GENERALIZED BEHAVIOR LEARNING FROM DIVERSE DEMONSTRATIONS

Anonymous authors
Paper under double-blind review

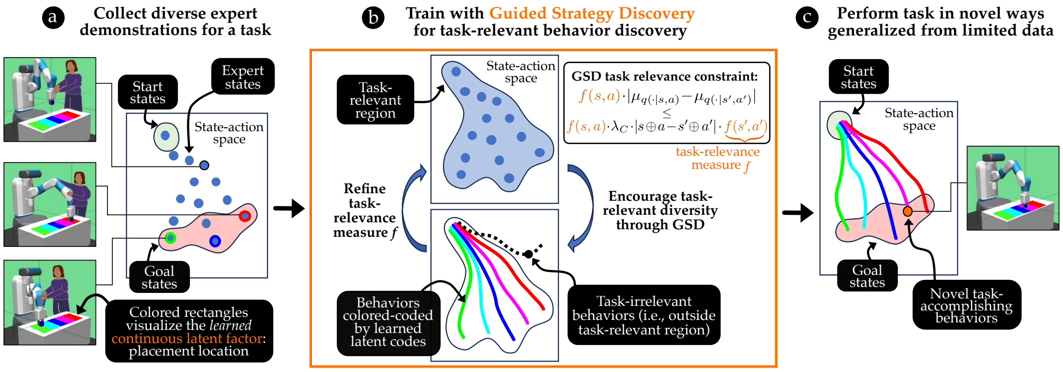

a
Collect diverse expert demonstrations for a task
Start states
Expert states
State-action space
Goal states
Colored rectangles visualize the learned continuous latent factor: placement location
b
Train with Guided Strategy Discovery for task-relevant behavior discovery
Task-relevant region
State-action space
GSD task relevance constraint:
$f(s,a)\cdot|\mu_{q(\cdot|s,a)}-\mu_{q(\cdot|s',a')}| \leq f(s,a)\cdot\lambda_C\cdot|s\oplus a-s'\oplus a'|\cdot \underbrace{f(s',a')}_{\text{task-relevance measure } f}$
Refine task-relevance measure f
Encourage task-relevant diversity through GSD
Behaviors colored-coded by learned latent codes
Task-irrelevant behaviors (i.e., outside task-relevant region)
c
Perform task in novel ways generalized from limited data
Start states
State-action space
Goal states
Novel task-accomplishing behaviors

Figure 1: The figure overviews our framework: Guided Strategy Discovery. (a) Given diverse task demonstrations with underlying latent factors, (b) our framework optimizes a task-relevance guided diversity objective, (c) to discover behaviors generalizing to unseen latent factor values.

ABSTRACT

Diverse behavior policies are especially valuable in domains requiring quick test-time adaptation or personalized human-robot interaction. Human demonstrations provide rich information regarding task objectives and individual preferences, which can be used to characterize *useful* diversity and learn diverse performant policies. However, we show that prior work that builds naive representations of demonstration heterogeneity fails in generating successful novel behaviors that generalize over preferences. We propose Guided Strategy Discovery (GSD), which introduces a novel diversity formulation based on a learned task-relevance measure that prioritizes behaviors exploring modeled latent factors. We empirically validate across three continuous control benchmarks for generalizing to in-distribution (interpolation) and out-of-distribution (extrapolation) preferences that GSD outperforms baselines in novel behavior discovery by $\sim$21%. Finally, we demonstrate that GSD can generalize striking behaviors for table tennis in a virtual testbed while leveraging human demonstrations collected in the real world.

1 INTRODUCTION

Intelligent agents encountered with a novel variation of a learned task should be able to adapt their default decision making to suit the variation at hand. To adapt on-the-fly, agents must learn a concise set of variations to quickly tune their behaviors. Such adaptability is valuable in applications such as few-shot learning (Duan et al., 2017) and personalized human-robot interaction (Wang et al., 2022) where limited examples or interaction must inform compatible approaches for task completion.

Behaviors that adaptable agents learn must be meaningfully diverse such that novel task variations can be addressed sufficiently. In the past, unsupervised reinforcement learning (RL) (Laskin et al., 2021) has been used to learn diverse behaviors or "skills". However, learned behaviors intended to explore the agent's environment may not directly be useful towards task completion. Such approaches are further limited by their inability to identify and exhibit meaningful variations that are useful during deployment. While supervised RL can be employed, reward specification to align diverse behaviors with user expectations can be cumbersome (Soares & Fallenstein, 2014).

In contrast to RL, learning from demonstration (LfD) methods enable agents to learn decision-making policies directly from human examples. The distinct latent preferences that individuals often have, even when pursuing the same task objectives (Sanderson, 1989) can provide rich information to learn diverse behaviors. The preferences reflect creative ways in which tasks are completed, which imparts useful diversity in behaviors that adaptable agents can exploit.

Prior work such as heterogeneous[1] or interpretable Imitation Learning (IL) (Li et al., 2017; Chi et al., 2023) largely focuses on generating behaviors corresponding to preferences in training datasets or inferring representations of test behaviors. In this work, we address the challenge of generating behaviors with novel latent variations. We specifically study the ability to inter- and extrapolate from a set of demonstrations to effectively produce new behaviors, that correspond to latent factor values not seen in the training dataset, while also accomplishing the task. For example, consider a robot quadruped that runs at different speeds. We seek policies that run at 2m/s or 4m/s from demonstrations with speeds of 1m/s and 3m/s. Such a generalization ability can provide task-accomplishing behaviors with desirable characteristics directly through latent prior sampling. Generalization over continuous latent factors is challenging, as we need to accurately identify the latent dimensions along which demonstrations vary and locate individual behaviors in the corresponding space before extending to novel behaviors.

We focus on novel behavior generation in the setting of online IL (Ho & Ermon, 2016) due to its data-sample efficiency. We show that prior approaches that utilize mutual information (MI)-based diversity objectives (Li et al., 2017) fail to produce novel behaviors. We draw inspiration from recent work in unsupervised RL (Park et al., 2022; 2024; 2023) that modify MI-based objectives and structure latent representations in order to induce specific behavioral traits (e.g., high Euclidean or temporal distances between states, controllability, etc). We propose to modify representation learning by restricting the latent space from capturing state-action space regions irrelevant to the task, identified through distillation of preference-specific occupancy measures. We find that our formulation encourages diversity specifically along traits that vary across demonstrations. We refer to this objective as task-relevant diversity as it produces behaviors that retain task-performance.

We present a novel approach to learn diverse task-accomplishing behaviors from demonstrations that generalize over latent preferences. Our contributions are four-fold:

- We show the need for a novel formulation of diversity for generalization in IL from diverse demonstrations through experiments in a 2D Point Maze domain (Sec. 4).
- We formulate task-relevant diversity, an objective to encourage diversity along factors of variation among demonstrations by restricting representations from capturing irrelevant regions. We propose Guided Strategy Discovery (GSD), an algorithm that optimizes diversity alongside imitation to discover novel task-accomplishing behaviors (Sec. 5).
- We demonstrate that GSD generalizes to novel behaviors with 21% average error reduction in preferences (known during evaluation) over four baselines across two preference splits (interpolation and extrapolation) in three domains spanning robot control, driving, and manipulation (Sec. 6.1).
- We demonstrate that GSD generalizes from physical human demonstrations to capture diverse stroke styles in a simulated Table Tennis domain (Sec. 6.3).

2 RELATED WORK

Generalization in Behavior Learning Prior works have studied generalization when agents are faced with task specifications from test distributions (Benjamins et al., 2022; Silva et al., 2021; Padalkar et al., 2023; Nair et al., 2022; Shridhar et al., 2023; Xu et al., 2022; Driess et al., 2020), or deployment settings different from training environments (Fu et al., 2017; Kumar et al., 2020; Packer et al., 2018; Kumar et al., 2021; Xie et al., 2023; Cobbe et al., 2019; Osa et al., 2022; Zahavy et al., 2022). In IL, generalization has been considered when demonstrators operate with diverse conditions (Qiu et al., 2023; Tangkaratt et al., 2020; Chen et al., 2021; Paleja et al., 2020; Schrum et al., 2023b; Li et al., 2017; Chen et al., 2020; Wang et al., 2017; Li et al., 2017; Hausman et al., 2017; Peng et al., 2022). Our work focuses on the latter, where we study heterogeneous demonstrators with latent preferences. Among these, prior works either attempt to characterize heterogeneity through latent representations (Paleja et al., 2020; Schrum et al., 2023b; Li et al.,

---

[1] Prior work has attributed heterogeneity to (i) suboptimality, where humans perform tasks in varying suboptimal ways (Ramachandran & Amir, 2007), and (ii) latent preferences, where tasks are performed differently based on individual preferences (Sanderson, 1989). Our work focuses solely on latent preferences.

2017; Chen et al., 2020), learn performant behaviors from diverse demonstrators (Qiu et al., 2023; Tangkaratt et al., 2020; Chen et al., 2021) or learn multiple behaviors (Wang et al., 2017; Li et al., 2017; Hausman et al., 2017; Peng et al., 2022). To the best of our knowledge, our work is the first to address all three to learn diverse performant behaviors that generalize demonstrator preferences.

Behavior learning from demonstrations Prior works in LfD utilize demonstrations to learn rewards (Abbeel & Ng, 2004; Ziebart et al., 2008; Fu et al., 2017; Chen et al., 2020), task-performant policies from optimal and suboptimal demonstrations (Ross et al., 2011; Ho & Ermon, 2016; Qiu et al., 2023; Tangkaratt et al., 2020; Chen et al., 2021) or diverse behaviors (Wang et al., 2017; Li et al., 2017; Hausman et al., 2017; Chen et al., 2020; Peng et al., 2022). Our work focuses on achieving behavior diversity, particularly with offline expert demonstrations with continuous latent factors. Among those, we study MI-based diversity objectives alongside online adversarial IL (Li et al., 2017; Hausman et al., 2017; Peng et al., 2022) as they can learn infinite behaviors with continuous latent spaces. We address the limitations of MI in capturing latent factors specific to demonstrations.

Diversity in unsupervised RL Diverse behavior learning has been employed for exploration, pretraining, and generalization to novel environments (Laskin et al., 2021). Our work is closely related to competence-based methods (Sharma et al., 2019; Hansen et al., 2019; Park et al., 2022; 2023; 2024) that learn latent spaces to represent heterogeneous behaviors or ‘skills’. Works focus on different aspects of diversity, such as state coverage (Eysenbach et al., 2018; Park et al., 2022; Laskin et al., 2022; Mendonca et al., 2021), dynamics (Sharma et al., 2019), controllability (Park et al., 2023; 2024), etc. Our work adopts ideas of regularization (Park et al., 2022; 2023; 2024) for designing diversity objectives to address drawbacks in diverse behavior learning in LfD.

3 PROBLEM STATEMENT AND PRELIMINARIES

We consider an infinite horizon, discounted, and reward-free Markov Decision Process (MDP\R), $(S, A, P, \rho_0, \gamma)$, where S and A represent state and action spaces, $P\colon S \times A \times S \to \mathbb{R}$, the transition probabilities, $\rho_0\colon S \to \mathbb{R}$, the initial state distribution, and γ, the discount factor. An optimal expert policy π^{ξ} is governed by a continuous ground truth factor, $\omega \in \Omega$. The factor space, Ω, is split into **disjoint** train and test regions, $\mathrm{Tr}(\Omega)$ and $\mathrm{Te}(\Omega)$, respectively. Given a set of demonstrations, $\mathscr{D}$, consisting of trajectories $\tau_i^{\xi} = \{s_0, a_0, s_1, a_1, ...\}$, $a_t \sim \pi^{\xi}(\cdot|s_t, \omega_i)$, $s_{t+1} \sim P(\cdot|s_t, a_t)$, and $\omega_i \in \mathrm{Tr}(\Omega)$, we aim to learn a policy π that captures the expert behavior π^{ξ} over the entire factor space without access to ω_i or Ω.

