Multi-Objective Reinforcement Learning Using Routed Ensembles and Trajectory Attribution

Zhenshuo Zhang
Northeastern University
Boston, MA

Ruoxuan Xiong
Emory University
Atlanta, GA

Hongyang R. Zhang
Northeastern University
Boston, MA

Abstract

Multi-objective reinforcement learning is a fundamental problem for modern AI, with applications spanning post-training, reasoning models, and robotic control. A central challenge is to learn policies that balance competing objectives, such as helpfulness versus honesty in safety and alignment, or speed versus energy in robotics. The optimal policy can vary sharply with objective-weighting combinations, and the extent of this variation depends crucially on the degree of conflict among objectives, making per-preference policy learning sample-inefficient. In this paper, we introduce objective-conflict coefficients and derive near-optimal regret bounds for online settings using this measure, thereby quantifying the impact of misalignment on learning efficiency. Then, we design routed ensemble policy learning, an algorithm that maintains a small set of basis policies and dynamically routes trajectories to the policy with the highest potential reward. Finally, we consider trajectory selection as another approach to mitigate objective conflict, and design a kernel surrogate trajectory attribution algorithm that estimates how feedback signals from a set of trajectories influence policy learning on other trajectories. The kernel can capture nonlinear interactions such as synergy and antagonism between trajectories. We evaluate our approach across diverse environments and benchmarks: routed ensembles improve average performance by 4.3% over baselines for online environments from Meta-World and MO-Gymnasium, and kernel surrogate attribution improves correlation with leave-one-out analysis over existing baselines by 50%. When applied to trajectory selection, kernel surrogate attribution yields a 27% improvement in reward over baseline attribution methods.

1 Introduction

Given m potentially competing objectives in a reinforcement learning (RL) environment, how can we learn a small set of policies that maximizes expected reward under any specified preference weighting? This problem is fundamental in multi-agent systems [29], and is ubiquitous in modern AI systems. In robotics, agents often need to balance speed and energy efficiency. In reinforcement learning with human feedback, reasoning models, and post-training, models are often evaluated under competing criteria such as helpfulness, safety, and honesty [20, 37].

A core challenge in multi-objective RL (MORL) is that the optimal policy can vary drastically with objective weight combinations. For example, even small changes in preference can lead to qualitatively different optimal policies [34, 55]. Prior work has focused on approximating the Pareto frontier of the optimal reward trade-offs [42, 52, 55], or learning one policy for each objective [44, 53], and it remains unclear *whether or when* transferring trajectories across objectives is effective.

A key yet underexplored question is that the difficulty of learning across preferences is fundamentally tied to the degree of misalignment among objectives. When objective weightings are well aligned (e.g., rewarding both final and intermediate progress), additional objectives can facilitate learning,

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

and data collected under one preference can benefit policy learning for others. When objectives are misaligned or adversarially-injected (e.g., RLHF with corrupted reward components), naive transfer can become inefficient or even harmful [20, 27]. This problem is especially acute in high-dimensional settings, making it a critical open challenge to identify objective alignment [21].

To address this challenge, we introduce a new theoretical framework based on measuring *objective conflict coefficients* that quantify the degree of misalignment among objectives through the χ^2-divergence. Our analysis shows that this coefficient interacts with the *coverability coefficient* [46]—which captures how well a single exploration policy can cover the preference space—to determine the sample complexity of online MORL. Our regret bound generalizes single-objective results [1, 19, 46, 7, 9] and is the first to formalize how objective misalignment affects learning efficiency across preferences. Our theoretical framework builds on recent advances in transfer learning under distribution shifts [50]. However, the technical challenge in MORL lies in quantifying the discrepancy in policy learning. Our main result, Theorem 3.6, relies on a low-rank basis assumption, along with a coverability condition that ensures sufficient exploration of the basis.

Motivated by our theoretical framework, we propose *routed ensemble policy learning*, an online algorithm for learning a small set of basis policies, which can be applied to a real-world RL environment. This algorithm alternates between two steps: (i) routing each trajectory to the basis policy with the highest expected return under the corresponding preference weight, and (ii) partitioning trajectories into disjoint clusters and updating each basis policy using its assigned trajectories with a standard policy learning algorithm such as proximal policy optimization.

Next, we consider an alternative perspective for tackling objective conflicts, based on selecting trajectories. This motivates the problem of quantifying the influence of individual trajectories upon the learned optimal policy, a problem for which we term *trajectory attribution*. We design a scalable *kernel-based trajectory attribution* procedure that estimates how trajectories help or hinder policy learning. Our procedure fits a kernel ridge regression surrogate to approximate policy performance across trajectory subsets. This can capture both individual contributions and higher-order interactions between trajectories. These interactions are common in high-dimensional settings and often underlie unexpected transfer behavior [21]. However, they are not captured by first-order influence approximations such as influence functions [36], linear surrogate models [26, 56], or gradient-similarity-based analysis [45, 17]. Our approach goes beyond prior work by leveraging efficient approximations grounded in policy gradients and random projections [55]. The results from more accurate trajectory attribution can guide trajectory selection in downstream policy learning.

Lastly, we empirically validate our approach on the Meta-World robotics benchmark and high-dimensional MO-Gymnasium tasks. Our algorithm consistently outperforms multi-objective baselines, achieving a significant reduction in cumulative regret in environments with structured objective conflicts. Specifically, routed ensemble learning outperforms the strongest baseline by 4.3% on the Meta-World benchmark and achieves an 8% average improvement in hypervolume on MO-Gymnasium tasks. Similar results are also observed in reinforcement learning with human feedback (RLHF) settings. Additionally, kernel-based attribution provides a 50% improvement in terms of the Spearman correlations with ground-truth leave-one-out influence scores. When applied to downstream task selection, kernel-based selection yields a 27% higher reward than existing attribution baselines. We present detailed ablation studies to validate the theoretical insights from our proposed framework. For instance, after the routed ensemble procedure, the coverability metric drops from 1.3 for a single policy to 0.2 for the ensemble policies, indicating that routing mitigates the exploration bottleneck caused by objective conflicts. We also conduct a detailed analysis on the number of basis policies in the ensemble, as well as validate a low-rank basis assumption in our theoretical setup.

