When Dynamics Shift, Robust Task Inference Wins: Offline Imitation Learning with Behavior Foundation Models Revisited

Rishabh Agrawal[1] **Rahul Jain**[1,2*] **Ashutosh Nayyar**[1]
[1]University of Southern California, [2]Google DeepMind
rishabha@usc.edu, rahul.jain@usc.edu, ashutosn@usc.edu

Abstract

Behavior Foundation Models (BFMs) enable scalable imitation learning (IL) by pretraining task-agnostic representations that can be rapidly adapted to new tasks. However, existing BFMs assume fixed environment dynamics, limiting their robustness under real-world shifts such as changes in friction, actuation, or sensor noise. We address this by formulating BFM task-inference as a robust minimax optimization problem, enabling adaptation to worst-case dynamics perturbations without modifying pretraining. To the best of our knowledge, this is the first BFM-based framework that achieves robustness to dynamics shifts while relying solely on offline data from a single nominal environment. Our approach significantly outperforms standard BFM and robust offline IL baselines under dynamics shifts. These results demonstrate that robust policies can be achieved entirely at task-inference time, improving the practicality of BFMs in dynamic settings.

1 Introduction

Deploying autonomous agents in complex, unstructured environments demands learning behavioral policies that generalize broadly without hand-crafted reward functions or controllers [26, 4]. Imitation Learning (IL), wherein an agent acquires behavior directly from expert demonstrations, removes the need for such hand design and has enabled remarkable successes in robotic manipulation [9], language generation [1, 48], and autonomous control [19]. Yet conventional IL methods learn each task in isolation, requiring retraining or large demonstration sets for every new behavior [23, 18], limiting their scalability.

Behavior Foundation Models (BFMs) [57, 58] address this scalability bottleneck by pretraining a single model on diverse exploratory offline data in an unsupervised manner, capturing task-agnostic environment representations via occupancy measures or successor measures. Forward-backward (FB) decompositions [57, 45] factorize these successor measures into policy-dependent and global components, yielding linear representations that generalize across tasks. At inference time, BFMs adapt rapidly to new tasks from limited demonstrations by optimizing a low-dimensional task embedding z, producing competitive policies by imitation within seconds to minutes without full retraining [45]. This *pretrain once, adapt anywhere* paradigm is a compelling solution for scalable multi-task IL in fixed environments.

However, BFMs fundamentally assume that the dynamics at pretraining and deployment coincide, an assumption routinely violated in practice due to sim-to-real discrepancies, mechanical wear, sensor noise, and environmental variation [56, 27]. Recent work demonstrates that standard BFMs fail to adapt when environment dynamics shift [13, 32]. The proposed remedies, dynamics-conditioned belief encoders [13] and memory-augmented architectures [32], both require pretraining data spanning

*Work done at University of Southern California.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

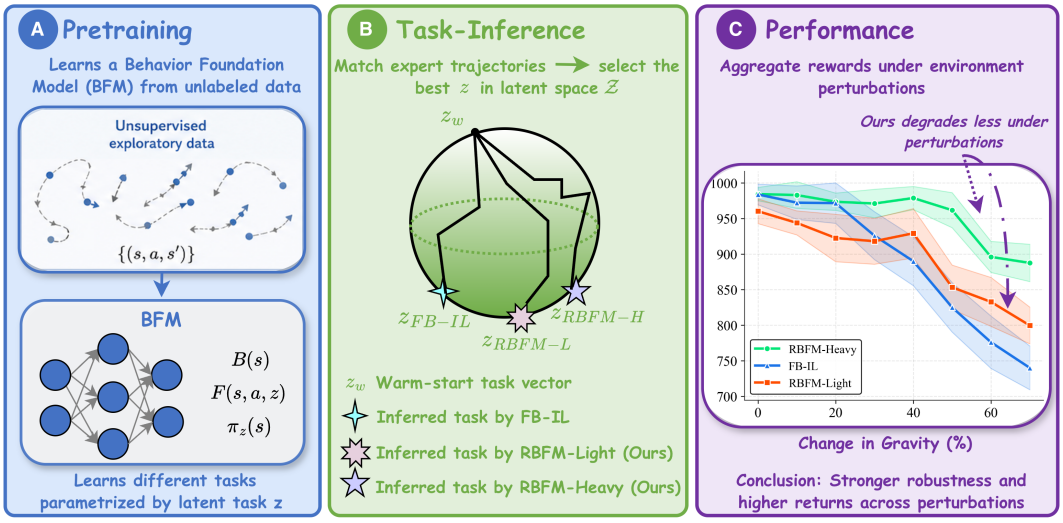


Figure 1: Overview of the proposed RBFM framework. FB-IL, RBFM-Light, and RBFM-Heavy share an identical pretraining procedure; robustness is introduced only at the task inference stage. RBFM variants produce policies significantly more robust to dynamic perturbations than FB-IL.

diverse dynamics variations to learn meaningful representations, a requirement that is expensive, time-consuming, and often infeasible in real-world settings [44, 7]. This raises the following question:

Can BFMs be made robust to environment dynamics shift when offline data from only a single nominal environment is available?

We answer affirmatively by formulating BFM task inference as a robust minimax optimization problem. Keeping the pretrained model fixed, we let an adversary perturb the environment within a specified uncertainty set while the learner minimizes the induced adversarial imitation loss, yielding a task embedding that hedges against plausible dynamics shifts. To preserve the fast task-inference characteristic of BFMs, we propose two variants (Figure 1): *RBFM-Light* augments the inference objective with a single robustness parameter optimized jointly with z, achieving robustness with minimal computational overhead. *RBFM-Heavy* further constrains the policy-induced occupancy measure to lie within the feasible distribution set, enforcing realizability and improving robustness at the cost of a larger optimization problem. Both variants require only lightweight modifications to the task-inference stage, with no changes to pretraining. Empirically, RBFM-Light and RBFM-Heavy consistently and substantially outperform standard BFMs (FB-IL) under diverse dynamics perturbations across multiple environments, while remaining competitive in the nominal setting (data-collecting environment) and, per task, orders of magnitude faster than single-task robust IL baselines such as DRBC [41] and BE-DROIL [3]. To summarize, our contributions are as follows:

1. We propose a robust formulation of BFM task inference via minimax optimization over an uncertainty set of environment dynamics, enabling robustness to dynamics shift without any modification to BFM pretraining.
2. We introduce *RBFM-Light* and *RBFM-Heavy*, two algorithms that trade computational efficiency against robustness while preserving the fast task-inference property of BFMs.
3. We demonstrate consistent and substantial robustness gains across three morphologically diverse environments under a wide range of perturbations. Our approach requires only a single exploratory dataset collected in the nominal setting for pretraining, unlike [13, 32], which rely on data from multiple diverse environments, and only a small set of expert demonstrations for task inference to produce a robust policy. Importantly, no expert data from perturbed environments is required at any stage.
4. To the best of our knowledge, this is the first work to achieve robustness to environment dynamics variation within the BFM framework without requiring pretraining or expert data from multiple dynamics regimes. This increases the range of real-world deployments for BFM-based methodology.

2 Preliminaries

Markov decision processes. We consider a discounted, reward-free Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \gamma, \mu)$. Here, $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $P(\mathrm{d}s' \mid s, a)$ is the transition kernel, $\gamma \in (0, 1)$ is the discount factor, and the initial state is drawn as $s_0 \sim \mu$. A policy π defines a distribution over actions conditioned on the state, denoted by $\pi(a|s)$. Starting from $s_0 \sim \mu$, trajectories $(s_t, a_t)_{t\geq 0}$ evolve according to $s_t \sim P(\mathrm{d}s_t \mid s_{t-1}, a_{t-1})$ and $a_t \sim \pi(\cdot \mid s_t)$.

Successor measure. To characterize the evolution of future states under a policy π, we consider the successor measure $M^\pi(\cdot \mid s, a)$ when taking action a in state s, which aggregates discounted future state visitation. For any measurable set $X \subset \mathcal{S}$,

$$M^\pi(X \mid s, a) = \sum_{t=0}^{\infty} \gamma^t \mathbb{P}_\pi(s_{t+1} \in X \mid s, a). \tag{1}$$

This construction naturally separates the effect of the dynamics from the specification of a task. In particular, for any reward function $r : \mathcal{S} \to \mathbb{R}$, the associated Q-function can be written as

$$Q_r^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_{t+1}) \,\middle|\, s, a \right] = \int_{\mathcal{S}} M^\pi(\mathrm{d}s' \mid s, a)\, r(s'). \tag{2}$$

Thus, once the successor measure is known, value functions for different rewards can be obtained without modifying the policy-dependent component.

Forward–backward (FB) parameterization. In practice, directly representing M^π is intractable in large or continuous state spaces. A common approach is therefore to approximate it using a low-rank decomposition. Concretely, Behavior Foundation Models (BFMs) represent M^π as

$$M^\pi(X \mid s, a) \approx \int_{s' \in X} F^\pi(s, a)^\top B(s')\, \phi(\mathrm{d}s'), \tag{3}$$

where $F^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ and $B : \mathcal{S} \to \mathbb{R}^d$ denote forward and backward embeddings, respectively, and ϕ is a reference distribution over states. Under this factorization, the dependence on the reward reduces to a linear projection:

$$Q_r^\pi(s, a) = F^\pi(s, a)^\top z, \qquad z = \mathbb{E}_{s\sim\phi}[B(s)\, r(s)]. \tag{4}$$

To enable generalization across tasks, both the policy and the forward embedding are conditioned on a latent vector $z \in \mathcal{Z} \subseteq \mathbb{R}^d$ [58]. This yields the parameterization

$$\begin{cases} M^{\pi_z}(X \mid s, a) \approx \int_{s' \in X} F(s, a, z)^\top B(s')\, \phi(\mathrm{d}s'), \\ \pi_z(s) = \arg\max_a F(s, a, z)^\top z. \end{cases} \tag{5}$$

In practice, the latent space $\mathcal{Z}$ is typically constrained (e.g., to a bounded subset of $\mathbb{R}^d$). In the *pre-training* stage, the embeddings are learned from an offline dataset with distribution ϕ by minimizing a temporal-difference objective derived from the Bellman equation [58]:

$$\begin{aligned} \mathcal{L}_{\mathrm{FB}}(F, B) = \mathbb{E}_{\substack{z\sim\nu,(s,a,s')\sim\phi \\ s^+\sim\phi,\, a'\sim\pi_z(\cdot|s')}} \Big[\big(F(s, a, z)^\top B(s^+) - \gamma \bar{F}(s', a', z)^\top \bar{B}(s^+) \big)^2 \Big] \\ - 2\, \mathbb{E}_{z\sim\nu,(s,a,s')\sim\phi} \big[F(s, a, z)^\top B(s') \big], \end{aligned} \tag{6}$$

where $\bar{F}$ and $\bar{B}$ denote stop-gradient operators and ν is a distribution over latent variables. For continuous action spaces, the maximization in π_z is replaced by an actor network trained to maximize:

$$\mathcal{L}_{\mathrm{actor}}(\pi) = -\mathbb{E}_{z\sim\nu,s\sim\phi,\, a\sim\pi_z(\cdot|s)} \big[F(s, a, z)^\top z \big]. \tag{7}$$

The overall procedure consists of two phases. In the first phase (pretraining), we learn the forward and backward embeddings, along with the actor (when applicable), offline from a fixed exploratory dataset by minimizing $\mathcal{L}_{\mathrm{FB}}$ and $\mathcal{L}_{\mathrm{actor}}$, without access to task-specific rewards. In the second phase, we perform task-inference by selecting a latent variable z that induces a policy π_z. The latent z can be inferred from rewards, goals, or expert demonstrations; since our focus is imitation learning, in this work, we will consider the reward-free setting with expert demonstration data.

Learning from demonstrations. Since no reward function is available for task-inference, learning must rely solely on expert demonstrations, without any additional interaction with the environment in the offline setting. To formalize this, we describe policies through their discounted occupancy measures under a nominal transition function T^o as $\rho^\pi_{T^o}(s,a,s') = (1-\gamma)\,\mathbb{E}_\pi\left[\sum_{t=0}^\infty \gamma^t\,\mathbf{1}_{s_t=s,a_t=a,s_{t+1}=s'}\right]$. The marginal state–action and state-only occupancy measures are then given by $\rho^\pi_{T^o}(s,a) = \sum_{s'}\rho^\pi_{T^o}(s,a,s')$ and $\rho^\pi_{T^o}(s) = \sum_{a,s'}\rho^\pi_{T^o}(s,a,s')$ respectively. Given a dataset of trajectories generated by an expert policy π_D, the goal is to identify a latent task-vector z such that the induced policy π_z matches the expert behavior. A natural objective is

$$\arg\min_{z\in\mathcal{Z}}\ \mathbb{E}_{s\sim\rho^{\pi_D}_{T^o}}\left[\mathcal{L}(\pi_z(\cdot|s),\pi_D(\cdot|s))\right], \tag{8}$$

where $\mathcal{L}$ is the *imitation loss* (e.g., mean squared error or KL divergence) measuring the gap between learner and expert policies. This is the FB-IL objective of [45], our primary baseline.

Robust formulation. Finally, we account for uncertainty in the transition dynamics by introducing an ambiguity or uncertainty set over transition kernels. Specifically, we consider

$$\mathcal{T}_{T^o}(\varepsilon') = \left\{(T_{s,a})_{(s,a)\in\mathcal{S}\times\mathcal{A}}\ \middle|\ T_{s,a}\in\Delta(\mathcal{S}),\ D_{\mathrm{TV}}\big(T_{s,a},T^o_{s,a}\big)\le\varepsilon',\ \forall(s,a)\in\mathcal{S}\times\mathcal{A}\right\}, \tag{9}$$

where $D_{\mathrm{TV}}(p,q) = \sup_A|p(A)-q(A)|$ is the total variation distance; the set is centred on the nominal kernel T^o.

This induces a distributionally robust objective in which performance is evaluated under the worst-case transition model within the ambiguity set. Formally, we consider [3]

$$\arg\min_{z\in\mathcal{Z}}\ \max_{T\in\mathcal{T}_{T^o}(\varepsilon')}\ \mathbb{E}_{s\sim\rho^{\pi_D}_{T}}\left[\mathcal{L}(\pi_z(\cdot|s),\pi_D(\cdot|s))\right], \tag{10}$$

where $\rho^{\pi_D}_T$ denotes the discounted state distribution induced by the expert policy π_D under transition kernel $T\in\mathcal{T}_{T^o}(\varepsilon')$. The objective admits a natural minimax interpretation: an adversary selects a transition model $T\in\mathcal{T}_{T^o}(\varepsilon')$ that maximizes the imitation loss, while the learner chooses a latent variable z (and corresponding induced-policy π_z) to minimize this worst-case loss. Crucially, this game is played in the offline regime, where only trajectories generated by π_D under T^o are available.

3 Fast Robust Offline Imitation Learning

We propose two complementary formulations for robust and computationally efficient task-inference of pre-trained Behavioral Foundation Models (BFMs) under transition uncertainty. The first variant, **RBFM-Light** (§3.1), relaxes dynamics perturbations into a state-occupancy uncertainty set, yielding a tractable offline objective. The second variant, **RBFM-Heavy** (§3.2), enforces robustness through explicit Bellman flow constraints, leading to less conservative policies at the expense of increased computational complexity.

3.1 RBFM-Light

We begin by characterizing the sensitivity of the discounted state-occupancy measure $\rho^\pi_T(s)$ to perturbations in the transition kernel T.

Lemma 1. *For any policy π and transition kernel $T\in\mathcal{T}_{T^o}(\varepsilon')$, the following holds:*

$$D_{\mathrm{TV}}(\rho^\pi_T(s),\rho^\pi_{T^o}(s)) \le \frac{\gamma\varepsilon'}{1-\gamma}.$$

Lemma 1 provides a uniform bound on the deviation between occupancy measures induced by perturbed and nominal dynamics. This enables replacing uncertainty over transition kernels with an uncertainty set defined directly in occupancy space, yielding a tractable relaxation of the robust objective in (10). Concretely, we define $\mathcal{D}^{\pi_D}_{\mathrm{light}}(\varepsilon_l) = \left\{\rho^{\pi_D}_T(s) : D_{\mathrm{TV}}(\rho^{\pi_D}_T(s),\,\rho^{\pi_D}_{T^o}(s)) \le \frac{\gamma\varepsilon'}{1-\gamma} = \varepsilon_l\right\}$, which contains all occupancy measures induced by admissible kernels $T\in\mathcal{T}_{T^o}(\varepsilon')$. By construction, optimizing over $\mathcal{D}^{\pi_D}_{\mathrm{light}}(\varepsilon_l)$ upper bounds the original worst-case objective over $\mathcal{T}_{T^o}(\varepsilon')$.

This reformulation removes explicit dependence on the unknown dynamics and provides a purely offline optimization problem. The resulting objective is given by

$$\text{RBFM-Light} := \min_{z} \max_{\rho_T^{\pi_D} \ll \rho_{T^o}^{\pi_D}} \mathbb{E}_{s\sim\rho_T^{\pi_D}}[\mathcal{L}(\pi_z(\cdot|s), \pi_D(\cdot|s))] \quad \text{s.t.} \quad D_{\text{TV}}(\rho_T^{\pi_D}(s) \,\|\, \rho_{T^o}^{\pi_D}(s)) \le \varepsilon_l. \tag{11}$$

The inner maximization admits a tractable dual reformulation, yielding the equivalent optimization problem provided in Proposition 1.