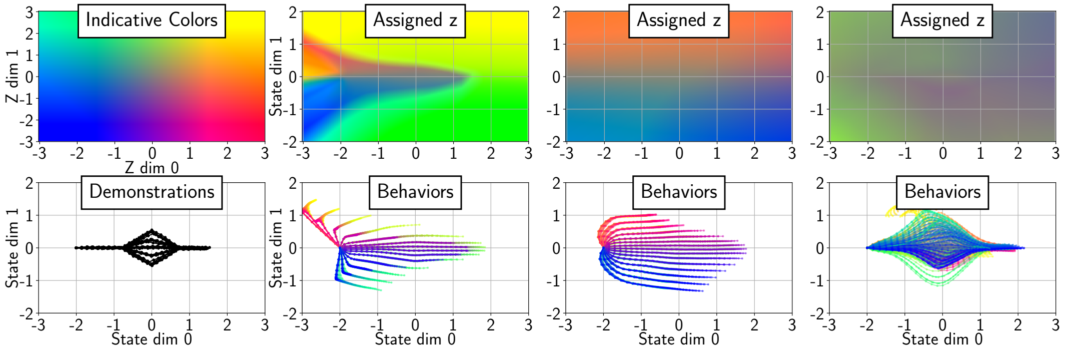
We ground our approach in InfoGAIL (Li et al., 2017), built upon Generative Adversarial Imitation Learning (GAIL) (Ho & Ermon, 2016) to imitate demonstrations: $J^{\mathrm{GAIL}} := E_{\pi^{\xi}}[\log D(s,a)] + E_{\pi}[\log(1 - D(s,a))]$, where D is a discriminator that distinguishes between the learned policy, π, and the expert policy, π^{ξ}. InfoGAIL additionally introduces a latent variable, $z \in Z$, to capture preferences underlying expert demonstrations. InfoGAIL optimizes MI by a variational lower bound (Barber & Agakov, 2004), $q(z|s,a)$. We refer to q as the *decoder* as it infers z from the state action pair. The InfoGAIL objective is $J^{\mathrm{InfoGAIL}} := J^{\mathrm{GAIL}} + \lambda_{\mathrm{I}} E_{z,\pi}[\log q(z|s,a)]$, where λ_{I} controls the diversity objective weight.

For formulating our diversity objective, we build on ideas from network distillation (Teh et al., 2017; Czarnecki et al., 2019; Chen et al., 2020). MSRD (Chen et al., 2020) learns task rewards from demonstrations over a finite set of distinct preferences, $\{\zeta_i\}$, by employing distillation with AIRL (Fu et al., 2017), a variant of GAIL that recovers a reward function, $r(s,a)$. MSRD then distills the reward functions for each preference, r_{ζ_i}, into a common reward function for the task, $\tilde{r}_0$. The distillation is done by formulating the reward for a preference as, $r_{\zeta_i}(s,a) := \tilde{r}_0(s,a) + \tilde{r}_{\zeta_i}(s,a)$. The preference-specific residual reward, $\tilde{r}_{\zeta_i}(s,a)$, is encouraged to be close to zero with the additional objective, $J^{\mathrm{MSRD}} := -E_{\zeta_i,\pi}[(\tilde{r}_{\zeta_i}(s,a))^2]$, to encourage the reward information common across demonstrations to be represented by the task reward function $\tilde{r}_0$.

4 NEED FOR REGULARIZATION

In this section, we show that prior diversity formulations from Li et al. (2017); Park et al. (2022) fail to produce novel, task-accomplishing behaviors, motivating the need for a new formulation.

No regularization: InfoGAIL’s diversity objective, MI, promotes diverse behaviors by rewarding the visitation of states associated with distinct latent vectors. In a 2D PointMaze domain with continuous state-action space (see Fig. 2), we show that an increased diversity objective’s weight,



a) Z, Demos **b) No regularization** **c) Lipschitz constraints** **d) GSD (Ours)**

Figure 2: Fig. 2a (Top): Map indicates the colors assigned to the 2D latent space. Fig. 2a (Bottom): The agent starts at (-2, 0), moves to (2, 0), passing through (-1, 0), (0, ω) and (1, 0) where $\omega \in [-1, 1]$. Figs. 2b, 2c, 2d: Latent vectors assigned to the state space, with a state-only decoder, alongside policy behaviors, under a high importance weight λ_I are visualized. Trajectories are colored according to conditioning latent vectors per Fig. 2a (Top). Baselines deviate from demonstrations arbitrarily (Fig. 2b, Bottom) or uniformly (Fig. 2c, Bottom), disregarding the goal. GSD (ours, Fig. 2d) discovers behaviors with novel latent variations (waypoints along $x = 0$) while reaching closer to the goal (2, 0)

.

λ_I, does not necessarily result in more diverse behaviors that accomplish the task. This result can be attributed to the decoder, q, a neural network (NN) that assigns latent vectors to state-action pairs, $\langle s, a\rangle$. Without regularization, the decoder, q, produces unconstrained latent assignments, with a high variety in smaller regions (see Fig. 2b, top, several distinct colors close to point (-2, 0)). This finding aligns with prior work (Choi et al., 2021; Park et al., 2022). Without regularization, related behaviors with close-by states can be mapped to unrelated far-away regions in the latent space without any meaningful structure. This behavior can cause insufficient (see Fig. 2b, bottom, several trajectories clump together along $y = 0$) or arbitrary (no pattern that governs deviation from demonstrations) behavior diversity.

Prior regularization methods produce misaligned behaviors: Prior works in unsupervised RL (Choi et al., 2021; Park et al., 2022) imposed Lipschitz constraints on the decoder, to enforce that for any two state-action pairs, $\langle s, a\rangle$, $\langle s', a'\rangle$, the assigned latent vectors (specifically the mean μ of the approximate posterior distribution $q(\cdot|s, a)$), differ by at most the Euclidean distance between the pairs, scaled by λ_C. Formally, $||\mu_{q(\cdot|s,a)} - \mu_{q(\cdot|s',a')}|| \leq \lambda_C \cdot ||s \oplus a - s' \oplus a'||$, where $\oplus$ denotes vector concatenation. The Lipschitz constraints ensure smooth latent vector assignments (see Fig.2c, top), which encourages behaviors to deviate from demonstrations uniformly over the state space. However, resulting behaviors do not necessarily proceed towards the goal (see Fig. 2c, bottom). Other diversity formulations focusing on controllability and temporal reachability (Park et al., 2023; 2024) will face similar issues if the auxiliary objectives are misaligned with behavior heterogeneity. We propose a general formulation that encourages behavior diversity along latent dimensions inferred from the demonstrations, without compromising task performance.

5 OUR METHOD: GUIDED STRATEGY DISCOVERY

We present our approach for achieving generalizable IL from diverse demonstrations.

5.1 ENCOURAGING f-RELEVANT DIVERSITY

First, we present a general approach for encouraging diversity selectively within state-action space regions indicated by high energy with respect to a scalar energy function, $f\colon S \times A \to [0, 1]$.

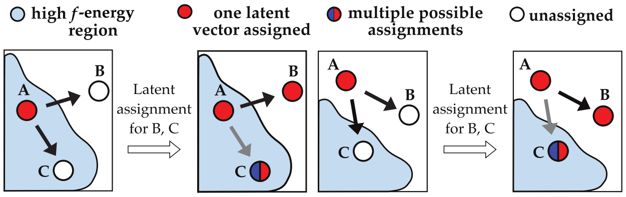
We design our approach by analysing transitions that occur during learning. Consider a scenario visualized in Fig. 3a, where an exploring agent is at a high f-energy state-action pair, $\langle s, a\rangle$, assigned a latent vector z, and it enters another pair, $\langle s', a'\rangle$. If a different vector, z' s.t. $z' \not\approx z$, were assigned to $\langle s', a'\rangle$, the diversity rewards, $\log q(z|s, a)$, $\log q(z'|s', a')$, would encourage behaviors $\pi(\cdot|\cdot, z)$, $\pi(\cdot|\cdot, z')$, to visit $\langle s, a\rangle$ and $\langle s', a'\rangle$ respectively. The behavior, $\pi(\cdot|\cdot, z')$, would be desirable if $\langle s', a'\rangle$ has high f-energy, as it would visit a high energy pair different from $\langle s, a\rangle$, increasing coverage of high energy regions. On the other hand, if $\langle s', a'\rangle$ were a low f-energy pair, the behavior $\pi(\cdot|\cdot, z')$ visiting a low energy pair would be undesirable. In this case, the assignment for $\langle s', a'\rangle$

could be constrained close to z, which would remove the incentive for a behavior distinct from $\pi(\cdot|\cdot, z)$ to specifically visit $\langle s', a' \rangle$.

Selectively allowing distinct latent vector assignments only in high-energy regions encourages behaviors that target these regions, thereby promoting diversity only in high-energy regions. Constraint shown in Eq. 1 formalizes this intuition: For a transition from $\langle s, a \rangle$ to $\langle s', a' \rangle$, the latter's latent vector can be far from the former's by at most the Euclidean distance between the two pairs, scaled by f-energy of the latter and a factor λ_C.

$$||\mu_{q(\cdot|s,a)} - \mu_{q(\cdot|s',a')}|| \leq \lambda_C \cdot ||s \oplus a - s' \oplus a'|| \cdot f(s', a') \tag{1}$$

The proposed constraint (Eq. 1) disregards the energy of the starting state-action pair, $f(s, a)$. The constraint enforces the same latent assignment for pairs in a low→low energy transition and allows different latent assignments for pairs in a low→high energy transition as visualized in Fig. 3b. The enforcement can lead to connected low energy regions being assigned the same latent vector which is different from that of reachable high energy regions. Distinct latent vectors for low energy regions can result in behaviors visiting those low-energy regions, which is **not** desirable.


high f-energy region
one latent vector assigned
multiple possible assignments
unassigned
A
B
C
Latent assignment for B, C
A
B
C
A
B
C
Latent assignment for B, C
A
B
C

(a) state-action pair A has *high f*-energy **(b) state-action pair A has *low f*-energy**

Figure 3: This figure shows a visualization of state transitions that an agent at state-action pair, A, may undergo and the effect of our constraint in Eq. 1. The light blue region indicates high f-energy and white state-action pairs do not yet have assigned latent vectors. The lightness of the arrows indicates the slackness of our constraint (Eq. 1) from scaling with f-energy on the right-hand side.

We rectify our constraints to prevent latent assignments for low energy regions, by enforcing them only with transitions with high energy starting pairs through scaling the constraint with f-energy of the starting pair, as shown in Eq. 2. Thus, when the starting pair is of low energy, the constraint implemented with a Lagrange multiplier is less effective due to a smaller violation.

The modified constraints encourage the decoder to effectively use the latent space to solely represent high-energy regions. We refer to the resulting objective as f-relevant diversity.

$$f(s, a) \cdot \left[||\mu_{q(\cdot|s,a)} - \mu_{q(\cdot|s',a')}||\right] \leq f(s, a) \cdot [\lambda_C \cdot ||s \oplus a - s' \oplus a'|| \cdot f(s', a')] \tag{2}$$

5.2 INFERRING A TASK-RELEVANCE MEASURE FROM DEMONSTRATIONS

Diverse task-accomplishing behaviors can be learned if an appropriate energy function f can be derived from demonstrations to instantiate our f-relevant diversity objective. f should indicate regions where agent occupation is favorable for the task while also capturing demonstrators' heterogeneity.

We categorize regions that diverse task-accomplishing demonstrations occupy into two types: (I) regions occupied by all experts, and (II) subspaces where different experts occupy distinct smaller regions. We propose that capturing these two types of regions in the energy function provides us with an objective that encourages task accomplishment and generalization over preferences.

While the discriminator, D, could be used to model f, it would capture type II subspace insufficiently. D is trained to capture the union of demonstration occupancies, covering all type I regions, but only certain regions in the type II subspace that correspond to training demonstrations. f modeled in this way would limit behavior discovery beyond the training dataset.