In summary, the contributions of this paper are threefold.

- First, we introduce a new theoretical framework for MORL based on a χ^2 measure that we call the objective conflict coefficient. The theoretical result can be relaxed to a setting with an approximate low-rank basis, and we empirically evaluate this assumption in various real-world environments.
- Second, we design two algorithms for tackling objective conflicts: (1) routed ensemble policy learning, and (2) kernel-based trajectory attribution to estimate inter-trajectory influence.
- Third, we demonstrate significant gains in sample efficiency and robustness across diverse MORL benchmarks covering language modeling, robotic control, and RLHF settings.

For the rest of this paper, we begin by stating the formal problem setup in Section 2. Afterward, we present our theoretical framework and proposed approach to tackle objective conflicts in Section 3. We will present our experiments to validate our proposed algorithms in Section 4. Lastly, we discuss the related work in Section A and conclude in Section 5. We provide the omitted algorithmic description and proofs in Appendix B and C. Omitted experimental setup and empirical evaluations are provided in Appendix D.

2 Preliminaries

We consider an RL environment modeled as a *multi-objective Markov decision process* (MOMDP), defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \Pr(\cdot \mid \cdot, \cdot), \mathbf{r}, \Omega, f_\Omega, H)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $\Pr(\cdot \mid s, a)$ is the transition kernel, $\mathbf{r} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^m$ is a vector-valued reward function with $m \geq 2$ objectives, and H is the horizon. The preference space is denoted as $\Omega \subseteq \mathbb{R}^m_+$. The scalarization operator f_w maps the return vector to a scalar utility under preference $w \in \Omega$. We evaluate performance under a target preference distribution $w \sim \mathcal{P}$.

A stochastic policy $\pi \in \Pi$ maps states to action distributions, and induces a trajectory $\tau = (s_0, a_0, \mathbf{r}_0, \ldots, s_H, a_H, \mathbf{r}_H)$, where $a_h \sim \pi(\cdot \mid s_h)$ and $s_{h+1} \sim \Pr[\cdot \mid s_h, a_t]$. More generally, the policy can be time dependent $\pi_1, \cdots, \pi_H$ with $a_h \sim \pi_h(\cdot \mid s_h)$ and induces a trajectory similarly.

The trajectory return is the cumulative reward vector $\mathbf{G}(\tau) = \sum_{h=0}^{H} \mathbf{r}(s_h, a_h) \in \mathbb{R}^m$. The expected return of policy π from an initial distribution $s_0 \sim \rho$ is given by

$$\mathbf{J}(\pi) = \mathop{\mathbb{E}}_{s_0 \sim \rho, \tau \sim \pi} [\mathbf{G}(\tau)] \in \mathbb{R}^m.$$

We primarily focus on linear operators, and thus the policy's expected return can be written as $\mathcal{J}(\pi; w) = \langle w, \mathbf{J}(\pi) \rangle$. We consider an episodic online MORL setting with episodes $t = 1, \ldots, T$. At the beginning of episode t, the system designer has access to the history $\mathcal{D}^{(t-1)} = \{(\tau^{(u)}, w_u)\}_{u=1}^{t-1}$ and observes the current preference vector $w_t \in \Omega$. Based on $(\mathcal{D}^{(t-1)}, w_t)$, a learner applies a possibly randomized policy learning algorithm to select a policy $\pi^{(t)}$, interacts with the MOMDP for H horizons, and observes a trajectory $\tau^{(t)} = (s_0^{(t)}, a_0^{(t)}, \mathbf{r}_0^{(t)}, \ldots, s_H^{(t)}, a_H^{(t)}, \mathbf{r}_H^{(t)})$.

After T episodes, the learner collects a set of trajectories $\mathcal{D} = \{(\tau^{(t)}, w_t)\}_{t=1}^{T}$ and aims to learn a preference-conditioned policy $\pi(\cdot; w)$ that attains high utility for target preferences w in the support of $\mathcal{P}$. In this paper, we focus on learning an ensemble policy corresponding to a small set of $\pi(\cdot; w)$.

Our theoretical framework builds on the notion of *coverability*, which is a complexity measure defined on a fixed policy class Π [1, 46]:

$$C_{\mathrm{cov}} := \inf_{\mu \in \Delta(\mathcal{S} \times \mathcal{A})} \sup_{\pi \in \Pi} \max_{h \in [H]} \sup_{(s,a)} \frac{d_h^\pi(s, a)}{\mu_h(s, a)},$$

where $d_h^\pi(s, a) = \Pr^\pi(s_h = s, a_h = a)$ denotes the state-action probability under policy π, and the infimum is over sequences of $\mu_h \in \Delta(\mathcal{S} \times \mathcal{A})$. Intuitively, C_{cov} captures how hard it is for a single exploration distribution to simultaneously cover the occupancy measures of all policies in Π, serving as a hardness measure for online exploration.

3 Our Approach

We first provide several examples to illustrate how multi-objective misalignment affects policy learning. We then formalize this observation through the notion of objective conflict coefficient and show how it interacts with coverability to derive online regret bounds. The regret bound is shown to be tight with a matching lower bound. These theoretical insights motivate our algorithm, routed ensemble policy learning, which learns a small set of basis policies and routes trajectories to the policies that give the highest reward. Lastly, to interpret objective conflict and routing behavior, we develop a kernel-surrogate trajectory attribution algorithm that quantifies how individual trajectories influence policy learning across preference weights.

3.1 Examples and motivation

We start with two common MOMDP use cases to motivate our theoretical framework.

