are given a single expert trajectory $\tau = (s_0, a_0, ..., s_T, a_T)$ at inference time, generated by some unknown expert policy π_E with trajectory distribution $p_{\pi_E}(\tau)$. No reward function is available. We must produce a policy $\hat{\pi}_{CIRL} \in \Pi$ that successfully reproduces the behavior of π_E defined by its unknown reward function, thereby achieving low regret. $\hat{\pi}_{CIRL}$ should be inferred from π_E with no additional environment interaction, test-time data, or gradient updates.*

To solve this problem, we will model the environment as a goal-conditioned MDP, defining a reward function that depends on a goal and thereby assuming that expert policies π_E have behaviors that can be described as goal-reaching. Then, we can infer the reward function associated with π_E via MaxEnt IRL. To do this, we will infer the goal $\hat{g}$ associated with π_E, and command a goal-conditioned policy to $\hat{g}$ that is trained with CRL. In the subsequent sections, we will prove that performing MaxEnt IRL with a goal-conditioned reward is equivalent to performing goal inference. We operate in the pure online RL setting, assuming no access to offline expert data during pretraining. This includes no access to the test-time goal distribution, a departure from CRL's oracle assumptions.

3.1 CONTRASTIVE RL

We define a goal-conditioned MDP by a tuple (S, A, G, P, r, ρ), where S is the state space, A is the action space, G is the goal space (equivalent to the state space in our formulation); $p : S \times A \times S \to [0, 1]$ describes the transition probabilities between states; $r : S \times A \times G \to \mathbb{R}$ is a goal-conditioned reward function, defined as $r(s_t, a_t, g) = (1-\gamma)p(s_{t+1} = s_g \mid s_t, a_t) = r_g(s_t, a_t)$, for some discount factor γ; $\rho_0(s_0)$ specifies the initial state distribution, and $p(g)$ specifies some test-time distribution over goals. We use τ to define a finite horizon trajectory as a sequence of states and actions: $\tau = (s_0, a_0, \cdots, s_T, a_T,)$, and write the likelihood of a trajectory under policy π as $p(\tau) = \rho_0(s_0) \prod_t p(s_{t+1} \mid s_t, a_t)\, \pi(a_t \mid s_t)$. We also define the discounted future state s_f occupancy measure (density) of goal-conditioned policy $\pi : \mathcal{S} \times \mathcal{G} \to \Delta(\mathcal{A})$ as $p^\pi_\gamma(s_f|s, a, g) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t p^\pi_t(s_t \mid s, a, g)$ and the marginal distribution as $p^\beta_\gamma(s_f) = \int p^\beta(s, a) p_G(g) p^\beta_\gamma(s_f \mid s, a, g) dsdadg$, where $\beta : \mathcal{S} \to \mathcal{A}$ is the behavioral policy. Using the contrastive RL algorithm, we can estimate the discounted state occupancy using Noise Contrastive Estimation (Oord et al., 2018) and obtain the critic function $f^\star_{\phi,\psi}(s, a, g) = \|\phi(s, a) - \psi(g)\|_2 = \log \frac{p^\pi_\gamma(s_f|s,a,g)}{p^\beta_\gamma(s_f)} = \frac{1}{p^\beta_\gamma(s_f)}$.

$Q_{s_f}^{\pi(\cdot|\cdot)}(s,a)$, where $Q_{s_f}^{\pi}(s,a) \triangleq \mathbb{E}_{\pi(\tau|s_f)}\left[\sum_{t'=t}^{\infty}\gamma^{t'-t}r_{s_f}\left(s_{t'},a_{t'}\right)|\, s_t=s,a_t=a\right]$ (Eysenbach et al., 2022).

3.2 Maximum Entropy Inverse Reinforcement Learning (MaxEnt IRL)

We will use the MaxEnt IRL framework to infer reward functions and policies from expert demonstrations. This framework assumes that demonstrations come from a MaxEnt RL policy $\tilde{\pi}^* = \arg\max_\pi \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T}\left(r_g\left(s_t,a_t\right)+\alpha\mathcal{H}\left(\pi\left(\cdot\mid s_t\right)\right)\right)\right]$ where α is an optional parameter to control the trade-off between reward maximization and entropy maximization. Without loss of generality, we can assume $\alpha = 1$ for notational simplicity. The trajectory likelihood under the optimal maximum entropy policy is then $p^\star\left(\tau=\{\boldsymbol{s}_{0:T},\boldsymbol{a}_{0:T}\}\mid g\right) = \frac{1}{Z_g}\left[\rho_0\left(s_0\right)\prod_{t=0}^{T}p\left(s_{t+1}\mid s_t,a_t\right)\right]\exp\left(\sum_{t=0}^{T}r_g\left(s_t,a_t\right)\right)$, where $Z_g = \int\rho\left(s_0\right)\prod_t P\left(s_{t+1}\mid s_t,a_t\right)e^{r_g(s_t,a_t)}d\tau$. We can then define the MaxEnt IRL problem as $\min_{g'}\mathbb{E}_{p(g)}\left[D_{\text{KL}}(p_E(\tau|g)\,\|\,p^\star\left(\tau=\{\boldsymbol{s}_{0:T},\boldsymbol{a}_{0:T}\}\mid g'\right)\right]$.

3.3 Goal Inference

The MaxEnt IRL problem involves inferring reward parameters from a demonstration, and our reward functions are completely parameterized by goals g. Therefore, we will perform inference to recover the latent goal of an actor from observed data. Applying Bayes' Rule to the trajectory likelihood of a MaxEnt RL policy, the posterior distribution over goals is $p^\star(g\mid\tau) = \frac{p^\star(\tau|g)p(g)}{p(\tau)} \propto p(g)e^{\sum_t r_g(s_t,a_t)-\log Z_g}$. The partition function Z_g is important for inferring goals, since it gives us a notion of average reward collected along all possible trajectories for a given reward function $r_g(s,a)$. If an expert demonstration collects more reward than this average over trajectories, it is more likely that the demonstration is associated with this particular goal (Eysenbach et al., 2020). The partition function is difficult to estimate, so we will instead fit a variational posterior $q_\xi(g|\tau)$ to perform goal inference (Dragan et al., 2013; Zurek et al., 2021).

4 Method

Our algorithm, CIRL, consists of the following components: (1) self-supervised contrastive RL pretraining to learn maximum entropy soft Q-values and a corresponding goal conditioned policy, (2) a goal inference model to learn the variational posterior, and (3) automatic goal sampling during pretraining. Our key contribution is in using goal inference and a goal-conditioned reward to couple IRL with CRL for a successful online imitation learning algorithm. However, certain components, such as the specific goal sampling method, could be substituted.