We employ the distillation of demonstration specific occupancy measures into a common measure, to model f and capture type II subspaces completely. For each training demonstration, we infer the latent vector, z, using the decoder, q. Our latent-conditioned discriminator, $D(s, a, z)$, is modeled as a linear combination of a latent-independent measure, i.e., our desired energy function, $f(s, a)$, and a dependent term, $g(s, a, z)$, in the logit space as shown in Eq. 3, where $g\colon S \times A \times Z \to [0, 1]$, σ is the logistic function, λ_S, a scaling constant, and b, a learnable bias. λ_S and b are introduced to transform the sum of bounded measures and enable D to capture most of the probability range $[0, 1]$. The discriminator is trained with an additional loss term, as shown in Eq. 4, to minimize the residual, g, to only capture preference-specific occupancy.

$$D(s, a, z) = \sigma(\lambda_S \cdot [f(s, a) + g(s, a, z)] + b) \tag{3}$$

Algorithm 1 Guided Strategy Discovery

```
Input: $\mathscr{D} = \{\tau_i^\xi\}$
Output: $\pi$
1: Initialize policy, $\pi$, task relevance, $f$, preference-specific residual, $g$, decoder, $q$, with parameters
   $\theta_\pi$, $\theta_f$, $\theta_g$, $\theta_q$, Lagrange multiplier, $\lambda$, bias, $b$, and learning rates $\eta_\pi$, $\eta_f$, $\eta_g$, $\eta_q$, $\eta_\lambda$, $\eta_b$
2: for $i \in \{0, 1, 2, ...\}$ epoch do
3:     Sample $z^\pi$ from prior, $\tau^\pi$ using policy $\pi(\cdot|\cdot, z^\pi)$; $\tau^\xi$ from $\mathscr{D}$, infer $z^\xi$ using decoder $q$
4:     Define objective for functions $f$, $g$ and bias $b$: $\forall \chi := (z, s, a, s', a') \in \tau$
       $D(\chi) = \sigma(\lambda_S \cdot [f(s,a) + g(s,a,z)] + b)$
       $J^{\mathrm{I}} \leftarrow \mathbb{E}_{\tau^\xi}[\log D(\chi^\pi)] + E_{\tau^\pi}[\log(1 - D(\chi^\xi))] - \mathbb{E}_{\tau^\pi}[(g(s,a,z))^2] - E_{\tau^\xi}[(g(s,a,z))^2]$
5:     Update $f$, $g$, $b$ using gradients: $[\theta_f, \theta_g, b] := [\theta_f, \theta_g, b] + [\eta_f \nabla_{\theta_f} J^{\mathrm{I}}, \eta_g \nabla_{\theta_g} J^{\mathrm{I}}, \eta_b \nabla_b J^{\mathrm{I}}]$
6:     Define objective for decoder $q$: $\forall \chi := (z, s, a, s', a') \in \tau$
       $\delta(\chi) \leftarrow \big[\lambda_C \cdot ||s \oplus a - s' \oplus a'|| \cdot f(s', a') - ||\mu_{q(\cdot|s,a)} - \mu_{q(\cdot|s',a')}||\big] \cdot f(s,a)$
       $q_L(\chi, z) \leftarrow \mathcal{N}(z|\mu_{q(\cdot|s,a)}, \Sigma_{q(\cdot|s,a)})$
       $J^{\mathrm{E}} \leftarrow \mathbb{E}_{\tau^\pi}[\log q_L(\chi, z) + \lambda \cdot \min(\delta(\chi), \epsilon)]$
7:     Update decoder $q$ and $\lambda$ using gradients: $[\theta_q, \lambda] := [\theta_q, \lambda] + [\eta_q \nabla_{\theta_q} J^{\mathrm{E}}, -\eta_\lambda \nabla_\lambda J^{\mathrm{E}}]$
8:     Update policy $\pi$ with RL using rewards: $r(s, a, s', z) = \log(D(\chi)) + \lambda_{\mathrm{I}} \cdot \log q_L(\chi, z)$
9: end for
```

$$J^{\mathrm{R}} := -E_{z,\pi}[(g(s,a,z))^2] \tag{4}$$

The objective, J^{R}, encourages f to completely capture both type I and II regions. Type I regions are captured by f, as g is driven to zero where occupancy is common across demonstrations and latent-independent. g is encouraged to be close to zero even in regions with demonstration-specific occupancy, causing it to capture minimal possible information and distilling the rest into f. We posit that f indicates entire subspaces where occupancy is demonstration dependent, i.e., type II subspaces, while g indicates regions in these subspaces specific to each demonstration. We call our procedure for deriving f conditioned distillation (ConDist), due to its use of latent conditioning and distillation, similar to prior reward distillation frameworks (Chen et al., 2020).

Algorithm Overview: We combine the two components to propose our approach Guided Strategy Discovery (GSD), with steps outlined in Algorithm 1. In each epoch, we sample behaviors using the policy conditioned on latent vectors from the prior (Line 3). Latent vectors for demonstrations are inferred using the decoder (Line 3). We compute the imitation and distillation objectives (Line 4) and update the energy function, f, residual, g, and bias, b, using gradients (Line 5). We compute the variational lower bound along with proposed constraints (Line 6) and update the decoder, q, and the Lagrange multiplier, λ, using gradients (Line 7). Finally we update the policy with rewards from the discriminator and decoder (Line 8).

We posit that f-relevant diversity and conditioned distillation are synergistic. An accurate f function representing demonstrations can guide latent assignments and associated behaviors to generalize beyond demonstrations. A latent space representing diverse demonstrations can help distillation capture regions beyond demonstrations that generalize latent preferences. Fig. 2d (bottom) shows that with GSD, the learned behaviors in 2D PointMaze capture novel latent variations by passing through waypoints along $x = 0$ while reaching closer to the goal of (2, 0) better than baselines in Figs. 2b, 2c. In addition, GSD produces a higher fraction of goal-reaching trajectories despite a low weight for the imitation objective and a weaker incentive to match the expert.

6 EVALUATION

We present empirical evaluation to answer the following research questions:

1. How do various methods perform in generalization to behaviors with novel latent factors while maintaining task performance? (Sec. 6.1)
2. How do various methods structure behaviors in the latent space? (Sec. 6.2)
3. How do various methods perform in learning diverse task-accomplishing behaviors from real-world human demonstrations? (Sec. 6.3)

Domains: For Sec. 6.1, 6.2, we use HalfCheetah (Wawrzyński, 2009), FetchPickPlace (Plappert et al., 2018) and DriveLaneshift (Leurent, 2018) as they provide well-defined tasks with distinct one-dimensional (1D) factors. We script expert policies based on these factors. In HalfCheetah, the robot runs at various speeds; in FetchPickPlace, the arm places the object at different table locations; and in DriveLaneshift, the ego-car changes lanes to overtake at varying headway distances. 1D factors help avoid multiple sources of heterogeneity allowing careful examination of learned policies.

Methods: We compare the following methods. We only consider InfoGAIL (Li et al., 2017) as our base method as other IL methods learn finite sets of policies with less scope for novel behaviors.

- IG: InfoGAIL (Li et al., 2017) with a continuous two dimensional latent variable.
- IG+Lipz: IG with Lipschitz constraints for decoder q to investigate the uniform diversity.
- IG+Con: IG with a conditioned discriminator $D(s, a, z)$ structure to investigate the effect of conditioning the discriminator.
- IG+ConDist: IG with our proposed conditioned distillation to investigate the effect of extraction of a task-relevance measure (Eqs. 4, 3).
- IG+ConDist+Lipz: IG+ConDist with Lipschitz to investigate the uniform diversity formulation alongside conditioned distillation.
- **GSD (Ours)**: IG+ConDist with our proposed task-relevant diversity formulation (Eq. 2).

6.1 QUANTITATIVE EVALUATION

We investigate whether learned behaviors can represent factor values in the disjoint test region $\mathrm{Te}(\Omega)$, after training on demonstrations, $\mathscr{D}$, from the train region, $\mathrm{Tr}(\Omega)$ (see Sec. 3). We consider factors that are directly measurable from trajectories (only for the sake of evaluation) to assess recovery performance, i.e., how well the learned latent space can represent expert behavior by comparing desired and measured factor values. When diverse expert behaviors form distinct modes, this framework checks if continuous factors underlying these modes can be accurately identified and generalized.

Splits: We divide the bounded 1D factor range into five consecutive equal-sized intervals:

- **Interpolation**: The first, third, and fifth intervals represent the train region, and the second and fourth are the test region. The split allows us to evaluate the ability to interpolate behaviors to two factor space intervals while providing three non-consecutive intervals to represent the factor.
- **Extrapolation**: The second and fourth intervals represent the train region, while the first and fifth intervals are the test region. We choose two non-consecutive intervals for the train region to have a sparse dataset while providing enough diversity to represent the factor.

These splits evaluate how well the latent space captures factors to interpolate and extrapolate behaviors. We use five demonstrations per interval (details in Appendix B).

Metrics: We search for desired behaviors using $K \in \{10, 20, 30, 40, 50\}$ latent vector samples from the prior $p_z(\cdot)$, where K represents the test time search sample-complexity, varied to investigate how well we generate desired behaviors from limited samples. We roll out policies conditioned on the sampled vectors, measure the factors of the sampled behaviors, and compute the least mean absolute error (MAE) between the desired and the K measured values, averaging over 1500/K rounds. We refer to this metric as the recovery error. We consider the midpoints of test intervals as the set of desired values. We report average and worst errors over desired values in the test region, providing estimates of closeness between desired values and closest available behavior's factor, on average and worst case. We report the mean and standard errors over five train seeds in Fig. 4. We also evaluate task performance by averaging environment returns over 1500 latent samples. We show recovery and task performance tradeoff in Fig. 5. Exact numbers are provided in Appendix F.

Lipschitz constraints: For HalfCheetah (Fig. 4, top row), IG+Lipz and IG+ConDist+Lipz have worse recovery errors compared to IG and IG+ConDist, indicated by the dark blue curve above magenta, and green above black, respectively. For DriveLaneshift (middle row), IG+Lipz and IG+ConDist+Lipz exhibit the same trend against IG and IG+ConDist: for interpolation, errors are improved shown by dark blue falling below magenta and green below black; and for extrapolation, the errors are worsened. For FetchPickPlace (bottom row), IG+Lipz and IG+ConDist+Lipz improve over IG and IG+ConDist indicated by dark blue and green falling consistently below magenta and black respectively. Lipschitz constraints seem to be benefiting FetchPickPlace alone, which might be due to "uniform diversity" aligning with object-position factors, that is absent in other domains. This supports our hypothesis that relevant factors for diversity must be inferred from demonstrations to benefit all domains.

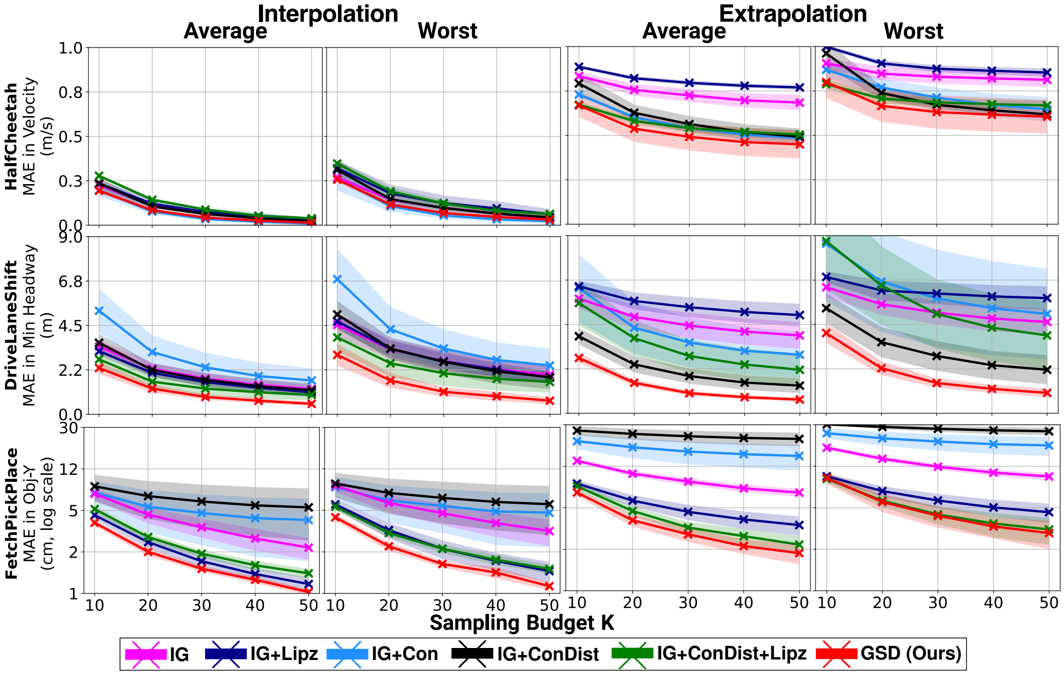


Figure 4: The figure shows average and worst-case recovery errors over the three domains and two factor space splits corresponding to interpolative and extrapolative generalization. Shaded regions are standard errors over five train seeds. GSD outperforms baselines in recovery of unseen latent factors across most domains and splits.

