Imitation Beyond Expectation Using Pluralistic Stochastic Dominance

Ali Farajzadeh, Danyal Saeed, Syed M. Abbas, Rushit Shah, Aadirupa Saha, Brian D. Ziebart
Department of Computer Science
University of Illinois Chicago
Chicago, IL 60607
{afaraj5,dsaeed3,sabbas33,rshah231,aadirupa,bziebart}@uic.edu

Abstract

Imitation learning seeks to estimate policies reflecting the values of demonstrated behaviors. Prevalent approaches learn to match or exceed the demonstrator's performance *in expectation* without knowing the demonstrator's reward function. Unfortunately, this does not induce pluralistic imitators that learn to support distinct demonstrations. We reformulate imitation learning using *stochastic dominance* over the demonstrations' reward distribution across a range of reward functions as our foundational aim. Our approach matches imitator policy samples (or support) with demonstrations using optimal transport theory to define an imitation learning objective over trajectory pairs. We demonstrate the benefits of pluralistic stochastic dominance (PSD) for imitation in both theory and practice.

1 Introduction

When learning from demonstrations, behaviors reflecting individual preferences and capabilities are often demonstrated. Existing imitation learning methods struggle to preserve these distinct behaviors while trying to improve beyond them. For example, inverse reinforcement learning (Abbeel & Ng, 2004; Ziebart, 2010) and discriminative imitation (Ratliff et al., 2006; Ho & Ermon, 2016) methods seek to match or outperform (Syed & Schapire, 2007) the demonstrations under a range of reward functions *in expectation*. As shown in Figure 1, this can be achieved by an imitator that never produces behavior that a demonstrator prefers over any of his or her more preferable demonstrations.

We seek a stronger *distributional* guarantee of **pluralistic stochastic dominance** (PSD)[1], which ensures the imitator a higher probability of achieving any level of reward than the demonstration distribution (i.e., stochastic dominance) for all reward functions (i.e., pluralism) within some defined set. This requires the imitator to *match or improve upon* the distinct properties of exceptional demonstrations rather than focusing on the average of demonstrations—often by randomizing between different modes (Figure 1). These guarantees support more complex applications of imitator policies (e.g., sampling many candidate trajectories and selecting the best) beyond the assumption that a single imitator trajectory is sampled and executed.

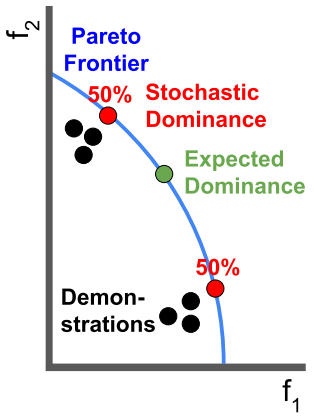

Figure 1: Dominance in expectation (green) guarantees better performance than the demonstration *average* for all conical sum reward functions. Pluralistic stochastic dominance (red) makes the *distribution of rewards* preferable by guaranteeing a higher probability of achieving any reward.

[1]Code available at `https://github.com/Ali199776/PSD`.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Our approach is based on the observation that if all demonstrations can be matched to supported imitator behaviors that are better than the paired demonstration for all reward functions (Figure 1), then PSD is achieved. We employ optimal transport theory (Rioux et al., 2024) to perform this matching and a margin-based upper bound (Ziebart et al., 2022) to define a loss over the demonstration-imitation behavior pair. We then optimize the imitator policy using the matched pairs in the fully realizable and model-based imitator policy settings. We provide stochastic dominance generalization guarantees to establish the theoretical benefits of pursuing pluralistic stochastic dominance, and experimentally demonstrate better support for distinct behavior that improves beyond the paired demonstrations.

The main contributions of this paper are three-fold: First, PSD improves upon imitation learning methods designed for specific risk-sensitivities (Majumdar et al., 2017; Singh et al., 2018; Santara et al., 2018; Lacotte et al., 2019) to provide guarantees simultaneously across all reasonable risk-sensitive performance measures (e.g., value-at-risk, conditional value-at-risk, and range value-at-risk, detailed in Appendix A.1). These performance guarantees provide broader support for more complex use cases of imitation learning, such as selecting the best trajectory from a set of samples. Second, PSD-based imitation provides an alternative justification to maximum entropy inverse reinforcement learning for stochastic imitation policies. Finally, we introduce novel evaluation metrics for imitation learning based on stochastic and Pareto dominance that can be applied in settings with known reward bases, but not singular motivating reward function.

2 Background and related work

2.1 Inverse reinforcement learning

In imitation learning settings, reward functions defining desirable behavior are unknown. Instead, foundational formulations assume that reward features are available that define a linear (Ng & Russell, 2000; Abbeel & Ng, 2004) or conical sum (Syed & Schapire, 2007) reward function.

Definition 2.1. Given **reward features** $\mathbf{f} : \Xi \to \mathbb{R}^K$, for behavior trajectories $\xi \in \Xi$, the family of feature-based **reward functions** is defined as $\mathrm{r}_\theta(\xi) = \theta \cdot \mathbf{f}(\xi)$ with parameters $\theta \in \mathbb{R}^K$ (**linear**) or $\theta \in \mathbb{R}^K_{\geq 0}$ (**conical sum**).

An imitation policy π that matches the feature moments of the demonstrator guarantees equal rewards in expectation—including the demonstrator's unknown (linear) reward function (Abbeel & Ng, 2004):

$$\forall \theta \in \mathbb{R}^K, \mathbb{E}_{\xi \sim \mathbb{P}_\pi}\Big[\mathbf{f}(\xi)\Big] = \mathbb{E}_{\tilde{\xi} \sim \mathbb{P}_{\tilde{\pi}}}\Big[\mathbf{f}(\tilde{\xi})\Big] \implies \mathbb{E}_{\xi \sim \mathbb{P}_\pi}\Big[\mathrm{r}_\theta(\xi)\Big] = \mathbb{E}_{\tilde{\xi} \sim \mathbb{P}_{\tilde{\pi}}}\Big[\mathrm{r}_\theta(\tilde{\xi})\Big] \tag{1}$$

where $\mathbb{P}_\pi$ denotes the distribution over trajectories ξ based on the interaction between the policy π and the dynamics of the decision process, which we assume are deterministic.

Many imitation learning approaches can be viewed as matching various moments (rewards, on-policy/off-policy state-action value functions) of the demonstrator (Swamy et al., 2021), including: behavior cloning (Pomerleau, 1988); maximum margin planning (Ratliff et al., 2006); maximum entropy inverse reinforcement learning (Ziebart, 2010); DAGGER (Ross et al., 2011) generative adversarial imitation learning (Ho & Ermon, 2016); and Value Dice (Kostrikov et al., 2019).

Entropy regularization methods for reinforcement learning (Neu et al., 2017)—also known as softmax decision policies (Sutton & Barto, 2018)—increase the diversity of the imitator's trajectories within these moment-matching techniques. However, these provide robust *predictive guarantees* for imitation learning (Ziebart, 2010) rather than *performance guarantees* for diverse demonstrators. Extensions of these methods attempt to model variations in preferences or quality of demonstrations with latent variables. These are then used to condition policy models (e.g., as mixture models) or focus imitation on more desirable demonstrations (Brown et al., 2020b; Chen et al., 2021; Wu et al., 2019; Zhang et al., 2021). We aim to avoid the computational challenges (e.g., difficult nonconvex optimizations) and/or strong assumptions underlying these approaches.

2.2 Outperformance and subdominance minimization

Our approach is closer in motivation to methods designed to outperform demonstrators. Early methods focus on policies that outperform in terms of expected rewards (Definition 2.2).

Definition 2.2. Policy π_1 has **expected dominance** over π_2 if the expected trajectory reward under π_1 is at least as much as the expected trajectory reward under π_2: $\mathbb{E}_{\xi_1 \sim \pi_1}\left[\mathrm{r}_\theta(\xi)\right] \geq \mathbb{E}_{\xi_2 \sim \pi_2}\left[\mathrm{r}_\theta(\xi)\right]$ for fixed θ.

MWAL (Syed & Schapire, 2007) and LPAL (Syed et al., 2008) guarantee outperforming the demonstrator in expectation under the assumption that the signs of the reward function weights are known (i.e., conical sum reward functions of Definition 2.1). In this setting, better expected reward features guarantee better expected rewards:

$$\forall \theta \in \mathbb{R}^K_{\geq 0}, \mathbb{E}_{\xi \sim \mathbb{P}_\pi}\left[\mathbf{f}(\xi)\right] \succeq \mathbb{E}_{\tilde{\xi} \sim \mathbb{P}_{\tilde{\pi}}}\left[\mathbf{f}(\tilde{\xi})\right] \implies \mathbb{E}_{\xi \sim \mathbb{P}_\pi}\left[\mathrm{r}_\theta(\xi)\right] \geq \mathbb{E}_{\tilde{\xi} \sim \mathbb{P}_{\tilde{\pi}}}\left[\mathrm{r}_\theta(\tilde{\xi})\right]. \tag{2}$$

Subdominance minimization (Ziebart et al., 2022) extends this idea of outperformance by seeking uniform dominance (Definition 2.3) across conical sum reward functions by minimizing a convex bound over the probability of violating uniform dominance.

Definition 2.3. Policy π_1 has **uniform dominance** over π_2 if all trajectory samples from π_1 have at least as much reward as all samples from π_2: $\mathbb{P}_{\xi_1 \sim \pi_1; \xi_2 \sim \pi_2}(\mathrm{r}_\theta(\xi_1) \geq \mathrm{r}_\theta(\xi_2)) = 1$ for fixed θ.

Unfortunately, uniform dominance encourages deterministic policies that are similar to the expected dominance policy in settings like Figure 1.

We pursue a less strict notion of dominance in this paper: stochastic dominance (Definition 2.4). It is based on having a better distribution of rewards.

Definition 2.4. Policy π_1 has **stochastic dominance** over π_2 if π_1 has at least as much probability of exceeding any reward threshold: $\forall c \in \mathbb{R}, \mathbb{P}_{\xi \sim \pi_1}(r_\theta(\xi) \geq c) \geq \mathbb{P}_{\xi \sim \pi_2}(r_\theta(\xi) \geq c)$ for fixed θ.

In terms of strictness, uniform dominance implies stochastic dominance, which implies expected dominance. However, uniform dominance is often infeasible (e.g., Figure 2a), while stochastic dominance is always feasible (e.g., $\pi = \tilde{\pi}$). In addition to expected reward bounds (2), stochastic dominance guarantees broad risk measure improvements (Ogryczak & Ruszczyński, 1999). We summarize some of these in Theorem 2.5.