4.1 Maximum Entropy Contrastive Reinforcement Learning

We build an extension of contrastive reinforcement learning under the Maximum Entropy assumption. While CRL just learns the sum of discounted future rewards, we also need to estimate the sum of discounted future entropy to optimize the MaxEnt RL objective. Following prior work (Haarnoja et al., 2018; Eysenbach et al., 2022), we define the the entropy regularized goal-conditioned reward function as $\tilde{r}_g(s_t,a_t) \triangleq (1-\gamma)\delta(s_t=g)-\alpha\log\pi(a\mid s,g)$, where $\delta(\cdot=g)$ is the delta measure at the goal g. Given a set of goals sampled from a goal distribution $g\sim p_{\mathcal{G}}(g)$, this new reward function allows us to rewrite the objective of the goal-conditioned policy as maximizing the entropy-regularized discounted state occupancy measure: $\max_\pi \mathcal{L}_{\text{Actor}}(\pi)$,

$$\mathcal{L}_{\text{Actor}}(\pi) = \mathbb{E}_{g\sim p_{\mathcal{G}}(g),\tau\sim\pi(\tau|g)}\left[(1-\gamma)\sum_{t=0}^{\infty}\gamma^t\left(r_g(s,a)-\alpha\log\pi(a\mid s,g)\right)\right] \tag{1}$$

$$= \mathbb{E}_{\substack{g\sim p_{\mathcal{G}}(g),s\sim\rho(s),a\sim\pi(a|s,g),\\ s_f\sim p_\gamma^\pi(s_f=g|s,a,g),a_f\sim\pi(a_f|s_f,g)}}\left[\delta(s_f=g)-\alpha\log\pi(a_f\mid s_f,g))\right] \tag{2}$$

$$\approx \mathbb{E}_{g\sim p_{\mathcal{G}}(g),s\sim p^\beta(s),a\sim\pi(a|s,g)}\left[\exp(f_{\phi,\psi}(s,a,g))-\alpha\log\pi(a\mid s,g)\right] = \tilde{Q}_g(s,a) \tag{3}$$

Thus, we augment CRL to optimize the soft Q function $\tilde{Q}_g(s,a)$ by optimizing the CRL loss $\min_{\phi,\psi} \mathcal{L}_{\text{Critic}}(\phi,\psi)$, where

$\mathcal{L}_{\text{Critic}}(\phi,\psi) = \mathbb{E}_{\mathcal{B}}\left[-\sum_{i=1}^{|\mathcal{B}|}\log\left(\frac{e^{f_{\phi,\psi}(s_i,a_i,g_i)}}{\sum_{j=1}^{K}e^{f_{\phi,\psi}(s_i,a_i,g_j)}}\right) - \sum_{i=1}^{|\mathcal{B}|}\log\left(\frac{e^{f_{\phi,\psi}(s_i,a_i,g_i)}}{\sum_{j=1}^{K}e^{f_{\phi,\psi}(s_j,a_j,g_i)}}\right)\right]$ for $\left\{f_{\phi,\psi}\left(s_i,a_i,g_j\right)_{i,j}\right\}$ over the elements of the batch $\mathcal{B}$ (Bortkiewicz et al., 2025). The critic function $f_{\phi,\psi}(s,a,g)$ estimates expected discounted future state occupancy, and the actor objective combines this with an additional term $\mathcal{L}_{\text{Entropy}}(\theta) = -\mathbb{E}_{g\sim p_{\mathcal{G}}(g),\tau\sim\pi(\tau|g)}\left[(1-\gamma)\sum_{t=0}^{\infty}\gamma^t\left(\alpha\log\pi(a\mid s,g)\right)\right]$ that estimates expected discounted future entropy. This term will be optimized with temporal difference updates. See Appendix B for more details on the algorithm.

4.2 Variational Goal Inference

Following the motivation of Section 3.3, we will learn a variational distribution $q_\xi(g|\tau)$ to match the true posterior $p^\star(g|\tau)$. We optimize the forward KL objective to achieve this (Ambrogioni et al., 2019; Yu et al., 2019): $\min_\xi D_{KL}\left(p^\star(g\mid\tau)\|q_\xi(g\mid\tau)\right) = \min_\xi \mathbb{E}_{p^\star(g,\tau)}\left[\log\frac{p^\star(g|\tau)}{q_\xi(g|\tau)}\right] = \max_\xi \mathbb{E}_{g\sim p(g);\tau\sim p^\star(\tau|g)}\left[\log q_\xi(g\mid\tau)\right] = \max_\xi \mathcal{L}_{\text{Info}}\left(\xi\right)$.

When our policy is trained to optimality, it will emit a trajectory distribution equivalent to $p^\star(\tau|g)$. Thus, we can use our online learned MaxEnt RL policy to sample trajectories both for contrastive RL pre-training and for learning the variational posterior.

Another way to model the variational posterior is with the mean field approximation: $q_\xi(g|\tau) = \prod_{t=0}^{T} q_\xi(g|s_t,a_t)$, where each local state-action independently influences the distribution over the goal. This form can be much easier to train since parameters ξ are now shared across state-action inputs. We can rewrite the expression for the true posterior as $p^\star(g\mid\tau) = \frac{p^\star(\tau|g)p(g)}{p(\tau)} \propto p(g)e^{\sum_{t=0}^{T} r_\theta(s_t,a_t,g)-\frac{1}{T}\log Z_\theta} \propto \prod_{t=0}^{T} e^{r_\theta(s_t,a_t,g)-\frac{1}{T}Z_\theta}$, and note that it precisely takes a mean field form when the input trajectory is finite. Thus, we can establish a corollary to motivate the use of the mean field approximation when optimizing $\mathcal{L}_{\text{Info}}\left(\xi\right)$ for our method, training a Gaussian MLP to perform amortized variational inference with the mean field approximation.

Corollary 1. *Without loss of generality, the class of mean field goal inference models includes the true posterior distribution.*

4.3 CIRL is Consistent

Our main theoretical result is to show that our method infers the correct distribution over expert goals. This statement is non-trivial because the most-frequented states may not be the user's intended state, so correctly performing goal inference requires reasoning about the relative difficulty of different goals. Proof can be found in Appendix A.1.

Lemma 1. *Let policy π_{demo} be given. CIRL produces policy π_{CIRL} that consistently infers rewards by converting the MaxEnt IRL problem into a goal inference problem:* $\min_\theta \mathbb{E}_{p(g)}\left[D_{KL}(p_E(\tau|g)\,\|\,p^\star(\tau|g)\right] \implies \max_\xi \mathbb{E}_{g\sim p(g);\tau\sim p^\star(\tau|g)}\left[\log q_\xi(g\mid\tau)\right]$

IRL with FB is Inconsistent FB (Touati & Ollivier, 2021) is presented as a method that can learn optimal policies for any task and proposes to imitate trajectories by inferring their reward and then using the corresponding reward-maximizing policy. In this section, we show that even if FB learns optimal policies for every reward function, it doesn't correctly identify which reward function a demonstrator is maximizing, thereby provably failing to perform zero-shot imitation. Proof can be found in Appendix A.2.

Lemma 2. *There exists an MDP with two unique reward-maximizing policies (π_1,π_2), where FB incorrectly demonstrates policy π_1 with policy π_2.*

4.4 Goal-Sampling

During training, we use states stored in the replay buffer to continually fit a Gaussian Kernel Density estimator (KDE) approximating the distribution of visited states. This buffer is pre-filled at the start

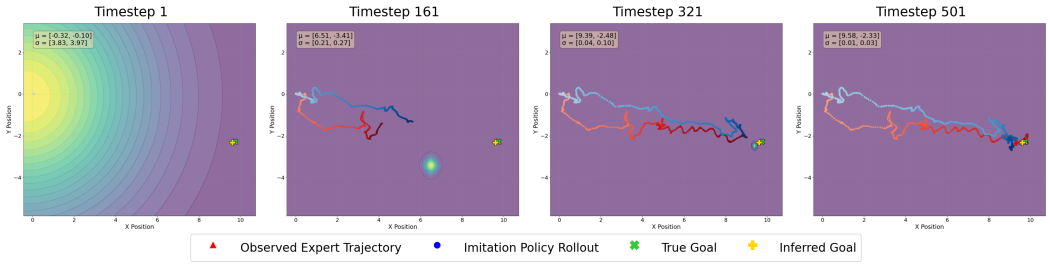

Figure 2: **Zero-shot imitation learning with CIRL via goal inference.** CIRL combines goal-conditioned contrastive RL pre-training, automatic goal sampling for exploration, and a mean field goal inference model to imitate expert demonstrations. Here we see how an Ant's imitation policy and posterior distribution over goal states evolve across timesteps toward a final maximum a posteriori (MAP) estimate.

of training with data from a randomly initialized policy, and at each iteration, we select the state from the buffer that has the lowest probability under the KDE to train the policy. We call this method of automatic exploration: GoalKDE. See Appendix B for a summary of the full CIRL algorithm.