Proposition 1. *The solution to the constrained optimization problem in* (11) *is given by solving the following, where* $(x)_+ := \max(x, 0)$*:*

$$\min_{z} \min_{\lambda\in\mathbb{R}} \left\{ \mathbb{E}_{s\sim\rho_{T^o}^{\pi_D}}\left[(\mathcal{L}(\pi_z(\cdot|s), \pi_D(\cdot|s)) - \lambda)_+\right] + \varepsilon_l \left(\max_{\rho_{T^o}^{\pi_D}(s)>0} \mathcal{L}(\pi_z(\cdot|s), \pi_D(\cdot|s)) - \lambda \right)_+ + \lambda \right\}$$

The proof of Proposition 1 is provided in Appendix A. Objective defined in Proposition 1 can be optimized to solve the robust imitation learning problem in (10), using the offline data only available from the nominal environment T^o. We further specialize the objective under standard assumptions commonly used in offline imitation learning.

Proposition 2. *Suppose that the expert and learner policies* π_D *and* π_z *are deterministic and that the action space* $\mathcal{A}$ *is bounded. Let* $\mathcal{L}(\pi_z(\cdot|s), \pi_D(\cdot|s)) = \|\pi_z(s) - \pi_D(s)\|_2^2$ *and* $L := \sup_{(s,a)\sim\rho_{T^o}^{\pi_D}} \|\pi_z(s) - \pi_D(s)\|_2^2$*. Then the optimization problem in Proposition 1 can be simplified to*

$$\min_{z} \min_{\lambda\in[0,(1+\varepsilon_l)L]} \left\{ \mathbb{E}_{s\sim\rho_{T^o}^{\pi_D}}\left[(\mathcal{L}(\pi_z(\cdot|s), \pi_D(\cdot|s)) - \lambda)_+\right] + \varepsilon_l \left(\max_{\rho_{T^o}^{\pi_D}(s)>0} \mathcal{L}(\pi_z(\cdot|s), \pi_D(\cdot|s)) - \lambda \right)_+ + \lambda \right\}$$

Proposition 2 restricts λ to a bounded interval. Since λ is the dual variable of the inner maximization and not a tuned hyperparameter, we find it by a one-dimensional search over $[0, (1 + \varepsilon_l)L]$ at every dual update. Proofs are in Appendix A.

3.2 RBFM-Heavy

While RBFM-Light provides a tractable offline solution by operating directly in occupancy space, it allows any state-occupancy measure within total variation distance ε_l of the nominal one. In particular, the uncertainty set $\mathcal{D}_{\text{light}}^{\pi_D}(\varepsilon_l)$ may include occupancy measures that assign probability mass to transitions that are not realizable under the expert policy in the underlying MDP. This can lead to overly conservative solutions. To address this limitation, we refine the uncertainty set by incorporating structural constraints that ensure realizability. We begin by considering the extension of Lemma 1 to joint state-action-next-state occupancy measures.

Lemma 2. *For any policy* π *and transition kernel* $T \in \mathcal{T}_{T^o}(\varepsilon')$*, the following holds:*

$$D_{\text{TV}}(\rho_T^{\pi}(s, a, s'), \rho_{T^o}^{\pi}(s, a, s')) \le \frac{\varepsilon'}{1-\gamma}.$$

Building on Lemma 2, we enforce a *Bellman flow conservation* condition [47, 8], ensuring that $\rho_T^{\pi_D}(s, a, s')$ corresponds to a valid stationary occupancy measure induced by the expert policy π_D under transition kernel T. This preserves the temporal structure of the MDP and rules out infeasible perturbations. In addition, we generalize the divergence constraint by replacing the TV distance with an f-divergence [6, 20] (see Appendix D.3 for details). Upon incorporating these changes, we obtain the following constrained minimax formulation:

$$\begin{aligned} \text{RBFM-Heavy} := \min_{z} \max_{\rho_T^{\pi_D}\ge 0} \; & \mathbb{E}_{s\sim\rho_T^{\pi_D}}[\mathcal{L}(\pi_z(\cdot|s), \pi_D(\cdot|s))] \\ \text{s.t.} \quad & (1) D_f(\rho_T^{\pi_D}(s, a, s') \,\|\, \rho_{T^o}^{\pi_D}(s, a, s')) \le \varepsilon, \\ & (2) \sum_{s'} \rho_T^{\pi_D}(s, a, s') = (1-\gamma)\mu(s)\pi_D(a|s) + \gamma\pi_D(a|s) \sum_{\tilde{s},\tilde{a}} \rho_T^{\pi_D}(\tilde{s}, \tilde{a}, s) \quad \forall (s, a). \end{aligned} \tag{12}$$

To solve (12), we fix the task vector z (hence the corresponding induced policy π_z) and consider the corresponding dual formulation:

$$\min_{Q,\tau\geq 0}\ \max_{\rho_T^{\pi_D}\geq 0}\ \mathbb{E}_{s\sim\rho_T^{\pi_D}}\left[L_{\pi_z}(s)\right]-\tau\big(D_f\big(\rho_T^{\pi_D}(s,a,s')\ \big\|\ \rho_{T^o}^{\pi_D}(s,a,s')\big)-\varepsilon\big) \\ -\sum_{s,a}Q(s,a)\left[\sum_{s'}\rho_T^{\pi_D}(s,a,s')-(1-\gamma)\mu(s)\pi_D(a|s)-\gamma\pi_D(a|s)\sum_{\tilde{s},\tilde{a}}\rho_T^{\pi_D}(\tilde{s},\tilde{a},s)\right]. \tag{13}$$

where $L_{\pi_z}(s)=\mathcal{L}(\pi_z(\cdot|s),\pi_D(\cdot|s))$, $Q(s,a)\in\mathbb{R}$ is the Lagrange multiplier for the Bellman flow constraint, and $\tau\geq 0$ is the multiplier for the f-divergence constraint. Since the inner maximization in (12) is convex and admits a strictly feasible point (e.g., $\rho_{T^o}^{\pi_D}$), Slater's condition [15] implies strong duality. Thus, (13) attains the same optimal value as the original problem. After algebraic simplification, we obtain:

$$\min_{Q,\tau\geq 0}\ \max_{\rho_T^{\pi_D}\geq 0}\ (1-\gamma)\,\mathbb{E}_{s\sim\mu,\,a\sim\pi_D(\cdot|s)}[Q(s,a)]-\tau\,D_f(\rho_T^{\pi_D}(s,a,s')\,\|\,\rho_{T^o}^{\pi_D}(s,a,s'))+\varepsilon\tau \\ +\mathbb{E}_{s,a,s'\sim\rho_T^{\pi_D}}\Big[L_{\pi_z}(s)+\gamma\,\mathbb{E}_{a'\sim\pi_D(\cdot|s')}[Q(s',a')]-Q(s,a)\Big]. \tag{14}$$

This formulation still depends on the unknown occupancy $\rho_T^{\pi_D}(s,a,s')$ for arbitrary $T\in\mathcal{T}_{T^o}(\varepsilon')$, and is therefore not directly amenable to offline optimization. To address this, we introduce the importance ratio $w(s,a,s')=\frac{\rho_T^{\pi_D}(s,a,s')}{\rho_{T^o}^{\pi_D}(s,a,s')}$. Using the definition of f-divergence, we rewrite (14) as

$$\min_{Q,\tau\geq 0}\ \max_{w\geq 0}\ (1-\gamma)\,\mathbb{E}_{s\sim\mu,\,a\sim\pi_D(\cdot|s)}[Q(s,a)]+\varepsilon\tau \\ +\mathbb{E}_{s,a,s'\sim\rho_{T^o}^{\pi_D}}\left[-\tau\,f(w(s,a,s'))+w(s,a,s')\,c_{Q,\pi_z}(s,a,s')\right], \tag{15}$$

where $c_{Q,\pi_z}(s,a,s')=L_{\pi_z}(s)+\gamma\,\mathbb{E}_{a'\sim\pi_D(\cdot|s')}[Q(s',a')]-Q(s,a)$.

This reformulation transforms the original maximization over unknown occupancy measures $\rho_T^{\pi_D}$ for any $T\in\mathcal{T}_{T^o}(\varepsilon')$ into an objective that depends only on samples from the nominal distribution $\rho_{T^o}^{\pi_D}$. As a result, the problem becomes tractable in a purely offline setting using expert demonstrations alone. We now turn to solving the inner maximization over the importance weights w, which admits a closed-form solution.

Proposition 3. *For $\tau>0$, the inner maximization in* (15) *admits the solution, with f' the derivative of the generator f,*

$$w^\star_{Q,\tau,\pi_z}(s,a,s')=\max\left(0,\ (f')^{-1}\left(\frac{c_{Q,\pi_z}(s,a,s')}{\tau}\right)\right).$$

For $\tau=0$, $w^\star_{Q,\tau,\pi_z}(s,a,s')=+\infty$ if $c_{Q,\pi_z}(s,a,s')>0$ and 0 otherwise.

The proof of Proposition 3 is provided in Appendix A. Substituting $w^\star_{Q,\tau,\pi_z}$ into (15) reduces the problem to a minimization over Q and τ. This results in a bilevel-like optimization with two coupled subproblems: estimating (Q,τ) and updating the task parameters.

$$\min_{Q,\tau\geq 0}\quad (1-\gamma)\,\mathbb{E}_{s\sim\mu,\,a\sim\pi_D(\cdot|s)}[Q(s,a)]+\varepsilon\tau \\ +\mathbb{E}_{(s,a,s')\sim\rho_{T^o}^{\pi_D}}\left[-\tau f\big(w^\star_{Q,\tau,\pi_z}(s,a,s')\big)+w^\star_{Q,\tau,\pi_z}(s,a,s')\,c_{Q,\pi_z}(s,a,s')\right] \tag{16}$$

$$\min_{z}\quad \mathbb{E}_{(s,a,s')\sim\rho_{T^o}^{\pi_D}}\left[w^\star_{Q,\tau,\pi_z}(s,a,s')\,L_{\pi_z}(s)\right].$$

Optimization alternates between updates of (Q,τ) and the task-vector z until convergence, yielding a policy π_z that is robust to worst-case transition perturbations in the uncertainty set.

Robustness radii. Both radii come from the kernel radius ε' in (9): $\varepsilon_l=\gamma\varepsilon'/(1-\gamma)$ for RBFM-Light (Lemma 1) and $\varepsilon=\varepsilon'/(1-\gamma)$ for RBFM-Heavy (Lemma 2), so $\varepsilon_l=\gamma\varepsilon$. We use $\varepsilon_l=\varepsilon=0.8$ throughout.

Uncertainty set and physical perturbations. A change in a physical parameter, such as joint friction loss, changes the transition kernel. Since the deployment-time change is unknown, we model it with a

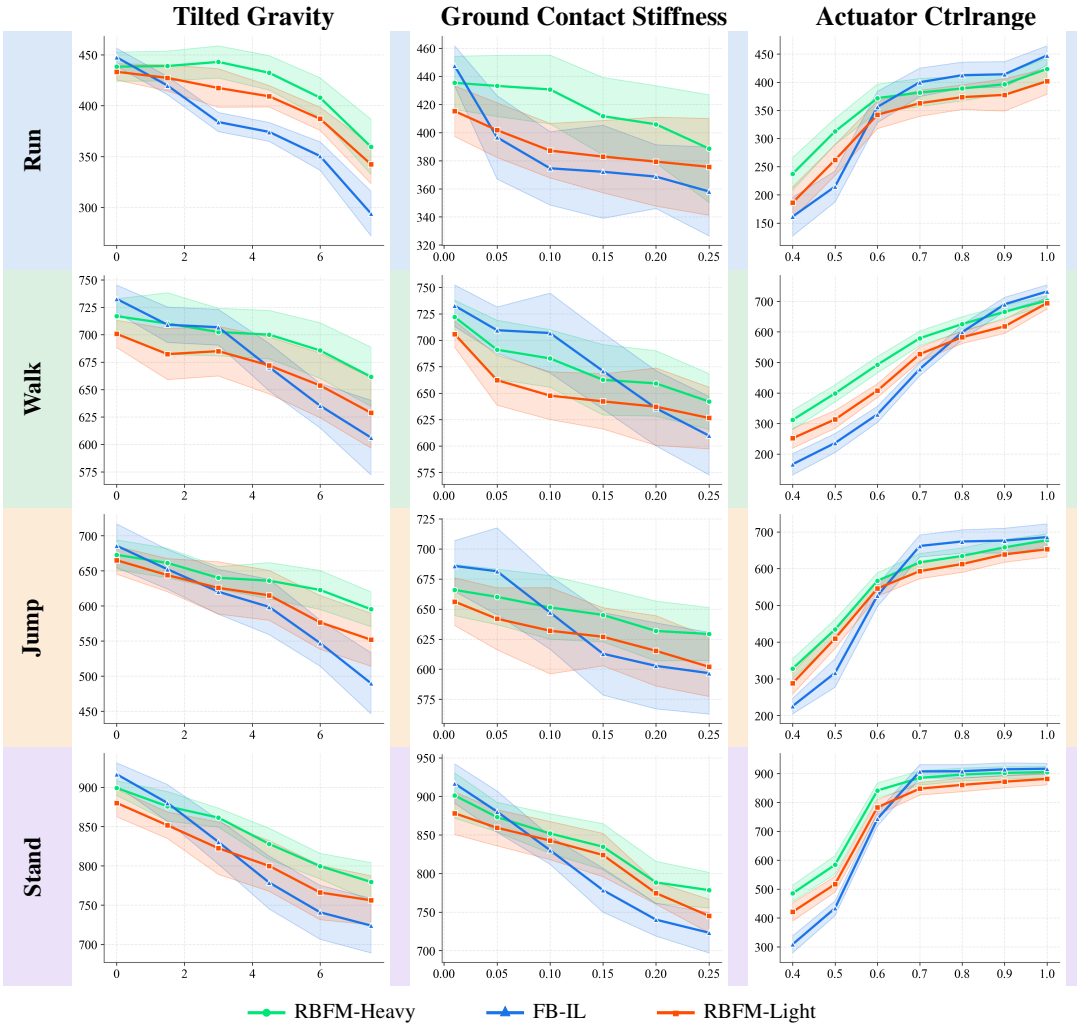


Figure 2: Quadruped under dynamics perturbations, pretrained on RND. Rows: tasks; columns: perturbation types, with the absolute value of the perturbed parameter on the x-axis. Nominal is the leftmost point for the first two perturbations and the rightmost for actuator control range. Mean return and 95% CI over 20 seeds.

divergence ball around T^o [29, 39], whose radius sets the level of conservatism rather than bounding a specific physical change. Transitions known to be physically infeasible can be excluded by setting $w = 0$ on them.

4 Experiments

We evaluate RBFM-Light (§3.1) and RBFM-Heavy (§3.2) on robustness to dynamics perturbations through six questions: **(Q1)** How does the robustness parameter ε affect performance, and how should it be chosen? **(Q2)** How do RBFM-Light and RBFM-Heavy compare with FB-IL under dynamics perturbations? **(Q3)** Do the gains persist when the dynamics drift within an episode or several parameters shift at once? **(Q4)** How do they compare with single-task robust offline IL algorithms? **(Q5)** What is their computational cost, including pretraining? **(Q6)** How do the quality and quantity of pretraining data affect robustness?

Setup. We evaluate on the ExORL benchmark [62], which provides datasets collected via unsupervised exploration on the DeepMind Control Suite [55]. Following prior BFM works [45, 31], we use pretraining datasets from RND [16], APS [37], and PROTO [61]. As in [51], experiments span three domains with twelve tasks: *Walker* (Stand, Walk, Run, Flip), *Quadruped* (Stand, Walk, Run,

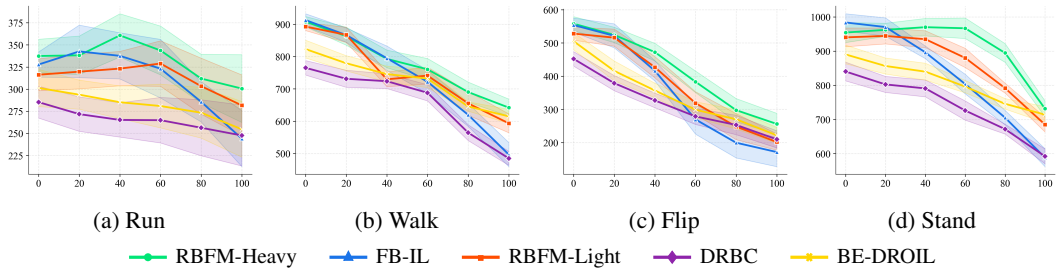


Figure 3: Walker under body-mass perturbation (% increase from nominal) across four tasks, pretrained on RND, compared with single-task robust offline IL. All methods use 4 expert trajectories per task. Mean and 95% CI over 20 seeds.