Theorem 2.5. *Stochastic dominance of $r_\theta(\pi) \succeq r_\theta(\tilde{\pi})$ for some fixed θ guarantees improved expected and risk-sensitive rewards for the imitator $\xi \sim P_\pi$ with respect to the demonstrator $\tilde{\xi} \sim P_{\tilde{\pi}}$:* $\forall\,(c \in [0,1], d \in (c,1])$, $\mathbb{E}_{\xi \sim \pi}[r_\theta(\xi)] \geq \mathbb{E}_{\tilde{\xi} \sim \tilde{\pi}}[r_\theta(\tilde{\xi})]$, *VaR*$_c(r_\theta(\xi)) \geq$ *VaR*$_c(r_\theta(\tilde{\xi}))$, *CVaR*$_c(r_\theta(\xi)) \geq$ *CVaR*$_c(r_\theta(\tilde{\xi}))$, *and RVaR*$_{c,d}(r_\theta(\xi)) \geq$ *RVaR*$_{c,d}(r_\theta(\tilde{\xi}))$.

Prior imitation learning research employs risk sensitivity narrowly to address safety concerns by targeting specific tail risks (and specific quantile levels). Extensions of generative-adversarial imitation learning (GAIL) (Ho & Ermon, 2016) match specific demonstrator risk-sensitivities (Majumdar et al., 2017; Santara et al., 2018; Lacotte et al., 2019). Bayesian estimation methods (Brown et al., 2020a; Javed et al., 2021) incorporate risk sensitivity to more robustly address uncertainty during reward function estimation, In contrast, we consider risk-sensitivity exhaustively—across a family of reward functions and over all sensitivity thresholds, as guaranteed by stochastic dominance (Theorem 2.5)—to incentivize high-quality coverage of diverse demonstrations.

2.3 Optimal transport

Optimal transport theory considers the minimum cost of transforming from one distribution, $\mathbb{P}_X$, to another $\mathbb{P}_Y$ under cost function $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$. The Kantorovich (1942) formulation defines this transformation using a joint probability measure $\gamma \in \Delta_{\mathcal{X} \times \mathcal{Y}}$ with $\gamma(x, y)$ representing the amount of probability mapped from x to y and marginals that match the source and target distributions. For discrete distributions (with $\mathbb{P}_X$ and $\mathbb{P}_Y$ supporting m and n values, respectively) , this can be expressed as a linear program:

$$\mathrm{OT}_c(\mathbb{P}_X, \mathbb{P}_Y) = \min_{\gamma \geq \mathbf{0}} \sum_{i,j} \gamma_{i,j} c(x_i, y_j) \text{ s.t. } \forall j, \sum_i \gamma_{i,j} = \mathbb{P}_Y(y_j)\ , \forall i, \sum_j \gamma_{i,j} = \mathbb{P}_X(x_i)\ . \tag{3}$$

The optimization is solved exactly by the Hungarian algorithm in $\mathcal{O}(|\mathcal{X}|^3)$ time or with ϵ error tolerance using specialized algorithms (Dvurechensky et al., 2018) in $\tilde{\mathcal{O}}(\max(|\mathcal{X}|, |\mathcal{Y}|)^2/\epsilon^2)$ time.

The optimal transport objective (commonly referred to as the Wasserstein distance for metric costs) has been popularized as an alternative to the Jensen-Shannon divergence in generative-adversarial

learning (Arjovsky et al., 2017). Previous investigations for imitation learning tasks (Xiao et al., 2019; Dadashi et al., 2021). include cross-domain imitation transfer (Nguyen et al., 2021; Fickinger et al., 2022), combining trajectory matching with behavioral cloning (Haldar et al., 2023), and matching reward-less trajectories with expert trajectories (Luo et al., 2023). Each of these are distinct from our approach and motivation.

We build upon a key relationship between optimal transport and stochastic dominance in this work:
Remark 2.6. $\mathbb{P}_Y \succeq \mathbb{P}_X$ (Definition 2.4) if and only if there is a mapping from $\mathbb{P}_X$ to $\mathbb{P}_Y$ that is non-decreasing in value: $\text{OT}_{\max(x-y,\,0)}(\mathbb{P}_X, \mathbb{P}_Y) = 0$, where $c(x, y)$ is positive only when $x > y$.

3 Approach

3.1 Pluralistic stochastic dominance

We consider imitation learning with multiple demonstrators. Each has their own reward function, r_θ, presumed to be from the family of conical sum cost functions (Definition 2.1). Feature moment methods (1, 2) can guarantee that each demonstrator is at least indifferent between the trajectory distributions of the demonstrators and the imitator *in expectation* (Swamy et al., 2021). However, this does not guarantee any chance of producing highly desirable behavior for any demonstrators. As a consequence, if preferences are based on higher quantiles of reward distributions rather than expectations, the demonstration distribution can be highly preferable. We introduce pluralistic stochastic dominance (PSD) to ensure that the imitator policy is no less preferable to the demonstration distribution for all conical sum reward functions and all reward quantiles (Definition 3.1).

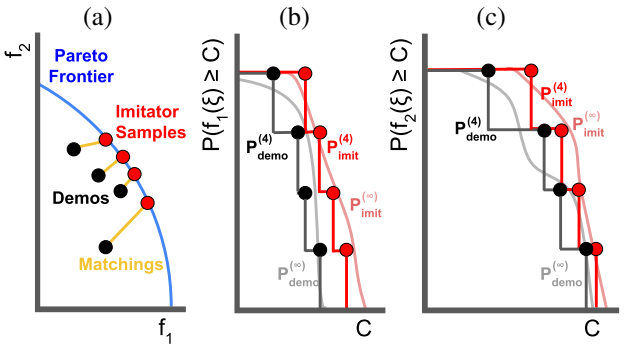

Figure 2: (a) Features (f_1, f_2) of four demonstrations (black points) matched (yellow lines) with four imitator samples (red points); the cumulative features for f_1 (b) and f_2 (c) for the sample distribution ($\text{P}^{(4)}$) and the full distribution ($\text{P}^{(\infty)}$). We seek to optimize the imitator policy using available demonstrator/imitator samples to achieve a full distribution for the imitator that stochastically dominates the demonstrator. This can be verified for each θ by sorting the samples, e.g., for $\theta = \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$ (b) and $\theta = \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$ (c). However, this is intractable for continuous sets of θ; our upper bound instead employs a single matching (a).

Definition 3.1. Distribution $\mathbb{P}_\pi$ provides (first-order) **pluralistic stochastic dominance** over $\mathbb{P}_{\tilde{\pi}}$ (for all conical sum reward functions), which we denote $\mathbb{P}_\pi \succeq_{\text{PSD}} \mathbb{P}_{\tilde{\pi}}$, iff:

$$\forall(\theta \in \mathbb{R}^K_{\geq \mathbf{0}}, C \in \mathbb{R}), \mathbb{P}_{\xi\sim\pi}(\text{r}_\theta(\xi) \geq C) \geq \mathbb{P}_{\tilde{\xi}\sim\tilde{\pi}}(\text{r}_\theta(\tilde{\xi}) \geq C). \tag{4}$$

Though exact replication ($\pi = \tilde{\pi}$) trivially achieves this, imitation learning often aims to be performant on withheld demonstrations, so exact replication of finite training demonstrations is not sufficient. Instead, imitator policies π that achieve better reward distributions (and better generalization) are desired. Figure 2(a) provides an example of strict stochastic dominance.

We extend stochastic dominance to the pluralistic setting by taking the maximum of optimal transport problems (Remark 2.6) for each conical sum reward function (Definition 3.2).

Definition 3.2. The **pluralistic stochastic subdominance** between imitator and demonstrator trajectory distributions is given by the worst-case reward function:

$$\max_{\theta\in[0,1]^K} \overbrace{\left(\min_{\gamma\succeq\mathbf{0}} \sum_{i,j} \gamma_{i,j} \left[\text{r}_\theta(\tilde{\xi}_j) - \text{r}_\theta(\xi_i)\right]_+ \text{ s.t. } \sum_j \gamma_{i,j} = \mathbb{P}_\pi(\xi_i)\ \forall i, \sum_i \gamma_{i,j} = \mathbb{P}_{\tilde{\pi}}(\tilde{\xi}_j)\ \forall j\right)}^{\text{OT}_{[\Delta\text{r}_\theta]_+}(\mathbb{P}_\pi, \mathbb{P}_{\tilde{\pi}})}. \tag{5}$$

Minimizing this entire set of optimal transport problems to zero guarantees PSD (Theorem 3.3). The proofs of this theorem and others are provided in Appendix B.

Theorem 3.3. *Zero maximum optimal transport in Def. 3.2 and pluralistic stochastic dominance are equivalent:* $\max_{\theta\in[0,1]^K} OT_{[\Delta r_\theta]_+}(\mathbb{P}_\pi, \mathbb{P}_{\tilde{\pi}}) = 0 \iff \mathbb{P}_\pi \succeq_{PSD} \mathbb{P}_{\tilde{\pi}}$.

As a result, PSD extends the risk-sensitive properties of stochastic dominance (Theorem 2.5) to the entire set of conical sum reward functions.

Corollary 3.4. *PSD guarantees that π exhibits better risk-sensitive performance than $\tilde{\pi}$ under r_θ for the set of measures in Theorem 2.5 and all conical sum reward functions, i.e., $\theta \geq \mathbf{0}$.*

While the inner minimization (over γ) in (5), $\text{OT}_{[\Delta \text{r}_\theta]_+}(\mathbb{P}_\pi, \mathbb{P}_{\tilde{\pi}})$, is a standard optimal transport linear program, the outer maximization (over θ) is of a convex function (of θ). This family of convex maximization programs is known to be NP-hard (Raghavachari, 1969), suggesting computational challenges for our specific instances, unfortunately.

3.2 Matched subdominance minimization

Given the apparent computational challenges of exactly verifying pluralistic stochastic dominance (not only for Figure 2(b,c), but all $\theta \geq \mathbf{0}$), we instead derive a computationally efficient upper bound. We approach this by "pushing" the maximization of θ deeper into the original PSD expression:

$$\max_{\theta\in[0,1]^K} \min_{\substack{\gamma\geq\mathbf{0} \text{ s.t.} \\ \sum_j \gamma_{i,j}=\mathbb{P}_\pi(\xi_i)\ \forall i \\ \sum_i \gamma_{i,j}=\mathbb{P}_{\tilde{\pi}}(\tilde{\xi}_j)\ \forall j}} \sum_{i,j} \gamma_{i,j} \underbrace{\left[\text{r}_\theta(\tilde{\xi}_j) - \text{r}_\theta(\xi_i)\right]_+}_{\text{subdom}_{\mathbf{1},\mathbf{0}}(\xi_i,\tilde{\xi}_j)}, \tag{6}$$

This replaces a set of optimal transport problems for each θ with one single optimal transport problem (Figure 2a) and makes numerous independent θ maximization problems that are easy to solve. Specifically, the resulting inner maximization of θ is equivalent to a specific instance of the **subdominance** (Ziebart et al., 2022) introduced for imitation learning via uniform dominance:

$$\text{subdom}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\xi, \tilde{\xi}) = \sum_k \left[\alpha_k \left(f_k(\tilde{\xi}) - f_k(\xi)\right) + \boldsymbol{\beta}\right]_+, \tag{7}$$

which measures how far trajectory ξ is from Pareto-dominating $\tilde{\xi}$ (by a margin β with features weighted by α). We define our relaxed optimization problem as a linear program using the more general form of subdominance (Def. 3.5).