5 Experiments

Our method contains components for self-supervised RL pretraining, automatic goal sampling, and goal inference. We ablate each in turn, and show that CIRL (CRL Pre-training + GoalKDE Exploration + Mean Field Goal Inference Model) can learn good representations for imitation across several environments. We use the JaxGCRL and Unsupervised Reinforcement Learning Benchmark (URLB) environments (Bortkiewicz et al., 2025; Laskin et al., 2021). Details are provided in Appendix C.

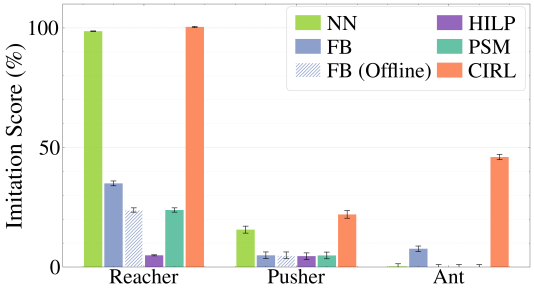

Figure 3: **Value of self-supervised RL pre-training** CIRL consistently outperforms the alternative FB representation zero-shot imitation method as well as the naive 1-NN policy baseline.

For our evaluation, we train an expert policy using CRL under oracle goal sampling. Using this policy, we sample 2000 goals from the oracle test distribution of goals and unroll the CRL expert policy toward each goal. For each expert demonstration, we perform zero-shot IL across our ablation setup, reporting imitation score (the ratio between the cumulative return of the algorithm and the average cumulative reward of the expert) (Pirotta et al., 2024). Unless otherwise noted, all methods were trained online. The only exceptions are FB (Offline) and goal-conditioned behavioral cloning (GCBC), which use the CRL expert policy data. FB (Offline) uses 2000 distinct trajectories of 1025 steps each sampled from the CRL expert policy. GCBC trains a goal-conditioned policy with the same number of update steps as the CRL expert. We also further evaluate using non-goal-conditioned policy demonstrations trained with URLB rewards on the Ant Forward, Ant Jump, and Ant Flip tasks and demonstrate the capability of CIRL to imitate these policies with low regret.

5.1 CIRL w/ Self-Supervised Pretraining Outperforms Baselines

We first compare CIRL against several baselines for imitation learning, including those with and without access to expert data during training. For each environment, we compared the reward earned by an expert policy (CRL) and the imitation learning method, reporting the fraction of expert reward achieved as the "imitation score." The baselines, both trained with no access to expert information, include the Nearest Neighbor baseline, which in a given state considers the 1-NN state in the expert demonstration and applies its corresponding action. We also include the URL baselines of FB representation (Touati & Ollivier, 2021), PSM (Agarwal et al., 2024), and HILP (Wu et al., 2018). The inferred latents for these methods were computed from expert demonstration states (Pirotta et al., 2024). As seen in Figure 3, CIRL consistently outperforms all baselines, regardless of en-

vironment difficulty, making it the most promising technique for learning to imitate in unfamiliar goal-conditioned environments. Even if we train the FB representation with data from the expert policy, this baseline is unable to achieve comparable imitation scores to our method.

5.2 CIRL Pre-Training Outperforms the FB Representation

CIRL and FB representation's algorithms have two main structural differences: the way it learns the successor representation and the way it infers intentions. To better understand why CIRL outperforms the FB representation, we hold the method of inferring intentions constant and only use information from the last state of the expert demonstration. Note that for tasks where the goal state is transient (e.g. tossing a ball to reach a particular height), the last state in a trajectory may not contain enough information about the true goal, but for Ant, Reacher, and Pusher, the agents are able to reach and stay at all possible goals. As seen in Figure 4, FB only achieves a small fraction of the imitation score of CIRL under these conditions. This result provides evidence that learning reward functions is indeed more expressive than summarizing behavior via goals, as it is easier for CIRL to learn a successor representation for reaching goals than it is for FB to learn more general reward functions. Additionally, in Figure 9 in Appendix D, even if we instead use the CIRL inferred goal to compute a latent for the FB representation, we are still not able to match CIRL's performance, though performance does improve compared to using the averaged backward representation of the entire demonstration or the backward representation of the last state.

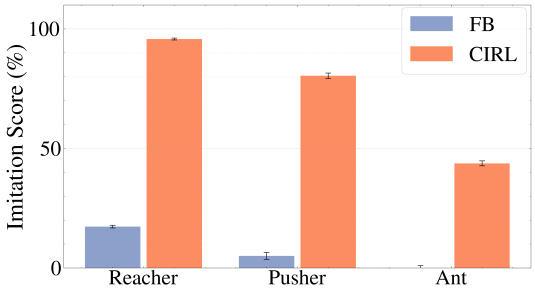

Figure 4: **Summarizing behavior via goals yields better imitation than reward-based explanations.** When using the last expert demonstration state as the goal, CIRL achieves high imitation scores on goal-conditioned environments while FB struggles to infer goal-conditioned reward functions from online learning.

5.3 Mean Field Approximation Improves Goal Inference

Our theory suggests that inferring goals using a mean field approximation should preserve predictive power compared to using the full τ as input to the context encoder. We also have fewer parameters to train under the mean field assumption, and thus hypothesize that it will outperform the full τ alternative. Testing this across environments with CIRL and GCBC, in Figure 5, we see that mean field goal inference universally outperforms the alternative of inferring goals, regardless of environment or training algorithm. These experiments validate our corollary of the preserved predictive power of mean field goal inference, with the added computational benefits of this simplified modeling choice. The mean field assumption also allows us to reliably infer goals from partial trajectories, as shown in Figure 2, where we see the posterior distribution hone in on the true goal as the imitator observes more of the demonstration.

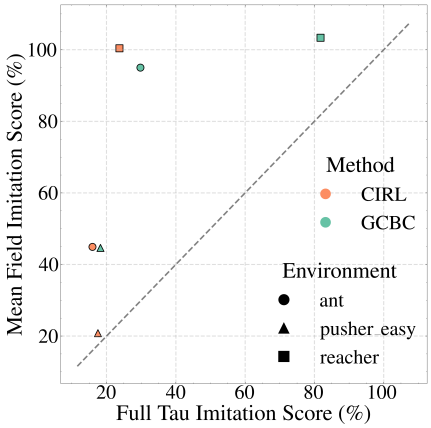

Figure 5: **Mean field goal inference models outperform alternative full τ input models.**

5.4 Better Automatic Goal Sampling Improves Imitation Scores

While we see that CIRL with GoalKDE automatic goal sampling outperforms our baselines with no expert data, we ablate our GoalKDE goal sampling method against oracle goal sampling (which

trains CRL on the test-time goal distribution) to experimentally quantify the gap between these methods. We see in Figure 6 that in the Reacher environment, training CRL policies with GoalKDE can yield near-perfect imitation scores, and that sometimes GoalKDE can better explore the state space for more generalizable policies. However, for the higher dimensional state spaces in Ant and Pusher, a combination of more sophisticated goal sampling techniques or more training steps on more automatically sampled goals could boost performance beyond oracle sampling. See Figure 8 in Appendix D for additional results ablating the CIRL goal inference method as well as Figure 10 for analysis of limitations of GoalKDE in increasingly complex PointMaze environments.