Conditioning: For HalfCheetah (top row), IG+Con improves errors compared to IG, indicated by the light blue curve below magenta. However, DriveLaneshift and FetchPickPlace, IG+Con seems to worsen performance, with light blue largely above magenta in the bottom two rows. Worsened errors may be a result of conditioning on a latent variable capturing arbitrary factors.

Conditioned Distillation: For HalfCheetah (top row), IG+ConDist and IG+ConDist+Lipz improve over IG and IG+Lipz for extrapolation, indicated by black and green curves below magenta and dark blue respectively. They remain on par for interpolation. For DriveLaneshift as well (middle row), IG+ConDist and IG+ConDist+Lipz improve over IG and IG+Lipz for extrapolation ($K \geq 30$) and remain on par for/slightly improve interpolation. For FetchPickPlace (bottom row), the trends are interesting. IG+ConDist worsens errors over IG for interpolation and extrapolation, indicated by black above magenta. However, with Lipz's addition, IG+ConDist+Lipz tends close to IG+Lipz for interpolation and outperforms it for extrapolation. The patterns firstly suggest that conditioned distillation can improve extrapolation performance. In addition, for FetchPickPlace where Lipschitz constraints are particularly effective, conditioned distillation can further improve extrapolation.

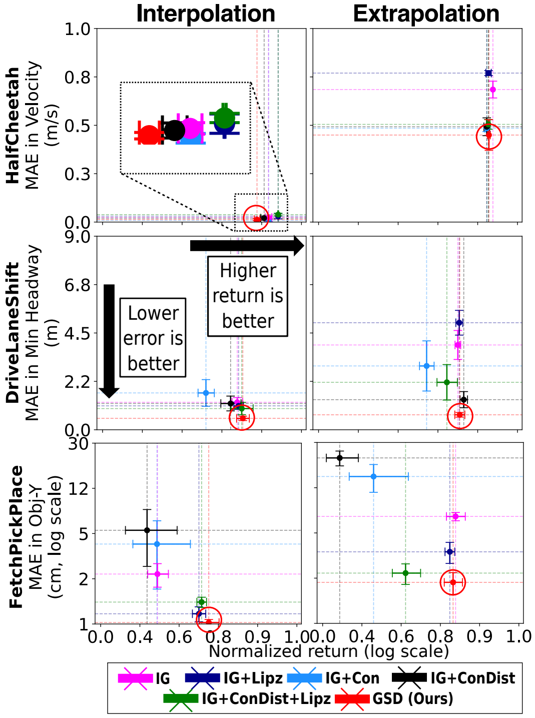


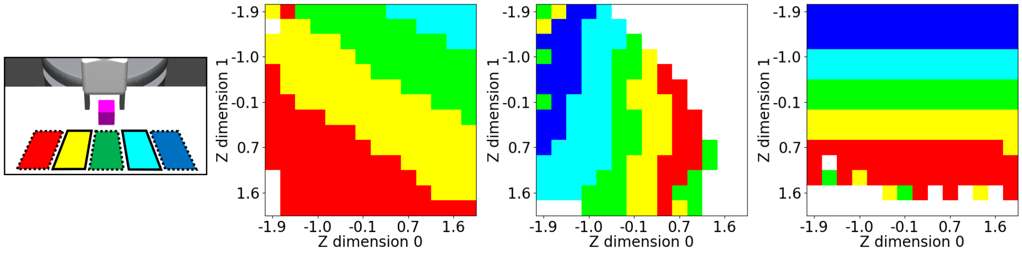
Figure 5: The figure shows the tradeoff between task and recovery performance for three domains and two splits. Error bars show standard errors over five seeds. High returns (x-axis) and low errors (y-axis) are better. GSD (circled in red) improves recovery while retaining or improving task performance across most domains and splits.

Task-relevant Diversity: GSD improves recovery over other approaches across most domains and splits, shown by the red curve below others in all plots, except for interpolation with HalfCheetah (top row, first two columns). In HalfCheetah (top row), the close performance across methods may be attributed to wide differences in gait styles across velocities that are challenging to interpolate or extrapolate. In DriveLaneshift (middle row), GSD reduces recovery error considerably over other approaches. In FetchPickPlace, GSD is closely matched by IG+Lipz or IG+ConDist+Lipz as Lipschitz constraints already capture relevant factors. Nevertheless, GSD can further improve recovery.

Tradeoff between Task and Recovery Performance: Across all domains, GSD either matches or improves average normalized returns over the latent prior, as indicated in Fig. 5 by the red cross generally being aligned with or positioned further right than others in all domain-split combinations but one. These results demonstrate the effectiveness of GSD's task-relevant diversity formulation in learning behaviors that reduce recovery error while maintaining task performance.

6.2 QUALITATIVE EVALUATION

We visualize the nature of the learned latent spaces for extrapolation in FetchPickPlace in Fig. 6. IG+ConDist learns a large behavior set for placing the object in the red test region (indicated by red cells in Fig. 6b), but ignores the dark-blue test region. While IG+ConDist+Lipz learns behaviors for all regions, it learns several that fail to place the object quickly enough (indicated by white cells in Fig. 6c). GSD learns behaviors that achieve the task (few white cells in Fig. 6d) while representing all place locations equally in proximity to each other (roughly equal number of cells across colors nearby each other). GSD exhibits potential for improving the accountability of policy learning by enabling well structured latent spaces.



(a) Fetch (b) IG+ConDist (c) IG+ConDist+Lipz (d) GSD (Ours)

Figure 6: Fig.6a The figure shows FetchPickPlace with possible object place locations visualized with colors. Solid and dotted boundaries indicate train and test regions respectively. Figs.6b,6c,6d: Policy behaviors are depicted in the 2D latent space through colors for resulting place-locations shown in 6a. White regions indicate behaviors that fail to place the object within the regions of interest or do so with low task reward. Behaviors with IG+ConDist (6b), IG+ConDist+Lipz (6c) either represent the relevant regions disproportionately or fail to accomplish the task. **Behaviors with GSD (6d) accomplish the task (low presence of white cells) and represent all regions well (roughly equal number of cells across colors)**.

6.3 EVALUATION WITH REAL-WORLD HUMAN DEMONSTRATIONS

We further evaluate our approach to test scalability to complex tasks with human demonstrations in a Table Tennis (TT) domain. TT represents a dynamic domain that requires precise robot motion and fast reaction times while acting on noisy observations. Our physical setup consists of a Barrett WAM Arm mounted to the ceiling in front of a TT table, and a racquet attached as the arm's effector. Balls are launched using a Butterfly Amicus launcher at a fixed orientation and velocity with some noise. Balls are detected and tracked using a YOLO object detector and a Kalman Filter. An

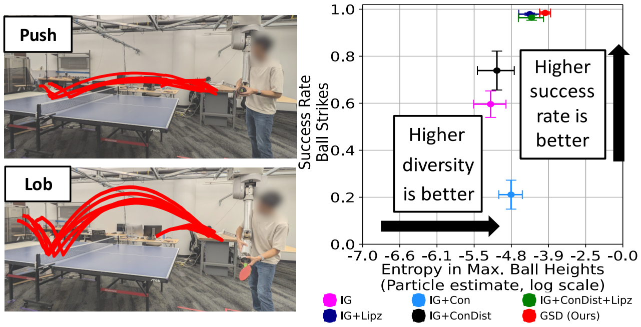


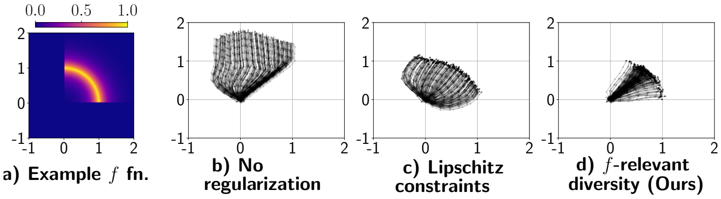
Figure 7: **Left:** The images visualize ball trajectories achieved by an expert kinesthetically demonstrating two types of strokes. **Right:** The figure shows the tradeoff between ball striking success rate and diversity in ball heights achieved. GSD outperforms baselines for both metrics.

expert provides kinesthetic demonstrations of push and lob strokes. We recreate the setup in simulation with PyBullet for behavior learning. Ball initialization and observation noise levels in the simulation match real data. Complete details are in Appendix D.

While multiple continuous factors may exist underlying TT stroke styles, we evaluate generalization for maximum ball height, which we assume to be one of the underlying continuous factors. We evaluate various methods in simulation for achieving high diversity in ball heights. We compute entropy in ball height values using particle estimates (Singh et al., 2003), after disregarding unsuccessful trials that fail to strike the ball over the table. We report the success rate traded off with diversity in ball heights in Fig. 7 (right). Our method GSD outperforms all baselines in both measures of success rate and entropy.

6.4 GENERALITY OF f-RELEVANT DIVERSITY

Our f-relevant diversity formulation discussed in Sec. 5.1 is designed to encourage behavior diversity with respect to any defined energy measure f. We briefly demonstrate the generality of our formulation in the simple 2D PointMaze domain with a user-defined energy function as shown in Fig. 8. Our formulation has a potential application in diverse solution discovery (Kumar et al., 2020; Osa et al., 2022), where a bounded form of the estimated Q function can be used as f to encourage diversity in high value regions.



a) Example f fn. **b) No regularization** **c) Lipschitz constraints** **d) f-relevant diversity (Ours)**

Figure 8: The figure visualizes behaviors in 2D PointMaze learned with a predefined energy function f shown in 8a. f is used to specify rewards in 8b-d and additionally formulate our diversity objective in 8d. With no regularization (8b) or Lipschitz constraints (8c), trajectories visit low f-energy regions of the state-space. However, with f-relevant diversity (ours, 8d), a larger portion of trajectories cover diverse high energy states.

7 CONCLUSION, LIMITATIONS AND FUTURE WORK

We study the problem of generalization from diverse demonstrations over underlying latent factors. We investigate the shortcomings of prior MI-based methods and propose a novel diversity formulation. Our empirical evaluation shows that our approach GSD improves the recovery of factors over the next best baseline (for K=50) by 18.3% and 24.6% for interpolation and extrapolation respectively while retaining task performance in three domains with synthetic demonstrations. Our qualitative analysis shows the potential of our approach in making learned policies accountable. Lastly, our experiments with real-world human demonstrations shows that our framework can capture a diverse range of task-accomplishing behaviors in a challenging domain requiring quick response times.

Limitations: Our experiments consider demonstrations with one-dimensional latent factors. We aim to test the scalability of our approach to higher dimensional factors. Our evaluation with human demonstrations is limited to quantitative metrics. We aim to conduct user studies to subjectively evaluate behaviors in human robot interaction settings. Our evaluation is further limited to simulated domains. We aim to explore the efficacy of our diversity formulation for learning novel behaviors in physical robot systems with improved data-sample efficiency. We further aim to explore the theoretical implications of our formulation and its alignment with the imitation objective.

REPRODUCIBILITY STATEMENT

Implementation details, hyperparameters and evaluation procedures are detailed in Appendices C, D.2. Data generation and collection is detailed in Appendices A, B, D.1.

REFERENCES

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the 21st International Conference on Machine learning*, pp. 1, 2004.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

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. *Advances in Neural Information Processing Systems*, 30, 2017.

David Barber and Felix Agakov. The IM algorithm: a variational approach to Information Maximization. *Advances in Neural Information Processing Systems*, 16(320):201, 2004.

Carolin Benjamins, Theresa Eimer, Frederik Schubert, Aditya Mohan, André Biedenkapp, Bodo Rosenhahn, Frank Hutter, and Marius Lindauer. Contextualize me–the case for context in reinforcement learning. *arXiv preprint arXiv:2202.04500*, 2022.

Marcel Binz and Dominik M. Endres. Where do human heuristics come from? *ArXiv*, abs/1902.07580, 2019. URL `https://api.semanticscholar.org/CorpusID:67769766`.

Gisela Böhm and Hans-Rüdiger Pfister. How people explain their own and others' behavior: a theory of lay causal explanations. *Frontiers in psychology*, 6:109763, 2015.

Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. arxiv. *arXiv preprint arXiv:1606.01540*, 10, 2016.

Zhangjie Cao, Yilun Hao, Mengxi Li, and Dorsa Sadigh. Learning feasibility to imitate demonstrators with different dynamics. *arXiv preprint arXiv:2110.15142*, 2021.

Letian Chen, Rohan Paleja, Muyleng Ghuy, and Matthew Gombolay. Joint goal and strategy inference across heterogeneous demonstrators via reward network distillation. In *Proceedings of the 2020 ACM/IEEE International Conference on human-robot interaction*, pp. 659–668, 2020.

Letian Chen, Rohan Paleja, and Matthew Gombolay. Learning from suboptimal demonstration via self-supervised reward regression. In *Conference on robot learning*, pp. 1262–1277. PMLR, 2021.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

Jongwook Choi, Archit Sharma, Honglak Lee, Sergey Levine, and Shixiang Shane Gu. Variational empowerment as representation learning for goal-based reinforcement learning. *arXiv preprint arXiv:2106.01404*, 2021.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*, 30, 2017.

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *International Conference on Machine Learning*, pp. 1282–1289. PMLR, 2019.

Antonio Coronato, Muddasar Naeem, Giuseppe De Pietro, and Giovanni Paragliola. Reinforcement learning for intelligent healthcare applications: A survey. *Artificial intelligence in medicine*, 109: 101964, 2020.

Wojciech M Czarnecki, Razvan Pascanu, Simon Osindero, Siddhant Jayakumar, Grzegorz Swirszcz, and Max Jaderberg. Distilling policy distillation. In *The 22nd International Conference on artificial intelligence and statistics*, pp. 1331–1340. PMLR, 2019.

Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R McKee, Joel Z Leibo, Kate Larson, and Thore Graepel. Open problems in cooperative ai. *arXiv preprint arXiv:2012.08630*, 2020.

Anca Dragan and Siddhartha Srinivasa. Generating legible motion. *Frontiers in psychology*, 2013.

Danny Driess, Jung-Su Ha, and Marc Toussaint. Deep visual reasoning: Learning to predict action sequences for task and motion planning from an initial scene image. *arXiv preprint arXiv:2006.05398*, 2020.

Yan Duan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. *Advances in Neural Information Processing Systems*, 30, 2017.

Marco Ewerton, Guilherme Maeda, Gerrit Kollegger, Josef Wiemeyer, and Jan Peters. Incremental imitation learning of context-dependent motor skills. In *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp. 351–358. IEEE, 2016.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2018.

Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *arXiv preprint arXiv:1611.03852*, 2016.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on machine learning*, pp. 1126–1135. PMLR, 2017a.

Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In *Conference on Robot Learning*, pp. 357–368. PMLR, 2017b.

Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on Robot Learning*, pp. 158–168. PMLR, 2022.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.

Kanishk Gandhi, Siddharth Karamcheti, Madeline Liao, and Dorsa Sadigh. Eliciting compatible demonstrations for multi-human imitation learning. In *Conference on Robot Learning*, pp. 1981–1991. PMLR, 2023.

Yapeng Gao, Jonas Tebbe, and Andreas Zell. Optimal stroke learning with policy gradient approach for robotic table tennis. *Applied Intelligence*, 53(11):13309–13322, 2023.

Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods. In *Conference on Robot Learning*, pp. 1259–1277. PMLR, 2020.

Diego Gomez, Michael Bowling, and Marlos C Machado. Proper laplacian representation learning. *arXiv preprint arXiv:2310.10833*, 2023.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in Neural Information Processing Systems*, 30, 2017.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International Conference on machine learning*, pp. 1352–1361. PMLR, 2017.

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.

Steven Hansen, Will Dabney, Andre Barreto, Tom Van de Wiele, David Warde-Farley, and Volodymyr Mnih. Fast task inference with variational intrinsic successor features. *arXiv preprint arXiv:1906.05030*, 2019.

Mahta HassanPour Zonoozi and Vahid Seydi. A survey on adversarial domain adaptation. *Neural Processing Letters*, 55(3):2429–2469, 2023.

Karol Hausman, Yevgen Chebotar, Stefan Schaal, Gaurav Sukhatme, and Joseph J Lim. Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets. *Advances in Neural Information Processing Systems*, 30, 2017.

Alexander Herzog, Kanishka Rao, Karol Hausman, Yao Lu, Paul Wohlhart, Mengyuan Yan, Jessica Lin, Montserrat Gonzalez Arenas, Ted Xiao, Daniel Kappler, et al. Deep rl at scale: Sorting waste in office buildings with a fleet of mobile manipulators. *arXiv preprint arXiv:2305.03270*, 2023.

Jonathan Ho and Stefano Ermon. Generative Adversarial Imitation Learning. *Advances in Neural Information Processing Systems*, 29, 2016.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Maxence Hussonnois, Thommen George Karimpanal, and Santu Rana. Controlled diversity with preference: Towards learning a diverse set of desired skills. *arXiv preprint arXiv:2303.04592*, 2023.

Sagar Imambi, Kolla Bhanu Prakash, and GR Kanagachidambaresan. Pytorch. *Programming with TensorFlow: Solution for Edge Computing Applications*, pp. 87–104, 2021.

Rohit Jena, Changliu Liu, and Katia Sycara. Augmenting gail with bc for sample efficient imitation learning. In *Conference on Robot Learning*, pp. 80–90. PMLR, 2021.

Xiaogang Jia, Denis Blessing, Xinkai Jiang, Moritz Reuss, Atalay Donat, Rudolf Lioutikov, and Gerhard Neumann. Towards diverse behaviors: A benchmark for imitation learning with human demonstrations. *arXiv preprint arXiv:2402.14606*, 2024.

Liyiming Ke, Sanjiban Choudhury, Matt Barnes, Wen Sun, Gilwoo Lee, and Siddhartha Srinivasa. Imitation learning as f-divergence minimization. In *Algorithmic Foundations of Robotics XIV: Proceedings of the 14th Workshop on the Algorithmic Foundations of Robotics 14*, pp. 313–329. Springer, 2021.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of zero-shot generalisation in deep reinforcement learning. *Journal of Artificial Intelligence Research*, 76:201–264, 2023.

W Bradley Knox and Peter Stone. Tamer: Training an agent manually via evaluative reinforcement. In *7th IEEE International Conference on development and learning*, pp. 292–297. IEEE, 2008.

W Bradley Knox and Peter Stone. Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the 5th International Conference on Knowledge Capture*, pp. 9–16, 2009.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

Saurabh Kumar, Aviral Kumar, Sergey Levine, and Chelsea Finn. One solution is not all you need: Few-shot extrapolation via structured maxent rl. *Advances in Neural Information Processing Systems*, 33:8198–8210, 2020.

Michael Laskin, Denis Yarats, Hao Liu, Kimin Lee, Albert Zhan, Kevin Lu, Catherine Cang, Lerrel Pinto, and Pieter Abbeel. Urlb: Unsupervised reinforcement learning benchmark. *arXiv preprint arXiv:2110.15191*, 2021.

Michael Laskin, Hao Liu, Xue Bin Peng, Denis Yarats, Aravind Rajeswaran, and Pieter Abbeel. Cic: Contrastive intrinsic control for unsupervised skill discovery. *arXiv preprint arXiv:2202.00161*, 2022.

Edouard Leurent. An environment for autonomous driving decision-making. `https://github.com/eleurent/highway-env`, 2018.

Yunzhu Li, Jiaming Song, and Stefano Ermon. Infogail: Interpretable imitation learning from visual demonstrations. *Advances in Neural Information Processing Systems*, 30, 2017.

Minghuan Liu, Tairan He, Minkai Xu, and Weinan Zhang. Energy-based imitation learning. *arXiv preprint arXiv:2004.09395*, 2020.

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*, pp. 1113–1132. PMLR, 2020.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning*, pp. 1678–1690. PMLR, 2022.

Russell Mendonca, Oleh Rybkin, Kostas Daniilidis, Danijar Hafner, and Deepak Pathak. Discovering and achieving goals via world models. *Advances in Neural Information Processing Systems*, 34:24379–24391, 2021.

Josh Merel, Leonard Hasenclever, Alexandre Galashov, Arun Ahuja, Vu Pham, Greg Wayne, Yee Whye Teh, and Nicolas Heess. Neural probabilistic motor primitives for humanoid control. *arXiv preprint arXiv:1811.11711*, 2018.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Eduardo F Morales and Claude Sammut. Learning to fly by combining reinforcement learning with behavioural cloning. In *Proceedings of the 21st International Conference on Machine learning*, pp. 76, 2004.

Suraj Nair, Eric Mitchell, Kevin Chen, Silvio Savarese, Chelsea Finn, et al. Learning language-conditioned robot behavior from offline data and crowd-sourced annotation. In *Conference on Robot Learning*, pp. 1303–1315. PMLR, 2022.

Tianwei Ni, Harshit Sikchi, Yufei Wang, Tejus Gupta, Lisa Lee, and Ben Eysenbach. f-irl: Inverse reinforcement learning via state marginal matching. In *Conference on Robot Learning*, pp. 529–551. PMLR, 2021.

OpenAI. o1. `https://openai.com/index/learning-to-reason-with-llms/`, 2024. Learning to reason with Large Language Models.

Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2): 1–179, 2018.

Takayuki Osa, Voot Tangkaratt, and Masashi Sugiyama. Discovering diverse solutions in deep reinforcement learning by maximizing state–action-based mutual information. *Neural Networks*, 152:90–104, 2022.

Charles Packer, Katelyn Gao, Jernej Kos, Philipp Krähenbühl, Vladlen Koltun, and Dawn Song. Assessing generalization in deep reinforcement learning. *arXiv preprint arXiv:1810.12282*, 2018.

Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

Rohan Paleja, Andrew Silva, Letian Chen, and Matthew Gombolay. Interpretable and personalized apprenticeship scheduling: Learning interpretable scheduling policies from heterogeneous user demonstrations. *Advances in Neural Information Processing Systems*, 33:6417–6428, 2020.

Seohong Park, Jongwook Choi, Jaekyeom Kim, Honglak Lee, and Gunhee Kim. Lipschitz-constrained unsupervised skill discovery. In *International Conference on Learning Representations*, 2022.

Seohong Park, Kimin Lee, Youngwoon Lee, and Pieter Abbeel. Controllability-aware unsupervised skill discovery. In *International Conference on Machine Learning*, pp. 27225–27245. PMLR, 2023.

Seohong Park, Oleh Rybkin, and Sergey Levine. METRA: Scalable unsupervised RL with metric-aware abstraction. In *The Twelfth International Conference on Learning Representations*, 2024.

Tim Pearce, Tabish Rashid, Anssi Kanervisto, Dave Bignell, Mingfei Sun, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Ida Momennejad, Katja Hofmann, et al. Imitating human behaviour with diffusion models. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*, 2023.

Jian-Wei Peng, Min-Chun Hu, and Wei-Ta Chu. An imitation learning framework for generating multi-modal trajectories from unstructured demonstrations. *Neurocomputing*, 500:712–723, 2022.

Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018.

Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science advances*, 4(7):eaap7885, 2018.

Yiwen Qiu, Jialong Wu, Zhangjie Cao, and Mingsheng Long. Out-of-dynamics imitation learning from multimodal demonstrations. In *Conference on Robot Learning*, pp. 1071–1080. PMLR, 2023.

Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *IJCAI*, volume 7, pp. 2586–2591, 2007.

Allen Z Ren, Justin Lidard, Lars L Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. *arXiv preprint arXiv:2409.00588*, 2024.

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*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

Stuart Russell. *Human compatible: Artificial intelligence and the problem of control*. Penguin, 2019.

Penelope M Sanderson. The human planning and scheduling role in advanced manufacturing systems: An emerging human factors domain. *Human Factors*, 31(6):635–666, 1989.

Mariah L Schrum, Erin Hedlund-Botti, and Matthew Gombolay. Reciprocal MIND MELD: Improving learning from demonstration via personalized, reciprocal teaching. In *Conference on Robot Learning*, pp. 956–966. PMLR, 2023a.

Mariah L Schrum, Emily Sumner, Matthew C Gombolay, and Andrew Best. Maveric: A data-driven approach to personalized autonomous driving. *arXiv preprint arXiv:2301.08595*, 2023b.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*, 2019.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pp. 785–799. PMLR, 2023.