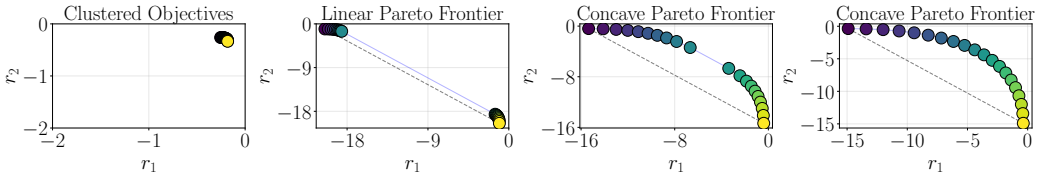


Figure 1: We visualize the objective space returns (r_1, r_2) for policies optimized under varying preference weights $w \in [0, 1]$ in three canonical regimes — clustered, linear and concave Pareto frontier. Light yellow indicates that w is closer to 1, while dark blue indicates w is closer to 0.

Example 3.1 (Learning the Pareto frontier of a MOMDP)**.** *The Pareto frontier consists of policies for which no objective can be improved without worsening at least one other objective. Formally, a policy π dominates π' (denoted $\pi \succ \pi'$) if $J_i(\pi) \geq J_i(\pi')$ for all objectives $i \in [m]$, with strict inequality for at least one i, where $J_i(\pi)$ is the i-th entry of $\mathbf{J}(\pi)$.*

Example 3.2 (Online exploration under corrupted objectives)**.** *Consider an MOMDP with $m = 2$, where r_1 is the clean task reward and r_2 may be corrupted (possibly adversarial). The target preference is $w = [1, 0]^\top$, so the goal is to optimize r_1. However, if corruption is present, trajectories collected online may be influenced by the spurious signal in r_2. This motivates the need to identify corrupted trajectories and to attribute their alignment with w.*

We illustrate that learning in an MOMDP depends on how objectives relate to one another. In the two-objective case ($m = 2$), there are three canonical regimes, as shown in Figure 1: (i) *aligned objectives*, (ii) *conflicting objectives with an (approximately) linear Pareto frontier*, and (iii) *conflicting objectives with a strictly concave frontier*. Aligned objectives admit a single policy that is optimal for all preferences. Linear conflict is captured by two corner policies whose mixtures span the frontier. Concave conflict requires additional intermediate policies beyond the corner solutions. When m grows, different objectives can simultaneously fall into different regimes—some aligned, linearly conflicting, and others exhibiting concave trade-offs. These induce discontinuous shifts and make policy optimization harder.

3.2 Theoretical characterization through conflict coefficients

With the above examples in mind, next, we quantify how objective conflict impacts sample complexity in MORL. We assume that there exists a set of K latent basis policies that (approximately) represent optimal policies across preferences for $w \in \Omega$. The number of basis policies K captures the intrinsic complexity of the objective trade-offs. We capture the extent of conflict via the *objective conflict coefficient* Γ, which measures the diversity of basis occupancies relative to their mixture.

Definition 3.3. *Let the basis-mixture occupancy be the uniform average of the K basis policies, $\overline{d}_h(s, a) := \frac{1}{K} \sum_{k=1}^{K} d_h^{\pi_{\mathrm{bk}}^*}(s, a)$. The objective conflict coefficient is*

$$\Gamma := \max_{h \in [H]} \max_{k \in [K]} \chi^2 \left(d_h^{\pi_{\mathrm{bk}}^*} \| \overline{d}_h \right). \tag{1}$$

Smaller Γ indicates substantial overlap among basis policies and easier transfer across preferences, while large values correspond to near-disjoint occupancies and greater learning difficulty. Moreover, Γ is always bounded by $K - 1$ (Lemma C.5), so its magnitude scales with the number of basis policies. For instance, $\Gamma = 0$ in Case 1, and Γ is larger in Case 3 than in Case 2.

We now analyze online MORL through cumulative regret:

$$\mathsf{Reg}(T) = \sum_{t=1}^{T} \left(\mathcal{J}(\pi_{w_t}^\star; w_t) - \mathcal{J}(\pi^{(t)}; w_t) \right).$$

We impose two structural assumptions. The first states that optimal policies lie in the convex hull of the basis occupancies.

Assumption 3.4. *There exists a finite set of latent basis policies $\{\pi_1^\star, \ldots, \pi_K^\star\}$ such that for any $w \in \Omega$, there exists a mixing vector $\nu_w \in \Delta^{K-1}$ (the probability simplex over K elements) satisfying $d_h^{\pi_w^*}(s, a) = \sum_{k=1}^{K} \nu_w(k)\, d_h^{\pi_{\mathrm{bk}}^*}(s, a)$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$.*

This assumption is imposed to yield a clean characterization of the role of K in regret. We relax this condition in Appendix C.4 by allowing optimal occupancies to lie within an ϵ-neighborhood of the convex hull, under which the same qualitative conclusions continue to hold. The second assumption ensures sufficient exploration of all basis components.

Assumption 3.5. *For each $t \geq 1$, there exists $\nu^{(t)} \in [0,1]^K$ such that $d_h^{\pi^{(t)}}(s,a) = \sum_{k=1}^K \nu^{(t)}(k)\, d_h^{\pi^\star_{\mathrm{bk}}}(s,a)$ for all (s,a) and H. Letting $\overline{\nu}^{(t)}(k) = \frac{1}{t-1}\sum_{i=1}^{t-1} \nu^{(i)}(k)$, we assume there exist $c_0 \in (0,1]$ and a burn-in time t_0 such that $\min_{k\in[K]} \overline{\nu}^{(t)}(k) \geq c_0/(C_{\mathrm{cov}}K)$ for all $t \geq t_0$.*

This condition holds, for example, when preferences are sampled with full support over Ω, ensuring persistent coverage of all basis policies.