5.5 CIRL Supports Imitation Beyond Goal-Conditioned Environments

We run further experiments on the standard URLB benchmark, which is not designed for goal-reaching, to show that CIRL outperforms prior methods for zero-shot imitation when imitating policies (1) trained with more general reward functions and (2) which require expanding the goal hypothesis space. Following the URLB Benchmark, we train expert policies on the Ant Forward, Ant Jump, and Ant Flip tasks with PPO on non-goal-conditioned reward functions, and report regret of CIRL inferred policies compared to these expert policies. We see in Figure 7 that CRL pre-training methods can achieve lower regret than FB, HILP, or PSM imitation policies by inferring goals involving the ant's torso 3D position and linear/angular velocity. CRL + Oracle goal sampling could perform better in some environments due to sampling fewer infeasible goals, and extensions to CIRL's exploration scheme based on related work could overcome this difficulty (OpenAI et al., 2021). Thus, CIRL can scale to more complex reward functions as long as we sufficiently expand the goal space to capture the task.

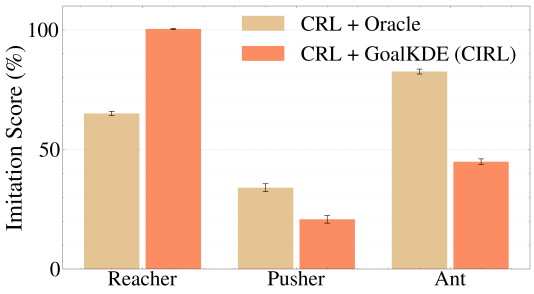

Figure 6: **GoalKDE exploration vs. oracle goal sampling during CRL pre-training.** Holding the goal inference method constant (mean field inference), we find that GoalKDE sampling can achieve a significant fraction of imitation score compared to the oracle baseline, and can even outperform this baseline in some environments.

6 Limitations and Conclusion

Since not all reward functions are goal reaching, future work could close the gap between these reward hypothesis classes by exploring richer goal representations, such as language or multi-modal spaces, and consider summarizing behavior with multiple sub-goals. Our method also requires access to a simulator and would require further research to evaluate applicability in safety-critical settings, settings where collecting data is prohibitively expensive, or human interaction settings. With more complex goal spaces, related work in exploration could be applied as a substitute for our GoalKDE method. A full comparison of goal-sampling methods is outside of the scope of this paper. Our main aim is to propose a full pipeline for enabling imitation via an imagine-and-practice loop in the complete absence of expert data. We introduced a framework for goal-conditioned maximum entropy inverse reinforcement learning that leverages self-supervised contrastive RL pretraining, automatic goal sampling, and a mean field variational goal inference model to enable zero-shot imitation from a single demonstration without access to an offline expert data during training. By re-framing reward inference as goal state inference and coupling this with CRL, our method learns transferable policies across diverse task distributions.

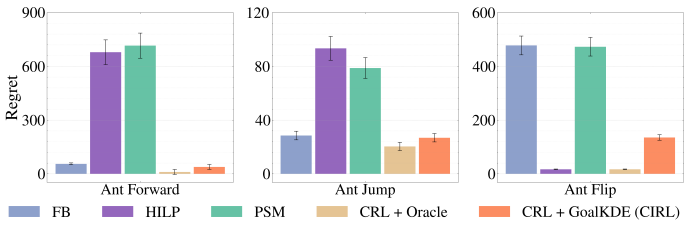

Figure 7: **CIRL inferred goals efficiently summarize complex rewards**. CIRL achieves lower regret than FB when imitating URLB policies with non-goal-reaching rewards.

7 Reproducibility Statement

All experiments in this paper are completely reproducible by running the experiments in our code: https://anonymous.4open.science/r/cirl-3CD7/README.md.. Background information on the environments used and algorithm implementations can be found in the Appendix, and anything not noted can be assumed to follow the defaults of the JaxGCRL and URLB benchmarks (Bortkiewicz et al., 2025; Laskin et al., 2021). Our method is based on open source Brax (Freeman et al., 2021) and Jax (Bradbury et al., 2018) libraries. We also used LLMs in completing this work for two purposes: to polish the paper, including generating some of the icons used in Figure 1 and for the additional purpose of aiding in code writing (via the Cursor AI application).

References

Siddhant Agarwal, Harshit S. Sikchi, Peter Stone, and Amy Zhang. Proto successor measure: Representing the space of all possible solutions of reinforcement learning. *ArXiv*, abs/2411.19418, 2024. URL https://api.semanticscholar.org/CorpusID:274423014.

Luca Ambrogioni, Umut Güçlü, Julia Berezutskaya, Eva van den Borne, Yağmur Güçlütürk, Max Hinne, Eric Maris, and Marcel van Gerven. Forward amortized inference for likelihood-free variational marginalization. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 777–786. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/ambrogioni19a.html.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Francis Christiano, John Schulman, and Dandelion Mané. Concrete problems in ai safety. *ArXiv*, abs/1606.06565, 2016. URL https://api.semanticscholar.org/CorpusID:10242377.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/453fadbd8a1a3af50a9df4df899537b5-Paper.pdf.

André Barreto, Will Dabney, R Munos, Jonathan J Hunt, T Schaul, David Silver, and H V Hasselt. Successor features for transfer in reinforcement learning. *Neural Inf Process Syst*, abs/1606.05312, June 2016.

Léonard Blier, Corentin Tallec, and Yann Ollivier. Learning successor states and goal-dependent values: A mathematical viewpoint. *ArXiv*, abs/2101.07123, 2021. URL https://api.semanticscholar.org/CorpusID:231632451.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *arXiv [cs.CV]*, April 2016.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir P. Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert

Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. *ArXiv*, 2021. URL https://crfm.stanford.edu/assets/report.pdf.

Elizabeth Bonawitz, Patrick Shafto, Hyowon Gweon, Noah D. Goodman, Elizabeth Spelke, and Laura Schulz. The double-edged sword of pedagogy: Instruction limits spontaneous exploration and discovery. *Cognition*, 120(3):322–330, 2011. ISSN 0010-0277. doi: https://doi.org/10.1016/j.cognition.2010.10.001. URL https://www.sciencedirect.com/science/article/pii/S0010027710002258. Probabilistic models of cognitive development.

Diana Borsa, André Barreto, John Quan, Daniel Jaymin Mankowitz, Rémi Munos, H. V. Hasselt, David Silver, and Tom Schaul. Universal successor features approximators. *ArXiv*, abs/1812.07626, 2018. URL https://api.semanticscholar.org/CorpusID:56475997.

Michał Bortkiewicz, Władek Pałucki, Vivek Myers, Tadeusz Dziarmaga, Tomasz Arczewski, Łukasz Kuciński, and Benjamin Eysenbach. Accelerating Goal-Conditioned RL Algorithms and Research. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/pdf/2408.11052.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/jax-ml/jax.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *ArXiv*, abs/1606.01540, 2016. URL https://api.semanticscholar.org/CorpusID:16099293.

Victor Campos, Alexander Trott, Caiming Xiong, Richard Socher, Xavier Giro-I-Nieto, and Jordi Torres. Explore, discover and learn: Unsupervised discovery of state-covering skills. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1317–1327. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/campos20a.html.

Jiayu Chen, Dipesh Tamboli, Tian Lan, and Vaneet Aggarwal. Multi-task hierarchical adversarial inverse reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024.

Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Comput.*, 5(4):613–624, July 1993. ISSN 0899-7667. doi: 10.1162/neco.1993.5.4.613. URL https://doi.org/10.1162/neco.1993.5.4.613.