Jump), and *Cheetah* (Walk, Run, Walk_Backward, Run_Backward). We perturb the physics of each domain: gravity, body mass, and joint friction loss for Walker; tilted gravity, ground contact stiffness, and actuator control range for Quadruped; and actuator strength, joint friction loss, and range of motion for Cheetah. Each perturbation changes one physical parameter at a time; (Q3) relaxes this. For Walker we use all three pretraining datasets, which gives 36 tasks; Quadruped and Cheetah use RND-pretrained data only, for a total of 60 tasks across all domains. A pretrained BFM generates 200 expert trajectories per task, and imitation learning uses 4 randomly sampled trajectories per task. Each seed's score is the average return over 100 evaluation episodes; we report the mean over 20 seeds per environment–task–perturbation combination, with 95% confidence intervals across seeds (shaded regions in all plots). We compare against FB-IL throughout and against DRBC [41] and BE-DROIL [3] in (Q4). Appendix C gives the full experimental setup.

Implementation Details. For all BFM-based methods, we follow [45] and warm-start z via $z_w = \mathbb{E}_\tau\left[\frac{\sum_{t\geq 0} B(s_{t+1})}{l(\tau)}\right]$. The term $(1-\gamma)\mathbb{E}_{s\sim\mu,\, a\sim\pi_D(\cdot|s)}[Q(s,a)]$ in (16) requires samples from the initial-state distribution μ. Each expert dataset contains only 4 initial states, so we follow [3] and treat all trajectory states as effective initial states, which improves coverage and stabilizes the estimate. Appendix D gives the complete implementation details.

Results. **(Q1)** The radius ε is RBFM's only robustness hyperparameter; λ, Q, and τ are dual variables optimized during inference. We vary $\varepsilon \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ and evaluate RBFM-Heavy on the Quadruped jump task under ground contact stiffness perturbations (Figure 4). Smaller values of ε give higher nominal returns but degrade sharply under perturbation, because a smaller uncertainty set hedges less against the shift. Larger values decay more gradually, at a small cost in nominal return. Performance is flat near the optimum: $\varepsilon \in \{0.6, 0.8, 1.0\}$ stay within 3.5% of each other at every perturbation level. We selected $\varepsilon = 0.8$ on this setting and used it unchanged in all 60 settings, without per-environment or per-task tuning. For a new system, ε can be set from prior knowledge of how far the physical parameters can vary, such as manufacturing tolerances or the error of system-identification estimates.

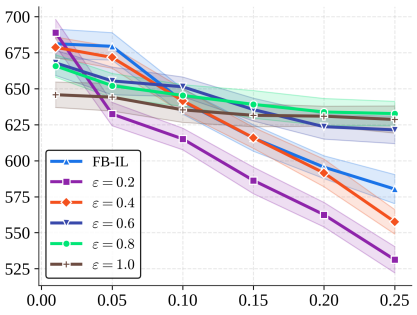


Figure 4: Average return of RBFM-Heavy vs. ground contact stiffness on Quadruped jump for five values of ε. Mean and 95% CI over 20 seeds.

(Q2) Across all three morphologies and perturbation types, both RBFM variants consistently outperform FB-IL as perturbations become more severe. In most settings, RBFM-Heavy reaches the highest returns, RBFM-Light comes second, and FB-IL degrades the fastest (Figure 2 for Quadruped; Walker and Cheetah in Figures 6 and 7, Appendix F.1). With 20 seeds, the confidence intervals of RBFM-Heavy and FB-IL separate as the perturbation grows.

The gap widens monotonically with the size of the perturbation, which is consistent with our robust objectives: RBFM optimizes z against the worst-case distribution in an uncertainty set around the empirical expert distribution, so it accounts for a shift in the transition distribution at evaluation time.

Table 1: Robustness beyond the stationary single-factor protocol. Wins: settings in which the method's mean return exceeds that of FB-IL. Gain: mean per-setting relative improvement over FB-IL. The task vector is inferred once under nominal dynamics.

		RBFM-Light		**RBFM-Heavy**	
Shift	**Settings**	Wins	Gain	Wins	Gain
Gradual within-episode drift	12	11	+10.9%	12	+21.6%
Abrupt within-episode shift	12	11	+11.7%	12	+24.7%
Compound, two parameters	9	9	+20.2%	9	+43.1%
Compound, three parameters	3	3	+40.8%	3	+83.5%

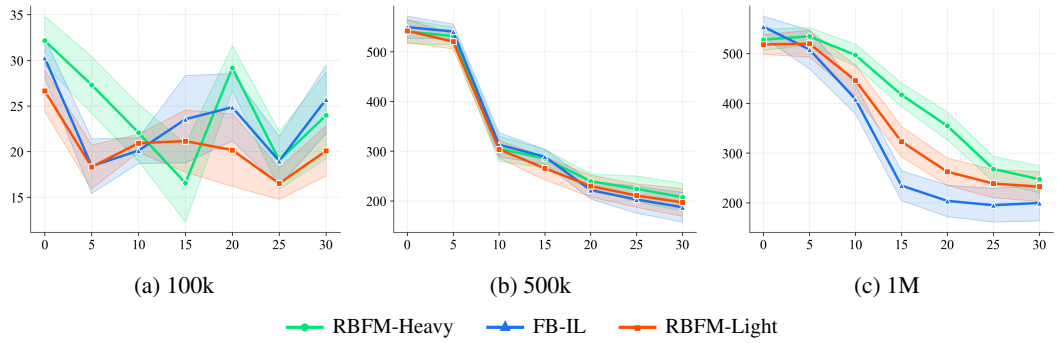


Figure 5: Average return vs. joint friction loss (absolute Nm per joint) on Walker flip for RND pretraining sets of 100k, 500k, and 1M transitions. Mean and 95% CI over 20 seeds.

RBFM-Heavy outperforms RBFM-Light because its realizability constraint gives a less conservative solution, while RBFM-Light's unconstrained objective over-penalizes transitions that are unlikely under the true perturbation distribution. FB-IL fits only the empirical expert distribution, so it has no margin for model mismatch and degrades the most. The pattern holds across perturbations with distinct physical causes and across morphologies (Appendix F.1).

(Q3) We test two harder regimes. In both, the task vector is inferred once under nominal dynamics and never updated, so any difference comes from the inferred z alone. In *within-episode drift*, the perturbation parameter (Walker joint friction loss, Quadruped ground contact stiffness, Cheetah actuator strength) either increases in equal increments every 10 steps from nominal to its maximum (gradual) or jumps from nominal to its maximum at mid-episode (abrupt). In *compound shifts*, two or three physical parameters are perturbed at once. RBFM-Heavy outperforms FB-IL in all 36 settings, and RBFM-Light does so in 34 (Table 1; per-task results in Appendix F.4). RBFM-Heavy's margin over FB-IL grows with the number of parameters perturbed at once: $+43.1\%$ with two and $+83.5\%$ with three.

(Q4) Figure 3 compares the RBFM variants and FB-IL with the single-task distributionally robust offline IL baselines DRBC and BE-DROIL on Walker under body mass perturbation. All methods receive the same 4 expert trajectories per task; the BFM-based methods also use reward-free exploratory data, which the single-task methods have no pretraining stage to exploit. At nominal and moderate perturbation levels, all BFM-based methods substantially outperform the single-task baselines. We attribute this to the multi-task pretraining of the FB models, whose task-agnostic representation generalizes across reward structures, consistent with the nominal-setting findings of [45]. More expert data does not close this gap: trained on 1,000 expert trajectories ($250\times$ more), DRBC and BE-DROIL still fall below RBFM-Heavy trained on 4 in all eight task–mass settings (Appendix F.5). At higher perturbation levels, BE-DROIL matches or slightly outperforms FB-IL, so per-task distributional robustness does give an advantage over non-robust BFM inference when the dynamics shift is non-trivial. RBFM, which combines multi-task BFM pretraining with robust task inference, outperforms both FB-IL and the single-task baselines across the full perturbation sweep.

(Q5) Table 2 reports the wall-clock time to compute an imitation policy for one task, averaged across environments and tasks, along with the one-time pretraining cost (1M steps). The single-task robust baselines take hours per task, while BFM-based methods pretrain once per environment and then take

minutes per task. For N tasks, RBFM-Heavy therefore costs $14\text{h} + 10N$ minutes, against $6\text{h}\,10\text{m} \times N$ for BE-DROIL and $5\text{h}\,23\text{m} \times N$ for DRBC. It is cheaper end-to-end from $N = 3$ tasks onward; at $N = 100$ it is $20\times$ cheaper than BE-DROIL and $18\times$ cheaper than DRBC, and it is also more robust (Q2–Q4).

Table 2: Wall-clock time per task, averaged across environments and tasks. †Once per environment, shared by all BFM-based methods.

Algorithm	**Time**
BFM pretraining†	14h
FB-IL	1m
RBFM-Light	5m
RBFM-Heavy	10m
DRBC	5h23m
BE-DROIL	6h10m

(Q6) RBFM changes only task inference, so it inherits FB's assumptions on the pretraining data and adds none: a dataset without rewards, task labels, or demonstrations that is large and diverse enough to cover the states downstream tasks visit [57, 31]. On Walker, the methods rank as in (Q2) for all three pretraining sources (RND, APS, PROTO), whose coverage differs substantially (Appendix F.2, Figures 8 and 9). For the data quantity, we vary the RND pretraining set over 100k, 500k, and 1M transitions on Walker flip under joint friction loss (Figure 5). The standard setting in the BFM literature is 1M transitions [45, 31]; we include 100k to find where the paradigm breaks. At 100k, FB pretraining itself does not converge: its IQM return stays far below the 500k run on all four Walker tasks (Figure 10, Appendix G). All three inference methods then degrade, FB-IL included, with no separation between them. At 500k the trends begin to stabilize, and at 1M a consistent separation emerges. The sensitivity to data quantity therefore comes from the pretrained BFM: on the same representation, RBFM degrades no more than FB-IL.

5 Related Work

Offline imitation learning is commonly approached via Behavioral Cloning (BC) [46], which treats imitation as supervised learning but suffers from covariate shift and compounding errors due to ignoring environment dynamics [49]. The DICE family [35, 33, 38] incorporates transition structure by estimating occupancy ratios from offline data. In parallel, robust reinforcement learning addresses transition uncertainty via worst-case optimization over ambiguity sets [39, 29, 22, 63, 42], but typically requires rewards and online interaction. Offline robust RL [40] removes interaction but still relies on known rewards, limiting applicability to imitation learning.

Robust imitation learning has largely focused on imperfect demonstrations or training stability [60, 54, 36, 50], while domain randomization and meta-IL methods assume access to simulators or fixed dynamics [44, 28, 24]. Recent approaches improve BC robustness under distributional shift [59, 53], but do not address transition uncertainty. Among methods that do, DRBC [41] and BE-DROIL [3] formulate robust offline imitation learning under transition ambiguity. However, these approaches operate in a per-task manner, requiring separate optimization for each task. This limits their applicability in multi-task settings, where rapid adaptation to new tasks is essential, thereby motivating the need for a unified and robust framework capable of generalizing across tasks. We refer to Appendix B for an extended discussion on related works.

6 Conclusion

We revisited Behavior Foundation Models (BFMs) under dynamics shift and showed that standard task inference is inherently brittle to transition mismatch. To address this, we proposed a robust minimax formulation over an uncertainty set of dynamics, and introduced two practical algorithms, *RBFM-Light* and *RBFM-Heavy*, trading off efficiency and robustness without modifying pretraining or requiring multi-environment data. Empirically, both variants significantly improve robustness over standard BFMs and outperform single-task robust offline IL baselines, while preserving fast task inference. Our results highlight a promising direction for scalable imitation learning, where a single pretrained model can be robustly adapted to new tasks using only offline data.

Limitations. The uncertainty set is a divergence ball around the nominal kernel, and robustness to physical shifts is established empirically. RBFM does not adapt online: z is inferred once. It inherits FB's need for a large, diverse reward-free pretraining dataset, and all evaluations are in simulation.

7 Acknowledgments and Disclosure of Funding

This project was partially supported by the following grants: Office of Naval Research MURI-N00014-24-1-274, and National Science Foundation (NSF) grant 2448885.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Rishabh Agrawal, Nathan Dahlin, Rahul Jain, and Ashutosh Nayyar. Policy optimization for strictly batch imitation learning. In *OPT 2024: Optimization for Machine Learning*, 2024. URL `https://openreview.net/forum?id=5L3qmI0XPz`.

[3] Rishabh Agrawal, Yusuf Alvi, Rahul Jain, and Ashutosh Nayyar. Balance equation-based distributionally robust offline imitation learning. *arXiv preprint arXiv:2511.07942*, 2025.

[4] Rishabh Agrawal, Nathan Dahlin, Rahul Jain, and Ashutosh Nayyar. Markov balance satisfaction improves performance in strictly batch offline imitation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 15311–15319, 2025.

[5] Rishabh Agrawal, Nathan Dahlin, Rahul Jain, and Ashutosh Nayyar. Conditional kernel imitation learning for continuous state environments. In Necmiye Ozay, Laura Balzano, Dimitra Panagou, and Alessandro Abate, editors, *Proceedings of the 7th Annual Learning for Dynamics & Control Conference*, volume 283 of *Proceedings of Machine Learning Research*, pages 469–483. PMLR, 04–06 Jun 2025. URL `https://proceedings.mlr.press/v283/agrawal25a.html`.

[6] Syed Mumtaz Ali and Samuel D Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological)*, 28(1):131–142, 1966.

[7] Elie Aljalbout, Jiaxu Xing, Angel Romero, Iretiayo Akinola, Caelan Reed Garrett, Eric Heiden, Abhishek Gupta, Tucker Hermans, Yashraj Narang, Dieter Fox, et al. The reality gap in robotics: Challenges, solutions, and best practices. *Annual Review of Control, Robotics, and Autonomous Systems*, 9, 2025.

[8] Eitan Altman. *Constrained Markov decision processes*. Routledge, 2021.

[9] Shan An, Ziyu Meng, Chao Tang, Yuning Zhou, Tengyu Liu, Fangqiang Ding, Shufang Zhang, Yao Mu, Ran Song, Wei Zhang, et al. Dexterous manipulation through imitation learning: A survey. *arXiv preprint arXiv:2504.03515*, 2025.

[10] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[11] André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P Van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

[12] Léonard Blier, Corentin Tallec, and Yann Ollivier. Learning successor states and goal-dependent values: A mathematical viewpoint. *arXiv preprint arXiv:2101.07123*, 2021.

[13] Maksim Bobrin, Ilya Zisman, Alexander Nikulin, Vladislav Kurenkov, and Dmitry V. Dylov. Zero-shot adaptation of behavioral foundation models to unseen dynamics. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=dBDBg4WF4F`.

[14] Diana Borsa, André Barreto, John Quan, Daniel Mankowitz, Rémi Munos, Hado Van Hasselt, David Silver, and Tom Schaul. Universal successor features approximators. *arXiv preprint arXiv:1812.07626*, 2018.

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

[16] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

[17] Jongseong Chae, Seungyul Han, Whiyoung Jung, Myungsik Cho, Sungho Choi, and Youngchul Sung. Robust imitation learning against variations in environment dynamics. In *International Conference on Machine Learning*, pages 2828–2852. PMLR, 2022.

[18] Seongwoong Cho, Donggyun Kim, Jinwoo Lee, and Seunghoon Hong. Meta-controller: Few-shot imitation of unseen embodiments and tasks in continuous control. *Advances in Neural Information Processing Systems*, 37:134250–134286, 2024.

[19] Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9329–9338, 2019.

[20] Imre Csiszár. On information-type measure of difference of probability distributions and indirect observations. *Studia Sci. Math. Hungar.*, 2:299–318, 1967.

[21] Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural computation*, 5(4):613–624, 1993.

[22] Esther Derman and Shie Mannor. Distributional robustness and regularization in reinforcement learning. *arXiv preprint arXiv:2003.02894*, 2020.

[23] Yan Duan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. *Advances in neural information processing systems*, 30, 2017.

[24] Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In *Conference on robot learning*, pages 357–368. PMLR, 2017.

[25] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR, 2019.

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

[27] Xiaoyi Hu, Qiao Sun, Bailin He, Haojie Liu, Xueyi Zhang, Chunpeng Lu, and Jiangwei Zhong. Impact of static friction on sim2real in robotic reinforcement learning. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 17107–17114. IEEE, 2025.

[28] Wenlong Huang, Igor Mordatch, Pieter Abbeel, and Deepak Pathak. Generalization in dexterous manipulation via geometry-aware multi-task learning. *arXiv preprint arXiv:2111.03062*, 2021.

[29] Garud N Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2): 257–280, 2005.

[30] Stephen James, Michael Bloesch, and Andrew J Davison. Task-embedded control networks for few-shot imitation learning. In *Conference on robot learning*, pages 783–795. PMLR, 2018.

[31] Scott Jeen, Tom Bewley, and Jonathan M Cullen. Zero-shot reinforcement learning from low quality data. *Advances in Neural Information Processing Systems*, 37:16894–16942, 2024.

[32] Scott Jeen, Tom Bewley, and Jonathan M Cullen. Zero-shot reinforcement learning under partial observability. *arXiv preprint arXiv:2506.15446*, 2025.

[33] Geon-Hyeong Kim, Seokin Seo, Jongmin Lee, Wonseok Jeon, HyeongJoo Hwang, Hongseok Yang, and Kee-Eung Kim. DemoDICE: Offline imitation learning with supplementary imperfect demonstrations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=BrPdX1bDZkQ.

[34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[35] Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. Imitation learning via off-policy distribution matching. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hyg-JC4FDr.

[36] Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. Dart: Noise injection for robust imitation learning. In Sergey Levine, Vincent Vanhoucke, and Ken Goldberg, editors, *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 143–156. PMLR, 13–15 Nov 2017. URL https://proceedings.mlr.press/v78/laskey17a.html.