Definition 3.5. Matched Subdominance Minimization given $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is obtained from:

$$\text{OT}_{\text{subdom}_{\alpha,\beta}}(\mathbb{P}_\pi, \mathbb{P}_{\tilde{\pi}}) = \min_{\substack{\gamma\geq\mathbf{0} \text{ s.t.} \\ \sum_j \gamma_{i,j}=\mathbb{P}_\pi(\xi_i)\ \forall i \\ \sum_i \gamma_{i,j}=\mathbb{P}_{\tilde{\pi}}(\tilde{\xi}_j)\ \forall j}} \sum_{i,j} \gamma_{i,j} \text{subdom}_{\alpha,\beta}(\xi_i, \tilde{\xi}_j). \tag{8}$$

This allows for a margin $\beta > 0$ requiring strict improvement and avoiding the trivial $\pi = \tilde{\pi}$ solution. The sets of trajectories should be sufficiently large to cover the distinct demonstrated behaviors. Given tens or hundreds of trajectories in each set, the optimal transport problem is not a critical computational bottleneck in practice.

As a result of being an upper bound, stochastic dominance can be guaranteed (Theorem 3.6).

Theorem 3.6. *For any $\alpha > 0$ and $\beta \geq 0$,*

$$OT_{subdom_{\alpha,\beta}}(\mathbb{P}_\pi, \mathbb{P}_{\tilde{\pi}}) = 0 \Longrightarrow \mathbb{P}_\pi \succeq_{PSD} \mathbb{P}_{\tilde{\pi}}.$$

We note that this is a special case of a recently established family of losses for which an optimal transport distance of zero implies multivariate stochastic dominance (Rioux et al., 2024).

3.3 Policy learning algorithms

We consider two imitation learning settings: **fully realizable** with any distribution over trajectories possible to learn; and **policy model** with a parametric policy model, π_θ, optimized.

In the fully realizable setting, we consider a set of candidate trajectories, ξ_i, (ideally from the Pareto frontier) and learn the imitator's distribution over those trajectories (Def. 3.7).

Definition 3.7. For a given candidate set of trajectories, fixed α weights, and a distribution of demonstrations, the **matched minimal subdominance imitator policy** is obtained from:

$$\min_{\gamma \geq 0} \sum_{i,j} \gamma_{i,j} \text{subdom}_{\alpha,\mathbf{1}}(\xi_i, \tilde{\xi}_j) + \lambda \text{Reg}(\gamma) \text{ such that: } \sum_i \gamma_{i,j} = \mathbb{P}_{\tilde{\pi}}(\tilde{\xi}_j) \ \forall j, \tag{9}$$

where regularizer $\text{Reg}(\gamma) = ||\gamma||$ or $\sum_i ||\gamma_{i,*}||$ encourages more uniform assignments and imitation trajectory distributions, respectively. The imitator trajectory distribution is then obtained by marginalizing: $\mathbb{P}_\pi(\xi_i) = \sum_j \gamma_{i,j}$.

Learned policy models enable generalization to different tasks within the same environment or to other environments. We leverage (deep) reinforcement learning methods (e.g., policy gradient optimization) using the subdominance-based optimal transport solution to determine a training signal for a policy model. This allows stochastically dominant policy optimization without first identifying a set of candidate trajectories (Def. 3.7). The model update procedure is described in Algorithm 1.

Algorithm 1 Policy model update

Input: M imitator samples $\{\xi_i\}$, N demonstrations $\{\tilde{\xi}_j\}$, policy/parameters π_ϕ, and learning rate η
Output: Updated policy/parameters π_ϕ

1: Set $\mathbb{P}_\pi(\xi_i) = \frac{1}{M}$
2: Solve $\text{OT}_{\text{subdom}}$ given $\mathbb{P}_\pi(\xi_i)$ and $\mathbb{P}_{\tilde{\pi}}(\tilde{\xi}_j)$ (Def. 3.2)
3: Construct training signals $\{a_i\}$ from OT solution
4: Update model parameters ϕ using variables $\mathbf{a}$ from (10) or (11): $\phi \leftarrow \phi + \eta \sum_{i=1}^M a_i \nabla_\phi \log \mathbb{P}_\pi(\xi_i)$

Step 4 of the Algorithm parallels policy gradient methods (Williams, 1992) with $\{a_i\}$ replacing other improvement signals. These are obtained from the OT matching, γ, using **demonstration normalization** (10) or **weighted best match** (11), which emphasizes the best trajectories more:

$$a_i = \sum_j \gamma_{i,j} \left(\text{subdom}(\xi_i, \tilde{\xi}_j) - \sum_{i'} \gamma_{i',j} \text{ subdom}(\xi_{i'}, \tilde{\xi}_j) \right), \tag{10}$$

$$a_i = \sum_j \left(\gamma_{i,j} \text{ subdom}(\xi_i, \tilde{\xi}_j) - \mathbb{I}\left[i = \text{argmin}_{i'} \text{subdom}(\xi_{i'}, \tilde{\xi}_j) \right] \sum_{i'} \gamma_{i',j} \text{ subdom}(\xi_{i'}, \tilde{\xi}_j) \right). \tag{11}$$

Additionally, the α values of the subdominances for optimal transport (8) can either remain fixed, as implied by Algorithm 1, or be be simultaneously updated using stochastic gradient optimization.

3.4 Generalization analysis

We characterize stochastic dominance guarantees for the population of demonstrations based on a finite, IID training sample using the Dvoretzky et al. (1956) inequality.

Theorem 3.8. *Given the cumulative mass function (CMF) in the K-dimensional reward feature space obtained by shifting the empirical demonstration CMF (with N IID sampled trajectories): $F_{\tilde{\pi}}^{N+}(\mathbf{f}) = \left[F_{\tilde{\pi}}^N(\mathbf{f}) - \epsilon\right]_+ + \epsilon \mathbb{I}[\mathbf{f} = \infty]$, and its corresponding probability mass function: $\mathbb{P}_{\tilde{\pi}}^{N+}(\mathbf{f})$, then:*

$$OT_{subdom_{\mathbf{1},\mathbf{0}}}(\mathbb{P}_\pi, \mathbb{P}_{\tilde{\pi}}^{N+}) = 0 \implies \mathbb{P}(\mathbb{P}_\pi \succeq_{PSD} \mathbb{P}_{\tilde{\pi}}) \geq 1 - NKe^{-2N\epsilon^2}.$$

This requires the convex hull of the Pareto frontier to be supported by a small number of points that each: have at least ϵ imitator probability; and Pareto dominate at least ϵ of $\mathbb{P}_{\tilde{\pi}}^{N+}$. Cases in which $\mathbb{P}_{\tilde{\pi}}^{N+}(\mathbf{f})$ assigns probability to "unrealizable" features $\mathbf{f}$ are also addressed in Appendix B.

4 Experiments

4.1 Baseline imitators and evaluation metrics

As baseline methods for comparison, we evaluate: Maximum Entropy Inverse Reinforcement Learning (MaxEnt IRL) (Ziebart, 2010); Linear Programming Apprenticeship Learning (LPAL) (Syed et al., 2008); Generative Adversarial Imitation Learning (GAIL) (Ho & Ermon, 2016); an oracle version of InfoGAIL (Li et al., 2017) trained on pre-determined mode clusters (InfoGAIL*); Risk-Averse Imitation Learning (RAIL) (Santara et al., 2018); and regret-based Bayesian Robust Optimization for Imitation Learning (BROIL) (Brown et al., 2020a; Javed et al., 2021). Additional experimental

details are described in Appendix C. We use withheld testing demonstrations to evaluate the diversity and quality of imitator policies in two ways:

Stochastic Dominance estimates whether the imitation reward distribution is better than the demonstration reward distribution. We randomly select a set of weight vectors to induce various reward functions. For each weight vector, we evaluate whether the imitation policy stochastically dominates the testing demonstrations (Def. 2.4) and report the rate of stochastic dominance for each approach.

Pareto Dominance estimates when imitation trajectories are unambiguously better than demonstrations. We use the exact imitator policy or randomly sample a set of rollouts. We measure $\mathbb{P}(\mathbf{f}(\xi_{\text{imit}}) \succeq \mathbf{f}(\tilde{\xi}_{\text{demo}}))$ for each demonstration and report the minimum, average, and maximum.

4.2 Illustrative grid world experiments

We first consider `Lava World`, a deterministic grid environment from the robust imitation literature (Brown et al., 2020a). Each trajectory starts from the same initial state and seeks to reach a fixed goal state in the bottom-right corner of the grid. At each time step, the agent can move in any of the four cardinal directions. Trajectories are characterized by two features: the number of white and red cells traversed (Figure 3). The cost of a trajectory is computed as a weighted sum of these features. A trajectory terminates either when the agent arrives at the goal state or when a fixed time horizon (e.g., 10) is reached.

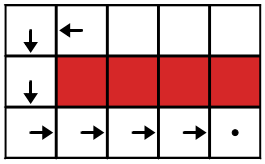

Figure 3: Sample demonstration in the `Lava World` grid environment with white and red (lava) grid cells.

While the cost feature weights for this environment are unknown, we are provided with a set of demonstrated trajectories. To train and evaluate our approach, we first divide a set of trajectories (with unique features) reaching the goal within 10 timesteps into imitator candidates (when on or near the Pareto frontier) and demonstrations (when less optimal). We then further divide the demonstrations into two random, equally-sized subsets for training and testing.

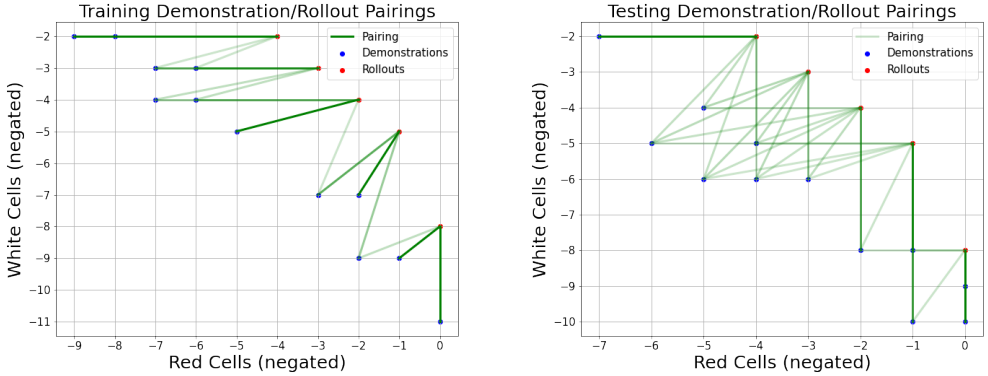

Figure 4: For training (left), demonstrations (blue) are paired with rollouts (red) via (9) to construct the imitator distribution. For testing (right), withheld demonstrators are paired with that imitator distribution via (8). Darker green pairings correspond to larger γ values.