Anca D. Dragan, Kenton C.T. Lee, and Siddhartha S. Srinivasa. Legibility and predictability of robot motion. In *Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction*, HRI '13, pp. 301–308. IEEE Press, 2013. ISBN 9781467330558.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *ArXiv*, abs/1802.06070, 2018. URL https://api.semanticscholar.org/CorpusID:3521071.

Benjamin Eysenbach, Xinyang Geng, Sergey Levine, and Ruslan Salakhutdinov. Rewriting history with inverse rl: hindsight inference for policy improvement. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. The information geometry of unsupervised reinforcement learning. *ArXiv*, abs/2110.02719, 2021. URL https://api.semanticscholar.org/CorpusID:238408056.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Chelsea Finn, Sergey Levine, and P. Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference on Machine Learning*, 2016. URL https://api.semanticscholar.org/CorpusID:8121626.

Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In Sergey Levine, Vincent Vanhoucke, and Ken Goldberg (eds.), *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pp. 357–368. PMLR, 13–15 Nov 2017. URL https://proceedings.mlr.press/v78/finn17a.html.

Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1515–1528. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/florensa18a.html.

C Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax – a differentiable physics engine for large scale rigid body simulation. *arXiv [cs.RO]*, June 2021.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adverserial inverse reinforcement learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rkHywl-A-.

Dibya Ghosh, Chethan Anand Bhateja, and Sergey Levine. Reinforcement learning from passive data via latent intentions. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11321–11339. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/ghosh23a.html.

Alison Gopnik. Childhood as a solution to explore–exploit tensions. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 375(1803):20190502, 2020. doi: 10.1098/rstb.2019.0502. URL https://royalsocietypublishing.org/doi/abs/10.1098/rstb.2019.0502.

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *ArXiv*, abs/1611.07507, 2016. URL https://api.semanticscholar.org/CorpusID:2918187.

Hyowon Gweon and Laura Schulz. From exploration to instruction: Children learn from exploration and tailor their demonstrations to observers' goals and competence. *Child Development*, 90(1):e148–e164, 2019. doi: https://doi.org/10.1111/cdev.13059. URL https://srcd.onlinelibrary.wiley.com/doi/abs/10.1111/cdev.13059.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 4572–4580, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. BC-z: Zero-shot task generalization with robotic imitation learning. In *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=8kbp23tSGYv.

Leslie Pack Kaelbling. Learning to achieve goals. In *International Joint Conference on Artificial Intelligence*, 1993. URL https://api.semanticscholar.org/CorpusID:5538688.

Seongun Kim, Kyowoon Lee, and Jaesik Choi. Variational curriculum reinforcement learning for unsupervised discovery of skills. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Martin Klissarov and Marlos C. Machado. Deep laplacian-based options for temporally-extended exploration. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Misha Laskin, Denis Yarats, Hao Liu, Kimin Lee, Albert Zhan, Kevin Lu, Catherine Cang, Lerrel Pinto, and Pieter Abbeel. Urlb: Unsupervised reinforcement learning benchmark. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/091d584fced301b442654dd8c23b3fc9-Paper-round2.pdf.

Chen Ma, Dylan R. Ashley, Junfeng Wen, and Yoshua Bengio. Universal successor features for transfer reinforcement learning. *ArXiv*, abs/2001.04025, 2018. URL https://api.semanticscholar.org/CorpusID:86513851.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *ArXiv*, abs/2210.00030, 2022. URL https://api.semanticscholar.org/CorpusID:252683397.

Marlos C. Machado, Marc G. Bellemare, and Michael H. Bowling. A laplacian framework for option discovery in reinforcement learning. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *JMLR Workshop and Conference Proceedings*, pp. 2295–2304. JMLR.org, 2017. URL http://proceedings.mlr.press/v70/machado17a.html.

Russell Mendonca, Oleh Rybkin, Kostas Daniilidis, Danijar Hafner, and Deepak Pathak. Discovering and achieving goals via world models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 24379–24391. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/cc4af25fa9d2d5c953496579b75f6f6c-Paper.pdf.

Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. Learning Temporal Distances: Contrastive Successor Features Can Provide a Metric Structure for Decision-Making. In *International Conference on Machine Learning*, 2024. URL https://proceedings.mlr.press/v235/myers24a.html.

Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/7ec69dd44416c46745f6edd947b470cd-Paper.pdf.

Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pp. 663–670, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1558607072.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv:1807.03748 [cs, stat]*, July 2018. URL http://arxiv.org/abs/1807.03748. arXiv: 1807.03748.

OpenAI OpenAI, Matthias Plappert, Raul Sampedro, Tao Xu, Ilge Akkaya, Vineet Kosaraju, Peter Welinder, Ruben D'Sa, Arthur Petron, Henrique Pondé de Oliveira Pinto, Alex Paino, Hyeonwoo Noh, Lilian Weng, Qiming Yuan, Casey Chu, and Wojciech Zaremba. Asymmetric self-play for automatic goal discovery in robotic manipulation. *ArXiv*, abs/2101.04882, 2021. URL https://api.semanticscholar.org/CorpusID:231592353.

Seohong Park, Kimin Lee, Youngwoon Lee, and Pieter Abbeel. Controllability-aware unsupervised skill discovery. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27225–27245. PMLR, 23–29 Jul 2023a. URL https://proceedings.mlr.press/v202/park23h.html.

Seohong Park, Oleh Rybkin, and Sergey Levine. Metra: Scalable unsupervised rl with metric-aware abstraction. *ArXiv*, abs/2310.08887, 2023b. URL https://api.semanticscholar.org/CorpusID:264128272.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2778–2787. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/pathak17a.html.

Deepak Pathak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Fred Shentu, Evan Shelhamer, Jitendra Malik, Alexei A. Efros, and Trevor Darrell. Zero-shot visual imitation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2131–21313, 2018. doi: 10.1109/CVPRW.2018.00278.

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5062–5071. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/pathak19a.html.

Matteo Pirotta, Andrea Tirinzoni, Ahmed Touati, Alessandro Lazaric, and Yann Ollivier. Fast imitation via behavior foundation models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Representation Learning*, volume 2024, pp. 12685–12724, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/370645ae400005f888e418f46b594539-Paper-Conference.pdf.

Silviu Pitis, Harris Chan, Stephen Zhao, Bradly Stadie, and Jimmy Ba. Maximum entropy gain exploration for long horizon multi-goal reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Francesco Poli, Marlene Meyer, Rogier B. Mars, and Sabine Hunnius. Exploration in 4-year-old children is guided by learning progress and novelty. *Child Development*, 96(1):192–202, 2025. doi: https://doi.org/10.1111/cdev.14158. URL https://srcd.onlinelibrary.wiley.com/doi/abs/10.1111/cdev.14158.

Dean A. Pomerleau. Alvinn: an autonomous land vehicle in a neural network. In *Proceedings of the 2nd International Conference on Neural Information Processing Systems*, NIPS'88, pp. 305–313, Cambridge, MA, USA, 1988. MIT Press.

Vitchyr H. Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew-fit: state-covering self-supervised reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Sai Rajeswar, Pietro Mazzaglia, Tim Verbelen, Alexandre Piché, B. Dhoedt, Aaron C. Courville, and Alexandre Lacoste. Mastering the unsupervised reinforcement learning benchmark from pixels. In *International Conference on Machine Learning*, 2022. URL https://api.semanticscholar.org/CorpusID:258887708.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *arXiv [cs.AI]*, May 2022.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/ross11a.html.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1312–1320, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/schaul15.html.

Seyed Kamyar Seyed Ghasemipour, Shixiang (Shane) Gu, and Richard Zemel. Smile: Scalable meta inverse reinforcement learning through context-conditional policies. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/2b8f621e9244cea5007bac8f5d50e476-Paper.pdf.