[37] Hao Liu and Pieter Abbeel. Aps: Active pretraining with successor features. In *International Conference on Machine Learning*, pages 6736–6747. PMLR, 2021.

[38] Liyuan Mao, Haoran Xu, Weinan Zhang, and Xianyuan Zhan. ODICE: Revealing the mystery of distribution correction estimation via orthogonal-gradient update. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=L8UNn7Llt4.

[39] Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

[40] Kishan Panaganti, Zaiyan Xu, Dileep Kalathil, and Mohammad Ghavamzadeh. Robust reinforcement learning using offline data. *Advances in neural information processing systems*, 35: 32211–32224, 2022.

[41] Kishan Panaganti, Zaiyan Xu, Dileep Kalathil, and Mohammad Ghavamzadeh. Distributionally robust behavioral cloning for robust imitation learning. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 1342–1347. IEEE, 2023.

[42] Kishan Panaganti, Zaiyan Xu, Dileep Kalathil, and Mohammad Ghavamzadeh. Bridging distributionally robust learning and offline rl: An approach to mitigate distribution shift and partial data coverage. In Necmiye Ozay, Laura Balzano, Dimitra Panagou, and Alessandro Abate, editors, *Proceedings of the 7th Annual Learning for Dynamics & Control Conference*, volume 283 of *Proceedings of Machine Learning Research*, pages 619–634. PMLR, 04–06 Jun 2025. URL https://proceedings.mlr.press/v283/panaganti25a.html.

[43] Seohong Park, Tobias Kreiman, and Sergey Levine. Foundation policies with hilbert representations. *arXiv preprint arXiv:2402.15567*, 2024.

[44] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018.

[45] Matteo Pirotta, Andrea Tirinzoni, Ahmed Touati, Alessandro Lazaric, and Yann Ollivier. Fast imitation via behavior foundation models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=qnWtw3l0jb.

[46] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1:305–313, 1988.

[47] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[48] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

[49] Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668. JMLR Workshop and Conference Proceedings, 2010.

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

[51] Thomas Rupf, Marco Bagatella, Marin Vlastelica, and Andreas Krause. Optimistic task inference for behavior foundation models. *arXiv preprint arXiv:2510.20264*, 2025.

[52] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International conference on machine learning*, pages 1312–1320. PMLR, 2015.

[53] Seokin Seo, Byung-Jun Lee, Jongmin Lee, HyeongJoo Hwang, Hongseok Yang, and Kee-Eung Kim. Mitigating covariate shift in behavioral cloning via robust stationary distribution correction. *Advances in Neural Information Processing Systems*, 37:109177–109201, 2024.

[54] Voot Tangkaratt, Nontawat Charoenphakdee, and Masashi Sugiyama. Robust imitation learning from noisy demonstrations. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 298–306. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/tangkaratt21a.html.

[55] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[56] Rajesh Tiwari, Shailesh Khapre, and Avantika Singh. Reinforcement learning in robotic systems: A review on sim-to-real transfer. *Robotics and Autonomous Systems*, page 105327, 2026.

[57] Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. *Advances in Neural Information Processing Systems*, 34:13–23, 2021.

[58] Ahmed Touati, Jérémy Rapin, and Yann Ollivier. Does zero-shot reinforcement learning exist? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=MYEap_OcQI.

[59] Shili Wu, Yizhao Jin, Puhua Niu, Aniruddha Datta, and Sean B Andersson. Robust behavior cloning via global lipschitz regularization. *arXiv preprint arXiv:2506.19250*, 2025.

[60] Yueh-Hua Wu, Nontawat Charoenphakdee, Han Bao, Voot Tangkaratt, and Masashi Sugiyama. Imitation learning from imperfect demonstration. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6818–6827. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/wu19a.html.

[61] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, pages 11920–11931. PMLR, 2021.

[62] Denis Yarats, David Brandfonbrener, Hao Liu, Michael Laskin, Pieter Abbeel, Alessandro Lazaric, and Lerrel Pinto. Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning. *arXiv preprint arXiv:2201.13425*, 2022.

[63] Zhuodong Yu, Ling Dai, Shaohang Xu, Siyang Gao, and Chin Pang Ho. Fast bellman updates for wasserstein distributionally robust mdps. *Advances in Neural Information Processing Systems*, 36:30554–30578, 2023.

[64] Kexin Zheng, Lauriane Teyssier, Yinan Zheng, Yu Luo, and Xianyuan Zhan. Breeze: Towards robust zero-shot reinforcement learning. https://github.com/Whiterrrrr/BREEZE, 2026. GitHub repository, accessed May 7, 2026.

[65] Allan Zhou, Eric Jang, Daniel Kappler, Alex Herzog, Mohi Khansari, Paul Wohlhart, Yunfei Bai, Mrinal Kalakrishnan, Sergey Levine, and Chelsea Finn. Watch, try, learn: Meta-learning from demonstrations and rewards. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJg5J6NtDr.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We claim to make BFMs robust when dataset is available only in one environment. Through analysis, we show how learning can be done in such offline settings and then do extensive empirical study across 60 tasks to validate our claim.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: It is provided in Appendix G.

   Guidelines:

   - The answer [N/A] 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: We provide the full-set of assumptions needed to prove any of our results included in this work.

Guidelines:

- The answer [N/A] 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: Full details are provided in Appendices C and D.

   Guidelines:

   - The answer [N/A] 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: We have included code along with our submission.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

Answer: [Yes]

Justification: Extensive discussion has been provided in Appendices C and D.

Guidelines:

- The answer [N/A] 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: [Yes]

Justification: All our main results consider a 95% confidence interval when reporting.

Guidelines:

- The answer [N/A] 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: [Yes]

   Justification: Details have been provided in Appendix C.5.

   Guidelines:

   - The answer [N/A] 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: It conforms with the Code of Conduct.

   Guidelines:

   - The answer [N/A] 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: [N/A]

    Justification: Our work focuses on making the system robust to changes in dynamics. This will only help in deploying such models to help avoid failure scenarios.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: There is no risk associated with our work.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: We have cited the references we have used for our experiments.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: We think it does not apply to us.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: We do not work with human subject.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: Does not apply to us.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

Answer: [N/A].

Justification: We only use LLMs for improving writing, as mentioned in Appendix H.

Guidelines:

- The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.

Appendices

A Proofs **25**

B Extended Related Work **28**

C Experimental Setup **29**

C.1 ExORL Domains 29

C.1.1 Walker 29

C.1.2 Quadruped 30

C.1.3 Cheetah 30

C.2 ExORL Datasets 30

C.3 Environment Perturbations 30

C.3.1 Walker 30

C.3.2 Quadruped 30

C.3.3 Cheetah 31

C.4 Evaluation Protocol 32

C.4.1 Pre-training Evaluation 32

C.4.2 Downstream Task Inference 32

C.5 Computational Resources 32

D Implementation Details **33**

D.1 Forward-Backward Representations 33

D.1.1 Architecture 33

D.1.2 Forward Representation $F(s, a, z)$ 33

D.1.3 Backward Representation $B(s)$ 33

D.1.4 Actor $\pi_z(s)$ 33

D.1.5 Additional Details. 33

D.2 Task Sampling Distribution $\mathcal{Z}$ 33

D.3 Discussion on usage of f-divergence in RBFM-Heavy 34

D.4 BFM-based Hyperparameters 34

D.4.1 Pretraining 34

D.4.2 Task-Inference 34

D.5 Single Task Robust Offline IL baselines 34

D.5.1 DRBC 35

D.5.2 BE-DROIL 35

E Algorithm **35**

F Extended Results **37**

F.1 Further discussion of (Q2) 37

F.2 Q6: Different Sources of Pretraining Data 39

F.3 Q1: Robustness radius 39
F.4 Q3: Non-stationary and compound perturbations 40
F.5 Q4: Single-task baselines with more demonstrations 40

G Discussion and Limitation **41**

H LLM Usage **41**

A Proofs

Lemma 1. *For any policy π and transition kernel $T \in \mathcal{T}_{T^o}(\varepsilon')$, the following holds:*

$$D_{\mathrm{TV}}(\rho^\pi_T(s), \rho^\pi_{T^o}(s)) \leq \frac{\gamma\varepsilon'}{1-\gamma}.$$

Proof. See Lemma 7 in [41] for details. □

Lemma 3. *For any policy π and transition kernel $T \in \mathcal{T}_{T^o}(\varepsilon')$, the following holds:*

$$D_{\mathrm{TV}}(\rho^\pi_T(s,a), \rho^\pi_{T^o}(s,a)) \leq \frac{\gamma\varepsilon'}{1-\gamma}.$$

Proof. We build upon the Lemma 1 that quantifies how uncertainty in the transition model influences the induced occupancy measure on the state-space for a fixed policy.

Since $\rho^\pi_T(s,a) = \pi(a|s)\rho^\pi_T(s)$, it follows that

$$\begin{aligned} D_{\mathrm{TV}}(\rho^\pi_T(s,a), \rho^\pi_{T^o}(s,a)) &= \frac{1}{2}\sum_{s,a} |\rho^\pi_T(s,a) - \rho^\pi_{T^o}(s,a)| \\ &= \frac{1}{2}\sum_{s,a} \pi(a|s)|\rho^\pi_T(s) - \rho^\pi_{T^o}(s)| \\ &= D_{\mathrm{TV}}(\rho^\pi_T(s), \rho^\pi_{T^o}(s)) \\ &\leq \frac{\gamma\varepsilon'}{1-\gamma}. \end{aligned}$$

□

Lemma 2. *For any policy π and transition kernel $T \in \mathcal{T}_{T^o}(\varepsilon')$, the following holds:*

$$D_{\mathrm{TV}}(\rho^\pi_T(s,a,s'), \rho^\pi_{T^o}(s,a,s')) \leq \frac{\varepsilon'}{1-\gamma}.$$

Proof. We have

$$\rho^\pi_T(s,a,s') = \rho^\pi_T(s,a)T_{s,a}(s'), \quad \rho^\pi_{T^o}(s,a,s') = \rho^\pi_{T^o}(s,a)T^o_{s,a}(s').$$

Then,

$$\begin{aligned} &\|\rho^\pi_T(s,a,s') - \rho^\pi_{T^o}(s,a,s')\|_1 \\ &= \sum_{s,a,s'} \left|\rho^\pi_T(s,a)T_{s,a}(s') - \rho^\pi_{T^o}(s,a)T^o_{s,a}(s')\right| \\ &\leq \sum_{s,a,s'} \rho^\pi_T(s,a)\left|T_{s,a}(s') - T^o_{s,a}(s')\right| + \sum_{s,a,s'} |\rho^\pi_T(s,a) - \rho^\pi_{T^o}(s,a)|\, T^o_{s,a}(s'). \end{aligned}$$

For the first term, using $\|T_{s,a} - T^o_{s,a}\|_1 = 2D_{\mathrm{TV}}(T_{s,a}, T^o_{s,a}) \leq 2\varepsilon'$,

$$\sum_{s,a,s'} \rho^\pi_T(s,a)\left|T_{s,a}(s') - T^o_{s,a}(s')\right| = \sum_{s,a} \rho^\pi_T(s,a)\|T_{s,a} - T^o_{s,a}\|_1 \leq 2\varepsilon'.$$

For the second term, since $\sum_{s'} T^o_{s,a}(s') = 1$, we have

$$\begin{aligned} \sum_{s,a,s'} |\rho^\pi_T(s,a) - \rho^\pi_{T^o}(s,a)|\, T^o_{s,a}(s') &= \sum_{s,a} |\rho^\pi_T(s,a) - \rho^\pi_{T^o}(s,a)| \\ &= \|\rho^\pi_T(s,a) - \rho^\pi_{T^o}(s,a)\|_1. \end{aligned}$$

By lemma 3,

$$D_{\mathrm{TV}}(\rho^{\pi}_{T}(s,a),\rho^{\pi}_{T^o}(s,a)) \le \frac{\gamma\varepsilon'}{1-\gamma},$$

which implies

$$\|\rho^{\pi}_{T}(s,a)-\rho^{\pi}_{T^o}(s,a)\|_1 \le \frac{2\gamma\varepsilon'}{1-\gamma}.$$

Combining both terms,

$$\|\rho^{\pi}_{T}(s,a,s')-\rho^{\pi}_{T^o}(s,a,s')\|_1 \le 2\varepsilon' + \frac{2\gamma\varepsilon'}{1-\gamma} = \frac{2\varepsilon'}{1-\gamma}.$$

Hence,

$$D_{\mathrm{TV}}\big(\rho^{\pi}_{T}(s,a,s'),\rho^{\pi}_{T^o}(s,a,s')\big) = \frac{1}{2}\|\rho^{\pi}_{T}(s,a,s')-\rho^{\pi}_{T^o}(s,a,s')\|_1 \le \frac{\varepsilon'}{1-\gamma}.$$

□

Proposition 1. *The solution to the constrained optimization problem in* (11) *is obtained by solving the following:*

$$\min_{z}\min_{\lambda\in\mathbb{R}}\left\{\mathbb{E}_{s\sim\rho^{\pi_D}_{T^o}}\big[(\mathcal{L}(\pi_z(\cdot|s),\pi_D(\cdot|s))-\lambda)_+\big] + \varepsilon_l\left(\max_{\rho^{\pi_D}_{T^o}(s)>0}\mathcal{L}(\pi_z(\cdot|s),\pi_D(\cdot|s))-\lambda\right)_+ + \lambda\right\}.$$

Proof. Fix the task vector z, and hence the learner policy π_z. Write $P := \rho^{\pi_D}_{T^o}$ for the nominal expert occupancy measure, $\mathcal{S}_o := \{s\in\mathcal{S}:\ P(s)>0\}$ for its support, and

$$\ell_z(s) := \mathcal{L}\big(\pi_z(\cdot|s),\pi_D(\cdot|s)\big)\ \ge 0, \qquad M_z := \max_{s\in\mathcal{S}_o}\ell_z(s).$$

As in the statement of the proposition, which involves ℓ_z only on $\mathcal{S}_o$, the inner maximization in (11) is over distributions Q supported on $\mathcal{S}_o$, i.e., $Q \ll P$, with $D_{\mathrm{TV}}(Q,P)\le\varepsilon_l$; these are the perturbations of the expert distribution that can be evaluated from offline expert data. For such Q, let $w(s) := Q(s)/P(s)$ on $\mathcal{S}_o$. Then $w\ge 0$, $\mathbb{E}_P[w]=1$, $\mathbb{E}_Q[h]=\mathbb{E}_P[w\,h]$ for any function h, and, since $\mathbb{E}_P[w-1]=0$,

$$D_{\mathrm{TV}}(Q,P) = \tfrac{1}{2}\,\mathbb{E}_P\big[\,|w-1|\,\big] = \mathbb{E}_P\big[(w-1)_+\big].$$

For $\lambda\in\mathbb{R}$, define

$$g(\lambda) := \mathbb{E}_P\big[(\ell_z-\lambda)_+\big] + \varepsilon_l\,(M_z-\lambda)_+ + \lambda.$$

We show that $\max_Q \mathbb{E}_Q[\ell_z] = \min_{\lambda\in\mathbb{R}} g(\lambda)$, where the maximum is over the feasible set above.

Upper bound. Let Q be feasible and $\lambda\in\mathbb{R}$. Since $w\ge 0$ and $\mathbb{E}_P[w]=1$,

$$\mathbb{E}_Q[\ell_z] = \lambda+\mathbb{E}_P\big[w\,(\ell_z-\lambda)\big] \le \lambda+\mathbb{E}_P\big[w\,(\ell_z-\lambda)_+\big] = \lambda+\mathbb{E}_P\big[(\ell_z-\lambda)_+\big]+\mathbb{E}_P\big[(w-1)\,(\ell_z-\lambda)_+\big].$$

On $\mathcal{S}_o$, $0\le(\ell_z-\lambda)_+\le(M_z-\lambda)_+$, so

$$\mathbb{E}_P\big[(w-1)\,(\ell_z-\lambda)_+\big] \le \mathbb{E}_P\big[(w-1)_+\,(\ell_z-\lambda)_+\big] \le (M_z-\lambda)_+\,D_{\mathrm{TV}}(Q,P) \le \varepsilon_l\,(M_z-\lambda)_+.$$

Hence $\mathbb{E}_Q[\ell_z]\le g(\lambda)$ for every feasible Q and every $\lambda\in\mathbb{R}$.