We employ our fully-realizable training approach (Definition 3.7) for PSD. We first prune the candidate trajectory set by removing trajectories that are Pareto-dominated by others in the set. This ensures that only trajectories with potentially optimal rewards remain to define the imitator's policy. Subsequently, we match the training set demonstrations with the pruned candidate trajectories by solving a quadratic program based on Eq. (9) with fixed subdominance variables $\alpha = 1$ and $\beta = 0.5$, and L_2 regularization of the imitator trajectory distribution $\mathbb{P}(\xi)$ to promote greater uniformity over the set of candidates, resulting in improved generalization to unseen demonstrations. Figure 4 (left) shows this matching for a particular training sample.

To verify generalized stochastic dominance after training, withheld demonstrations are matched to the imitator's trajectory distribution using Equation (8). If the objective of the matching problem

is zero (equivalently, the imitation trajectories all dominate their paired demonstration trajectories), then stochastic dominance is guaranteed, as Figure 4 (right) shows.

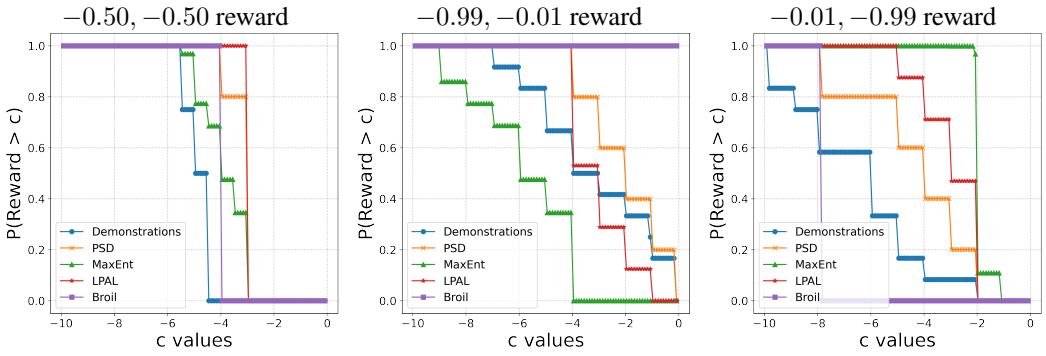

Figure 5: Cumulative rewards of the trajectory distributions of the demonstrations, PSD, MaxEnt IRL, and Regret methods for three reward functions.

Figure 5 shows the excess reward distributions for three reward functions. Other imitation methods produce curves that are worse for some portions of some reward functions, reflecting the trade-offs they make for better performance in other portions of these curves. In contrast, PSD produces excess reward curves that are strictly better than the demonstration curves for all three reward functions on the train-test split of Figure 4, indicating better mode coverage.

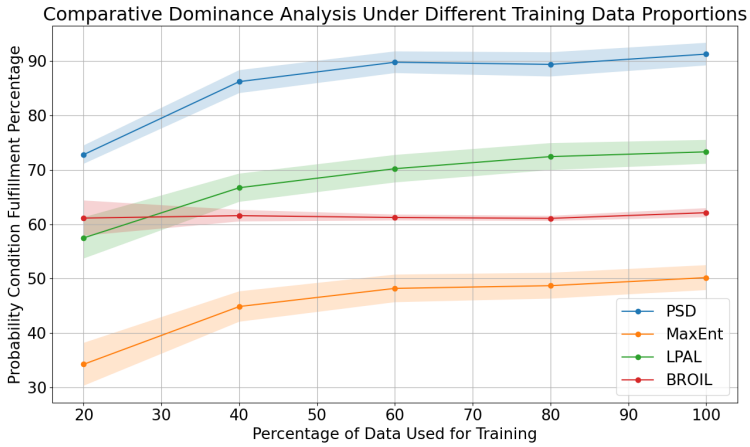

Figure 6: Average dominance (with standard error) for different training data amounts.

Figure 6 shows the percentage of randomly produced reward functions for which each imitation learning method stochastically dominates the withheld demonstration set, and how it changes with the amount of training demonstrations. PSD provides a higher rate of stochastic dominance across the entire range of training data sizes compared to other methods.

Table 1: Frequency of Pareto dominance over demonstrations in `Lava World`.

Policy	min	avg	max
Demonstrations	**0.08**	0.16	0.50
MaxEnt IRL	0.00	0.33	**1.00**
LPAL	0.00	0.46	**1.00**
BROIL	0.00	0.15	**1.00**
PSD	**0.08**	**0.47**	0.92

Table 1 provides summary statistics for how frequently demonstrations are Pareto dominated by the imitator's trajectory distribution (or withheld demonstrations as an additional baseline). PSD Pareto-dominates all demonstrations with at least 8% probability. In contrast, all other imitators fail to have any probability of Pareto dominating at least one demonstration. Additionally, PSD provides the highest average Pareto dominance, with nearly half of the imitator trajectories dominating (i.e., being unambiguously better) than the demonstrations.

4.3 Policy model optimization

Our second set of experiments considers policy model optimization (Alg. 1) in the `Point Bot` (Javed et al., 2021) and `Reacher` Todorov et al. (2012) environments. `Point Bot` is a continuous robotic task for navigating a point mass (subject to noisy, velocity-based air resistance) in a two-dimensional plane from a starting position to a pre-defined and stationary goal, ostensibly without passing through a gray region (obstacles). The robot moves by applying a force in a cardinal direction. `Reacher` is a robotic arm with two rigid links and two joints (Figure 7). The end of one link is fixed to the center of the environment. In our multi-modal variant, the goal is to move the robot's end effector to one of two targets (red or yellow) by applying appropriate sequences of torques.

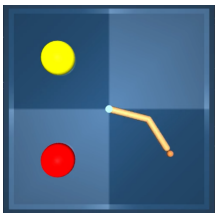

Figure 7: `Reacher` environment.

For the `Point Bot` environment, the imitators learn from the set of human-demonstrated trajectories shown in Figure 8 (left). Some demonstrations completely avoid the obstacles (gray areas), others partially avoid obstacles, while many appear entirely oblivious of obstacles. The trajectories are characterized by the number of timesteps in gray areas, the number of timesteps in white areas, and the sum of distances to the goal location over the trajectory. To facilitate multi-modal policy learning, we increase the number of layers of the policy model from two to four (each with 64 fully connected hidden nodes) compared to prior work (Javed et al., 2021). For PSD, we train this model using the *demonstration normalization* variant of Algorithm 1.

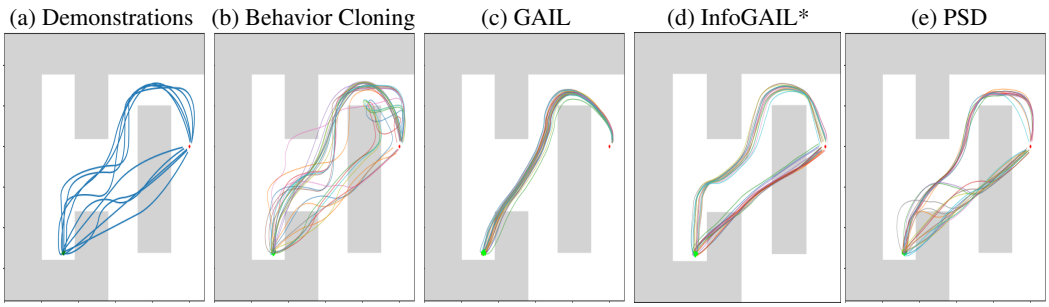

Figure 8: `Point Bot` training demonstrations starting from a lower left origin and moving to the goal in the upper right (a); and sample trajectories from policies learned using each method (b)-(e).

Behavior cloning produces trajectories that are similar to the demonstrations, but often of lower quality. Specifically, the sample trajectories in Figure 8b encounter more obstacles and sometimes fail to efficiently reach the goal. We initialize the other imitation learning methods from this behavior cloning policy as a starting point. GAIL (Figure 8c) suffers from mode collapse, ultimately producing trajectories that all encounter obstacles. InfoGAIL* (Figure 8d) is given a binary mode membership (obstacle-avoiding or obstacle-oblivious) of each trajectory and learns a separate model for each mode. By matching each demonstrated mode's means, the resulting trajectories tend to be suboptimal compared with the best demonstrations of the corresponding mode. PSD imitation (Figure 8e) produces trajectories that are of higher quality while still covering the spectrum of trade-offs between grays cells and white cells/distance of the demonstrations. Significant support is provided to these two modes: obstacle avoidance and obstacle obliviousness. However, some trajectories also cover trade-offs between the two modes (e.g., avoiding the first obstacle, but not the second). This illustrates the flexibility of PSD to operate in settings without clearly specified latent spaces of modes (e.g., binary-valued modes).

For the `Reacher` environment, demonstrations are synthetically produced using reinforcement learning—more specifically, the soft actor-critic (SAC) algorithm (Haarnoja et al., 2018)—to learn policies for two different targets, the yellow circle and the red circle. A SAC-learned policy (for the yellow or the red goal) and a uniformly random policy are sampled with complementary probabilities to produce the trajectories of the demonstration set with an equal number of red goal trajectories and yellow goal trajectories. The features we incorporate are the sum of distances from each of the targets over the entire trajectory. The policy model that the imitators train is a Gaussian multi-layer perceptron with four layers of 64 neurons. For PSD, we train this policy model using the *weighted best match* variant of Algorithm 1 and also report PSD with optimized α parameters (PSD-α^*), as described in §3.3.

Table 2: Frequency of imitator Pareto and stochastic dominance of demonstrations.

	`Point Bot`				`Reacher`			
	Pareto			Stochastic	Pareto			Stochastic
Policy	min	avg	max	avg	min	avg	max	avg
Demonstrations	0.000	0.222	0.444	0.000	0.000	0.144	0.333	0.000
Behavior Cloning	0.001	0.180	0.353	0.000	0.001	0.154	0.290	0.000
GAIL	0.000	0.015	0.031	0.002	0.000	0.005	0.023	0.425
RAIL	0.000	0.004	0.031	0.000	0.000	0.000	0.000	0.000
InfoGAIL*	0.000	0.227	0.496	0.000	0.409	0.474	0.500	0.071
PSD	0.070	0.326	0.493	0.420	0.452	0.498	**0.547**	0.561
PSD-α^*	**0.080**	**0.387**	**0.642**	**0.657**	**0.466**	**0.500**	0.534	**0.662**

Table 2 provides statistics for how well demonstrations are supported by the imitator policy (Pareto dominance) using 1000 policy rollouts, and whether the reward distributions of the imitator are strictly better than the demonstration reward distribution (stochastic dominance) averaged over 1000 random reward functions for both `Point Bot` and `Reacher`. Similarly to the fully realizable experiments, there are some demonstrations that are very difficult for the baseline methods to outperform (near zero minimum Pareto dominance values). The `Point Bot` demonstrations pose a significant challenge because some are distinct from the two main demonstrated modes, causing InfoGAIL* to also perform poorly in terms of minimum Pareto dominance. In contrast, the PSD policy produces trajectories that provide coverage of all demonstrations. More broadly, PSD excels across all metrics because it tends to produce trajectories that are: **in proportion** with the demonstrated behavior modes and often of **higher quality** than the demonstrations comprising that mode. PSD's high frequency of stochastic dominance illustrates the effectiveness of our policy gradient optimization in achieving the PSD objective. The other imitation methods are unfortunately unable to maintain the modes of the demonstrations and generally exhibit poor performance across all of these metrics, with a few exceptions, as a result.