We now derive a regret bound that explicitly quantifies the impact of K and Γ. We study a K-policy variant of the optimistic confidence-set algorithm [19]. The learner maintains K confidence sets $\mathcal{F}_k^{(t-1)}(\beta)$, one for each basis index $k \in [K]$, and at episode t executes the optimistic policy $\pi^{(t)} = \pi_{f^{(t)}}$ associated with some $f^{(t)} \in \mathcal{F}_{k_t}^{(t-1)}(\beta)$ for the routed basis k_t. A detailed description is provided in Appendix C.1. Each $\mathcal{F}_k$ is given by $\mathcal{F}_{k,1} \times \cdots \times \mathcal{F}_{k,H}$, with $\mathcal{F}_{k,h} \subseteq (\mathcal{S}\times\mathcal{A}\times\Omega \to [0,H])$. We assume that $\mathcal{F}_k$ is finite and write $|\mathcal{F}| = \max_{k\in[K]} |\mathcal{F}_k|$. For infinite classes, one can replace $\log|\mathcal{F}|$ via covering numbers [19]. Our main result is stated as follows.

Theorem 3.6. *Suppose scalarized rewards are bounded in $[0,1]$ per step, Bellman realizability and completeness hold for each per-basis class $\mathcal{F}_k$, and Assumptions 3.4–3.5 hold. Let $\beta = c_1 \log(K|\mathcal{F}|T/\delta)$ be the confidence-set radius after a union bound across the K basis classes. After $t_0 = \mathcal{O}(C_{\mathrm{cov}}K\beta)$, then with probability at least $1-\delta$, there exists an algorithm with K policies such that the cumulative regret satisfies*

$$\mathsf{Reg}(T) \leq \mathcal{O}_T\left(H\sqrt{C_{\mathrm{cov}}(1+\Gamma)\log(K\,|\mathcal{F}|\,T\delta^{-1})T}\right). \tag{2}$$

Theorem 3.6 decomposes learning difficulty into (i) exploration hardness, governed by C_{cov}, and (ii) transfer hardness induced by objective conflict, captured by Γ. When objectives are well aligned, both Γ and K are small, and the regret approaches the single-objective rate; when basis occupancies are nearly disjoint, Γ increases, reflecting the cost of transferring information across preferences. The full proof of Theorem 3.6 is given in Appendix C.1.

Nearly-tight regret lower bound. We complement this upper bound with *a matching lower bound* in Theorem C.6 in Appendix C.2, in the order of $\Omega_T\big(H\sqrt{(1+\Gamma)T}\big)$, establishing the tightness of the dependence on Γ and T up to C_{cov} and logarithmic factors.

Remark 3.7. *Although Theorem 3.6 is developed for a stylized online algorithm, similar proof techniques can be extended to more general settings. Proposition C.10 in Appendix C.4 shows that, when the low-rank basis Assumption 3.4 is relaxed to approximate low-rankness, the resulting regret bound preserves the same qualitative dependence on K and Γ.*

Furthermore, empirical analysis in Appendix D.3 demonstrates that these insights extend beyond the theoretical setting. To assess the approximation empirically, we measure the convex-hull residual over a preference grid. In a tabular concave setting, the mean residual decreases from 0.186 at $K = 3$ to 1.45×10^{-4} at $K = 21$; on MO-Hopper, it decreases from 0.443 at $K = 3$ to 0.231 at $K = 15$ using candidate-selected bases. The analysis below uses the exact assumption, while Appendix C.4 treats nonzero residuals. We also find that increasing the size of N gradually reduces the approximation error imposed by the low-rank condition (See Table 8 for the evaluation).

3.3 Learning routed ensemble policies

In Sections 3.1 and 3.2, the basis policies are assumed latent. Next, we design a scalable *routed ensemble policy learning* algorithm that:

(i) explicitly learn a set of N basis policies, with each trajectory routed to the highest-reward basis, and

(ii) learn a preference-conditioned transfer operator $f_w : \Pi^N \to \Pi$ that maps these basis policies to a composite policy for any given preference w.

We also clarify that the ensemble size N may or may not be the same as K as stated in Assumption 3.4. Thus, the algorithms studied will apply to a real-world MORL environment.

Our proposed Algorithm 1 alternates between scoring, routing, and updating: (i) *Scoring*: For each trajectory and each basis policy, compute a score using its estimated scalarized return (cf. Equation (4) below). (ii) *Routing*: Assign each trajectory to the basis policy that achieves the highest score for the corresponding preference, inducing a partition of the data into N clusters. (iii) *Updating*: Update each basis policy using only the trajectories assigned to its cluster via a standard RL subroutine (e.g., upper-confidence bound value iteration for MOMDPs [44]). (iv) *Iteration*: Repeat the above steps across online batches until both the cluster assignments and the learned policies stabilize. We formalize the learning objective as:

$$\max_{\pi_1,\pi_2,\ldots,\pi_N} \mathop{\mathbb{E}}_{s_0\sim\rho, w\sim\mathcal{P}} \left[\left\langle w, \sum_{h=0}^{H} \mathbf{r}(s_h, f_w) \right\rangle \right], \tag{3}$$

which maximizes the expected scalarized return $\mathcal{J}(\pi; w)$ of the composed policy $f_w(\pi_1, \ldots, \pi_N)$ under the target preference distribution $\mathcal{P}$.

Importantly, the routing step is *optimal* under this formulation: when f_w lies in the convex hull of a fixed set of basis policies, the optimal solution reduces to deterministically selecting the basis policy that maximizes the return under w. This follows from the linearity of $\mathcal{J}(\pi; w)$ in w, and is formalized in the following lemma.