Matching lower bound. Let $s^\star\in\mathcal{S}_o$ satisfy $\ell_z(s^\star)=M_z$. If $\varepsilon_l\ge 1$, the constraint is inactive because $D_{\mathrm{TV}}\le 1$; the point mass $Q=\delta_{s^\star}$ is feasible and $\mathbb{E}_Q[\ell_z]=M_z=g(M_z)$, where the last equality uses $\ell_z\le M_z$ on $\mathcal{S}_o$. If $0\le\varepsilon_l<1$, let

$$\lambda^\star := \min\big\{t\in\ell_z(\mathcal{S}_o):\ P(\ell_z\le t)\ge\varepsilon_l\big\}\ \le M_z,$$

so that $P(\ell_z<\lambda^\star)\le\varepsilon_l\le P(\ell_z\le\lambda^\star)$ and $P(\ell_z=\lambda^\star)>0$. Choose $\theta\in[0,1]$ with $P(\ell_z<\lambda^\star)+\theta\,P(\ell_z=\lambda^\star)=\varepsilon_l$, set $r(s) := \mathbf{1}\{\ell_z(s)<\lambda^\star\}+\theta\,\mathbf{1}\{\ell_z(s)=\lambda^\star\}\in[0,1]$, so that $\mathbb{E}_P[r]=\varepsilon_l$, and define

$$Q^\star(s) := P(s)\big(1-r(s)\big) + \varepsilon_l\,\mathbf{1}\{s=s^\star\}, \qquad s\in\mathcal{S}_o.$$

$Q^\star$ moves mass ε_l from the states with the smallest loss to a state with the largest loss. It is a probability distribution on $\mathcal{S}_o$, and it is feasible because

$$D_{\mathrm{TV}}(Q^\star, P) = \sum_{s\in\mathcal{S}_o} \big(Q^\star(s) - P(s)\big)_+ = \sum_{s\in\mathcal{S}_o} \big(\varepsilon_l\, \mathbf{1}\{s = s^\star\} - P(s)\, r(s)\big)_+ \le \varepsilon_l.$$

Its value is $\mathbb{E}_{Q^\star}[\ell_z] = \mathbb{E}_P[\ell_z] - \mathbb{E}_P[r\,\ell_z] + \varepsilon_l M_z$. On the other hand, using $(x)_+ = x + (-x)_+$, $\lambda^\star \le M_z$, $\mathbb{E}_P[r\,\ell_z] = \mathbb{E}_P\big[\ell_z \mathbf{1}\{\ell_z < \lambda^\star\}\big] + \theta\lambda^\star P(\ell_z = \lambda^\star)$, and $P(\ell_z < \lambda^\star) = \varepsilon_l - \theta P(\ell_z = \lambda^\star)$,

$$\begin{aligned} g(\lambda^\star) &= \mathbb{E}_P[\ell_z] - \lambda^\star + \mathbb{E}_P\big[(\lambda^\star - \ell_z)_+\big] + \varepsilon_l\,(M_z - \lambda^\star) + \lambda^\star \\ &= \mathbb{E}_P[\ell_z] + \lambda^\star P(\ell_z < \lambda^\star) - \mathbb{E}_P\big[\ell_z \mathbf{1}\{\ell_z < \lambda^\star\}\big] - \varepsilon_l\lambda^\star + \varepsilon_l M_z \\ &= \mathbb{E}_P[\ell_z] - \mathbb{E}_P[r\,\ell_z] + \varepsilon_l M_z \;=\; \mathbb{E}_{Q^\star}[\ell_z]. \end{aligned}$$

In both cases some feasible Q and some λ satisfy $\mathbb{E}_Q[\ell_z] = g(\lambda)$. Together with the upper bound,

$$\max_{Q \ll \rho^{\pi_D}_{T^o}:\, D_{\mathrm{TV}}(Q, \rho^{\pi_D}_{T^o}) \le \varepsilon_l} \mathbb{E}_{s\sim Q}[\ell_z(s)] = \min_{\lambda\in\mathbb{R}} \Big\{ \mathbb{E}_{s\sim\rho^{\pi_D}_{T^o}}\big[(\ell_z(s) - \lambda)_+\big] + \varepsilon_l\,(M_z - \lambda)_+ + \lambda \Big\},$$

and both the maximum and the minimum are attained. This identity holds for every z. Minimizing both sides over z and substituting $\ell_z(s) = \mathcal{L}(\pi_z(\cdot|s), \pi_D(\cdot|s))$ shows that (11) and the problem in Proposition 1 have the same optimal value and the same minimizers z. □

Remark. Finiteness of $\mathcal{S}_o$, as for the empirical expert distribution used in practice, only ensures that the maxima are attained. For a general $\rho^{\pi_D}_{T^o}$, the identity holds with M_z the essential supremum of ℓ_z under $\rho^{\pi_D}_{T^o}$ and a supremum on the left. The upper bound is unchanged. In the lower bound, $\lambda^\star$ becomes the lower ε_l-quantile of ℓ_z (its essential infimum if $\varepsilon_l = 0$), and the point mass at $s^\star$ is replaced by $\rho^{\pi_D}_{T^o}(\,\cdot \mid \ell_z > M_z - \delta)$; this gives feasible Q_δ with $\mathbb{E}_{Q_\delta}[\ell_z] \ge \min_\lambda g(\lambda) - \delta$ for every $\delta > 0$.

Proposition 2. *Suppose that the expert and learner policies π_D and π_z are deterministic and that the action space $\mathcal{A}$ is bounded. Let $\mathcal{L}(\pi_z(\cdot|s), \pi_D(\cdot|s)) = \|\pi_z(s) - \pi_D(s)\|_2^2$ and $L := \sup_{(s,a)\sim\rho^{\pi_D}_{T^o}} \|\pi_z(s) - \pi_D(s)\|_2^2$. Then the optimization problem in Proposition 1 can be simplified to*

$$\min_z \min_{\lambda\in[0,(1+\varepsilon_l)L]} \left\{ \mathbb{E}_{s\sim\rho^{\pi_D}_{T^o}}\big[(\mathcal{L}(\pi_z(\cdot|s), \pi_D(\cdot|s)) - \lambda)_+\big] + \varepsilon_l \left(\max_{\rho^{\pi_D}_{T^o}(s) > 0} \mathcal{L}(\pi_z(\cdot|s), \pi_D(\cdot|s)) - \lambda \right)_+ + \lambda \right\}.$$

Proof. Let us fix the learner's task vector z and the corresponding policy π_z and define the point-wise loss

$$\ell_z(s) := \mathcal{L}(\pi_z(\cdot|s), \pi_D(\cdot|s)) = \|\pi_z(s) - \pi_D(s)\|_2^2.$$

Since the expert and learner policies are deterministic and the action space $\mathcal{A}$ is bounded, the loss is bounded on the support of $\rho^{\pi_D}_{T^o}$. In particular,

$$0 \le \ell_z(s) \le L \qquad \text{for all } s \text{ with } \rho^{\pi_D}_{T^o}(s) > 0.$$

Now, Proposition 1 provides the following optimization problem:

$$\min_z \min_{\lambda\in\mathbb{R}} \left\{ \mathbb{E}_{s\sim\rho^{\pi_D}_{T^o}}\big[(\mathcal{L}(\pi_z(\cdot|s), \pi_D(\cdot|s)) - \lambda)_+\big] + \varepsilon_l \left(\max_{s:\, \rho^{\pi_D}_{T^o}(s) > 0} \mathcal{L}(\pi_z(\cdot|s), \pi_D(\cdot|s)) - \lambda \right)_+ + \lambda \right\}.$$

Next, we define $M_z := \max_{s\in\mathcal{S}:\, \rho^{\pi_D}_{T^o}(s) > 0} \ell_z(s)$ and $g(\lambda)$ as

$$g(\lambda) := \mathbb{E}_{s\sim\rho^{\pi_D}_{T^o}}\big[(\ell_z(s) - \lambda)_+\big] + \varepsilon_l (M_z - \lambda)_+ + \lambda.$$

If $\lambda < 0$, then $\ell_z(s) - \lambda > 0$ and $M_z - \lambda > 0$, so

$$g(\lambda) = \mathbb{E}_{s\sim\rho^{\pi_D}_{T^o}}[\ell_z(s) - \lambda] + \varepsilon_l (M_z - \lambda) + \lambda,$$

which is strictly decreasing in λ because $\varepsilon_l > 0$. Hence no minimizer lies below 0. If $\lambda \ge M_z$, then $\ell_z(s) - \lambda \le 0$ for all s in the support of $\rho^{\pi_D}_{T^o}$, and also $M_z - \lambda \le 0$, so

$$g(\lambda) = \lambda,$$

which is strictly increasing in λ. Hence no minimizer lies above M_z. Therefore every minimizer belongs to $[0, M_z]$. Since $M_z \leq L$, we may restrict the minimization further to the larger interval $[0, (1+\varepsilon_l)L]$ without changing the value. Thus, we have

$$\min_z \min_{\lambda \in [0,(1+\varepsilon_l)L]} \left\{ \lambda + \mathbb{E}_{s \sim \rho^{\pi_D}_{T^o}} \left[(\mathcal{L}(\pi_z(\cdot|s), \pi_D(\cdot|s)) - \lambda)_+ \right] + \varepsilon_l \left(\max_{s \in \mathcal{S}: \rho^{\pi_D}_{T^o}(s) > 0} \mathcal{L}(\pi_z(\cdot|s), \pi_D(\cdot|s)) - \lambda \right)_+ \right\}$$

□

Proposition 3. *For $\tau > 0$, the inner maximization in* (15) *admits the solution, with f' the derivative of the generator f,*

$$w^\star_{Q,\tau,\pi_z}(s,a,s') = \max\left(0,\ (f')^{-1}\left(\frac{c_{Q,\pi_z}(s,a,s')}{\tau}\right)\right).$$

For $\tau = 0$, $w^\star_{Q,\tau,\pi_z}(s,a,s') = +\infty$ if $c_{Q,\pi_z}(s,a,s') > 0$ and 0 otherwise.

Proof. We first consider the case $\tau > 0$. For fixed (Q, τ), the optimization over $w \geq 0$ in (15) reduces to

$$\max_{w \geq 0}\ \mathbb{E}_{s,a,s' \sim \rho^{\pi_D}_{T^o}} \left[-\tau f(w(s,a,s')) + w(s,a,s')\, c_{Q,\pi}(s,a,s') \right].$$

Since the expectation is separable across (s,a,s'), the maximization decomposes pointwise. For each (s,a,s'), we solve

$$\max_{w \geq 0}\ -\tau f(w) + c_{Q,\pi}(s,a,s')\, w.$$

Because f is strictly convex, the objective is strictly concave in w, and the problem admits a unique maximizer. Ignoring the constraint $w \geq 0$, the first-order optimality condition gives

$$-\tau f'(w) + c_{Q,\pi}(s,a,s') = 0 \quad \Rightarrow \quad w = (f')^{-1}\left(\frac{c_{Q,\pi}(s,a,s')}{\tau}\right).$$

Enforcing the constraint $w \geq 0$ amounts to projecting this unconstrained solution onto $\mathbb{R}_+$, yielding

$$w^\star_{Q,\tau,\pi}(s,a,s') = \max\left(0,\ (f')^{-1}\left(\frac{c_{Q,\pi}(s,a,s')}{\tau}\right)\right).$$

For tuples (s,a,s') with $\rho^{\pi_D}_{T^o}(s,a,s') = 0$, the corresponding terms do not affect the objective, and w can be defined arbitrarily.

We now consider the case $\tau = 0$. The objective reduces to

$$\mathbb{E}_{s,a,s' \sim \rho^{\pi_D}_{T^o}} \left[w(s,a,s')\, c_{Q,\pi}(s,a,s') \right],$$

which is linear in w. Maximizing over $w \geq 0$ yields

$$w^\star_{Q,\tau,\pi}(s,a,s') = \begin{cases} +\infty, & \text{if } c_{Q,\pi}(s,a,s') > 0, \\ 0, & \text{otherwise.} \end{cases}$$

□

B Extended Related Work

Offline Imitation Learning. Behavioral Cloning (BC) [46] reduces imitation to supervised regression from expert state-action pairs, but its independence from environment dynamics makes it susceptible to compounding errors under covariate shift [49]. The DICE family of methods [35, 33, 38] addresses this by estimating stationary occupancy ratios between expert and learner distributions, enabling off-policy imitation without explicit policy access. Recent work [2, 4, 5] establishes that offline IL must respect a Markov balance condition coupling the expert policy to the underlying transition dynamics. Our formulation shares this occupancy-based perspective but departs fundamentally: rather than correcting for policy mismatch under fixed dynamics, we introduce importance weights through

a robust inner maximization over a transition uncertainty set, explicitly accounting for dynamics shift at test time: a regime where BC, DICE, and Markov-balance methods uniformly fail.

Robust Reinforcement Learning. Robust RL optimizes worst-case returns over an ambiguity set of transition kernels, typically using rectangular uncertainty sets and minimax dynamic programming [39, 29]. Subsequent work extends this to distributionally robust formulations under total variation, f-divergence, and Wasserstein metrics [22, 63, 42]. Offline variants estimate worst-case value functions from fixed datasets via nominal-measure reformulations [40], but continue to require known reward signals, rendering them inapplicable to the imitation setting where only expert demonstrations are available. Our work transplants these robust foundations into the IL regime, constructing a fully offline, reward-free formulation that hedges against transition uncertainty using only nominal expert data.

Robust Imitation Learning. A substantial body of IL work addresses robustness to demonstration imperfections, suboptimal labels, noisy annotations, or covariate shift: through label denoising [60, 54], corrective querying [50], and noise injection [36], but these assume fixed transition dynamics throughout. Meta-IL and multi-task IL frameworks improve cross-task generalization via shared experience [24, 30, 65] but similarly do not account for dynamics mismatch. Domain randomization [44, 28] perturbs physical parameters during training to improve sim-to-real transfer, but requires a high-fidelity simulator and access to perturbed environments, assumptions that are incompatible with the strictly offline setting we consider. Among methods that explicitly target transition uncertainty, [59] enforce Lipschitz regularization against input perturbations under fixed dynamics, while DRIL-DICE [53] mitigates covariate shift via f-divergence regularization, also under nominal dynamics. RIME [17] learns policies robust across families of MDPs but requires online rollouts for robustness estimation. The most closely related single-task methods are DRBC [41] and BE-DROIL [3], which robustify offline IL against transition uncertainty; however, both are single-task methods that require independent per-task optimization, incurring prohibitive computational cost in the multi-task regime where policies for many tasks must be recovered efficiently.

Behavior Foundation Models. BFMs currently achieve state-of-the-art zero-shot performance in RL by pretraining task-agnostic representations that generalize across both goal-conditioned and dense-reward tasks [57, 58]. Conceptually rooted in successor representations [21], universal value function approximators [52], successor features [11], and successor measures [12], modern BFMs are instantiated as either universal successor features [14, 43] or forward-backward (FB) representations [57, 31, 45]. Recent work identifies key failure modes of standard BFMs under distributional shift: [13] show that BFMs conflate policies across dynamics regimes and cannot adapt to unobserved dynamics changes, while [32] demonstrate degradation under partial observability. Both propose architectural remedies: belief encoders and memory-augmented models, respectively, but require pretraining data from multiple dynamics regimes. To the best of our knowledge, this is the first BFM approach that achieves robustness to dynamics variation using only a single nominal offline dataset per environment, with no changes to pretraining and no expert data required under perturbed dynamics.

C Experimental Setup

C.1 ExORL Domains

We evaluate our methods on three environments from the ExORL benchmark [62], which is built on top of the DeepMind Control Suite [55].

C.1.1 Walker

This domain features a two-legged agent that must achieve stable locomotion starting from a crouched configuration. The state space is 24-dimensional and the action space is 6-dimensional, comprising joint positions, velocities, and torques. The benchmark defines four tasks: *stand*, *walk*, *run*, and *flip*. The *stand* task rewards maintaining an upright torso with extended legs. The *walk* and *run* tasks extend this objective by encouraging forward motion at different speeds, while *flip* promotes rotational motion of the torso after achieving a standing posture. All reward signals are dense.

C.1.2 Quadruped

This domain involves a four-legged robot navigating within a three-dimensional space. The state and action spaces are 78- and 12-dimensional, respectively. We consider four tasks. The *quadruped_stand* task rewards maintaining an upright posture. The *quadruped_walk* and *quadruped_run* tasks additionally encourage forward motion through velocity-based rewards, with *quadruped_walk* also including a term promoting a minimum center-of-mass height. Finally, *quadruped_jump* rewards achieving sufficient vertical displacement. Rewards are dense across all tasks.

C.1.3 Cheetah

This domain features a planar biped designed for high-speed locomotion. The state is represented by a 17-dimensional vector capturing joint positions and velocities, while actions are 6-dimensional. We evaluate four tasks. In *cheetah_walk* and *cheetah_run*, rewards scale with forward velocity up to target speeds of approximately 2 m/s and 10 m/s, respectively. The tasks *cheetah_walk_backward* and *cheetah_run_backward* instead incentivize achieving specified backward velocities.

C.2 ExORL Datasets

Training is performed using 1M transitions (unless otherwise mentioned) sampled uniformly from datasets generated by different unsupervised exploration strategies within ExORL. Specifically, we use data collected by RND [16], APS [37], and PROTO [61].

C.3 Environment Perturbations

We evaluate robustness across three environments: Walker, Quadruped, and Cheetah, each subject to three perturbation types chosen to span distinct physical failure modes. All perturbations are applied at evaluation time only; the BFM pretraining is performed exclusively under nominal dynamics. For each environment, we describe the perturbation mechanism, the parameter modified in the MuJoCo physics model, and the rationale for the chosen sweep range.

C.3.1 Walker

The Walker is a planar bipedal robot with six actuated joints (hip, knee, ankle $\times$ 2 legs) and four tasks: stand, walk, run, and flip.

Gravity (+0% to +35% of nominal -9.81 m/s²). We scale the vertical component of gravitational acceleration from its nominal value up to a 35% increase in effective gravitational load. This directly increases the ground reaction forces the robot must overcome to maintain posture and generate forward momentum, stressing both the stance and swing phases of the gait cycle. The range is selected to produce monotonic degradation across all four tasks without inducing physics instability; the flip task is most sensitive due to its requirement for sustained aerial rotation against gravity.