5 Discussion and conclusions

This paper introduces stochastic dominance as an important property of distributional alignment (Sorensen et al., 2024) for imitation learning when demonstrations reflect the differing preferences of distinct demonstrators. Stochastic dominance provides stronger guarantees for demonstrators than expectation-matching imitation methods: reward distributions for each demonstrator that are at least as good as the demonstrated distribution—despite not knowing each demonstrator's exact reward function—for all common risk-sensitive measures. This avoids policies that are compromises between competing objectives by maintaining stochasticity. Though directly achieving stochastic dominance appears computationally difficult, we establish a relaxation using optimal transport theory that leads to exact algorithms in the fully-realizable setting and policy gradient algorithms when training a policy model. Through qualitative and quantitative analyses we show that our imitation learning approach provides support to all demonstrated behavior modes, while aiming to produce better quality behavior within those modes, leveraging concepts of both Pareto and stochastic dominance.

There are multiple important directions for future research. We have focused on deterministic dynamics in this paper. While our policy model optimizations naturally extend to stochastic environments, additional analyses and experimental validation remain as future work. Next, though hand-engineered reward features are reasonable for engineered systems (e.g., self-driving vehicles, robotics), many imitation learning methods learn reward functions without such features being available. Integrating reward feature learning in a manner that leverages potential multi-modality of demonstrations using our framework is an important future direction to avoid the limitation of known reward features. Finally, one key challenge is that policy optimization based on distributional criteria appears more challenging than maximizing expected rewards. Exploration of both on-policy and off-policy reinforcement learning in this context is likely needed for scaling to larger environments.

Acknowledgments

This work was supported by the National Science Foundation under award #2312955.

References

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 1, 2004.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Brown, D., Niekum, S., and Petrik, M. Bayesian robust optimization for imitation learning. *Advances in Neural Information Processing Systems*, 33:2479–2491, 2020a.

Brown, D. S., Goo, W., and Niekum, S. Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In *Conference on robot learning*, pp. 330–359. PMLR, 2020b.

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

Dadashi, R., Hussenot, L., Geist, M., and Pietquin, O. Primal Wasserstein imitation learning. In *International Conference on Learning Representations*, 2021.

Dvoretzky, A., Kiefer, J., and Wolfowitz, J. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics*, pp. 642–669, 1956.

Dvurechensky, P., Gasnikov, A., and Kroshnin, A. Computational optimal transport: Complexity by accelerated gradient descent is better than by sinkhorn's algorithm. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1367–1376. PMLR, 2018.

Fickinger, A., Cohen, S., Russell, S., and Amos, B. Cross-domain imitation learning via optimal transport. In *International Conference on Learning Representations*, 2022.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. 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.

Haldar, S., Mathur, V., Yarats, D., and Pinto, L. Watch and match: Supercharging imitation with regularized optimal transport. In *Conference on Robot Learning*, pp. 32–43. PMLR, 2023.

Ho, J. and Ermon, S. Generative adversarial imitation learning. *Advances in Neural Information Processing Systems*, 29, 2016.

Javed, Z., Brown, D. S., Sharma, S., Zhu, J., Balakrishna, A., Petrik, M., Dragan, A., and Goldberg, K. Policy gradient Bayesian robust optimization for imitation learning. In *International Conference on Machine Learning*, pp. 4785–4796. PMLR, 2021.

Kantorovich, L. V. On the translocation of masses. In *Dokl. Akad. Nauk. USSR (NS)*, volume 37, pp. 199–201, 1942.

Kostrikov, I., Nachum, O., and Tompson, J. Imitation learning via off-policy distribution matching. In *International Conference on Learning Representations*, 2019.

Lacotte, J., Ghavamzadeh, M., Chow, Y., and Pavone, M. Risk-sensitive generative adversarial imitation learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 2154–2163, 2019.

Learned-Miller, E. and DeStefano, J. A probabilistic upper bound on differential entropy. *IEEE Transactions on Information Theory*, 54(11):5223–5230, 2008.

Li, Y., Song, J., and Ermon, S. Infogail: Interpretable imitation learning from visual demonstrations. *Advances in neural information processing systems*, 30, 2017.

Luo, Y., Cohen, S., Grefenstette, E., and Deisenroth, M. P. Optimal transport for offline imitation learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Majumdar, A., Singh, S., Mandlekar, A., and Pavone, M. Risk-sensitive inverse reinforcement learning via coherent risk models. In *Robotics: Science and Systems*, volume 16, pp. 117, 2017.

Naaman, M. On the tight constant in the multivariate dvoretzky–kiefer–wolfowitz inequality. *Statistics & Probability Letters*, 173:109088, 2021.

Neu, G., Jonsson, A., and Gómez, V. A unified view of entropy-regularized Markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

Ng, A. Y. and Russell, S. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning*, volume 1, pp. 2, 2000.

Nguyen, T., Le, T., Dam, N., Tran, Q. H., Nguyen, T., and Phung, D. Tidot: A teacher imitation learning approach for domain adaptation with optimal transport. In *International Joint Conference on Artificial Intelligence 2021*, pp. 2862–2868, 2021.

Ogryczak, W. and Ruszczyński, A. From stochastic dominance to mean-risk models: Semideviations as risk measures. *European journal of operational research*, 116(1):33–50, 1999.

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

Raghavachari, M. On connections between zero-one integer programming and concave programming under linear constraints. *Operations Research*, 17(4):680–684, 1969.

Ratliff, N. D., Bagnell, J. A., and Zinkevich, M. A. Maximum margin planning. In *International Conference on Machine Learning*, pp. 729–736, 2006.

Rioux, G., Nitsure, A., Rigotti, M., Greenewald, K., and Mroueh, Y. Multivariate stochastic dominance via optimal transport and applications to models benchmarking. *arXiv preprint arXiv:2406.06425*, 2024.

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 627–635, 2011.

Santara, A., Naik, A., Ravindran, B., Das, D., Mudigere, D., Avancha, S., and Kaul, B. RAIL: Risk-Averse Imitation Learning. In *International Conference on Autonomous Agents and Multi-Agent Systems*, pp. 2062–2063, 2018.

Singh, S., Lacotte, J., Majumdar, A., and Pavone, M. Risk-sensitive inverse reinforcement learning via semi-and non-parametric methods. *The International Journal of Robotics Research*, 37(13-14): 1713–1740, 2018.

Sorensen, T., Moore, J., Fisher, J., Gordon, M., Mireshghallah, N., Rytting, C. M., Ye, A., Jiang, L., Lu, X., Dziri, N., Althoff, T., and Choi, Y. A roadmap to pluralistic alignment. *arXiv preprint arXiv:2402.05070*, 2024.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018. ISBN 0262039249.

Swamy, G., Choudhury, S., Bagnell, J. A., and Wu, S. Of moments and matching: A game-theoretic framework for closing the imitation gap. In *International Conference on Machine Learning*, pp. 10022–10032, 2021.

Syed, U. and Schapire, R. E. A game-theoretic approach to apprenticeship learning. *Advances in Neural Information Processing Systems*, 20, 2007.

Syed, U., Bowling, M., and Schapire, R. E. Apprenticeship learning using linear programming. In *International Conference on Machine Learning*, pp. 1032–1039, 2008.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

Wu, Y.-H., Charoenphakdee, N., Bao, H., Tangkaratt, V., and Sugiyama, M. Imitation learning from imperfect demonstration. In *International Conference on Machine Learning*, pp. 6818–6827. PMLR, 2019.

Xiao, H., Herman, M., Wagner, J., Ziesche, S., Etesami, J., and Linh, T. H. Wasserstein adversarial imitation learning. *arXiv preprint arXiv:1906.08113*, 2019.

Zhang, S., Cao, Z., Sadigh, D., and Sui, Y. Confidence-aware imitation learning from demonstrations with varying optimality. *Advances in Neural Information Processing Systems*, 34:12340–12350, 2021.

Ziebart, B., Choudhury, S., Yan, X., and Vernaza, P. Towards uniformly superhuman autonomy via subdominance minimization. In *International Conference on Machine Learning*, pp. 27654–27670. PMLR, 2022.

Ziebart, B. D. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims are motivated in theory in § 3 with experimental results provided in § 4. This paper's contribution is compared to other popular methods with a summary of results in Table 1 and Table 2 where we measure stochastic dominance and Pareto dominance.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: Though we do not create a separate section to enumerate limitations, all our assumptions (e.g., about the conical sum cost functions of given cost features) are stated when we discuss the motivation of our approach. Additionally, we discuss expanding beyond these limitations as potential future work in our conclusions section.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All theoretical claims have their proofs provided in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

Justification: We have outlined in the paper the information that we believe is necessary to reproduce the results. Furthermore, we plan on releasing code which would enable exact replication.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code with instructions will be uploaded in supplementary material, and also later publicly released with paper publication.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Although our contribution is mainly theoretical, a summary of these details is available in Appendix C. However, we understand this might not be enough for complex experiments. For this reason, we will publish the code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Although error bars are reported for only the fully-realizable setting, the results in Tables 1 and 2 are the average of multiple runs as described in § 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [No]

Justification: Detailed compute times and requirements are not provided. But experiments are simple enough to run on typical modern laptops as stated in Appendix C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The work does not utilize human subjects and uses simulated environments. Public code was used that is referenced. Public datasets were not used. Our code will be released along with the demonstrations it was tested on.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: No direct negative societal impact expected.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not utilize models or datasets with high risk of misuse. General safety concerns that apply to any AI algorithm or model apply.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Although license terms are not explicitly mentioned, the original creators of repositories upon which we build our work are credited along with links to original repositories, where the detailed license terms are available. All repositories that our work primarily builds upon are under MIT License.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The only new assets are some demonstrator trajectories that will be released alongwith the code.

    Guidelines:

    - The answer NA means that the paper does not release new assets.

- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We only use human trajectories for some of our experiments. These were generated by the authors themselves 'playing' the simulated RL environment.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Study does not involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs not involved as a core or important part of study.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.

- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Additional Background Material

A.1 Risk-sensitivity

Limiting the probability and magnitude of undesirable outcomes is crucial in many applications. Popular metrics for assessing these include *value at risk* (Def. A.1), *conditional value at risk* (Def. A.2), and *range value at risk* (Def. A.3).