Archit Sharma, Shixiang Shane Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. *ArXiv*, abs/1907.01657, 2019. URL https://api.semanticscholar.org/CorpusID:195791369.

Harshit S. Sikchi, Siddhant Agarwal, Pranaya Jajoo, Samyak Parajuli, Caleb Chuck, Max Rudolph, Peter Stone, Amy Zhang, and Scott Niekum. Rlzero: Direct policy inference from language without in-domain supervision. 2024. URL https://api.semanticscholar.org/CorpusID:274597708.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016.

Aimee E. Stahl and Lisa Feigenson. Observing the unexpected enhances infants' learning and exploration. *Science*, 348(6230):91–94, 2015. doi: 10.1126/science.aaa3799. URL https://www.science.org/doi/abs/10.1126/science.aaa3799.

Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Ahmed Touati, Jérémy Rapin, and Yann Ollivier. Does zero-shot reinforcement learning exist? *ArXiv*, abs/2209.14935, 2022. URL https://api.semanticscholar.org/CorpusID:252596234.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *International Conference on Machine Learning*. PMLR, 2023.

Zizhao Wang, Jiaheng Hu, Caleb Chuck, Stephen Chen, Roberto Martín-Martín, Amy Zhang, Scott Niekum, and Peter Stone. Skild: unsupervised skill discovery guided by factor interactions. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

David Warde-Farley, Tom Van de Wiele, Tejas Kulkarni, Catalin Ionescu, Steven Hansen, and Volodymyr Mnih. Unsupervised control through non-parametric discriminative rewards. *arXiv [cs.LG]*, November 2018.

Yifan Wu, George Tucker, and Ofir Nachum. The laplacian in rl: Learning representations with efficient approximations. *arXiv preprint arXiv:1810.04586*, 2018.

Peter R Wurman, Samuel Barrett, Kenta Kawamoto, James MacGlashan, Kaushik Subramanian, Thomas J Walsh, Roberto Capobianco, Alisa Devlic, Franziska Eckert, Florian Fuchs, Leilani Gilpin, Piyush Khandelwal, Varun Kompella, Haochih Lin, Patrick MacAlpine, Declan Oller, Takuma Seno, Craig Sherstan, Michael D Thomure, Houmehr Aghabozorgi, Leon Barrett, Rory Douglas, Dion Whitehead, Peter Dürr, Peter Stone, Michael Spranger, and Hiroaki Kitano. Outracing champion gran turismo drivers with deep reinforcement learning. *Nature*, 602(7896):223–228, February 2022.

Lantao Yu, Tianhe Yu, Chelsea Finn, and Stefano Ermon. Meta-inverse reinforcement learning with probabilistic context variables. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/30de24287a6d8f07b37c716ad51623a7-Paper.pdf.

Tianhe Yu, Chelsea Finn, Annie Xie, Sudeep Dasari, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot imitation from observing humans via domain-adaptive meta-learning. *arXiv [cs.LG]*, February 2018.

Tom Zahavy, Yannick Schroecker, Feryal M. P. Behbahani, Kate Baumli, Sebastian Flennerhag, Shaobo Hou, and Satinder Singh. Discovering policies with domino: Diversity optimization maintaining near optimality. *ArXiv*, abs/2205.13521, 2022. URL https://api.semanticscholar.org/CorpusID:249097488.

Chongyi Zheng, Jens Tuyls, Joanne Peng, and Benjamin Eysenbach. Can a misl fly? analysis and ingredients for mutual information skill learning. *arXiv preprint arXiv:2412.08021*, 2024.

Brian D Ziebart, Andrew L Maas, J Bagnell, and A Dey. Maximum entropy inverse reinforcement learning. *National Conference on Artificial Intelligence*, pp. 1433–1438, July 2008.

Matthew Zurek, Andreea Bobu, Daniel S. Brown, and Anca D. Dragan. Situational confidence assistance for lifelong shared autonomy. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2783–2789. IEEE Press, 2021. doi: 10.1109/ICRA48506.2021.9561839. URL https://doi.org/10.1109/ICRA48506.2021.9561839.

A Theoretical Analysis

A.1 CIRL is Consistent

Proof. MaxEnt IRL corresponds to the following objective:

$$\arg\min_{\theta} D_{\mathrm{KL}}\left(p_{\pi_E(\tau)} \| p^\star(\tau)\right) = \arg\max_{\theta} \mathbb{E}_{p_{\pi_E}(\tau)}\left[\log p^\star(\tau)\right]$$

Under MaxEnt modeling, each goal g induces a trajectory model $p^\star(\tau \mid g) \propto \left[\rho_0(s_0)\prod_{t=0}^{T} p(s_{t+1} \mid s_t, a_t)\right] \exp\left(\sum_{t=0}^{T} r_g(s_t, a_t)\right)$ with log-partition $\log Z_g$. In a goal-conditioned setting, taking the reward to be entirely determined by g means the family $\{p^\star(\tau \mid g)\}_g$ is indexed by goals, and the learning objective can be posed as minimizing the average forward KL

$$\min_\theta \mathbb{E}_{p(g)}\left[D_{\mathrm{KL}}\left(p_E(\tau \mid g) \| p^\star(\tau \mid g)\right)\right],$$

where $p(g)$ is the goal prior used both in data collection and modeling.

Define the expert and model joints over (τ, g) as $p_E(\tau, g) = p(g)p_E(\tau \mid g)$ and $p^\star(\tau, g) = p(g)p^\star(\tau \mid g)$. When the same prior $p(g)$ is used, the average conditional KL equals a joint forward KL :

$$\mathbb{E}_{p(g)}\left[D_{\mathrm{KL}}\left(p_E(\tau \mid g) \| p^\star(\tau \mid g)\right)\right] = \mathbb{E}_{p(g)}\left[D_{\mathrm{KL}}\left(p_E(\tau, g) \| p^\star(\tau, g)\right)\right],$$

by applying Bayes Rule and canceling the identical priors.

Apply the KL chain rule to the joint KL:

$$D_{\mathrm{KL}}\left(p_E(\tau, g) \| p^\star(\tau, g)\right) = D_{\mathrm{KL}}\left(p_E(\tau) \| p^\star(\tau)\right) + \mathbb{E}_{\tau \sim p_E(\tau)}\left[D_{\mathrm{KL}}\left(p_E(g \mid \tau) \| p^\star(g \mid \tau)\right)\right],$$

Thus our MaxEnt IRL objective is

$$\min_\theta \mathbb{E}_{p(g)} D_{\mathrm{KL}}\left(p_E(\tau \mid g) \| p^\star(\tau \mid g)\right) = \min_\theta \left\{D_{\mathrm{KL}}\left(p_E(\tau) \| p^\star(\tau)\right) + \mathbb{E}_{p_E(\tau)} D_{\mathrm{KL}}\left(p_E(g \mid \tau) \| p^\star(g \mid \tau)\right)\right\} \tag{4}$$

Now we note that our marginal distribution $p^\star(\tau) = \int p(g)p^\star(\tau|g)dg$ is a difficult integral to compute and thus apply variational inference by introducing the amortized variational distribution $q_\xi(g; \tau)$. Then

$$\log p^\star(\tau) = ELBO(\theta, \xi; \tau) + D_{KL}(q_\xi(g|\tau) \| p^\star(g|\tau))$$

where

$$ELBO = \mathbb{E}_{q_\xi}\left[\log p(g) + \log p^\star(\tau|g) - \log q_\xi(g|\tau)\right]$$

Taking the expectation over expert trajectories:

$$\begin{aligned}
\min_\theta \left\{D_{\mathrm{KL}}\left(p_E(\tau) \| p^\star(\tau)\right)\right\} &= \max_\theta \mathbb{E}_{p_E(\tau)}\left[\log p^\star(\tau)\right] \\
&= \max_\xi \mathbb{E}_{q_\xi}\left[\log p(g) + \log p^\star(\tau|g) - \log q_\xi(g|\tau)\right] \\
&= \min_\xi \left[D_{KL}(q_\xi(g|\tau) \| p^\star(g|\tau))\right]
\end{aligned}$$

Now we see the major issue with using the ELBO/reverse KL is that it requires us to be able to evaluate the conditional likelihood $p^\star(\tau|g)$. This is impossible in our scenario, but we could sample from it since we can sample from the trajectory distribution of our MaxEnt RL policy. This motivates the use of **Forward Amortized Variational Inference (FAVI)**, which uses the forward KL instead of the reverse KL in its optimization (Ambrogioni et al., 2019).