Lemma 3.8. *Fix a set of N policies $\{\pi_1, \ldots, \pi_N\}$. Suppose $f_w \in \mathrm{conv}\{\pi_1, \ldots, \pi_N\}$ and $d_h^{f_w}(s,a) = \sum_{i=1}^{N} \alpha_i(w) d_h^{\pi_i}(s,a)$ for $\alpha(w) \in \Delta^{N-1}$. There exists an optimal routing function $\widehat{f}$ for Equation (3) such that, for every $w \in \Omega$, $\widehat{f}_w = \pi_j$, where $j \in \arg\max_{i\in\{1,2,\ldots,N\}} \langle w, J(\pi_i)\rangle$.*

Proof. For any w, since f_w lies in the convex hull of $\pi_1, \ldots, \pi_N$, the state-action occupancy probabilities satisfy $d_h^{f_w}(s,a) = \sum_i \alpha_i(w) d_h^{\pi_i}(s,a)$. Thus, the scalarized return satisfies

$$\begin{aligned} J_\alpha(f_w) &= \sum_h \sum_{a_h, s_h} d_h^{f_w}(s_h, a_h) \langle w, r_h(s,a)\rangle \\ &= \sum_h \sum_{a_h, s_h} \sum_{i=1}^{N} \alpha_i(w) d_h^{\pi_i}(s_h, a_h) \langle w, r_h(s,a)\rangle = \sum_{i=1}^{N} \alpha_i(w) \langle J(\pi_i), w\rangle . \end{aligned} \tag{4}$$

The maximum over $\alpha \in \Delta^{N-1}$ is therefore attained at an extreme point, corresponding to selecting a single basis policy. Thus, the optimal strategy is to choose the policy π_i that maximizes $\langle J(\pi_i), w\rangle$.

Lemma 3.8 motivates the scoring and routing steps: we compute scores using equation (4) and route each trajectory to the basis policy that maximizes the score. Note that in the updating step, each basis policy is trained only on its assigned trajectories. This is crucial because a policy that performs well for one preference w may perform poorly for a different w'. Consequently, mixing trajectories across preferences can degrade learning [44]. Moreover, Proposition C.8 in Appendix C.3 connects this algorithm to Γ, showing that the intuition underlying Theorem 3.6 carries over to its performance.

Finally, while our algorithm assumes N is given, in practice it can be selected via cross-validation (Section D.2). One may adaptively grow N by identifying new clusters when existing policies do not explain incoming trajectories, drawing on the clustering literature [8, 28]. Compared to previous work [44] that learns one optimal policy for each objective, the routed ensemble algorithm can leverage potential overlap that is latent in the underlying RL environment.

3.4 Trajectory attribution

Another approach to address the problem of objective conflict is through trajectory selection, for instance, removing trajectories that are far from the learned policy [55]. This motivates the question of quantifying how individual training trajectories contribute to policy performance, a problem that we term *trajectory attribution*. This complements our theoretical analysis above by providing a data-driven measure of which trajectories help or hinder learning and by improving the interpretability of routed ensemble policies.

Algorithm 1 Learning Routed Ensemble Policies for Multi-Objective Markov Decision Processes

Input: A MOMDP $\mathcal{M}$; Initial state distribution ρ; Preference distribution $\mathcal{P}$; Number of policies N
Require: A policy learning algorithm F; A policy evaluation function J; Number of iterations T; Horizon H; Number of observed trajectories B per iteration
Output: A set of N policies

$\pi_1^{(0)}, \pi_2^{(0)}, \ldots, \pi_N^{(0)} : \mathcal{S} \to \Delta(\mathcal{A})$, start from N random initializations
for $t = 1, \ldots, T$ **do**
  $w_t \sim \mathcal{P}$, $\mathcal{D}_t \leftarrow \emptyset$, $S \leftarrow N$ by B score matrix
  for $i = 1, \ldots, B$ **do**
    $j_i \leftarrow$ sample uniformly from $\{1, 2, \ldots, N\}$
    $s_0 \leftarrow \rho$: sample from the initial state distribution
    $\tau_i^{(t)} \leftarrow (s_0, a_0, \ldots, s_H, a_H)$: simulate from s_0 for H steps using $\pi_{j_i}^{(t-1)}$
    for $j = 1, \ldots, N$ **do**
      $S_{j,i} \leftarrow \sum_{h=0}^{H} \left\langle w_t, \sum_{a \in \mathcal{A}} \mathbf{r}(s_h, a) \pi_j^{(t-1)}(s_h, a) \right\rangle$ ▷ Score policy j on trajectory i
    end for
    $\mathcal{D}_t \leftarrow \mathcal{D}_t \cup \{(\tau_i^{(t)}, w_t)\}$
  end for
  for $j = 1, \ldots, N$ **do**
    $\mathcal{D}_{t,j} \leftarrow \left\{ (\tau_i^{(t)}, w_t) \in \mathcal{D}_t \mid j \in \arg\max_{1 \le j' \le N} S_{j',i} \right\}$ ▷ Route trajectories to policy j
    $\pi_j^{(t)} \leftarrow F\big(\pi_j^{(t-1)}, \mathcal{D}_{t,j}\big)$ ▷ Update policy j using algorithm F
  end for
end for
return $\pi_1^{(T)}, \pi_2^{(T)}, \ldots, \pi_N^{(T)}$

A natural starting point for tackling trajectory attribution is via leave-one-out (LOO) retraining, which measures the change in performance when trajectory t is excluded:

$$\mathcal{I}_t^{\text{LOO}} = \mathcal{V}\big([\mathbf{1}]_T\big) - \mathcal{V}\big([\mathbf{1}]_T - e_t\big),$$

where, for any subset indicator $S \in \{0,1\}^T$, we define $\mathcal{V}(S) := \mathbb{E}_{w \sim \mathcal{P}}\left[\mathcal{J}\big(\pi_S; w\big)\right], \pi_S := F(\mathcal{D}_S)$, with $\mathcal{D}_S := \{(\tau^{(j)}, w_j) \in \mathcal{D} : S_j = 1\}$. Here π_S is the policy trained on the subset selected by S using algorithm F, $[\mathbf{1}]_T$ is the all-ones vector, and e_t is the t-th standard basis vector.