Body Mass (+0% to +100% of nominal). We uniformly scale all non-world body masses: torso, thighs, shins, and feet, by up to 100% of their nominal values. Increased inertia raises the torque demand at every joint during acceleration and deceleration phases, directly stressing the actuator budget. At +100%, the total robot mass doubles, placing the system well outside the inertial regime experienced during expert demonstration.

Joint Friction Loss (0 to 30 Nm per joint). We add a constant passive resistive torque opposing motion at all six limb joints, drawn from the MuJoCo `dof_frictionloss` parameter (nominal: 0 Nm). Unlike viscous damping, which scales with joint velocity, frictionloss imposes a fixed load regardless of motion speed, effectively raising the torque threshold that must be exceeded before any useful joint motion occurs. At 30 Nm per joint, the six limb joints collectively impose 180 Nm of passive resistance, making fast locomotion tasks (run, flip) substantially more costly while moderately degrading slower tasks (walk, stand).

C.3.2 Quadruped

The Quadruped is a three-dimensional four-legged robot with twelve actuated joints (yaw, lift, extend $\times$ 4 legs) and four tasks: stand, walk, run, and jump.

Tilted Gravity (0 to 7.5 m/s² lateral component). We introduce a lateral gravitational component along the x-axis of the global frame while keeping the vertical component fixed at -9.81 m/s², producing an effective gravity vector tilted up to 37.4° from vertical. This creates a persistent lateral torque on the torso that the policy must continuously counteract to maintain uprightness, directly penalizing the upright reward component ($r = r_{\text{upright}} \times r_{\text{move}}$) that governs all four tasks. The range $[0, 7.5]$ m/s² is chosen as the stable regime; values above ≈ 8 m/s² induce numerical instability (`mjWARN_BADQACC`) in a subset of episodes due to the robot falling and entering degenerate contact configurations. Stand and walk are affected through postural destabilization; run and jump additionally suffer degraded forward propulsion as lateral compensation diverts actuator effort.

Ground Contact Stiffness (solref timeconst: 0.01 to 0.25 s). We increase the MuJoCo `geom_solref` timeconst of the floor geometry from its nominal value of 0.01 s to 0.25 s. The timeconst parameter controls the rise time of contact constraint forces: larger values produce slower, softer contact responses, effectively making the ground more compliant and energy-absorbing, analogous to transitioning from rigid pavement to soft terrain such as mud or foam. The quadruped's gait relies on brief, stiff ground contacts to deliver sharp propulsive impulses; increased compliance reduces the effective push-off force per step, degrading locomotion efficiency across all tasks with a stronger effect on dynamic tasks (run, jump) than static ones (stand).

Actuator Control Range (command bound: ± 1.0 to ± 0.3). We clip the control signal sent to all twelve actuators symmetrically to $[-v, v]$ where v decreases from the nominal 1.0 to 0.3. This is mechanistically distinct from gear ratio scaling: the actuator hardware remains at full torque capacity, but the command channel is truncated so that the policy cannot request torques in the upper $[v, 1]$ portion of its action space regardless of what the network outputs. At $v = 0.3$, the effective command range is reduced to 30% of nominal, which for the quadruped's gear-1 actuators directly caps the maximum controllable torque at 30% of the physical limit. This models command saturation or control signal degradation in the communication channel between the policy and the actuators.

C.3.3 Cheetah

The Cheetah is a planar six-joint runner (bthigh, bshin, bfoot, fthigh, fshin, ffoot) with nominal gear ratios of $120/90/60/90/60/30$ Nm/unit and four tasks: run, run_backward, walk, and walk_backward.

Actuator Strength (0% to -30% gear ratio reduction). We scale all six actuator gear ratios down from their nominal values by up to 30%, reducing the maximum joint torque to 70% of nominal. The gear ratios are large (120 Nm/unit for bthigh), so even a 30% reduction represents a substantial absolute torque loss of up to 36 Nm on the primary propulsive joint. The range is deliberately conservative: exploratory testing revealed a sharp performance cliff at -40% reduction where the actuators become insufficient to sustain any meaningful locomotion, making deeper perturbations uninformative for robustness comparison. The $[0\%, 30\%]$ range corresponds to the gradual degradation regime where policy quality rather than physical feasibility determines performance.

Joint Friction Loss (0 to 25 Nm per joint). We add a constant passive resistive torque opposing motion at all six actuated joints via the `dof_frictionloss` parameter (nominal: 0 Nm). Unlike viscous damping, which scales with joint velocity, frictionloss imposes a fixed load regardless of motion speed, effectively raising the torque threshold that must be exceeded before any useful joint motion occurs. At 25 Nm per joint, the six joints collectively impose 150 Nm of passive resistance per control cycle, substantial relative to the lighter actuators (ffoot: 30 Nm/unit nominal gear). The effect is particularly pronounced on the Cheetah due to the high-frequency nature of its gait: fast locomotion tasks (run, run_backward) require rapid joint reversals on every stride, paying the frictionloss penalty twice per cycle (opposing both flexion and extension). The range $[0, 25]$ Nm is chosen to span the regime from negligible degradation to near-collapse of high-speed locomotion while preserving meaningful signal across all four tasks.

Range of Motion (100% to 60% of nominal joint range retained). We shrink the mechanical range of motion of all six limb joints symmetrically around their center positions, reducing the allowed angular excursion to a specified fraction of the nominal range. For example, the bfoot joint with nominal range $[-4.01, 0.87]$ rad is restricted to $[-2.52, 0.37]$ rad at 60% retention. This models physical range restriction arising from joint damage, swelling, or mechanical interference. The Cheetah's fast gait requires large angular excursions, particularly at bfoot and fshin, to generate the leg sweeping motion that drives forward velocity; restricting these excursions forces the policy into a kinematically constrained regime that was not present during expert demonstration. The range

Hyperparameter	Value
Latent dimension d	50
F / ψ dimensions	(1024, 1024)
B / φ dimensions	(256, 256, 256)
Preprocessor dimensions	(1024, 1024)
Std. deviation for policy smoothing σ	0.2
Truncation level for policy smoothing	0.3
Learning steps	1,000,000
Batch size	512
Optimizer	Adam [34]
Learning rate	0.0001
Discount γ	0.98
Activations (unless otherwise stated)	ReLU
Target network Polyak smoothing coefficient	0.01
z-inference labels	10,000
z mixing ratio	0.5

Table 3: Hyperparameters for BFM methods.

Hyperparameter	Value
Optimization steps	3,000
Batch size	512
Learning rate	0.001
Optimizer	Adam [34]
Loss	MSE
Policy smoothing std. deviation σ	0.1

Table 4: Hyperparameters for FB-IL.

$[100\%, 60\%]$ is chosen to avoid the sharp feasibility cliff below 60% where joint constraints render normal locomotion physically impossible for all methods.

C.4 Evaluation Protocol

Our pipeline consists of two stages: (i) pre-training of the BFM and (ii) downstream task inference.

C.4.1 Pre-training Evaluation

Following [31], during BFM pre-training, we assess performance by measuring the cumulative reward (score) achieved on each task. Training is run for 10^6 steps, and evaluation is performed every $20{,}000$ steps. At each evaluation point, we execute 10 rollouts per task and compute the interquartile mean (IQM) of the resulting scores to obtain a robust estimate of performance. We select the checkpoint corresponding to the highest IQM aggregated across all tasks. This checkpoint is used for all subsequent downstream experiments.

C.4.2 Downstream Task Inference

Using the selected pretrained checkpoint, we run downstream imitation learning with 20 random seeds. Each seed's score is the cumulative return averaged over 100 evaluation episodes, and we report the mean across seeds with a 95% confidence interval.

C.5 Computational Resources

All experiments were conducted on a single NVIDIA Quadro P5000 GPU with 16GB memory. Pretraining of BFM took approximately 14 hours for 1M steps.

Hyperparameter	Value
Robustness parameter ε_l	0.8
Optimization steps	5,000
Batch size	512
Learning rate	0.0005
Optimizer	Adam [34]
Policy smoothing std. deviation σ	0.1
(Task Vector : Dual Update) Frequencies	1 : 1

Table 5: Hyperparameters for RBFM-Light.

D Implementation Details

D.1 Forward-Backward Representations

D.1.1 Architecture

Our implementation of the forward-backward framework is based on the design introduced by [31], with specific hyperparameter choices summarized in Table 3.

D.1.2 Forward Representation $F(s, a, z)$

The forward network operates on preprocessed inputs. In particular, the state-action pair (s, a) and the state-task pair (s, z) are each passed through separate MLPs, which project them into a shared embedding space of dimension 512. These two embeddings are then concatenated and fed into an additional MLP, which outputs an embedding vector of dimension d.

D.1.3 Backward Representation $B(s)$

The backward network is implemented as an MLP that takes the state s as input and produces a d-dimensional embedding vector.

D.1.4 Actor $\pi_z(s)$

The policy network adopts a similar design to the forward model. The inputs (s, a) and (s, z) are first encoded independently using separate MLPs into 512-dimensional embeddings. These embeddings are concatenated and passed through a final MLP that generates an action vector with the dimension of the action space. A Tanh activation is applied at the output layer to ensure bounded actions. During training, Gaussian noise with standard deviation σ is injected into the actions for smoothing, following [25].

D.1.5 Additional Details.

To improve training stability, layer normalization [10] is applied, and Tanh activations are used in the first layer of each MLP.

D.2 Task Sampling Distribution $\mathcal{Z}$

FB representations require a strategy for generating the task vector z during training. Following [58], we adopt a combination of two sampling procedures. First, z is drawn uniformly from the surface of a hypersphere with radius $\sqrt{d}$ in $\mathbb{R}^d$. Second, z is obtained by encoding states $s \sim \mathcal{D}$ through the backward model, i.e., $z = B(s)$. Due to the ℓ_2 normalization applied in the backward network, these vectors also lie on the hypersphere, although their distribution is not uniform. At each training step, we sample z using these two approaches with equal probability.

D.3 Discussion on usage of f-divergence in RBFM-Heavy

Lemma 2 provides a total variation (TV) bound on the mismatch between occupancy measures under nominal and perturbed dynamics. We extend this to general f-divergences by restricting to generators bounded by the TV generator.

Lemma 4. *Let $f : [0, \infty) \to \mathbb{R}$ satisfy $f(1) = 0$ and $f(t) \leq \alpha f_{\mathrm{TV}}(t)$ for all $t \geq 0$, where $f_{\mathrm{TV}}(t) = \frac{1}{2}|t - 1|$. Then, for any policy π and $T \in \mathcal{T}_{T^o}(\varepsilon')$,*

$$D_f\big(\rho^\pi_T \,\|\, \rho^\pi_{T_o}\big) \leq \alpha D_{\mathrm{TV}}\big(\rho^\pi_T, \rho^\pi_{T_o}\big) \leq \alpha \frac{\varepsilon'}{1-\gamma}. \tag{17}$$

Proof. See Lemma 2 in [3] for details. □

Our formulation further requires f to be differentiable with invertible f'. Since f_{TV} is non-differentiable at $t = 1$, we use the smooth SoftTV generator:

$$f_{\mathrm{SoftTV}}(x) = \frac{1}{2}\log\big(\cosh(x-1)\big), \quad (f'_{\mathrm{SoftTV}})^{-1}(y) = \tanh^{-1}(2y) + 1. \tag{18}$$

As $f_{\mathrm{SoftTV}}(x) \leq f_{\mathrm{TV}}(x)$ for all x, setting $\alpha = 1$ yields

$$D_{f_{\mathrm{SoftTV}}}\big(\rho^\pi_T \,\|\, \rho^\pi_{T_o}\big) \leq \frac{\varepsilon'}{1-\gamma}. \tag{19}$$

We use SoftTV in practice, though any generator satisfying the conditions in Lemma 4 is admissible.

D.4 BFM-based Hyperparameters

D.4.1 Pretraining

We use the publicly available code at [64] for BFM and RBFM-based approaches. The hyperparameters used for pretraining BFM model are provided in Table 3. Since FB-IL, RBFM-Light, and RBFM-Heavy share the same pretrained BFM, these hyperparameters are common to all these approaches.

D.4.2 Task-Inference

FB-IL. The hyperparameters used for inferring task vector in FB-IL are provided in Table 4.

RBFM-Light. The hyperparameters used for inferring task vector in RBFM-Light are provided in Table 5.

RBFM-Heavy. Q-network architecture. The Q-function in RBFM-Heavy is parameterized as a two-layer MLP with hidden dimension 64 and ReLU activations, operating on the frozen forward model features $\bar{F}(s, a, z_w)$ evaluated at a fixed warm-start task embedding z_w. Specifically, given the pretrained forward model F, we compute $\bar{F}(s, a, z_w) \in \mathbb{R}^d$ and pass it through the MLP to produce a scalar Q-value estimate. Critically, a key motivation for building on BFMs is their ability to infer task-specific policies within minutes; using a large Q-network over raw observations and actions would render inference computationally prohibitive, defeating this purpose entirely. By operating on the pretrained FB model instead, the Q-network remains lightweight while still leveraging the rich representational structure already encoded during pretraining. The hyperparameters used for task vector inference in RBFM-Heavy are provided in Table 6.

D.5 Single Task Robust Offline IL baselines

We compare against the following Robust Offline IL baselines in our experiments.

Hyperparameter	Value
Robustness parameter ε	0.8
Optimization steps	5,000
Batch size	512
γ	0.99
Learning rate	0.0003
Optimizer	Adam [34]
Policy smoothing std. deviation σ	0.1
Q-network architecture	MLP
Q-network hidden dimension	64
Q-network hidden layers	2
Q-network activation	ReLU
(Task Vector : Dual Update) Frequencies	1 : 1
Dual variables learning rate	0.0003

Table 6: Hyperparameters for RBFM-Heavy.

Environment	Robustness param ε_l	Policy layers	Activation	Max steps	Learning rate	LR decay	Decay rate	Decay freq.
Cheetah	0.3	(256, 256)	Tanh	2M	1×10^{-5}	True	0.9	10k
Walker	0.5	(256, 256)	Tanh	2M	1×10^{-4}	True	0.95	10k
Quadruped	0.6	(256, 256)	Tanh	2M	1×10^{-4}	True	0.9	10k

Table 7: Hyperparameter configuration for DRBC.

D.5.1 DRBC

We implement DRBC [41] in PyTorch using a policy network with two hidden layers of 256 units each and Tanh activations. Models are trained for 2×10^6 gradient steps with a batch size of 256, optimized using Adam. The learning rate follows the schedule in [41], with an initial value of 1×10^{-4} for all environments except HalfCheetah, where 1×10^{-5} is used. The learning rate is decayed exponentially every $10{,}000$ steps with a factor between 0.9 and 0.95. The robustness parameter ε_l is set to 0.3, 0.5, and 0.6 for Cheetah, Walker, and Quadruped, respectively (see Table 7).

D.5.2 BE-DROIL

We also implement BE-DROIL [3] in PyTorch. The policy is modeled as a TanhGaussian network with two fully connected layers of 256 units, producing a Gaussian distribution whose outputs are squashed via a Tanh transformation to respect action bounds. Both the policy and Q-function are trained using Adam with a learning rate of 5×10^{-5} and a batch size of 512. The Q-function is parameterized as a two-layer MLP with 256 hidden units and ReLU activations, and we use a discount factor of $\gamma = 0.99$. The policy objective minimizes the mean squared error between actions sampled from the learned policy and those from the expert i.e. $\mathcal{L}_\pi(s) = \mathrm{MSE}(a, a_D), \quad a \sim \pi(\cdot \mid s),\ a_D \sim \pi_D(\cdot \mid s)$, where π and π_D denote the learned and expert policies, respectively. Training is performed for 10^6 gradient steps, alternating between a policy update and a joint update of the Q-function and τ network. See Table 8 for additional details.

E Algorithm

Pseudo-code for pre-training of BFM is provided in Algorithm 1. This follows Algorithm 1 of [31] with the conservative term removed. Pseudo-code to extract the task vector for our RBFM-Light method is provided in Algorithm 2. Finally, Algorithm 3 provides the pseudo-code for our RBFM-Heavy algorithm.