The loss function of FAVI derives from the joint-contrastive variational inference objective and is expressed as:

$$\mathcal{L}_{\text{FAVI}}[p, q] = D(p^\star(g, \tau) \| q_\xi(g, \tau))$$

To approximate the intractable posterior $p^\star(g \mid \tau)$, we factorize the variational joint as the product of a variational posterior $q_\xi(g \mid \tau)$ and a sampling distribution of the data:

$$q_\xi(\tau, g) = q_\xi(g \mid \tau)k(\tau)$$

Now we note:

$$D_{KL}(p^\star(\tau, g) \| q_\xi(\tau, g)) = \mathbb{E}_{p^\star(\tau,g)}\left[\log \frac{p^\star(\tau, g)}{q_\xi(g \mid \tau)k(\tau)}\right] \tag{5}$$

$$= -\mathbb{E}_{p^\star(\tau,g)}[\log q_\xi(g \mid \tau)] + \mathbb{E}_{p^\star(\tau,g)}\left[\log \frac{p^\star(\tau, g)}{k(\tau)}\right] \tag{6}$$

Considering only the terms that depends on q, we can define the FAVI loss as follows:

$$\mathcal{L}_{\text{FAVI}} = -\mathbb{E}_{p^\star(\tau,g)}[\log q_\xi(g \mid \tau)]$$

This is precisely the loss function $\mathcal{L}_{\text{Info}}(\xi)$ we train. Therefore, for our goal-conditioned setting, the IRL problem can be reduced to one of learning a variational posterior with FAVI. Importantly, note that the partition function is implicit within the samples we generate from the joint distribution via $g \sim p(g), \tau \sim p^\star(\tau \mid g)$, allowing us to consistently infer goals where methods that ignore the partition function do not. □

A.2 FB is Inconsistent

We prove this by providing a counterexample. The key idea in the counterexample is that an infrequently visited state may nonetheless be the policy's desired goal. We illustrate this with a simple 2-state MDP.

Proof. We define an MDP with 2 states (s_1, s_2) and 2 actions (a_1, a_2) with the following dynamics:

$$p(s' \mid s, a) = \begin{cases} s_1, & \text{if } s = s_1, a = a_1 \\ s_1, & \text{w.p. } \frac{1}{2} \text{ if } s = s_1, a = a_2 \\ s_2, & \text{w.p. } \frac{1}{2} \text{ if } s = s_1, a = a_2 \\ s_2, & \text{if } s = s_2 \end{cases}. \tag{7}$$

Assume that the initial state is distributed $p_0(s) = \mathbb{1}(s_1)$. Note that state Y has just one action. The only decision to make is the action at initial state X. Since all MDPs have deterministic optimal policies, there are just two unique (potential) reward-maximizing policies for this MDP:

$$\pi_1(a \mid s) = \begin{cases} a_1 & \text{if } s = s_1 \\ \text{any action} & \text{if } s = s_2 \end{cases} \tag{8}$$

$$\pi_2(a \mid s) = \begin{cases} a_2 & \text{if } s = s_1 \\ \text{any action} & \text{if } s = s_2 \end{cases} \tag{9}$$

We will show that when data are collected from policy π_2, FB infers that data were collected with policy π_1. This policy is clearly different, achieving different amounts of rewards (for all non-trivial reward functions).

We next compute the occupancy measure for policy π_2. From the initial state x, the policy transitions to state y with probability $\frac{1}{2}$ at each time step. Thus, the probability of still being at state x after t time steps decays as $1/2^t$. The occupancy measure can thus be written as:

$$\rho^{\pi_2}(s = X) = (1 - \gamma) + [1 + \gamma\frac{1}{2} + \gamma^2\frac{1}{2^2} + \gamma^3\frac{1}{2^3} + \cdots] \tag{10}$$

$$= (1 - \gamma)\sum_{t=0}^{\infty}(\gamma/2)^t = \frac{1 - \gamma}{1 - \gamma/2}. \tag{11}$$

Then $\rho^{\pi_2}(s = Y) = 1 - \rho^{\pi_2}(s = X)$. Thus, when γ is small enough, policy π_2 "spends more time at" state x than state y:

$$\gamma < \frac{2}{3} \implies \rho^{\pi_2}(s = s_1) > \rho^{\pi_2}(s = s_2). \tag{12}$$

This will be a problem for FB, which infers rewards based not on the difficulty of maximizing them, but rather instead based on visitation counts:

$$z_R = \sum_t B(s_t). \tag{13}$$

Without loss of generality, we assume that $B(s_t) = \mathbb{1}(s_t)$, a one hot vector; this solution is always admissible if the representations have high-enough dimension. Thus, the inferred reward function is

$$r(s) = \begin{cases} \frac{1-\gamma}{1-\gamma/2} & \text{if } s = s_1 \\ \frac{\gamma/2}{1-\gamma/2} & \text{if } s = s_2 \end{cases} \tag{14}$$

Note that state s_1 has a higher reward than state s_2 with $\gamma < \frac{2}{3}$. Thus, the reward-maximizing policy for this reward function is π_1 (which stays in s_1), not π_2 (which sometimes transitions to the lower reward state s_2). □

This demonstrates that FB incorrectly identifies demonstrations from π_2 as coming from π_1. The fundamental issue is that FB uses the occupancy measure directly as the reward signal without considering the partition function or the policy's optimality under that reward. This leads to systematic misidentification of the demonstrator's true policy.

B Algorithm

We present pseudocode for training our zero-shot IL method based on contrastive RL pretraining:

Algorithm 1 Contrastive IRL

Input: CRL loss $\mathcal{L}_{\text{Critic}}$ and energy function $f_{\phi,\psi}(s, a, g) = \phi(s, a)^T \psi(g)$ (Eysenbach et al., 2022), Entropy-regularization value function $\mathcal{L}_{\text{Entropy}}$, actor objective $\mathcal{L}_{\text{Actor}}$, variational posterior loss $\mathcal{L}_{\text{info}}$
Initialize ϕ, ψ, θ, ξ, π and a pre-filled replay buffer $\mathcal{D}$
repeat
  in parallel over environments
    $g = \arg\min_g KDE(\mathcal{D})$
    Store $\tau \sim \pi(s, g)$ in $\mathcal{D}$
  for $j = 1, \ldots,$ `num_updates` **do**
    Randomly sample (with discount) a batch $\mathcal{B}$ from $\mathcal{D}$ of state-action pairs and goals from their future
    Update critic:
      $(\phi, \psi) \leftarrow (\phi, \psi) - \alpha \nabla_{\phi,\psi} \left[\mathcal{L}_{\text{Critic}}(\mathcal{B}; \phi, \psi)\right]$
    Update entropy-regularization value function:
      $(\theta) \leftarrow (\theta) - \alpha \nabla_{\theta} \left[\mathcal{L}_{\text{Entropy}}(\mathcal{B}; \theta)\right]$
    Update policy:
      $\pi \leftarrow \pi - \alpha \nabla_{\pi} \left[\mathcal{L}_{\text{Actor}}(\mathcal{B}; \phi, \psi, \pi)\right]$
    Update variational posterior:
      $q \leftarrow q - \alpha \nabla_{\xi} \left[\mathcal{L}_{\text{Info}}(\mathcal{B}; \phi, \psi, \pi)\right]$
until convergence