Thus, $\mathcal{I}_t^{\text{LOO}}$ reflects the marginal importance of trajectory t. However, exact LOO retraining becomes computationally infeasible as the number of trajectories grows [26]. Moreover, it only captures one-at-a-time effects and misses nonlinear interactions among trajectories.

To address these limitations, Algorithm 2 introduces a scalable *kernel-surrogate* approach [26, 56] that approximates the subset-performance function $\mathcal{V}(\cdot)$. This procedure proceeds in three steps. First, we randomly sample n trajectory subsets $S_1, \ldots, S_n$, where each $S_i \in \{0,1\}^T$ indicates which trajectories are included. Second, for each subset S_i, we estimate its performance using a surrogate loss $v(S_i) \approx \mathcal{V}(S_i)$. Third, we fit a kernel ridge regression surrogate: we form the radial basis function kernel $\Psi_{ij} = \exp(-\gamma \|S_i - S_j\|^2)$, set $V = [\widehat{v}(S_1), \ldots, \widehat{v}(S_n)]^\top$, and compute $\alpha = (\Psi + \lambda \,\mathrm{Id}_n)^{-1} V$. This yields the predictor

$$g_\alpha(S) = \sum_{i=1}^{n} \alpha_i \exp(-\gamma \|S_i - S\|^2), \text{ for } \mathcal{V}(S).$$

Because this kernel model $g_\alpha(S)$ is nonlinear in the subset indicator, it can capture higher-order interactions such as synergy and antagonism among trajectories. Provided with $g_\alpha(S)$, we could then use this to score each trajectory; for instance, by averaging over all the subsets that include a particular trajectory. Then, trajectories with the highest scores are selected for downstream applications.

The procedure above is formulated for trajectory attribution. However, the same procedure can be applied to objective attribution by sampling subsets of objectives. Each subset retains only the training contributions associated with its selected objectives. Removing an objective therefore excludes its contribution across all trajectories and allows us to estimate its effect on policy performance.

Algorithm 2 Kernel Surrogates for Trajectory Attribution Using Policy Gradient Estimation

Input: A multi-objective RL environment with m reward objectives
Requires: Subset sampling distribution $\mathcal{D}$; Number of subsets n; ℓ_2-regularization parameter λ; A policy learning algorithm F
Output: A kernel surrogate model $g_\theta : \{0,1\}^m \to \mathbb{R}$

1: $\theta^\star \leftarrow$ Learn a meta policy with all the m reward objectives simultaneously using F
2: $S_1, \ldots, S_n \leftarrow$ Sample n subsets from $\mathcal{D}$ independently
3: $\widehat{v}(S_1), \widehat{v}(S_2), \ldots, \widehat{v}(S_n) \leftarrow$ Apply PolicyGradEx (Algorithm 3) with $(S_1, S_2, \ldots, S_n; \theta^\star)$
4: $\Psi_{ij} \leftarrow \exp(-\gamma \left\|S_i - S_j\right\|^2)$, for all $1 \le i, j \le n$ ▷ Construct kernel matrix
5: $V \leftarrow [\widehat{v}(S_1), \ldots, \widehat{v}(S_n)]^\top$ ▷ Collect estimated outcomes
6: $\alpha \leftarrow (\Psi + \lambda \operatorname{Id}_n)^{-1} V$ ▷ Fit kernel ridge regression
7: **return** $g_\theta(S) = \sum_{i=1}^n \alpha_i \exp(-\gamma \|S_i - S\|^2)$

Implementation. In online RL, exact retraining for each sampled subset is often impractical because the training data evolve. We therefore estimate the subset performance $v(S_i)$ locally around a reference policy trained on all trajectories. Specifically, we use policy gradients from local rollouts to approximate the parameter update induced by each subset, avoiding the cost of full retraining. The algorithm and the corresponding derivation are presented in Appendix B.

The experimental results support this approximation. In Figure 3, Appendix D.3, we show that the fitted kernel model $g_\alpha([\mathbf{1}]_T - e_t)$ is strongly positively correlated with the LOO scores $\mathcal{I}_t^{\mathrm{LOO}}$ and strongly negatively correlated with the objective conflict coefficient Γ. Moreover, Algorithm 2 achieves a Spearman rank correlation of 0.75 with the true policy-gradient influence scores, confirming that it identifies influential trajectories. More details of the approximation procedure are presented in Appendix B.

4 Experiments

Our experiments are designed to address the following two aspects: (1) to verify the theoretical insights from the regret analysis, and (2) to evaluate the empirical performance of our algorithms across multi-objective RL settings, including synthetic environments, continuous robotic control, and RLHF settings. We highlight three findings. First, routing mitigates negative transfer and reduces cumulative regret in synthetic environments. Second, routed ensembles outperform baselines by a 4.3% relative margin in multi-task RL, up to 48% in multi-objective RL, and 17% in large language model benchmarks. Third, our kernel surrogates improve correlation with ground-truth marginal gains by 50% for trajectory attribution and improve downstream task-selection reward by 27%. Our code repository can be found at https://github.com/VirtuosoResearch/Multi-objective-reinforcement-learning-code/.

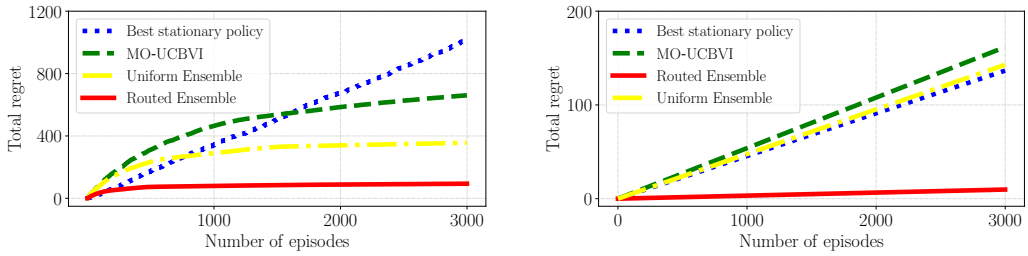
4.1 Evaluation in synthetic environments

We validate our theoretical insights in the following two environments (details can be found in Appendix D.1).