Definition A.1. The **Value-at-Risk (VaR)** for a random reward variable X and given quantile $c \in [0, 1]$, is the inverse of the cumulative density function $F(X)$ (if it exists) (12) or the generalized inverse (e.g., for discrete variables) (13):

$$\text{VaR}_c(X) = \nu_c(X) = \begin{cases} F_X^{-1}(c) & (12) \\ \inf_x \{x \mid \mathbb{P}(X \leq x) \geq c\}. & (13) \end{cases}$$

Definition A.2. The **Conditional Value-at-Risk (CVaR)**, measures the expected value within the tail of a reward distribution for a given quantile α: $\text{CVaR}_c(X) = \mathbb{E}_X [X | X \leq \nu_c(X)]$.

Definition A.3. The **Range Value-at-Risk (RVaR)** discards both tails (defined by c and d) of the reward distribution and computes the expectation for the range in between: $\text{RVaR}_{c,d} = \mathbb{E}_X [X \mid \nu_c(X) \leq X \leq \nu_d(X)]$.

A.2 Stochastic Dominance

We provide an alternative definition of stochastic dominance (Definition 2.4) that motivates its use in imitation learning.

Definition A.4. Random variable X **stochastically dominates** random variable Y *if and only if* $\mathbb{E}[u(X)] \geq \mathbb{E}[u(Y)]$ for any non-decreasing utility function $u : \mathbb{R} \to \mathbb{R}$.

From an imitation learning perspective, this guarantees that a demonstrator or user with an unknown non-decreasing utility function will prefer the distribution of random rewards from variable X over the distribution of random rewards from variable Y.

A.3 Optimal Transport

We provide a proof sketch for Remark 2.6, which establishes a direct correspondence between stochastic dominance and optimal transport. For simplicity, we assume a one-to-one mapping between distributions, but the argument easily extends when the mapping is not one-to-one.

Proof sketch for Remark 2.6. The (generalized) inverse ν_c of Definition A.1, provides notation for defining a mapping between distributions for X and Y, namely using the pairing $(\nu_c(X), \nu_c(Y))\ \forall c \in [0, 1]$, which maps between values at the same quantile. If Y stochastically dominates X, then $\nu_c(X) \leq \nu_c(Y)\ \forall c \in [0, 1]$. This corresponds to an optimal transport solution with value 0 for cost functions that are 0 when $x \leq y$.

In the other direction, if the optimal transport solution has a value of 0, then for some complete set of pairings (c_1, c_2), $\nu_{c_1}(X) \leq \nu_{c_2}(Y)$. If $c_1 \neq c_2$ for some of these pairings, then given one pairing (c_a, c_x) such that $c_a > c_x$, there must be another pairing (c_b, c_y) such that $c_b < c_y$ and $c_b < c_a$ by the pigeonhole principle, resulting in $\nu_{c_b}(X) \leq \nu_{c_a}(X) \leq \nu_{c_y}(Y) \leq \nu_{c_x}(Y)$. Re-pairing these as (c_a, c_y) and (c_b, c_x) does not increase any cost as an optimal transport solution since $\nu_{c_a}(X) \leq \nu_{c_y}(Y)$ and $\nu_{c_b}(X) \leq \nu_{c_x}(Y)$, and brings the pairing closer to sorted matching. This re-pairing procedure can be continued until the resulting pairing is exactly $(\nu_c(X), \nu_c(Y))\ \forall c \in [0, 1]$. This pairing is exactly the definition of stochastic dominance (Definition 2.4). □

B Proofs of Theorems and Lemmas

Proof of Theorem 3.3. We first prove that $\max_{\theta \in [0,1]^K} \text{OT}_{[\Delta \text{r}_\theta]_+}(\mathbb{P}_\pi, \mathbb{P}_{\tilde{\pi}}) = 0 \implies P_\pi(r_\theta(\xi)) \succeq_{\text{PSD}} P_{\tilde{\pi}}(r_\theta(\tilde{\xi}))$. First, consider the optimal assignment for a specific $\theta \in [0, 1]^K$

in Definition 3.2: $\gamma^*(\theta)$. Since the corresponding objective value is assumed to be zero, this implies that:

$$\forall \gamma_{i,j}(\theta)^* > 0, \mathrm{r}_\theta(\xi_i) \geq \mathrm{r}_\theta(\tilde{\xi}_j) \tag{14}$$

$$\implies \gamma_{i,j}(\theta)^* \mathbb{I}\left[\mathrm{r}_\theta(\tilde{\xi}_j) \geq C\right] = \gamma_{i,j}(\theta)^* \mathbb{I}\left[\mathrm{r}_\theta(\tilde{\xi}_j) \geq C\right] \times \mathbb{I}\left[\mathrm{r}_\theta(\xi_i) \geq C\right], \forall C \in \mathbb{R}, \tag{15}$$

where $\mathbb{I}[x]$ is an indicator function that evaluates to 1 if expression x is true and 0 otherwise. We leverage this equality (15) in (b) below to prove the implication:

$$\begin{aligned}
\forall (C, \theta) \in \mathbb{R} \times [0,1]^K, \quad \mathbb{P}_\pi(\mathrm{r}_\theta(\xi) \geq C) &\triangleq \sum_{i,j} \gamma^*_{i,j}(\theta) \mathbb{I}\left[\mathrm{r}_\theta(\xi_i) \geq C\right] \\
&\overset{(a)}{\geq} \sum_{i,j} \gamma^*_{i,j}(\theta) \mathbb{I}\left[\mathrm{r}_\theta(\xi_i) \geq C\right] \times \mathbb{I}\left[\mathrm{r}_\theta(\tilde{\xi}_j) \geq C\right] \\
&\overset{(b)}{=} \sum_{i,j} \gamma^*_{i,j}(\theta) \mathbb{I}\left[\mathrm{r}_\theta(\tilde{\xi}_j) \geq C\right] \\
&\triangleq \mathbb{P}_{\tilde{\pi}}(\mathrm{r}_\theta(\tilde{\xi}) \geq C),
\end{aligned} \tag{16}$$

where inequality (a) results from adding an additional condition. This inequality is then generalized to all positive values of θ:

$$\begin{aligned}
&\forall (\alpha, \theta) \in \mathbb{R}_{\geq 0} \times [0,1]^K,\ \mathbb{P}_\pi(\alpha r_\theta(\xi) \geq \alpha C) \geq \mathbb{P}_{\tilde{\pi}}(\alpha r_\theta(\tilde{\xi}) \geq \alpha C) \\
\implies &\forall \theta \in \mathbb{R}^K_{\geq 0},\ \mathbb{P}_\pi(r_\theta(\xi) \geq C) \geq \mathbb{P}_{\tilde{\pi}}(r_\theta(\tilde{\xi}) \geq C).
\end{aligned}$$

In the other direction ($P_\pi(r_\theta(\xi)) \succeq_{\text{PSD}} P_{\tilde{\pi}}(r_\theta(\tilde{\xi})) \implies \max_{\theta \in [0,1]^K} \mathrm{OT}_{[\Delta \mathrm{r}_\theta]_+}(\mathbb{P}_\pi, \mathbb{P}_{\tilde{\pi}}) = 0$), our proof is constructive. For any $\theta \in \mathbb{R}^K_{\geq 0}$, sort the trajectories ξ_i and $\tilde{\xi}_j$ according to $\mathrm{r}_\theta(\cdot)$. Then choose $\gamma^*_{i,j}(\theta)$ that matches ξ_i and $\tilde{\xi}_j$ according to the sorted order with the weight based on the remaining unmatched probabilities of $\mathbb{P}_\pi(\xi_i)$ and $\mathbb{P}_{\tilde{\xi}}(\tilde{\xi}_j)$. Since $\mathbb{P}_\pi(\mathrm{r}_\theta(\xi) \geq C) \geq \mathbb{P}_{\tilde{\pi}}(\mathrm{r}_\theta(\tilde{\xi}) \geq C)$, then $\gamma_{i,j} > 0 \implies [\mathrm{r}_\theta(\tilde{\xi}_j) - \mathrm{r}_\theta(\xi_i)]_+ = 0$, so $\mathrm{OT}_{[\Delta \mathrm{r}_\theta]_+}(\mathbb{P}_\pi, \mathbb{P}_{\tilde{\pi}}) = 0$. This holds for all $\theta \in [0,1]^K$. □

Proof of Theorem 2.5. Leveraging the results of Theorem 3.3, we consider the $\gamma_{i,j}(\theta)$ for which $\sum_{i,j} \gamma_{i,j} \left[\mathrm{r}_\theta(\tilde{\xi}_j) - \mathrm{r}_\theta(\xi_i)\right]_+ = 0$. This implies that:

$$\forall (i,j), \gamma_{i,j}(\theta)\ \mathrm{r}(\xi_{i)}) \geq \gamma_{i,j}(\theta)\ \mathrm{r}_\theta(\tilde{\xi}_j) \tag{17}$$

$$\implies \sum_{i,j} \gamma_{i,j}(\theta)\ \mathrm{r}(\xi_{i)}) \geq \sum_{i,j} \gamma_{i,j}(\theta)\ \mathrm{r}_\theta(\tilde{\xi}_j) \tag{18}$$

$$\implies \mathbb{E}_{\xi \sim \mathbb{P}_\pi}\left[\mathrm{r}(\xi)\right] \geq \mathbb{E}_{\tilde{\xi} \sim \mathbb{P}_\pi}\left[\mathrm{r}_\theta(\tilde{\xi})\right]. \tag{19}$$

Further, using the definition of pluralistic stochastic dominance (Definition 3.1), we have: $\mathbb{P}_\pi \succeq_{\text{PSD}} \mathbb{P}_{\tilde{\pi}} \iff \forall \theta \in \mathbb{R}^K_{\geq \mathbf{0}}, C \in \mathbb{R}$:

$$\mathbb{P}_\pi \succeq_{\text{PSD}} \mathbb{P}_{\tilde{\pi}} \text{ iff: } \forall (\theta \in \mathbb{R}^K_{\geq \mathbf{0}}, C \in \mathbb{R}), \tag{20}$$

$$\mathbb{P}_\pi(\mathrm{r}_\theta(\xi) \geq C) \geq \mathbb{P}_{\tilde{\pi}}(\mathrm{r}_\theta(\tilde{\xi}) \geq C) \tag{21}$$

$$\mathbb{P}_\pi(\mathrm{r}_\theta(\xi) \leq C) \leq \mathbb{P}_{\tilde{\pi}}(\mathrm{r}_\theta(\tilde{\xi}) \leq C). \tag{22}$$