Algorithm 1 Unsupervised Pre-training of Forward-Backward Representations

Require: Exploratory Dataset $\mathcal{D}$ of trajectories

Initialize F_{θ_F}, B_{θ_B}, and policy π_z
Set number of steps N, z-distribution $\mathcal{Z}$, Polyak factor ν, batch size b
for $n = 1$ to N **do**
  Sample mini-batch $\{(s_i, a_i, s_{i+1})\}_{i=1}^{b} \sim \mathcal{D}$
  Sample $\{z_i\}_{i=1}^{b} \sim \mathcal{Z}$
  // FB Update
  Sample $a_{i+1} \sim \pi(s_{i+1}, z_i)$
  Update F, B using $\{(s_i, a_i, s_{i+1}, a_{i+1}, z_i)\}$ using Eq. (6)
  // Actor Update
  Sample $a_i \sim \pi(s_i, z_i)$
  Update actor to maximize Eq. (7), i.e.,

$$\mathbb{E}\big[F(s_i, a_i, z_i)^\top z_i\big]$$

  // Target network updates (Polyak averaging)
  $\theta_F^- \leftarrow (1-\nu)\theta_F^- + \nu\theta_F$
  $\theta_B^- \leftarrow (1-\nu)\theta_B^- + \nu\theta_B$
end for

Algorithm 2 RBFM-Light

Require: Expert dataset $\mathcal{D}_e = \{(s_j, a_j)\}_{j=1}^{N}$, uncertainty radius ε_l

Warm-start $z \leftarrow z_w$ as defined in Implementation details in Section 4
for iteration $k = 1, \ldots, K$ **do**
  Sample mini-batch $\{(s_i, a_i)\}_{i=1}^{b} \sim \mathcal{D}_e$
  //RBFM-Light objective

$$L := \max_{i \in [b]} \mathcal{L}(\pi_z(\cdot \mid s_i), a_i),$$

$$\mathcal{L}_{\text{RBFM-Light}}(z) = \min_{\lambda \in [0, (1+\varepsilon_l)L]} \Bigg[\frac{1}{b} \sum_{i=1}^{b} \big(\mathcal{L}(\pi_z(\cdot \mid s_i), a_i) - \lambda\big)_+ + \varepsilon_l \big(\max_{i \in [b]} \mathcal{L}(\pi_z(\cdot \mid s_i), a_i) - \lambda \big)_+ + \lambda \Bigg].$$

  // Dual update
  $\lambda \leftarrow \arg\min_{\lambda \in [0, (1+\varepsilon_l)L]} \ \mathcal{L}_{\text{RBFM-Light}}(z, \lambda)$
  // Policy update
  $z \leftarrow \arg\min_{z} \ \mathcal{L}_{\text{RBFM-Light}}(z, \lambda)$
end for
return z

Component	Architecture	Hidden units	Activation	Batch size	Learning rate	Steps	γ	Update ratio (policy : Q, τ)
Policy π	TanhGaussian MLP	(256, 256)	ReLU	512	5×10^{-5}	1M	0.99	1:1
Q-function Q	MLP	(256, 256)	ReLU	512	5×10^{-5}	1M	0.99	1:1

Table 8: Hyperparameter configuration for BE-DROIL.

Algorithm 3 RBFM-Heavy

Require: Expert Dataset $\mathcal{D}_e = \{(s_j, a_j, s'_j, a'_j)\}_{j=1}^N$, initial state distribution μ, discount γ, robustness parameter ε

Initialize Q_θ, scalar $\tau \geq 0$, and warm-start $z \leftarrow z_w$ as defined in Implementation details in Section 4

for iteration $k = 1, \ldots, K$ **do**

  Sample mini-batch $\{(s_i, a_i, s'_i, a'_i)\}_{i=1}^b \sim \mathcal{D}_e$

  Sample $\{(s_0^{(i)}, a_0^{(i)})\}_{i=1}^b \sim \mathcal{D}_e$ (See Implementation details in Experiments section 4)

  // Step 1: Compute optimal density ratio $w^\star$

  Compute $\{c_{Q,\pi_z}(s_i, a_i, s'_i)\}_{i=1}^b$ as defined in Eq. (15)

  Compute $\{w^\star_{Q,\tau,\pi_z}(s_i, a_i, s'_i)\}_{i=1}^b$ via closed-form in Proposition 3

  // Step 2: Update critic (Q, τ)

  Estimate gradient of:

$$\mathcal{L}_{\theta,\tau} = \frac{1-\gamma}{b} \sum_{i=1}^{b} Q_\theta(s_0^i, a_0^i) + \varepsilon\tau + \frac{1}{b} \sum_{i=1}^{b} \Big[-\tau\, f\big(w^\star_{Q_\theta,\tau,\pi_z}(s_i, a_i, s'_i)\big) + w^\star_{Q_\theta,\tau,\pi_z}(s_i, a_i, s'_i)\, c_{Q_\theta,\pi_z}(s_i, a_i, s'_i) \Big]$$

  Update $\theta \leftarrow \theta - \eta_Q \nabla_\theta \mathcal{L}_{Q_\theta,\tau}$

  Update $\tau \leftarrow \max(0, \tau - \eta_\tau \nabla_\tau \mathcal{L}_{Q_\theta,\tau})$

  // Step 3: Policy update (actor)

  Estimate:

$$\mathcal{L}_{\pi_z} = \sum_{i=1}^{b} \frac{w^\star_{Q_\theta,\tau,\pi_z}(s_i, a_i, s'_i) L_{\pi_z}(s_i)}{b}$$

  Update $z \leftarrow z - \eta_\pi \nabla_z \mathcal{L}_{\pi_z}$

end for

return (Q_θ, τ, z)

F Extended Results

F.1 Further discussion of (Q2)

Walker. Figure 6 shows that the advantage of the RBFM variants over FB-IL is largest under gravity and body mass perturbations. The gap peaks on flip, a demanding task that requires coordinated full-body rotation and where the perturbations shift the transition distribution the most. Under joint friction loss, all three methods degrade monotonically as the resistive torque grows, but RBFM-Heavy keeps the highest return throughout the sweep and RBFM-Light stays in between. FB-IL has the highest variance at severe perturbation levels, which indicates that its inferred z becomes unstable once the dynamics deviate far from the nominal expert distribution. Walker gives the clearest support for our hypothesis: DRO-based task inference produces task embeddings that remain viable under transition distributions overlapping the worst-case support identified during inference, while the FB-IL embedding is brittle outside the nominal support.

Quadruped. The quadruped is a harder robustness benchmark because its redundant four-leg morphology passively compensates for many single-parameter perturbations. Even so, Figure 2 shows that under tilted gravity and ground contact stiffness, RBFM-Heavy keeps the highest return on all four tasks, followed by RBFM-Light, while FB-IL degrades the most. Tilted gravity adds a persistent lateral torque that directly penalizes the upright reward term, and the contact stiffness

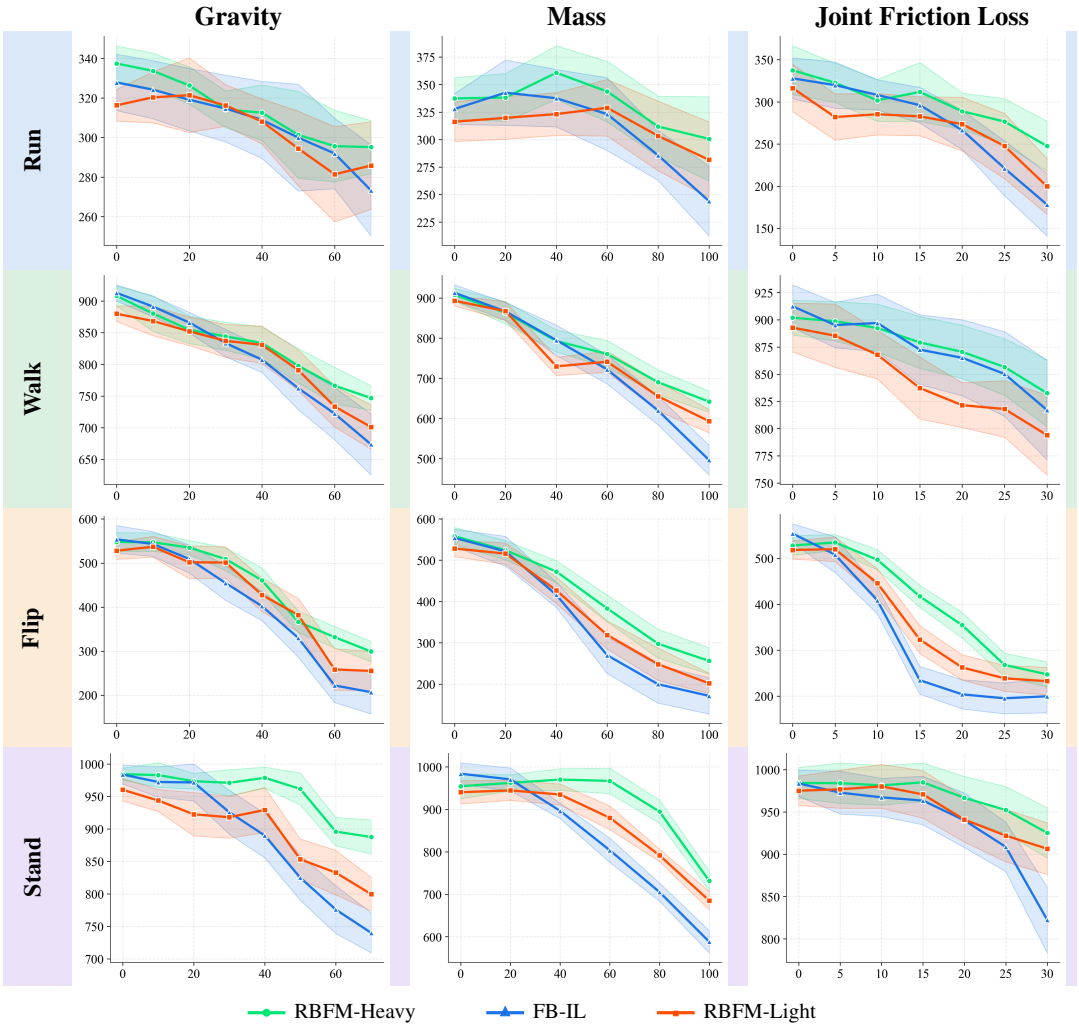


Figure 6: Walker performance with 95% confidence interval over 20 seeds across four tasks (rows) and three perturbation types (columns): gravity and body mass increase the physical load on the robot (% change from nominal); joint friction loss adds passive resistive torque at each joint, simulating mechanical wear (absolute Nm per joint). Pretrained on RND data.

perturbation softens contact impulses, which reduces push-off efficiency. The advantage is largest on the dynamic tasks (run, jump), where the perturbations shift the distribution more, and smaller on stand and walk, where the quadruped's passive stability partly compensates for the mismatch. Under actuator control-range clipping, the advantage of RBFM-Heavy is clearest at intermediate levels. Beyond a critical clipping threshold all methods collapse, which suggests that DRO-based robustness works within the recoverable dynamics regime but cannot compensate for command truncation that makes the task infeasible for any task embedding.

Cheetah. Figure 7 shows the sharpest three-way separation of the three domains, across all four tasks and all three perturbation types. Joint friction loss applies a fixed passive resistive torque at every joint on every control step, and its effect compounds across the six actuated degrees of freedom. Under it, FB-IL collapses to near-zero return at moderate perturbation levels, while RBFM-Heavy keeps substantial locomotion performance and RBFM-Light lands in between. Under reduced actuator strength, the separation is present from the first perturbation level, so even a small reduction in gear ratio exposes the brittleness of the FB-IL task embedding. Range of motion restriction is the only kinematic perturbation in our experiments: it constrains the feasible configurations, not force generation or dissipation. Under it, RBFM-Heavy keeps the highest return throughout, while FB-IL shows higher variance, which suggests greater sensitivity to the hard feasibility constraints of reduced

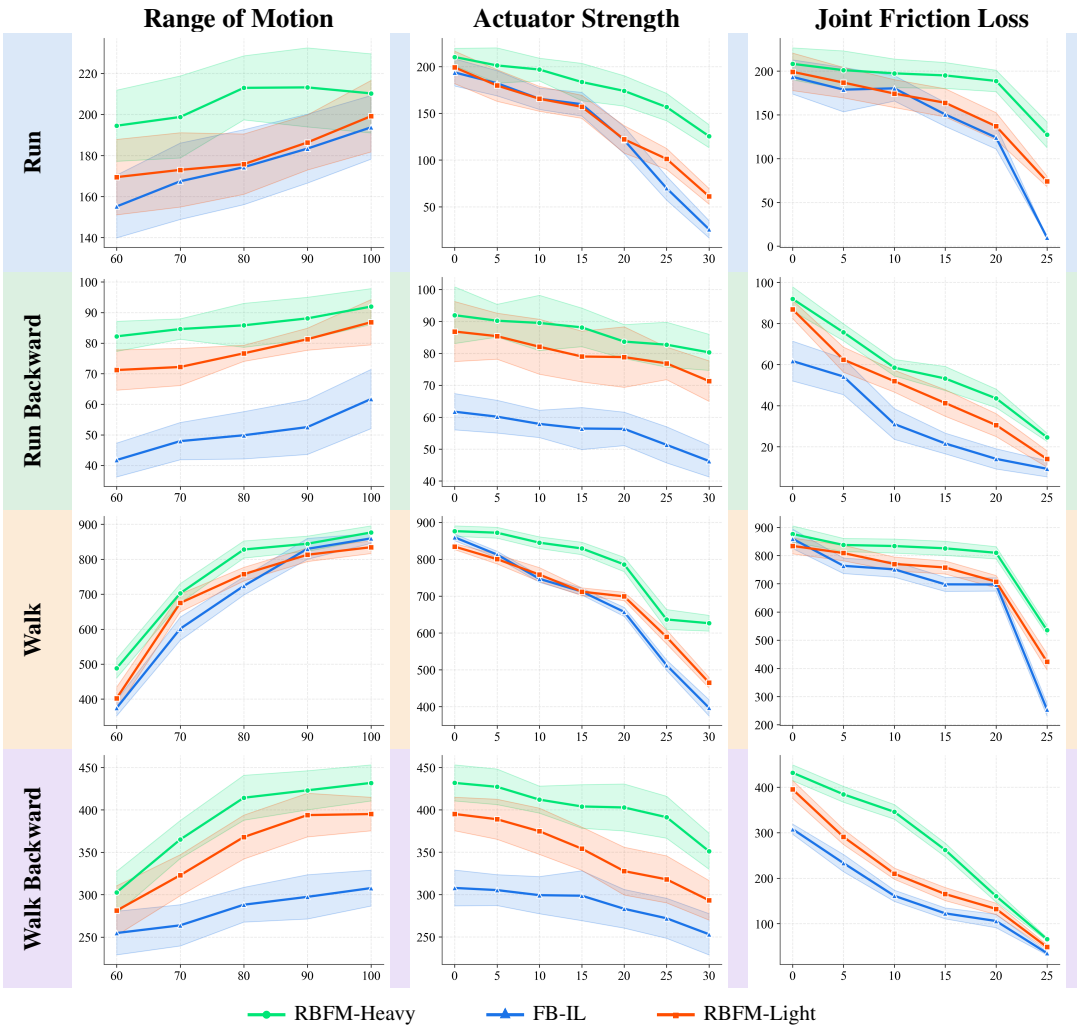


Figure 7: Cheetah performance with 95% confidence interval over 20 seeds across four tasks (rows) and three perturbation types (columns): range of motion restricts how far each joint can move, simulating mechanical damage (% of nominal retained); actuator strength reduces peak joint torque, simulating motor degradation (% reduction); joint friction loss adds passive resistive torque at each joint, simulating mechanical wear (absolute Nm per joint). Pretrained on RND data.

joint limits. The separation is sharper on the cheetah than on the walker and quadruped, likely because fast asymmetric locomotion is very sensitive to dynamics mismatch. The cheetah's gait involves large joint excursions and brief ground contacts, which concentrates the nominal transition distribution in a narrow dynamical regime where perturbations more easily push transitions outside the expert support.

F.2 Q6: Different Sources of Pretraining Data

Figures 8 and 9 compare RBFM-Heavy and RBFM-Light with FB-IL on Walker pretrained on APS and PROTO data, respectively. The trends match those under RND pretraining: RBFM-Heavy is the most robust, followed by RBFM-Light and then FB-IL.

F.3 Q1: Robustness radius

Table 9 lists the values behind Figure 4. For $\varepsilon \in \{0.6, 0.8, 1.0\}$, returns stay within 3.5% of each other at every stiffness level, and $\varepsilon = 0.8$ gives the highest return from 0.15 onward.