1. Learning Pareto frontier (Example 3.1). The environment has $m = 4$ conflicting objectives. The reward functions are designed such that maximizing one objective necessitates sacrificing the others, creating a complex, high-dimensional Pareto frontier. We sample preference vectors w uniformly from the simplex to evaluate the approximation of the frontier.

2. Learning under corrupted examples (Example 3.2). The environment has $m = 2$ objectives, where r_1 represents a clean task signal and r_2 represents a corrupted signal. The target preference is fixed at $[1, 0]$, meaning the agent solves a single-objective problem but is exposed to conflicting gradients from r_2 during exploration. The training preference is sampled uniformly in the simplex.

3. Routing under varying goal separation. We keep the environment dynamics fixed and vary the separation between goals. For each setting, we compute Γ from the resulting occupancies and measure the reduction in cumulative regret achieved by routing relative to the comparison method.



(a) Learning Pareto frontier

(b) Learning under corrupted examples

Figure 2: We compare the cumulative regret of routed ensembles (red) against MO-UCBVI (green dashed), uniform ensembles (yellow), and the best stationary policy (blue dotted).

For Pareto frontier learning, Figure 2a shows that routed ensemble learning substantially reduces cumulative regret relative to MO-UCBVI [44], using one policy per objective. This suggests that routed ensembles maintain diverse basis policies that cover different regions of the frontier, while the single-policy baseline fails to capture the complex four-way trade-off.

For learning under corrupted examples, Figure 2b shows that our approach mitigates negative transfer from corrupted inputs and achieves near-optimal performance. This suggests that routing can filter out harmful interference and assign trajectories to support optimization of the clean objective r_1.

4.2 Evaluation on robotic control and language models

Next, we present our evaluations applied to several robotic control and language modeling settings. We first describe the setups.

Environments and models. We evaluate on four regimes, each with a different source of conflict: (i) *shared dynamics* on the 10-task MT10 benchmark from Meta-World [54]; (ii) *explicit Pareto trade-offs* on the continuous MO-HalfCheetah/Hopper/Ant and discrete Deep-Sea-Treasure/Fruit-Tree environments of MO-Gymnasium [12]; (iii) *preference-induced conflict* on HelpSteer2 [43], with Qwen3-1.7B fine-tuned with PPO via per-basis LoRA adapters; and (iv) *trajectory attribution* on Multitask Minigrid, MT10, and a synthetic synergy MTRL environment (See Appendix D.1). In all regimes, routing is applied as a drop-in module on top of the published base learner.

Baselines. We compare to the following baselines: (i) For multi-task RL, Soft Modularization [51], PaCo [39], CARE [38]; (ii) For continuous multi-objective RL, PGMORL [49], Q-Pensieve [18], CAPQL [31], GPI-LS [3], and learned-gate MoE; for discrete multi-objective RL, PQL [42], MP-MOQL [42], and Envelop [52]; (iii) For RLHF, a single-policy model; (iv) For trajectory attribution, Influence Functions [23], TracIn [35] adapted to online RL setting [17], and linear surrogate models [26]. The MoE and GPI-LS comparisons hold the base policy-learning algorithm and interaction budget fixed to examine different routing and policy-selection mechanisms. For the first three regimes, we also include *uniform assignment*, which randomly assigns trajectories to basis policies.

Evaluation. We use the following evaluation metrics: (i) success rate on MT10; (ii) hypervolume (HV) and utility (UT) on continuous MO-Gymnasium, and HV and cardinality (Card.) on discrete MO-Gymnasium; (iii) preference-induced conflict on HelpSteer2 [43] and UltraFeedback [10], using Qwen3-1.7B and Llama-3.2-1B fine-tuned with PPO via per-basis LoRA adapters; (iv) for trajectory attribution, linear data modeling score (LDS) and selection performance. LDS is the Spearman rank correlation between predicted and ground-truth marginal gains. Selection performance is the target objective return on a target objective when training within the top-3 most influential source tasks selected by each procedure.

Results for policy optimization. Our proposed routed ensemble Algorithm 1 consistently outperforms baselines in multi-task and multi-objective settings. The comparison is presented in Table 1. On MT10, it achieves an average success rate of 85.6%, outperforming soft modularization (82.0%) by a relative margin of 4.3%.

Our routing algorithm also improves RLHF performance on both the synthetic environment and real-data benchmarks. In Table 9, we compare Algorithm 1, uniform assignment, and learning a single

Table 1: Comparison of our approach with several baseline methods. We report the average success rate on Meta-World, HV and UT on MuJoCo, as well as HV and Cardinality on Grid-World.