For convenience, we define these probabilities as p and q:

$$p = \mathbb{P}_\pi(\mathrm{r}_\theta(\xi) \leq C) \leq \mathbb{P}_{\tilde{\pi}}(\mathrm{r}_\theta(\tilde{\xi}) \leq C) = q \tag{23}$$

$$\nu_p(\mathrm{r}_\theta(\xi)) = C \qquad \text{and} \qquad \nu_q(\mathrm{r}_\theta(\tilde{\xi})) = C \tag{24}$$

but $p \leq q$ and as VaR monotonically increases with increasing confidence level (if $\alpha' \geq \alpha$ then $\nu_{\alpha'}(X) \geq \nu_\alpha(X)$):

$$\nu_q(\mathrm{r}_\theta(\xi)) = C' \geq C \tag{25}$$

As pluralistic stochastic dominance holds for all C, VaR guarantee holds for all α

$$\therefore, \qquad \text{VaR}_\alpha(\text{r}_\theta(\xi)) \geq \text{VaR}_\alpha(\text{r}_\theta(\tilde{\xi})) \qquad \forall \alpha \in [0,1]. \tag{26}$$

The proof for CVaR trivially follows from (26). If a function is always smaller than the other, its definite integral and average (with restricted domain) will also be smaller:

$$\rho_\alpha(\text{r}_\theta(\xi)) = \frac{1}{\alpha}\int_0^\alpha \nu_\gamma(\text{r}_\theta(\xi))\, d\gamma \geq \frac{1}{\alpha}\int_0^\alpha \nu_\gamma(\text{r}_\theta(\tilde{\xi}))\, d\gamma = \rho_\alpha(\text{r}_\theta(\tilde{\xi})) \tag{27}$$

$$\text{CVaR}_\alpha(\text{r}_\theta(\xi)) \geq \text{CVaR}_\alpha(\text{r}_\theta(\tilde{\xi})) \qquad \forall \alpha \in [0,1]. \tag{28}$$

And for the same reason, if we have different limits of the definite integral, the inequality still holds:

$$\eta_{\alpha,\beta}(\text{r}_\theta(\xi)) = \frac{1}{\beta-\alpha}\int_\alpha^\beta \nu_\gamma(\text{r}_\theta(\xi))\, d\gamma \geq \frac{1}{\beta-\alpha}\int_\alpha^\beta \nu_\gamma(\text{r}_\theta(\tilde{\xi}))\, d\gamma = \eta_{\alpha,\beta}(\text{r}_\theta(\tilde{\xi})) \tag{29}$$

$$\text{RVaR}_{\alpha,\beta}(\text{r}_\theta(\xi)) \geq \text{RVaR}_{\alpha,\beta}(\text{r}_\theta(\tilde{\xi})) \qquad 0 \leq \alpha \leq \beta \leq 1. \tag{30}$$

□

Lemma B.1. *The matched subdominance minimization value (Def. 3.5) upper bounds the worst-case reward difference optimal transport value:* $OT_{subdom_{\mathbf{1},\mathbf{0}}}(\mathbb{P}_\pi, \mathbb{P}_{\tilde{\pi}}) \geq \max_{\theta\in[0,1]} OT_{[\Delta r_\theta]_+}(\mathbb{P}_\pi, \mathbb{P}_{\tilde{\pi}})$.

Proof. Starting from Definition 3.2:

$$\max_{\theta\in[\mathbf{0},\mathbf{1}]} \min_{\gamma\succeq 0} \sum_{i,j} \gamma_{i,j} \left[\theta\cdot\mathbf{f}(\tilde{\xi}_j) - \theta\cdot\mathbf{f}(\xi_i)\right]_+ \overset{(a)}{\leq} \min_{\gamma\succeq 0} \max_{\theta\in[\mathbf{0},\mathbf{1}]} \sum_{i,j} \gamma_{i,j} \left[\theta\cdot\mathbf{f}(\tilde{\xi}_j) - \theta\cdot\mathbf{f}(\xi_i)\right]_+ \tag{31}$$

$$\overset{(b)}{\leq} \min_{\gamma\succeq 0} \sum_{i,j} \gamma_{i,j} \max_{\theta\in[\mathbf{0},\mathbf{1}]} \left[\theta\cdot\mathbf{f}(\tilde{\xi}_j) - \theta\cdot\mathbf{f}(\xi_i)\right]_+ \tag{32}$$

$$\overset{(c)}{=} \min_{\gamma\succeq 0} \sum_{i,j} \gamma_{i,j} \text{subdom}_{\mathbf{1},\mathbf{0}}(\xi_i, \tilde{\xi}_j) \tag{33}$$

$$\overset{(d)}{\leq} \min_{\gamma\succeq 0} \sum_{i,j} \gamma_{i,j} \text{subdom}_{\mathbf{1},\boldsymbol{\beta}}(\xi_i, \tilde{\xi}_j),$$

where: (a) follows from the maxmin-minmax inequality; (b) makes the maximizing choice of θ independently for each pair (i,j) with $\gamma_{i,j} > 0$; (c) results from the subdominance being the worst-case difference in rewards for the imitator; and (d) is nondecreasing as the subdominance margin increases. □

Proof of Theorem 3.6. Since the optimal matched subdominance (Definition 3.5) upper bounds the optimal pluralistic risk-sensitive matching (Definition 3.2) via Lemma B.1, an objective value of zero for the former implies an objective value of zero for the latter. Theorem 3.3 then implies pluralistic stochastic dominance. □

Figure 9 shows when minimizing the matched subdominance is unnecessary for pluralistic stochastic dominance (i.e., $\mathbb{P}_\pi \succeq_{\text{PSD}} \mathbb{P}_{\tilde{\pi}} \not\Longrightarrow \text{OT}_{\text{subdom}_{\mathbf{1},\mathbf{0}}}(\mathbb{P}_\pi, \mathbb{P}_{\tilde{\pi}}) = 0$).

Proof of Theorem 3.8. The right-sided multivariate Dvoretzky-Kiefer-Wolfowitz inequality (Dvoretzky et al., 1956; Naaman, 2021) provides probabilistic bounds on the deviation between the empirical multivariate cumulative mass function (CMF) with N samples ($F^N_{\tilde{\pi}}$) and the true population CMF (F) from the right as (See proof of Lemma 4.1 in (Naaman, 2021)):

$$\mathbb{P}\big(D_n^+ > \epsilon\big) = \mathbb{P}\left(\sup_{\mathbf{f}\in\mathbb{R}^K} (F^N_{\tilde{\pi}}(\mathbf{f}) - F(\mathbf{f})) > \epsilon\right) \leq NKe^{-2N\epsilon^2}, \tag{34}$$

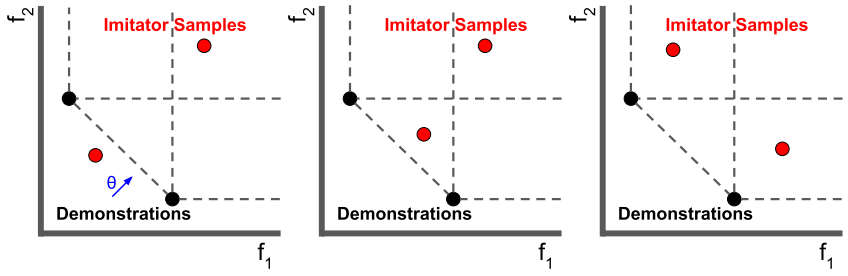

Figure 9: (left) Imitation not being a pluralistic stochastic dominator of demonstrations (and the θ maximizing the improvement violation); (center) imitation being a pluralistic stochastic dominator, but paired subdominance is nonzero; (right) imitation being a pluralistic stochastic dominator and paired subdominance is zero.

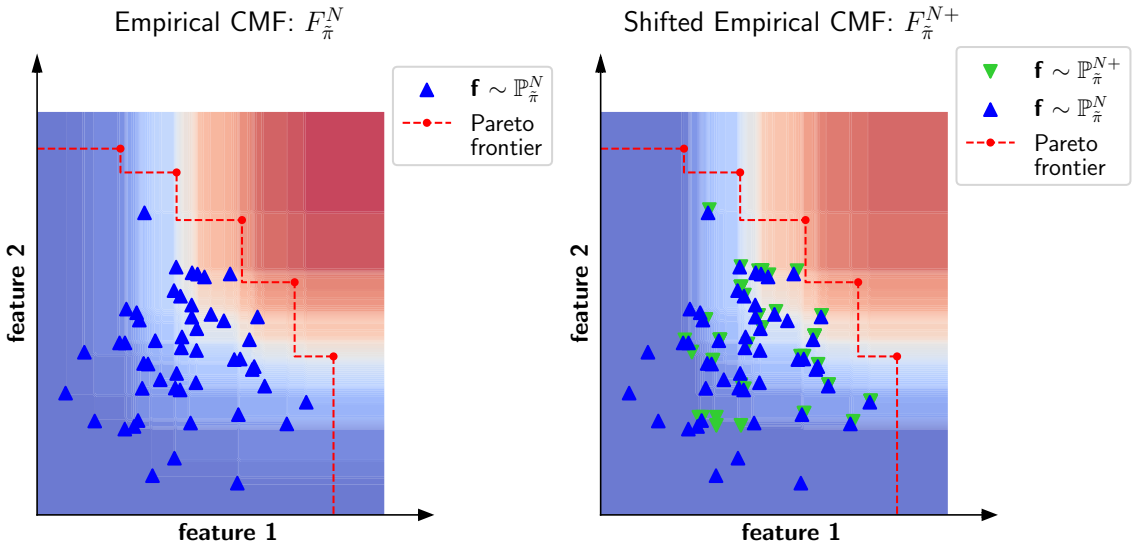

Figure 10: (Left) A sample population of demonstration feature vectors (in blue) in two dimensions overlaying the heatmap of its empirical CMF, and (Right) the heatmap of the empirical CMF shifted to the right by $\epsilon = 0.05$, along with feature vector samples from its corresponding PMF obtained via finite differences (in green).

which is the same as saying

$$\mathbb{P}\left(\sup_{\mathbf{f}\in\mathbb{R}^K}\left(F^N_{\tilde{\pi}}(\mathbf{f}) - F(\mathbf{f})\right) \leq \epsilon\right) \geq 1 - NKe^{-2N\epsilon^2} \tag{35}$$

since $\sup_{\mathbf{f}\in\mathbb{R}^K}\left(F^N_{\tilde{\pi}}(\mathbf{f}) - F(\mathbf{f})\right) \leq \epsilon \iff \forall \mathbf{f} \in \mathbb{R}^K, \left(F^N_{\tilde{\pi}}(\mathbf{f}) - F(\mathbf{f})\right) \leq \epsilon$, we have:

$$\mathbb{P}\left(\left(F^N_{\tilde{\pi}}(\mathbf{f}) - \epsilon\right) \leq F(\mathbf{f})\right) \geq 1 - NKe^{-2N\epsilon^2}\ \forall \mathbf{f} \in \mathbb{R}^K \tag{36}$$

Our approach considers the worst-case distribution within these bounds and characterizes when stochastic dominance of the worst-case can be guaranteed. This perspective is inspired by entropic analysis under these bounds (Learned-Miller & DeStefano, 2008).