Andrew Silva, Nina Moorman, William Silva, Zulfiqar Zaidi, Nakul Gopalan, and Matthew Gombolay. Lancon-learn: Learning with language to enable generalization in multi-task manipulation. *IEEE Robotics and Automation Letters*, 7(2):1635–1642, 2021.

David Silver, Satinder Singh, Doina Precup, and Richard S Sutton. Reward is enough. *Artificial Intelligence*, 299:103535, 2021.

Özgür Şimşek and Andrew G Barto. Using relative novelty to identify useful temporal abstractions in reinforcement learning. In *Proceedings of the 21st International Conference on Machine learning*, pp. 95, 2004.

Harshinder Singh, Neeraj Misra, Vladimir Hnizdo, Adam Fedorowicz, and Eugene Demchuk. Nearest neighbor estimates of entropy. *American journal of mathematical and management sciences*, 23(3-4):301–321, 2003.

Nate Soares and Benja Fallenstein. Aligning superintelligence with human interests: A technical research agenda. *Machine Intelligence Research Institute (MIRI) technical report*, 8, 2014.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Andrew Szot, Amy Zhang, Dhruv Batra, Zsolt Kira, and Franziska Meier. Bc-irl: Learning generalizable reward functions from demonstrations. *arXiv preprint arXiv:2303.16194*, 2023.

Adrien Ali Taiga, Rishabh Agarwal, Jesse Farebrother, Aaron Courville, and Marc G Bellemare. Investigating multi-task pretraining and generalization in reinforcement learning. In *The 11th International Conference on Learning Representations*, 2022.

Chen Tang, Ben Abbatematteo, Jiaheng Hu, Rohan Chandra, Roberto Martín-Martín, and Peter Stone. Deep reinforcement learning for robotics: A survey of real-world successes. *arXiv preprint arXiv:2408.03539*, 2024.

Voot Tangkaratt, Bo Han, Mohammad Emtiyaz Khan, and Masashi Sugiyama. Variational imitation learning with diverse-quality demonstrations. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 9407–9417, 2020.

Yee Teh, Victor Bapst, Wojciech M Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. *Advances in Neural Information Processing Systems*, 30, 2017.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in Neural Information Processing Systems*, 30, 2017.

Chen Wang, Claudia Pérez-D'Arpino, Danfei Xu, Li Fei-Fei, Karen Liu, and Silvio Savarese. Co-gail: Learning diverse strategies for human-robot collaboration. In *Conference on Robot Learning*, pp. 1279–1290. PMLR, 2022.

Yawei Wang and Xiu Li. Reward function shape exploration in adversarial imitation learning: an empirical study. In *2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pp. 52–57. IEEE, 2021.

Ziyu Wang, Josh S Merel, Scott E Reed, Nando de Freitas, Gregory Wayne, and Nicolas Heess. Robust imitation of diverse behaviors. *Advances in Neural Information Processing Systems*, 30, 2017.

Paweł Wawrzyński. A cat-like robot real-time learning to run. In *Adaptive and Natural Computing Algorithms: 9th International Conference, Kuopio, Finland, Revised Selected Papers 9*, pp. 380–390. Springer, 2009.

Annie Xie, Dylan Losey, Ryan Tolsma, Chelsea Finn, and Dorsa Sadigh. Learning latent representations to influence multi-agent interaction. In *Conference on robot learning*, pp. 575–588. PMLR, 2021.

Annie Xie, Lisa Lee, Ted Xiao, and Chelsea Finn. Decomposing the generalization gap in imitation learning for visual robotic manipulation. *arXiv preprint arXiv:2307.03659*, 2023.

Mengdi Xu, Yikang Shen, Shun Zhang, Yuchen Lu, Ding Zhao, Joshua Tenenbaum, and Chuang Gan. Prompting decision transformer for few-shot policy generalization. In *International Conference on machine learning*, pp. 24631–24645. PMLR, 2022.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, pp. 1094–1100. PMLR, 2020.

Luyao Yuan, Xiaofeng Gao, Zilong Zheng, Mark Edmonds, Ying Nian Wu, Federico Rossano, Hongjing Lu, Yixin Zhu, and Song-Chun Zhu. In situ bidirectional human-robot value alignment. *Science Robotics*, 7(68):eabm4183, 2022.

Tom Zahavy, Yannick Schroecker, Feryal Behbahani, Kate Baumli, Sebastian Flennerhag, Shaobo Hou, and Satinder Singh. Discovering policies with domino: Diversity optimization maintaining near optimality. *arXiv preprint arXiv:2205.13521*, 2022.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Zeyu Zhu and Huijing Zhao. A survey of deep rl and il for autonomous driving policy learning. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14043–14065, 2021.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

A POINT MAZE

The PointMaze domain considered in Sec. 4 is presented in Fig. 9. PointMaze is a two-dimensional navigation environment with continuous state and action spaces. The state vector represents the agent's current location's x- and y- coordinates in $[-\infty, \infty]^2$. The action space is a velocity command, a two-dimensional vector in $[-1, 1]^2$. The episode length is fixed to 25 steps. Expert demonstrations are collected from a PD controller parameterized with a one-dimensional (1D) factor, ω that determines the waypoint through which the agent passes $(0, \omega)$ on its way to the goal $(2, 0)$.

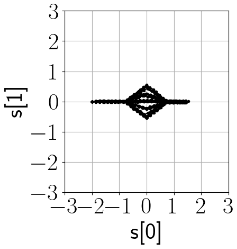


Figure 9: The figure visualizes expert demonstrations in PointMaze with varying waypoints along $x = 0$.

B DOMAINS AND DEMONSTRATIONS

The bounded factor range is divided into 5 intervals for each domain, as explained in Sec. 6. For each interval, we add Gaussian noise to the mean value of the interval to generate five samples. We condition the expert policy on the five samples to obtain five demonstrations for each interval.

B.1 HALFCHEETAH

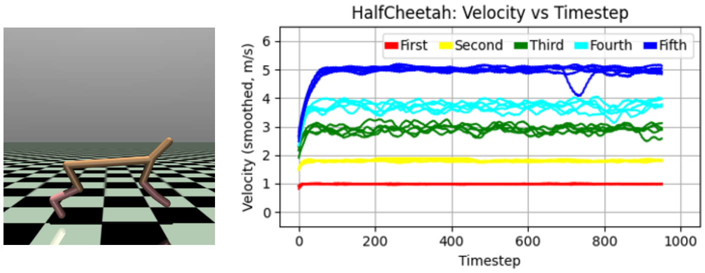


Figure 10: **Left:** The images visualizes the HalfCheetah robot. **Right:** The figure shows the (smoothed) velocity plotted against the timestep of the demonstration. The trajectories are colored to indicate the factor interval in which they belong.

The HalfCheetah environment considered in Sec. 6 is from the gym library (Brockman et al., 2016). The observation vector consists of the positions and velocities of the robot joints, along with height and velocities in the vertical and horizontal directions. The reward is modified as shown in Eq. 5, where r_t is the reward at the time, t, and x_t is the position of the center of mass of the robot along the x-axis at time t, and I, the indicator function that outputs 1 if and only if (iff) its argument evaluates to true. The undiscounted episode return counts the number of steps in which the cheetah moves forward by a non-zero amount. The return is normalized using the range, $[0, 1050]$. The episode length is fixed at 1000 steps. The environment is stochastic with the robot initialized at random configurations.

$$r_{step} = I(x_{t+1} - x_t > 0) \tag{5}$$

The factor is the mean velocity measured as the net change in the x-coordinate over the elapsed time. Due to environment stochasticity, we use five sampled trajectories per conditioning latent vector during evaluation and consider the mean value. Demonstrations consist of the robot running at different mean velocities $[1, 2, 3, 4, 5]$ m/s, collected using RL policies trained using SAC (Haarnoja et al., 2018) and auxiliary rewards for target velocities.

B.2 DRIVELANESHIFT

The DriveLaneshift environment is built from the highway-env library (Leurent, 2018). The highway consists of two lanes. The scenario includes the ego-car in the right lane controlled by the agent, and another car in front, in the same lane, that maintains a constant speed of 25 m/s. The task of the ego-car is to shift to the left lane, overtake the other car, and reach the target speed of 30 m/s. The reward at each step is as shown in Eq. 6, where r_t is the reward at the time, t, b_{onroad}, evaluates to true iff the car is within the road bounds at time t, b_{safe} evaluates to true iff the ego-car has not crashed until time t, $b_{leftlane}$ evaluates to true iff the ego-car is in the left lane at time t, v_t is the speed at time t, and $clip(x, a, b)$ clips the value x to lie between a and b. The return is normalized using the range $[0, 175]$. The state vector includes positions (absolute for the ego-car, relative for the other), velocities, heading angles, and longitudinal, latitudinal, and angular offsets to the closest lane for both cars. The episode length is fixed at 50 steps. The environment is deterministic.

$$r_{step} = I(b_{onroad}) + I(b_{safe}) + I(b_{leftlane}) + clip(\frac{|v_t - 30|}{5}, 0, 1) \tag{6}$$

The factor is the min headway distance, i.e., the distance between the ego-car and the other, at which the ego-car shifts to the left lane before overtaking. Demonstrations consist of the ego-car performing overtaking maneuvers at varying min headway distances $[10.92, 18.28, 25.62, 32.91, 40.27]$ m, collected using a scripted PD controller.

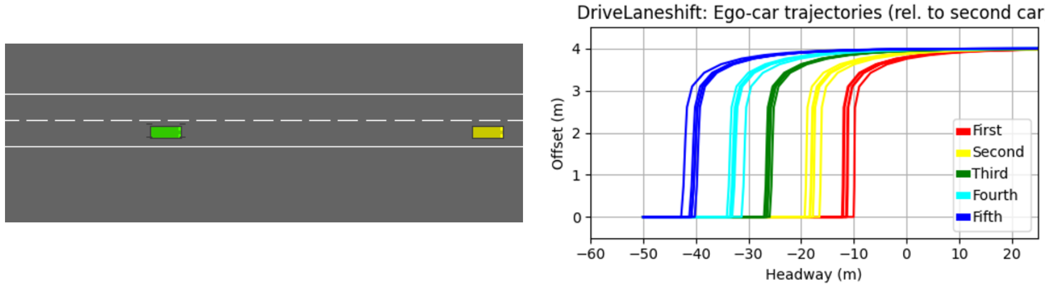


Figure 11: **Top:** The images visualize the highway overtaking scenario with the ego-car (green) starting behind the other car (yellow) in the right lane. **Bottom:** The figure visualizes the position of the ego-car relative to the other car as recorded in the demonstrations. The trajectories are colored to indicate the factor interval in which they belong.

B.3 FETCHPICKPLACE

The FetchPickPlace environment considered is from the gym library (Brockman et al., 2016). The task is to move the object from its initial location on the table at $[1.20, 0.75]$ to along the line $x = 1.40$, with reward at each step measured as shown in Eq. 7, where r_t is the reward at the time, t, and x_t is the position of the center of mass of the object along the x-axis at time t. The return is normalized using the range $[-20, -5]$. The state vector includes the end effector position and velocity, object position and velocity, finger gripper position and velocity, and object position relative to the gripper. The episode length is fixed at 100 steps. The environment is deterministic.

$$r_{step} = -|x_{t+1} - x_t| \tag{7}$$

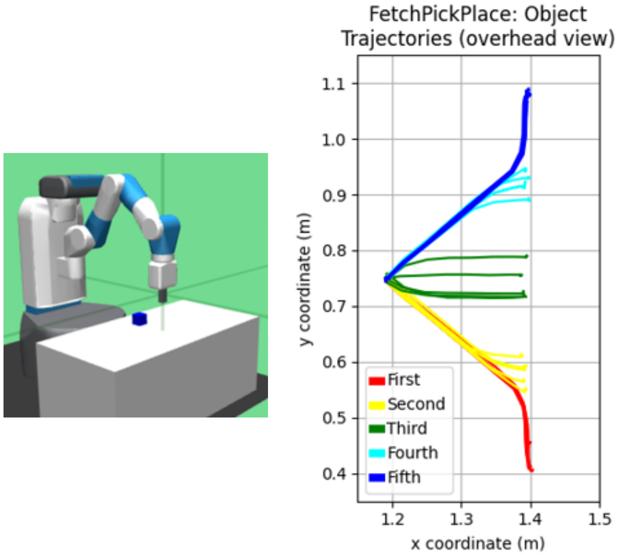


Figure 12: **Left:** The images visualize the Fetch robot with an object on the table. **Right:** The figure shows the object trajectories (from the top down) recorded in the demonstrations. The trajectories are colored to indicate the factor interval in which they belong.