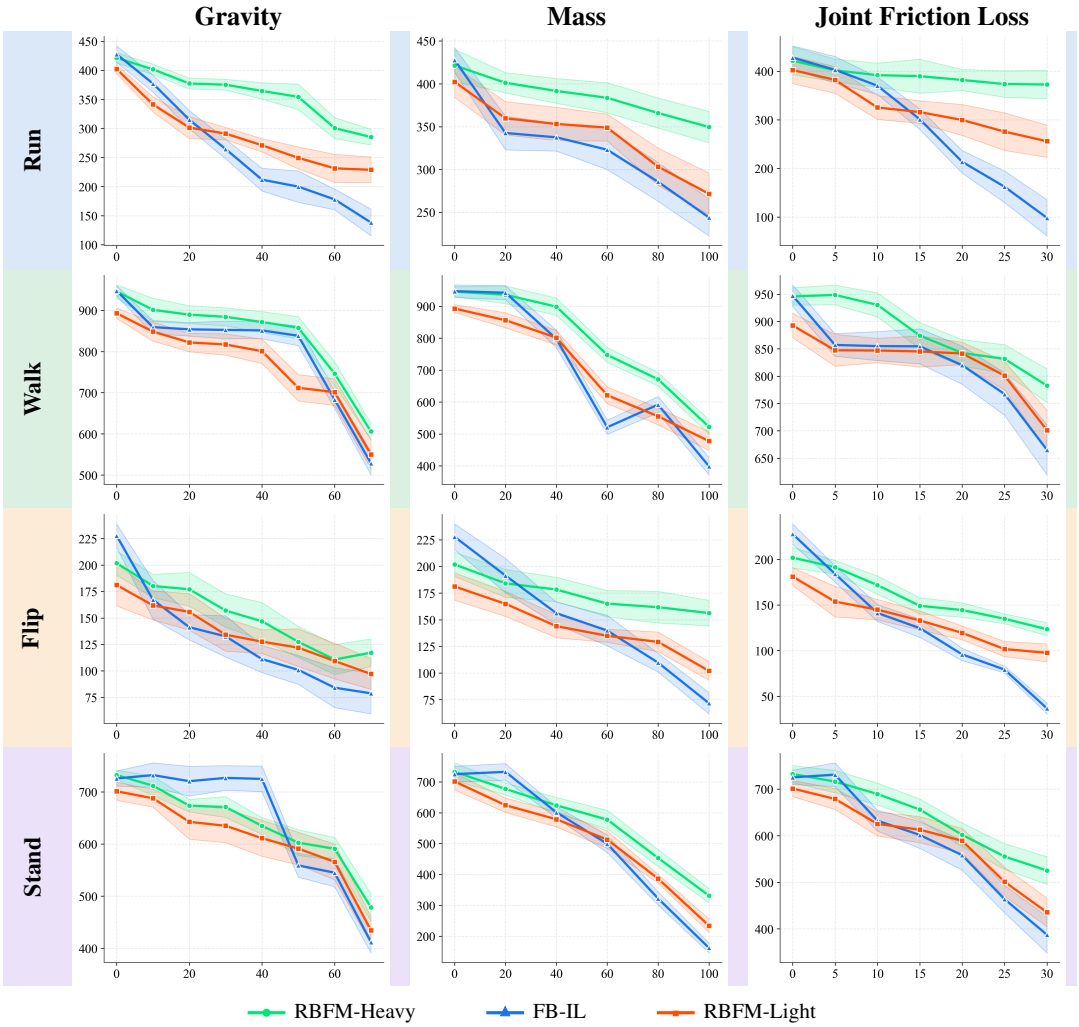


Figure 8: Walker performance with 95% confidence interval over 20 seeds across four tasks (rows) and three perturbation types (columns): gravity and body mass increase the physical load on the robot (% change from nominal); joint friction loss adds passive resistive torque at each joint, simulating mechanical wear (absolute Nm per joint). Pretrained on APS data.

F.4 Q3: Non-stationary and compound perturbations

We test two regimes beyond the stationary single-factor protocol. In both, the task vector is inferred once under nominal dynamics, with no re-inference or online adaptation. In *within-episode drift*, the perturbation parameter either increases in equal increments every 10 steps from nominal to its maximum by the end of the episode (gradual), or stays nominal for the first half of the episode and then jumps to its maximum (abrupt); Table 10 reports the results. In *compound perturbations*, two or three physical parameters are perturbed at once (Table 11).

F.5 Q4: Single-task baselines with more demonstrations

To test whether DRBC and BE-DROIL underperform because of data starvation, we retrain both with 1,000 expert trajectories (≈1M expert transitions, 250× the 4 available to RBFM). Table 12 reports the results.

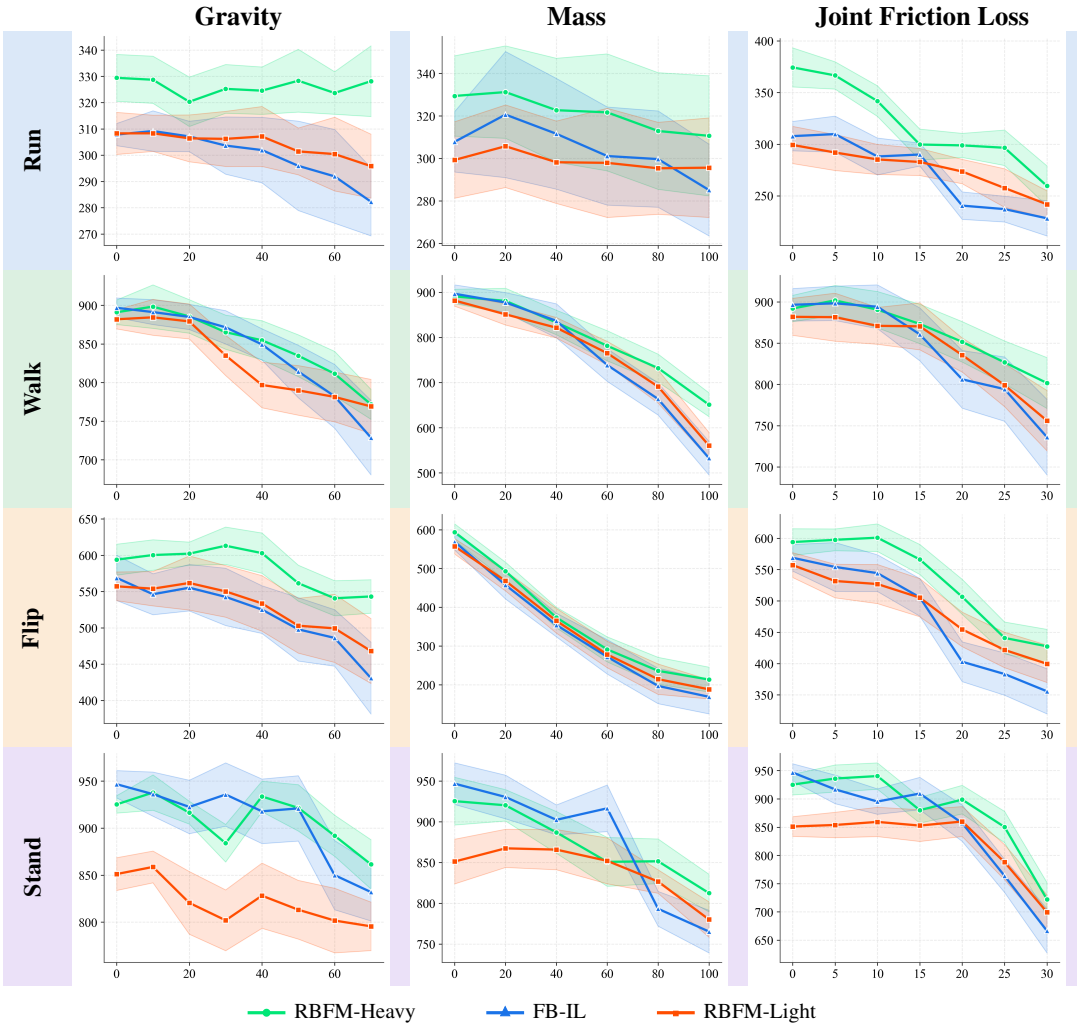


Figure 9: Walker performance with 95% confidence interval over 20 seeds across four tasks (rows) and three perturbation types (columns): gravity and body mass increase the physical load on the robot (% change from nominal); joint friction loss adds passive resistive torque at each joint, simulating mechanical wear (absolute Nm per joint). Pretrained on PROTO data.

G Discussion and Limitation

RBFM changes only task inference, so it needs what FB needs: enough exploratory data to learn a well-structured representation. The quantity study in (Q6) probes this on Walker flip, with 1M transitions as the standard operating point [45, 31] and 100k included only to find where the paradigm breaks. Figure 10 tracks the IQM return during FB pretraining on Walker with RND datasets of 100k and 500k transitions. With 100k transitions the representation does not converge, and its IQM on run, stand, flip, and walk stays far below that of the 500k run. Every inference method built on it then degrades, FB-IL included, with no separation between them (Figure 5a). The dependence is therefore on the pretrained BFM, and robust task inference adds no fragility beyond what FB-IL shows on the same representation; [31] study this degradation of FB under limited and low-quality data. Meeting the requirement is cheap in the BFM setting: the pretraining data needs no rewards, task labels, or demonstrations, and it is collected once per environment and amortized over all downstream tasks.

H LLM Usage

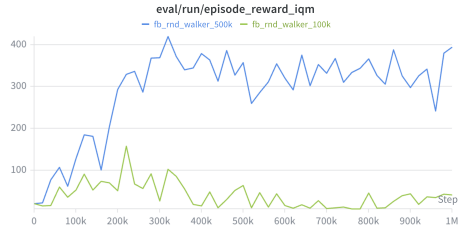
We used LLMs only to assist with the grammar and readability of the paper.

Table 9: RBFM-Heavy on Quadruped jump under ground contact stiffness perturbation for five values of ε, with FB-IL for reference. Mean $\pm$ 95% CI over 20 seeds. Best per row in bold.

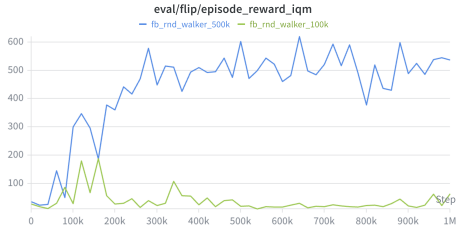
Stiffness	**FB-IL**	$\varepsilon = 0.2$	$\varepsilon = 0.4$	$\varepsilon = 0.6$	$\varepsilon = 0.8$	$\varepsilon = 1.0$
0.01 (nominal)	681.2 ± 10.2	$\mathbf{688.9} \pm 9.2$	678.7 ± 7.2	667.9 ± 8.7	665.8 ± 7.8	645.8 ± 8.8
0.05	$\mathbf{679.4} \pm 9.5$	632.4 ± 8.1	671.7 ± 7.5	655.4 ± 9.6	652.1 ± 8.5	644.1 ± 9.6
0.10	641.2 ± 8.8	615.1 ± 7.5	641.6 ± 8.4	$\mathbf{651.3} \pm 6.8$	645.3 ± 7.5	635.3 ± 8.7
0.15	615.8 ± 9.2	586.3 ± 9.2	615.9 ± 7.3	635.5 ± 7.9	$\mathbf{639.1} \pm 9.6$	631.5 ± 7.9
0.20	595.5 ± 7.9	562.5 ± 8.5	591.5 ± 9.6	623.6 ± 8.6	$\mathbf{633.8} \pm 9.4$	630.9 ± 6.9
0.25	580.3 ± 10.1	531.3 ± 9.1	557.6 ± 8.5	621.4 ± 9.5	$\mathbf{632.7} \pm 8.5$	628.6 ± 9.4

Table 10: Within-episode non-stationary dynamics. *Gradual*: the perturbation parameter increases in equal increments every 10 steps from nominal to its maximum by episode end. *Abrupt*: nominal for the first half of the episode, then maximum. The task vector is inferred once under nominal dynamics (no re-inference, no online adaptation). Total episode return, mean $\pm$ 95% CI over 20 seeds. Best per row and profile in bold.

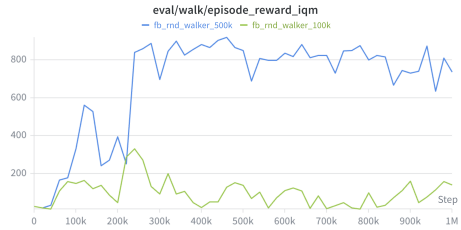
		Gradual drift			**Abrupt shift**		
Domain (perturbation)	**Task**	FB-IL	RBFM-Light	RBFM-Heavy	FB-IL	RBFM-Light	RBFM-Heavy
Walker Joint friction loss, 0→30 Nm	stand	851.4 ± 9.4	871.2 ± 8.3	$\mathbf{953.2} \pm 7.4$	881.4 ± 3.4	931.2 ± 3.2	$\mathbf{961.3} \pm 3.9$
	walk	848.8 ± 6.7	812.6 ± 7.0	$\mathbf{881.3} \pm 7.3$	811.3 ± 4.3	771.6 ± 4.2	$\mathbf{835.3} \pm 3.7$
	run	275.2 ± 2.0	287.3 ± 2.1	$\mathbf{305.4} \pm 1.9$	214.7 ± 3.1	231.2 ± 4.3	$\mathbf{271.3} \pm 3.5$
	flip	356.4 ± 4.5	395.2 ± 3.2	$\mathbf{412.4} \pm 3.8$	311.0 ± 2.7	342.8 ± 3.1	$\mathbf{382.2} \pm 2.8$
Quadruped Contact stiffness, 0.01→0.25	stand	809.1 ± 8.2	837.2 ± 9.8	$\mathbf{871.3} \pm 7.1$	827.2 ± 3.7	845.0 ± 4.0	$\mathbf{871.2} \pm 4.1$
	walk	631.6 ± 2.8	642.2 ± 3.1	$\mathbf{655.4} \pm 4.1$	660.2 ± 4.3	677.3 ± 3.9	$\mathbf{694.3} \pm 4.2$
	run	382.7 ± 6.4	401.2 ± 5.8	$\mathbf{423.2} \pm 4.3$	396.6 ± 3.3	401.1 ± 1.2	$\mathbf{415.3} \pm 2.1$
	jump	637.0 ± 2.3	644.2 ± 2.8	$\mathbf{661.2} \pm 2.1$	637.8 ± 3.8	646.3 ± 1.8	$\mathbf{662.5} \pm 2.4$
Cheetah Actuator strength, 0→ −30%	run	111.2 ± 3.4	131.1 ± 2.9	$\mathbf{172.3} \pm 2.1$	99.1 ± 1.9	123.2 ± 2.6	$\mathbf{171.3} \pm 2.1$
	run_backward	55.0 ± 1.1	81.1 ± 1.2	$\mathbf{90.1} \pm 0.6$	51.4 ± 1.1	81.2 ± 1.5	$\mathbf{88.3} \pm 1.2$
	walk	611.0 ± 7.8	633.2 ± 5.4	$\mathbf{681.2} \pm 5.6$	596.7 ± 4.1	631.5 ± 3.2	$\mathbf{742.7} \pm 4.2$
	walk_backward	255.8 ± 4.5	351.2 ± 4.9	$\mathbf{411.2} \pm 3.2$	271.2 ± 2.3	341.9 ± 2.7	$\mathbf{399.6} \pm 2.8$



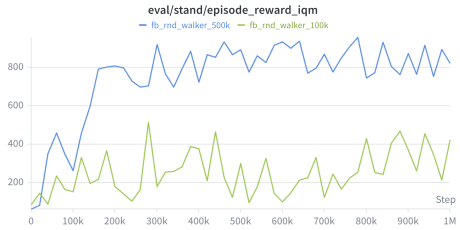
(a) Run



(b) Flip



(c) Walk



(d) Stand

Figure 10: FB pretraining on Walker with RND datasets of 100k and 500k transitions. Every 20,000 steps, the model is evaluated with 10 rollouts and the IQM return is recorded.

Table 11: Compound perturbations: two or three physical parameters shifted simultaneously. Mean $\pm$ 95% CI over 20 seeds. RBFM-Heavy's mean relative gain over FB-IL is +43.1% with two simultaneous perturbations and +83.5% with three.

Domain	**Perturbations applied simultaneously**	FB-IL	RBFM-Light	RBFM-Heavy
Walker	Gravity +20%, Mass +40%	281.5 ± 15.4	371.2 ± 13.2	**443.5** ± 12.8
	Gravity +20%, Friction 10 Nm	301.2 ± 12.8	405.6 ± 11.8	**484.3** ± 10.8
	Mass +40%, Friction 10 Nm	268.2 ± 13.2	365.9 ± 12.9	**442.3** ± 13.1
	Gravity +20%, Mass +40%, Friction 10 Nm	191.6 ± 10.9	282.2 ± 11.2	**356.3** ± 11.9
Quadruped	Tilted gravity 3.0, Contact stiffness 0.10	398.5 ± 12.2	441.7 ± 10.9	**481.2** ± 9.8
	Tilted gravity 3.0, Ctrlrange 0.7	361.0 ± 15.4	432.8 ± 14.9	**479.8** ± 10.2
	Contact stiffness 0.10, Ctrlrange 0.7	412.2 ± 12.9	444.2 ± 11.4	**499.0** ± 8.9
	Tilted gravity 3.0, Contact stiffness 0.10, Ctrlrange 0.7	211.3 ± 14.5	273.6 ± 14.3	**361.7** ± 11.9
Cheetah	Actuator -15%, Friction 10 Nm	155.7 ± 13.2	176.2 ± 11.1	**213.6** ± 9.8
	Actuator -15%, Range of motion 80%	141.3 ± 12.8	161.5 ± 10.9	**209.8** ± 8.8
	Friction 10 Nm, Range of motion 80%	161.2 ± 14.3	181.3 ± 9.9	**232.1** ± 7.4
	Actuator -15%, Friction 10 Nm, Range of motion 80%	91.2 ± 15.6	132.9 ± 12.1	**176.3** ± 7.8

Table 12: Single-task robust IL baselines with 250$\times$ more expert data (Walker, body mass). DRBC and BE-DROIL trained on 1,000 expert trajectories ($\approx$1M expert transitions) remain below RBFM-Heavy trained on 4. Mean return over 20 seeds.

Task	**Mass**	DRBC (4)	DRBC (1000)	BE-DROIL (4)	BE-DROIL (1000)	RBFM-Heavy (4)
stand	0%	840.1	860.3	890.1	902.4	**954.2**
stand	+100%	592.1	611.3	714.1	715.5	**731.5**
walk	0%	765.1	782.2	823.1	841.3	**908.0**
walk	+100%	485.1	499.1	615.1	621.2	**642.1**
run	0%	285.3	289.8	301.3	313.9	**337.4**
run	+100%	247.6	255.3	254.6	266.5	**300.6**
flip	0%	452.1	461.6	505.1	516.3	**557.9**
flip	+100%	210.0	211.1	222.0	231.0	**256.3**