C Experimental Details

We ran our experiments building off the JaxGCRL benchmark (Bortkiewicz et al., 2025). Unless otherwise mentioned, we used the same hyperparameters as that implementation. α used for Maximum Entropy IRL was 1e-5. For the FB representation, PSM and HILP baselines, we use the same

encoder networks as in JaxGCRL and the same actor and critic learning rates. The metric value net for the HILP baseline similarly uses the JaxGCRL encoder architecture for the phi network. For the context encoder, we also use the JaxGCRL encoder and train to predict the mean and variance of a Gaussian. The backbone MLP of the JaxGCRL encoder networks has a hidden width of 256 units, hidden depth of 2 layers, and 1 skip connection. We use the Swish activation after each hidden Dense layer. We used the seed 1 to train each type of policy.

Table 1: Reacher environment hyperparameters

hyperparameter	value
batch size	1024
num timesteps	20,000,000
num environments	256

Table 2: Pusher environment hyperparameters (goal: 3D position and 3D linear velocity)

hyperparameter	value
batch size	256
num timesteps	60,000,000
num environments	512

Table 3: Ant environment hyperparameters (goal: 2D position)

hyperparameter	value
batch size	512
num timesteps	30,000,000
num environments	1024

Table 4: Ant environment hyperparameters (goal: 3D position and 3D linear velocity)

hyperparameter	value
batch size	256
num timesteps	600,000,000
num environments	512
healthy z range	(0.0, 4.0)
target z	Uniform over range (0.2, 2.0)
target 3D linear velocity	Uniform over (-1.0, 1.0)

Table 5: Ant environment hyperparameters (goal: 3D angular velocity)

hyperparameter	value
batch size	256
num timesteps	600,000,000
num environments	512
target 3D angular velocity	Uniform over (-2.5, 2.5)

Table 6: PointMaze environment hyperparameters (goal: 2D position)

hyperparameter	value
batch size	1024
num environments	256
num timesteps	20,000,000 (U-Maze) 40,000,000 (Big Maze) 60,000,000 (Hardest Maze)

C.1 Environments

Reacher: This environment is a 2D manipulation task involving a two-jointed robotic arm. The goal is to move the arm's end effector to a sampled 2-dimensional target located randomly within a workspace disk. The 11-dimensional state space includes joint angles and velocities along with the position of the end effector. The 2-dimensional action represents torques applied at the arm's hinge joints.

Pusher: This features a 3D robotic arm and a movable object resting on a surface. The objective is to push the object into a 2D goal location randomly sampled at each episode reset. The 23-dimensional state space includes the arm's joint angles, velocities, and the position of the movable object. The 7-dimensional action space controls the robotic arm via continuous motor torques at its joints.

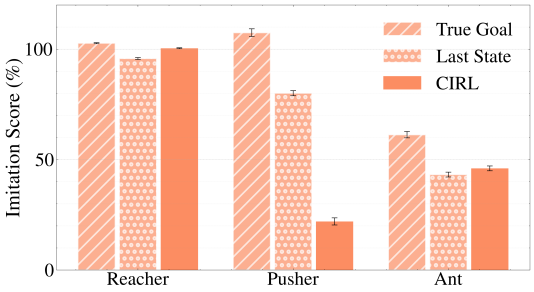

Figure 8: **Ablating CIRL.** For Pusher, most of the performance gap is due to goal inference, but for the Ant environment, most of the performance gap is likely due to distribution shift induced by GoalKDE.

Ant: This locomotion task involves a quadruped navigating towards target XY positions randomly sampled from a circle around its starting position. The 29-dimensional state space comprises the robot's joint positions, orientations, and velocities, and the 8-dimensional action space consists of torques applied to each of the multiple leg joints. When using CIRL to infer URLB rewards, we expand the goal space to include the 3D position and 3D linear velocity or 3D angular velocity.

PointMaze: This navigation task involves a point mass navigating towards a target XY position in a constructed maze. The 4-dimensional state space includes the position and linear velocity of the point mass, and the 2D action space controls linear force along slide joints of the point mass in the environment. The possible configurations include U-Maze (20 x 20 maze layout), Big Maze (32 x 32 maze layout), and Hardest Maze (36 x 48 maze layout), in order of increasing state space size.

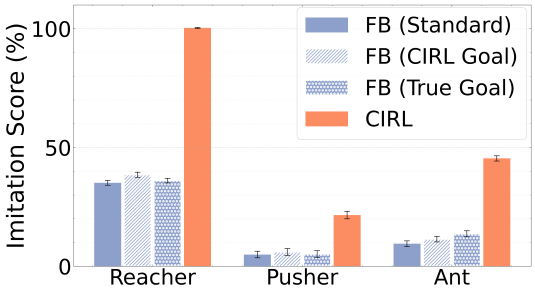

Figure 9: **Use the backward representation of the CIRL inferred goal to command the FB policy.** Comparing alternate methods of computing the FB latent against the standard FB baseline, CIRL still achieves higher imitation scores.

D Additional Results

Figure 8 ablates CIRL performance against alternate goal inference methods: knowing the true goal or inferring the goal to be the last state of the expert demonstration. When we provide the true goal to a policy trained with CRL and GoalKDE exploration, we get a significant boost in imitation score, with any gap in imitation score from 100% likely due to distribution shift between

goals sampled via GoalKDE and those from the oracle test distribution, and alternative methods for goal exploration are a promising area for future work in GCRL. For goal-conditioned expert policies, inferring the last state to be the goal can be a strong baseline, but would fail when we try to imitate a task such as an Ant jumping.

In Figure 9, we imagine if the FB representation had access to the CIRL mean field inferred goal. Using this state to compute a backward representation latent to command the FB policy, FB is still not able to match the performance of CIRL. This is likely because FB must perform more exploration to learn a zero-shot policy for all possible rewards, including goal-reaching rewards, while CIRL has a smaller hypothesis space of rewards to explore and learn. However, we do note that FB with the inferred goal does outperform FB with the latent computed from either the last state of the demonstration or the averaged backward representation of all states in the demonstration.

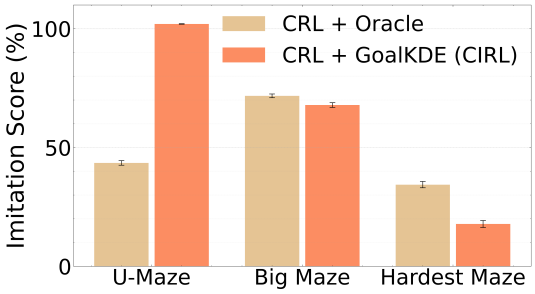

Figure 10: **Testing the limits of GoalKDE as a goal-sampling method for CIRL.** We see that the performance gap between CRL + Oracle and CIRL widens as the state space of PointMaze environments grows.

In Figure 10, we perform further analysis of GoalKDE's capabilities across increasingly complex PointMaze environments, with U Maze having the smallest state space and Hardest Maze having the largest. We see that as the state space grows, the performance gap between GoalKDE and Oracle goal sampling widens.