The factor is the y-coordinate of the final object position. Demonstrations consist of the robot arm picking the object up from the initial location and placing it at the target x-coordinate and varying y-coordinates, $0.75 + [-0.32, -0.16, 0, 0.16, 0.32]$ m, collected using a scripted state-based PD controller.

C EVALUATION

C.1 IMPLEMENTATION

We implement our approach on top of the public code-base for VILD (Tangkaratt et al., 2020) that implements adversarial IL algorithms using PyTorch (Imambi et al., 2021): github.com/voot-t/vild_code. We use implementation tricks from their codebase to ensure convergence across methods, such as gradient penalty (Gulrajani et al., 2017) with a weight of 0.1 for the discriminator/task relevance, and the positive logarithmic function (Wang & Li, 2021) for discriminator rewards, i.e., $r(s, a) = -\log(1 - D(s, a))$ instead of $r(s, a) = \log(D(s, a))$.

We use a normal prior for the latent space across all approaches. The decoder q outputs the mean and diagonal elements of the covariance matrix of the approximate posterior distribution, which is assumed to be Gaussian.

Conditioned Discriminator To infer latent code τ^{ξ} for a demonstration trajectory τ^{ξ}, we make a simplifying assumption that the posterior distributions across transitions are independent. Thus, the product of the individual distributions gives us the posterior distribution for the demonstration trajectory.

We add expert transitions with mismatched demonstration latent vectors as fake samples to the discriminator dataset to ensure that the conditioned discriminator, $D(s, a, z)$, does not ignore the input latent vector. We upsample "real" data points in a batch to avoid imbalanced classes for discriminator gradient updates.

Decoder Regularization We perform spectral normalization (Miyato et al., 2018) using the PyTorch function `nn.utils.parametrizations.spectral_norm`. We scale the inputs to the decoder to implement Lipschitz constraint scaling with λ_S.

C.2 DOMAIN-SPECIFIC VARIATIONS AND TUNING

The hyperparameters used in our optimization are listed in Tables 1, 2. Each method is independently tuned for λ_I (and λ_C for Lipz, GSD) over the specified ranges, to maximize MAE over the test split for K=10 over averaged over four rounds of evaluation and five train seeds. All hyperparameters omitted from the tables are set to default values from our base implementation.

Hyperparameter	Value
NN update minibatch size	256
Policy learning rate	3e-4
Entropy bonus	0.0001
Gamma	0.99
GAE coefficient (Schulman et al., 2015)	0.97
NN architectures	FCN
Policy activation	Tanh
BC warmstart epochs	10000
Disc. activation	Tanh
Disc. learning rate	1e-3
Disc. gradient steps	5
Dec. hidden dimensions	[100, 256]
Dec. activation	ReLU
Dec. gradient steps	5
λ_S	10
b initial value	-5
Constraint slack (ϵ)	1e-6
Lambda learning rate	1e-3
Optimizers	Adam

Table 1: The table contains the list of hyperparameters, common across domains and generalization settings. GAE: Generalized Advantage Estimation, NN: Neural Network, FCN: Fully connected network, BC: Behavior cloning, Disc.: Discriminator, Dec.: Decoder

Hyperparam. \ Domain	**HalfCheetah**	**DriveLaneshift**	**FetchPickPlace**
Env. steps	1.5e7	0.5e7	1e7
RL algorithm	PPO	PPOBC	PPOBC
BC halflife, weight (PPOBC (Jena et al., 2021))	-	0.1, 0.2	0.1, 0.1
NN update interval (steps)	10000	1000	10000
NN hidden dimensions	[100, 100]	[100, 100]	[32, 32]
Observation norm. (w/ demos.)	False	False	True
Policy weight decay	(0, 1e-3)	1e-4	(1e-4, 5e-5)
Dec. learning rate	1e-3	1e-4	1e-3
Lambda initial value	(100, 500)	500	5000
Dec. gradient norm clip	∞	25	∞
Dec. rewards clip	[-∞, ∞]	[-20, 5]	[-∞, ∞]
Distillation objective weight	(0.02, 0.001)	(0.001, 0.001)	(0.0001, 0.0005)
λ_S sweep list	[0.02, 0.05, 0.1, 0.2]	[0.1, 0.5, 1.0, 5.0]	[0.1, 0.5, 1.0, 5.0]
λ_I sweep list	([0.9, 0.8], [0.8, 0.7])	([0.99, 0.97], [0.99, 0.97])	([0.9, 0.8], [0.8, 0.7])

Table 2: The table contains the list of hyperparameters, specific to each domain (indicated by the column) and generalization setting (indicated by a 2-tuple (left, right), where left and right correspond to interpolation and extrapolation respectively).

C.3 COMPARISON AGAINST OFFLINE IL APPROACHES

We compare our approach against offline IL approaches that learn solely from data without any environment interaction to provide a comprehensive evaluation. We rely on the implementations open-sourced by D3IL (Jia et al., 2024). To maintain focus on multimodal action distribution modeling, we exclude architectures that incorporate state histories or predict action sequences, such as action chunking (Zhao et al., 2023). We use fully connected neural network backbones with two hidden layers each containing 100 units. Unless otherwise specified, we use default hyperparameters that are most common across tasks in D3IL. We use demonstrations corresponding to held-out factor intervals as the validation dataset for early stopping.

- Behavior cloning (BC): The NN takes state as input and outputs actions. The NN is trained with mean squared error (MSE) loss.
- BC with VAE (BC-VAE): We utilize a state-conditioned encoder-decoder setup to model the action distribution (Sohn et al., 2015). We use a latent space dimension of size 2, similar to our approach. We use a weight of 5.0 for scaling KL-divergence loss.
- Implicit BC (IBC): Florence et al. (2022) propose energy models that implicitly capture the action distribution at each state. The action is inferred by optimizing the energy function using Markov chain Monte-carlo sampling at each inference step.

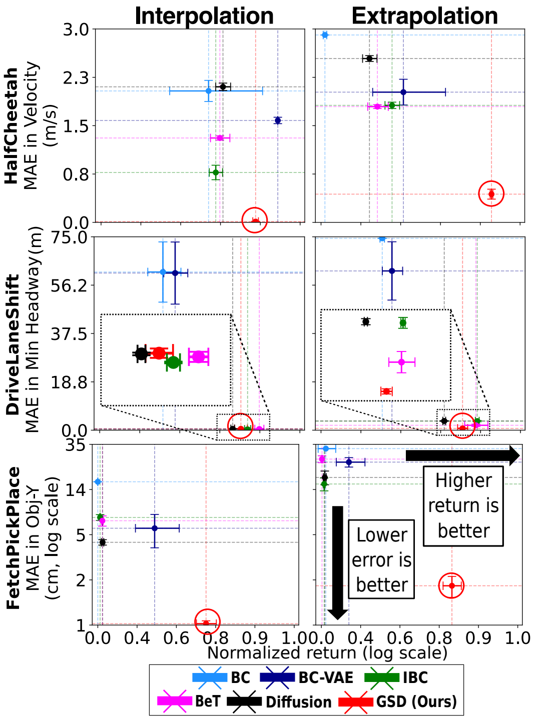


Figure 13: The figure shows the tradeoff between task and recovery performance for three domains and two splits for offline IL approaches and GSD (indicated with red circles). Error bars show standard errors over five seeds.

- K-Means Discretization (BeT): Shafiullah et al. (2022) propose an approach to capture multimodal action distributions using a learned discretization with K predicted action means and offsets. We use $K = 64$.
- Diffusion Policy (Diffusion): Chi et al. (2023); Pearce et al. (2023) propose modeling action distributions with a diffusion model conditioned on the state. We use timestep embeddings of size 16 and 24 denoising steps.

We show the recovery and performance tradeoff of offline IL and our approach in Fig. 13. With regard to task performance (x-axis), for HalfCheetah (top row) and FetchPickPlace (bottom), GSD outperforms offline approaches (except for interpolation in HalfCheetah) indicated by the red cross being positioned further right than others. For DriveLaneShift (middle row), offline approaches other than BC and BC-VAE are competitive with GSD. The result suggests that in domains with complex dynamics like HalfCheetah and FetchPickPlace, environment interaction is necessary for task completion when learning from few demonstrations.

With regard to recovery performance (y-axis), for HalfCheetah (top row) and FetchPickPlace (bottom), GSD outperforms all offline approaches, indicated by the red cross being positioned below others. For interpolation in DriveLaneShift (middle row, left), approaches IBC, BeT and Diffusion are comparable to GSD. However, for extrapolation (middle row, right), GSD outperforms all methods. Poor performance of offline approaches in domains with complex dynamics like HalfCheetah and FetchPickPlace may be attributed to the absence of environment interaction. In simpler domains, IBC, BeT or Diffusion may be able to interpolate diverse behaviors. However, for extrapolation, environment interaction is necessary, even for simpler domains.

D EVALUATION WITH HUMAN DEMONSTRATIONS IN TABLE TENNIS

D.1 SETUP

Demonstrations: The WAM arm is enabled with gravity compensation for collecting kinesthetic demonstrations. Messages published to a ROS interface are collected for two seconds (starting after the ball is detected to move over the table) which is enough time to capture the return trajectory. Joint states ($\mathbb{R}^7$) and ball positions ($\mathbb{R}^3$) are matched over time, and concatenated to construct state vectors ($\mathbb{R}^{10}$). Action vectors are constructed by calculating the displacements at corresponding timesteps. We collect five demonstrations for each of the two-stroke types considered.

Simulation: We use position control for the WAM arm at 100hz with the control gains tuned to match real robot demonstrations visually when replaying action commands open loop. We tune ball flight parameters such that the ball flight paths (before being struck) visually match those from the real demonstrations when launched from a similar position, velocity, and noise as the ball launcher. We add Gaussian noise to the ball positions in the observation vector to mimic real recorded ball positions. We use an episode length of 200 steps that corresponds to a real-life execution period of two seconds.

D.2 EVALUATION

We detect if the ball has been returned by checking if the velocity component along the long edge of the TT table has reversed. Once a returning ball is detected, we check if the ball remains above the table plane and within 10 cm beyond the sides of the table for the following 0.5 seconds. If the trajectory satisfies both criteria, we deem it a success. We calculate the factor value for each successful return which is the maximum height the ball reaches in the return trajectory.

We sample five trajectories per sampled latent vector due to the stochastic nature of the ball observations. However, not every sampled trajectory for a particular latent vector is guaranteed to succeed due to the stochasticity in the domain and optimization. Thus, we consider a latent vector successful if the ball is returned in at least three out of five trials. We consider the factor value for the successful latent vector to be the mean of the values of the successful trajectories.

For each method and train seed, we sample 200 latent vectors from the prior, and sample five trajectories per latent vector. We report the fraction of successful latent vectors to evaluate if behaviors can accomplish the underlying task. Among the set of latent vectors, we subsample 100 successful latent vectors (after ensuring each method has at least 50% success rate) and report the entropy (based on particle estimates (Singh et al., 2003)) among the calculated factors using the equation shown in Eq. 8 where $V = \{v_i\}_{i=0}^{M}$ is the set of factor values v_i, $M = 100$, $K = 50$ and $\mathrm{NNe}_{K,V}(v_i)$ returns the Kth nearest neighbor to v_i from the set of values V. The entropy measure is up to a proportionality constant, as we use it to compare diversity achieved in return ball trajectory heights across methods.

$$H_K(V) = \frac{1}{M} \sum_{i=0}^{M} \log ||v_i - \mathrm{NNe}_{K,V}(v_i)|| \tag{8}$$

E FURTHER LIMITATIONS

Our work pertains to the limited realm of adversarial IL frameworks that employ diversity objectives in the form of MI. The generalization capabilities of other multimodal IL frameworks based on non-adversarial IL should be explored.

The scope of generalization considered in this work pertains to variations in the demonstrated expert preferences. Other forms of generalization to altered environment dynamics, adversarial perturbations, etc., should also be considered in the context of IL.