Consider the worst-case CMF based on this bound:

$$F^{N+}_{\tilde{\pi}}(\mathbf{f}) = \left[F^N_{\tilde{\pi}}(\mathbf{f}) - \epsilon\right]_+ + \epsilon\mathbb{I}[\mathbf{f} = \infty], \tag{37}$$

where ϵ is reduced from every $\mathbf{f} \in \mathbb{R}^K$ and added to the largest possible value of $\mathbf{f}$. The corresponding probability mass function, $\mathbb{P}^{N+}_{\tilde{\pi}}(\mathbf{f})$ may be obtained from the CMF via finite differences.

If $\mathbb{P}^{N+}_{\tilde{\pi}}(\mathbf{f}) > 0$, but if $\mathbf{f}$ is not Pareto-dominated by any realizable demonstrator trajectory, we call $\mathbf{f}$ *unrealizable*. Let $u \triangleq \mathbb{P}^{+}_{\tilde{\pi}}(\mathbf{f} \text{ unrealizable})$ denote the probability under $\mathbb{P}^{N+}_{\tilde{\pi}}$ of cost features that are not possible to Pareto-dominate (or equal) by any realizable demonstrator trajectory. The sum of these unrealizable probabilities is an integer multiple of ϵ, to which we add one more ϵ (the one subtracted from the CMF at every input value). We assume that these unrealizable probabilities u get assigned, or "matched", to the worst-case realizable demonstrator trajectory ξ_{wc} during optimal transport. See Figure 10 for an example of the discussed quantities in two dimensions with $N = 50$ and $\epsilon = 0.05$ for the case when $u = 0$.

From (36), we see that the random variable with CMF $F^{N+}_{\tilde{\pi}}(\mathbf{f})$ stochastically dominates the random variable with CMF F by a margin of ϵ with probability at least $(1 - NKe^{-2N\epsilon^2})$.

Now if $\max_{\xi_{\text{wc}}} \text{OT}(\mathbb{P}_\pi - u\delta_{\xi_{\text{wc}}}, \mathbb{P}^{N+}_{\tilde{\pi}} - u\delta_{\xi_{\text{wc}}}) = 0$ for the remaining distributions after making the worst-case assignment of unrealizable probability to ξ_{wc}, in other words, if our approach achieves pluralistic stochastic dominance over the trajectory distribution of the worst-case CMF, i.e., $\mathbb{P}_\pi \succeq_{\text{PSD}} \mathbb{P}^{N+}_{\tilde{\pi}}$ (by Theorem 3.6), then by (36) we have pluralistic stochastic dominance over the demonstrator *population* trajectory distribution $\mathbb{P}_{\tilde{\pi}}$ with probability at least $(1 - NKe^{-2N\epsilon^2})$, i.e., $\mathbb{P}(\mathbb{P}_\pi \succeq_{\text{PSD}} \mathbb{P}_{\tilde{\pi}}) \geq (1 - NKe^{-2N\epsilon^2})$. □

C Additional Experimental Details

We provide additional experimental details in this section, including expanded interpretations of our evaluation measures (Section C.1), implementation details (Section C.2), and supplementary experimental evaluations (Section C.3).

C.1 Evaluation Measure Interpretations

Since assessing stochastic dominance across the entire set of conical sum reward functions directly is computationally challenging (Section 3.1), we instead assess the dominance of the imitator over the demonstrator from two different perspectives.

Our **stochastic dominance** measure considers the frequency of stochastic dominance over randomly sampled cost functions rather than the guarantee for the entire set of reward functions. For each sample reward function, stochastic dominance guarantees:

- That for all non-decreasing utility functions applied to the sampled reward function, the imitator is preferable to the demonstrations (Def A.4); and
- For any reward threshold, the imitator has an equal or higher probability of exceeding that threshold compared to the demonstration distribution (Def 2.4).

Our **Pareto dominance** evaluation measure considers uniform dominance (Definition 2.3) between sampled imitator trajectories and the demonstration set of trajectories, indicating how frequently the imitator trajectory is preferred for all conical sum reward functions. Large values across the entire set of samples indicate good alignment with the demonstrations. Low values do not prevent stochastic dominance (as shown in Figure 9, center), but they tend to indicate some degree of misalignment. We report the minimum, average, and maximum of these samples to assess how well the imitator trajectory samples align with demonstrations.

C.2 Implementation Details

Our implementation builds upon OpenAI Spinning Up[2], PG-BROIL (Javed et al., 2021)[3], and BROIL (Brown et al., 2020a)[4] repositories.

For Lava world experiments, for reporting the results comparing different approaches with different amounts of training data and frequency of imitator Pareto dominance, we have randomly split the

[2] `https://github.com/openai/spinningup`

[3] `https://github.com/zaynahjaved/pg-broil`

[4] `https://github.com/dsbrown1331/broil`

whole set containing 24 demonstrations into two equal splits of training and testing 100 times, and the represented results are averaged. We have used the time horizon of 10 for trajectories. For solving the Quadratic Program (QP), we have used MOSEK optimizer, and have set the regularization parameter λ to 0.001. For subdominance calculation during training, we have set the β parameter to 0.5. The threshold for achieving the goal in `Point Bot` was originally in the default setting of 1, however, we have increased it to 10. The initial features for this environment are the number of gray and white cells, however, we added another feature that takes into account the distance from the target.

We use REINFORCE algorithm for policy optimization. For our policy network, we have used a Gaussian Multi-Layer Perceptron (MLP) with 4 hidden layers each having 64 neurons with the Tanh activation function. The network receives the agent's observations and produces a mean and standard deviation for each action dimension, and the agent takes actions by sampling from this Gaussian distribution. For optimization, we used the Adam optimizer with a learning rate of $3e-5$, and for the subdominance calculation, we set the β parameter to 0.001. We solve the QP using MOSEK optimizer. Training goes on for 2000 iterations, and the best model is saved according to the lowest QP objective value. We use 10 demonstrations for training, they come from two main modes, with each mode having 5 demonstrations. During each iteration, we rollout 30 trajectories. For PSD-α^*, we have used Adam optimizer with a learning rate of $5e-4$ for learning alpha values. Alpha values are initialized uniformly, sum to 1, and are always at least equal to 0.1. Training goes on for 4000 iterations and similarly the best model is saved.

The `reacher` environment we use has two targets with fixed positions, one labeled with a yellow circle and the other red. We use two sets of demonstrations, each containing 15 trajectories, for two different modes of behavior. The modes include moving the robot's end effector to the yellow or red circle. For Reacher, we have used the same policy optimization algorithm we used for `Point bot`, REINFORCE. The policy network architecture and QP optimizer are also the same. We have used Adam optimizer with a learning rate of $3e$-3, a β parameter value of 0.001, and with the MOSEK regularization parameter λ set to 1000. During each iteration, we rollout 60 trajectories. Training goes on for 400 iterations and the best model is saved based on the lowest QP objective value. For PSD-α^*, we have used Adam optimizer with a learning rate of $1e-2$ for learning alpha values. Similar to `Point Bot`, alpha values are also initialized uniformly, sum to 1, and are at least 0.1. Training goes on for 1000 iterations and similarly the best model is saved.

For `Point Bot`, baseline experiments (GAIL, RAIL, InfoGAIL, BC) were run on several different personal computers and the slowest one took less than 5 hours to converge (e.g. on a laptop with 2.6GHz 10-core CPU, 32GB RAM). For `Reacher`, GAIL seemed to converge much earlier than `Pointbot`, taking close to an hour to converge (showing minimal improvements with longer training). RAIL was trained for 2-3 hours on `Reacher` but failed to display any improvement in our metrics. Experiments for PSD were run on an in-house server with GPU acceleration (equipped with two Nvidia GTX 1080 Ti GPUs), taking close to 1 hour and 1.5 hours each for convergence with `Pointbot` and `Reacher`, respectively.

C.3 Supplemental evaluations

We provide additional experimental results in this section.

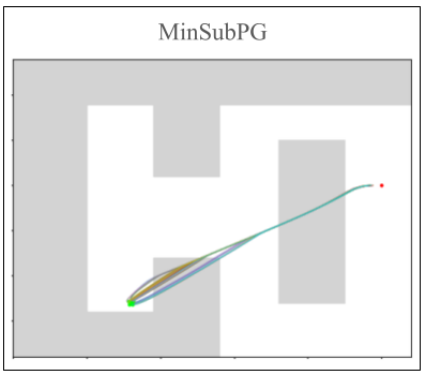

Figure 11: Trajectory samples from a policy learned using subdominance minimization (Ziebart et al., 2022).

Figure 11 shows the resulting trajectory samples from a policy learned using uniform subdominance minimization (Ziebart et al., 2022) in the `Point Bot` experiment of Section 4.3. Like GAIL (Ho & Ermon, 2016) (Figure 8(c)), it converges to a single mode of behavior. In this case, the mode is a "compromise" between the values reflected by the demonstrations. Unfortunately, since other modes are ignored, the resulting imitation policy does not provide good performance guarantees relative to the entire demonstration distribution.

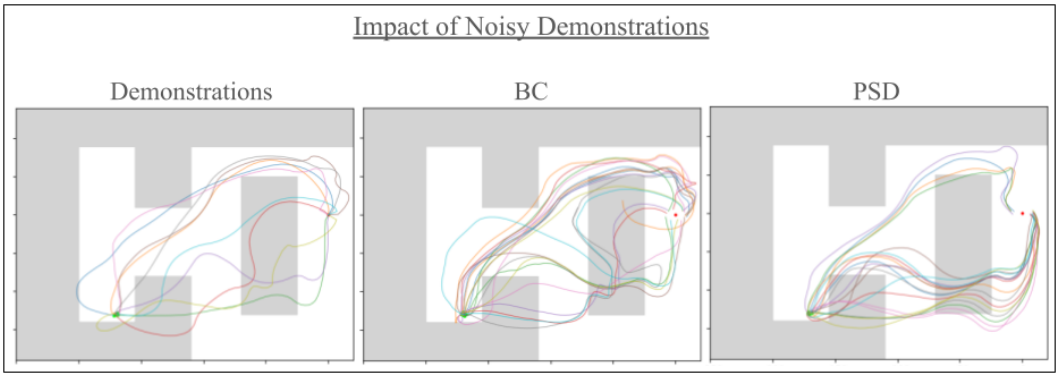

Figure 12: Noisy demonstration trajectories (left), trajectory samples from a behavioral-cloned policy (center) and trajectory samples from policy learned using PSD (right).

We next investigate imitation learning using a "noisier" set of demonstrations (Figure 12, left). Learning from these noisier demonstrations exacerbates the suboptimalities of behavior cloning (Figure 12, center), with far greater amounts of trajectory in the gray portions of the environment for many of the trajectories compared to the demonstrations (Figure 12, left) and to behavior cloning from less noisy demonstrations (Figure 8b). The PSD policy (Figure 12, right) produces better trade-offs of distance and obstacle with its trajectories, though with noticeable suboptimality compared to the PSD policy learned from less noisy data (Figure 8e). This suggests that incorporating a larger margin may be needed when learning from noisier demonstrations.