We employ distillation to extract a task-relevance measure. Other approaches that learn generalizable reward functions (Szot et al., 2023) could also be explored. Our work pertains to online adversarial IL frameworks that employ diversity objectives in the form of MI. The efficacy of our constraints could be explored with offline and non-MI-based diversity frameworks.

F EVALUATION RESULTS

We provide the numerical figures for recovery errors used to plot the graphs in Fig. 4 below. We further abbreviate Con, ConDist and Lipz as CO, CD and LZ respectively due to width constraints.

HalfCheetah: Interpolation, Average:

Model	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$
IG	0.258 ± 0.007	0.147 ± 0.006	0.106 ± 0.004	0.080 ± 0.004	0.063 ± 0.004
IG+LZ	0.265 ± 0.009	0.156 ± 0.010	0.111 ± 0.008	0.089 ± 0.008	0.069 ± 0.006
IG+CO	0.229 ± 0.015	0.114 ± 0.007	0.074 ± 0.004	0.058 ± 0.003	0.049 ± 0.003
IG+CD	0.266 ± 0.014	0.142 ± 0.007	0.101 ± 0.006	0.077 ± 0.005	0.060 ± 0.005
IG+CD+LZ	0.307 ± 0.004	0.177 ± 0.005	0.124 ± 0.004	0.091 ± 0.003	0.077 ± 0.003
GSD (Ours)	0.226 ± 0.007	0.120 ± 0.003	0.081 ± 0.003	0.065 ± 0.001	0.052 ± 0.002

HalfCheetah: Interpolation, Worst:

Model	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$
IG	0.298 ± 0.004	0.177 ± 0.002	0.134 ± 0.003	0.103 ± 0.003	0.076 ± 0.003
IG+LZ	0.348 ± 0.020	0.212 ± 0.021	0.159 ± 0.019	0.131 ± 0.017	0.099 ± 0.013
IG+CO	0.291 ± 0.029	0.142 ± 0.014	0.092 ± 0.007	0.072 ± 0.007	0.061 ± 0.005
IG+CD	0.339 ± 0.024	0.181 ± 0.013	0.133 ± 0.011	0.102 ± 0.009	0.082 ± 0.008
IG+CD+LZ	0.373 ± 0.007	0.223 ± 0.010	0.158 ± 0.008	0.117 ± 0.006	0.100 ± 0.006
GSD (Ours)	0.287 ± 0.007	0.151 ± 0.005	0.106 ± 0.004	0.084 ± 0.002	0.069 ± 0.002

HalfCheetah: Extrapolation, Average:

Model	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$
IG	0.841 ± 0.010	0.766 ± 0.014	0.737 ± 0.016	0.710 ± 0.017	0.697 ± 0.019
IG+LZ	0.891 ± 0.004	0.829 ± 0.005	0.804 ± 0.005	0.788 ± 0.005	0.779 ± 0.005
IG+CO	0.741 ± 0.007	0.618 ± 0.002	0.560 ± 0.004	0.525 ± 0.006	0.505 ± 0.007
IG+CD	0.801 ± 0.024	0.642 ± 0.020	0.581 ± 0.019	0.538 ± 0.019	0.513 ± 0.020
IG+CD+LZ	0.686 ± 0.008	0.597 ± 0.008	0.559 ± 0.010	0.539 ± 0.010	0.525 ± 0.011
GSD (Ours)	0.682 ± 0.028	0.556 ± 0.032	0.512 ± 0.033	0.484 ± 0.034	0.472 ± 0.034

HalfCheetah: Extrapolation, Worst:

Model	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$
IG	0.910 ± 0.013	0.854 ± 0.016	0.837 ± 0.017	0.828 ± 0.017	0.821 ± 0.017
IG+LZ	0.999 ± 0.004	0.909 ± 0.007	0.880 ± 0.009	0.869 ± 0.009	0.860 ± 0.010
IG+CO	0.875 ± 0.019	0.779 ± 0.021	0.725 ± 0.024	0.685 ± 0.027	0.661 ± 0.030
IG+CD	0.963 ± 0.031	0.750 ± 0.017	0.686 ± 0.012	0.656 ± 0.008	0.634 ± 0.007
IG+CD+LZ	0.797 ± 0.012	0.719 ± 0.011	0.699 ± 0.011	0.689 ± 0.011	0.682 ± 0.011
GSD (Ours)	0.808 ± 0.039	0.679 ± 0.038	0.646 ± 0.040	0.632 ± 0.041	0.621 ± 0.041

DriveLaneshift: Interpolation, Average:

Model	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$
IG	3.435 ± 0.148	2.305 ± 0.132	1.808 ± 0.122	1.489 ± 0.105	1.277 ± 0.098
IG+LZ	3.193 ± 0.096	2.101 ± 0.097	1.631 ± 0.101	1.345 ± 0.078	1.111 ± 0.073
IG+CO	5.246 ± 0.500	3.153 ± 0.375	2.377 ± 0.333	1.936 ± 0.302	1.712 ± 0.278
IG+CD	3.619 ± 0.200	2.246 ± 0.151	1.749 ± 0.154	1.419 ± 0.163	1.221 ± 0.154
IG+CD+LZ	2.785 ± 0.196	1.652 ± 0.172	1.302 ± 0.156	1.114 ± 0.142	0.985 ± 0.128
GSD (Ours)	2.343 ± 0.134	1.299 ± 0.094	0.877 ± 0.061	0.685 ± 0.058	0.530 ± 0.045

DriveLaneshift: Interpolation, Worst:

Model	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$
IG	4.528 ± 0.262	3.294 ± 0.226	2.693 ± 0.228	2.280 ± 0.193	2.008 ± 0.190
IG+LZ	4.702 ± 0.179	3.315 ± 0.187	2.675 ± 0.192	2.253 ± 0.152	1.859 ± 0.140
IG+CO	6.843 ± 0.690	4.304 ± 0.507	3.329 ± 0.441	2.754 ± 0.397	2.469 ± 0.380
IG+CD	5.057 ± 0.307	3.329 ± 0.249	2.657 ± 0.269	2.171 ± 0.282	1.878 ± 0.271
IG+CD+LZ	3.895 ± 0.314	2.581 ± 0.310	2.079 ± 0.291	1.797 ± 0.269	1.637 ± 0.241
GSD (Ours)	3.009 ± 0.246	1.709 ± 0.171	1.138 ± 0.104	0.911 ± 0.106	0.687 ± 0.079

DriveLaneshift: Extrapolation, Average:

Model	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$
IG	5.807 ± 0.231	4.889 ± 0.269	4.446 ± 0.288	4.144 ± 0.297	3.943 ± 0.302
IG+LZ	6.435 ± 0.124	5.692 ± 0.193	5.375 ± 0.224	5.134 ± 0.244	4.977 ± 0.257
IG+CO	6.347 ± 0.769	4.335 ± 0.584	3.588 ± 0.567	3.166 ± 0.542	2.967 ± 0.521
IG+CD	3.902 ± 0.185	2.480 ± 0.185	1.886 ± 0.172	1.559 ± 0.166	1.399 ± 0.168
IG+CD+LZ	5.588 ± 0.511	3.809 ± 0.400	2.902 ± 0.369	2.472 ± 0.362	2.206 ± 0.369
GSD (Ours)	2.803 ± 0.108	1.545 ± 0.074	1.019 ± 0.053	0.815 ± 0.046	0.695 ± 0.048

DriveLaneshift: Extrapolation, Worst:

Model	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$
IG	6.386 ± 0.219	5.527 ± 0.246	5.102 ± 0.268	4.805 ± 0.289	4.624 ± 0.291
IG+LZ	6.908 ± 0.125	6.222 ± 0.224	6.071 ± 0.244	5.923 ± 0.259	5.836 ± 0.269
IG+CO	8.599 ± 1.302	6.673 ± 1.140	5.827 ± 1.107	5.324 ± 1.069	5.033 ± 1.026
IG+CD	5.326 ± 0.338	3.606 ± 0.341	2.907 ± 0.339	2.438 ± 0.322	2.206 ± 0.328
IG+CD+LZ	8.723 ± 1.105	6.473 ± 0.890	5.038 ± 0.790	4.340 ± 0.764	3.945 ± 0.766
GSD (Ours)	4.067 ± 0.225	2.283 ± 0.156	1.534 ± 0.105	1.238 ± 0.091	1.034 ± 0.094

FetchPickPlace: Interpolation, Average:

Model	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$
IG	0.071 ± 0.004	0.045 ± 0.004	0.034 ± 0.003	0.027 ± 0.003	0.022 ± 0.002
IG+LZ	0.044 ± 0.002	0.024 ± 0.001	0.016 ± 0.001	0.012 ± 0.001	0.010 ± 0.001
IG+CO	0.072 ± 0.009	0.053 ± 0.010	0.046 ± 0.010	0.041 ± 0.011	0.040 ± 0.011
IG+CD	0.083 ± 0.011	0.067 ± 0.012	0.060 ± 0.012	0.055 ± 0.012	0.052 ± 0.012
IG+CD+LZ	0.050 ± 0.001	0.027 ± 0.001	0.019 ± 0.001	0.015 ± 0.001	0.012 ± 0.001
GSD (Ours)	0.037 ± 0.001	0.020 ± 0.001	0.014 ± 0.000	0.011 ± 0.000	0.008 ± 0.000

FetchPickPlace: Interpolation, Worst:

Model	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$
IG	0.084 ± 0.005	0.058 ± 0.005	0.046 ± 0.005	0.037 ± 0.004	0.031 ± 0.004
IG+LZ	0.056 ± 0.003	0.032 ± 0.002	0.021 ± 0.002	0.016 ± 0.001	0.013 ± 0.001
IG+CO	0.081 ± 0.009	0.061 ± 0.011	0.054 ± 0.011	0.048 ± 0.011	0.047 ± 0.011
IG+CD	0.088 ± 0.010	0.072 ± 0.012	0.064 ± 0.012	0.059 ± 0.012	0.056 ± 0.012
IG+CD+LZ	0.053 ± 0.002	0.030 ± 0.001	0.021 ± 0.001	0.017 ± 0.001	0.014 ± 0.001
GSD (Ours)	0.042 ± 0.001	0.022 ± 0.001	0.015 ± 0.001	0.013 ± 0.001	0.009 ± 0.000

FetchPickPlace: Extrapolation, Average:

Model	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$
IG	0.137 ± 0.003	0.103 ± 0.003	0.087 ± 0.003	0.075 ± 0.003	0.068 ± 0.003
IG+LZ	0.083 ± 0.003	0.057 ± 0.003	0.045 ± 0.003	0.038 ± 0.003	0.034 ± 0.003
IG+CO	0.210 ± 0.016	0.183 ± 0.018	0.167 ± 0.018	0.158 ± 0.018	0.152 ± 0.018
IG+CD	0.264 ± 0.012	0.246 ± 0.013	0.233 ± 0.014	0.225 ± 0.014	0.220 ± 0.015
IG+CD+LZ	0.078 ± 0.003	0.046 ± 0.002	0.032 ± 0.002	0.026 ± 0.002	0.022 ± 0.002
GSD (Ours)	0.068 ± 0.002	0.037 ± 0.002	0.027 ± 0.002	0.021 ± 0.002	0.018 ± 0.002

FetchPickPlace: Extrapolation, Worst:

Model	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$
IG	0.182 ± 0.004	0.143 ± 0.004	0.120 ± 0.003	0.105 ± 0.003	0.097 ± 0.003
IG+LZ	0.097 ± 0.003	0.071 ± 0.004	0.057 ± 0.004	0.049 ± 0.004	0.044 ± 0.004
IG+CO	0.249 ± 0.012	0.223 ± 0.015	0.208 ± 0.017	0.196 ± 0.017	0.191 ± 0.018
IG+CD	0.303 ± 0.003	0.286 ± 0.006	0.275 ± 0.008	0.266 ± 0.009	0.260 ± 0.010
IG+CD+LZ	0.092 ± 0.003	0.057 ± 0.003	0.042 ± 0.004	0.035 ± 0.004	0.031 ± 0.004
GSD (Ours)	0.094 ± 0.005	0.056 ± 0.005	0.041 ± 0.004	0.033 ± 0.003	0.028 ± 0.004