	Metrics	Soft Modularization	PaCo	CARE	Uniform Assignment	**Algorithm 1**
Meta-World	Success rate	$82.0_{\pm 1.1}\%$	$73.1_{\pm 1.1}\%$	$80.2_{\pm 1.8}\%$	$81.3_{\pm 1.4}\%$	$\mathbf{85.6}_{\pm 1.8}\%$
	Metrics	PGMORL	Q-Pensieve	CAPQL	Uniform Assignment	**Algorithm 1**
MO-HalfCheetah2d	HV ($\times 10^6$)	$0.11_{\pm 0.08}$	$4.85_{\pm 0.06}$	$3.29_{\pm 0.18}$	$3.17_{\pm 0.11}$	$\mathbf{5.17}_{\pm 0.16}$
	UT ($\times 10^2$)	$0.12_{\pm 0.13}$	$2.14_{\pm 0.13}$	$0.63_{\pm 0.09}$	$1.01_{\pm 0.12}$	$\mathbf{2.47}_{\pm 0.14}$
MO-Hopper	HV ($\times 10^7$)	$1.23_{\pm 0.04}$	$1.61_{\pm 0.32}$	$1.39_{\pm 0.13}$	$1.56_{\pm 0.16}$	$\mathbf{1.83}_{\pm 0.14}$
	UT ($\times 10^2$)	$0.04_{\pm 0.02}$	$1.47_{\pm 0.14}$	$1.29_{\pm 0.12}$	$1.17_{\pm 0.09}$	$\mathbf{1.56}_{\pm 0.06}$
MO-Ant	HV ($\times 10^7$)	$0.17_{\pm 0.11}$	$1.83_{\pm 0.28}$	$0.84_{\pm 0.05}$	$1.81_{\pm 0.15}$	$\mathbf{2.03}_{\pm 0.16}$
	UT ($\times 10^2$)	$0.21_{\pm 0.06}$	$1.75_{\pm 0.02}$	$1.76_{\pm 0.16}$	$1.68_{\pm 0.08}$	$\mathbf{1.82}_{\pm 0.12}$
	Metrics	PQL	MPMOQL	Envelop	Uniform Assignment	**Algorithm 1**
Deep-Sea-Treasure	HV ($\times 10^3$)	$0.76_{\pm 0.30}$	$0.25_{\pm 0.01}$	$0.79_{\pm 0.00}$	$0.62_{\pm 0.20}$	$\mathbf{1.10}_{\pm 0.00}$
	Card. ($\times 10$)	$0.83_{\pm 0.15}$	$0.17_{\pm 0.06}$	$0.10_{\pm 0.00}$	$0.80_{\pm 0.11}$	$\mathbf{1.00}_{\pm 0.00}$
Fruit-Tree	HV ($\times 10^3$)	$6.69_{\pm 0.95}$	$1.01_{\pm 0.24}$	$3.90_{\pm 0.00}$	$8.29_{\pm 0.38}$	$\mathbf{9.30}_{\pm 0.00}$
	Card. ($\times 10$)	$3.40_{\pm 0.36}$	$0.17_{\pm 0.06}$	$3.00_{\pm 0.00}$	$5.47_{\pm 0.06}$	$\mathbf{6.40}_{\pm 0.00}$

policy for each objective. Our algorithm consistently increases hypervolume on the induced frontier. In the synthetic environment, our method improves hypervolume by 17% over the single-policy baseline and 14% over the uniform ensemble. Averaged across the two models, the corresponding improvements are 5.0% and 8.2% on HelpSteer2, and 6.2% and 3.0% on UltraFeedback.

Next, we present comparisons in continuous MO-Gymnasium environments in Table 4. Our routing algorithm yields an 8% average improvement over Q-Pensieve and achieves higher HV than learned-gate MoE and GPI-LS on all three benchmarks. The capacity-matched MoE comparison supports the benefit of trajectory routing beyond additional expert capacity. These results suggest that routing trajectories across basis policies helps distinguish aligned from conflicting data and approximates the Pareto frontier more efficiently than uniform assignment.

Results for trajectory attribution. By applying Algorithm 2 at the objective level, we can achieve higher LDS scores and downstream task-selection performance. The results are presented in Table 6. The kernel surrogate improves LDS by 50% over attribution baselines across Multitask-Minigrid and MT10. Results on the MO-Gymnastics are shown in in Table 7. Moreover, when used to select source tasks for co-training, kernel-surrogate attribution improves the target reward by 27%, demonstrating that the learned attributions are useful for downstream policy learning.

5 Conclusion

In this paper, we study how to learn policies that balance competing objectives in multi-objective reinforcement learning. We first analyze the theoretical limits of transfer learning across preferences, highlighting that structural misalignment between objectives is the primary cause of negative transfer. To overcome this challenge, we propose a routed ensemble algorithm guided by a new measure, the objective conflict coefficient. Furthermore, we introduce a kernel-based trajectory-attribution method that captures widespread nonlinear interactions among trajectories and can improve subsequent policy learning. The experimental results demonstrate that our approach achieves superior sample efficiency and attribution accuracy compared to state-of-the-art baselines across diverse benchmarks.

One limitation of our theoretical framework is that the regret bounds do not directly apply to the routed ensemble algorithm. Further understanding the sample complexity of learning ensemble policies in multi-objective Markov decision processes is an interesting question for future work.

Acknowledgment

We are grateful to the anonymous reviewers for their constructive comments that greatly improved this paper. Thanks to Dongruo Zhou for pointing out references to prior work. This work received financial assistance from NSF award IIS-2412008 and a startup fund from Northeastern University. Zhenshuo Zhang received additional financial support from Northeastern University's Khoury PhD fellowships. R. Xiong is supported in part by the URC award at Emory University.

References

[1] A. Agarwal, N. Jiang, S. M. Kakade, and W. Sun. "Reinforcement learning: Theory and algorithms". In: *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep* 32 (2019), p. 96.

[2] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan. "On the theory of policy gradient methods: Optimality, approximation, and distribution shift". In: *Journal of Machine Learning Research* 22.98 (2021), pp. 1–76.

[3] L. N. Alegre, A. L. Bazzan, D. M. Roijers, A. Nowé, and B. C. da Silva. "Sample-efficient multi-objective learning via generalized policy improvement prioritization". In: *arXiv preprint arXiv:2301.07784* (2023).

[4] M. Bateni, V. Cohen-Addad, Y. Gu, S. Lattanzi, S. Meierhans, and C. Mohri. "Algorithmic Thinking Theory". In: *arXiv preprint arXiv:2512.04923* (2025).

[5] A. Beygelzimer, S. Kale, and H. Luo. "Optimal and adaptive algorithms for online boosting". In: *International Conference on Machine Learning*. PMLR. 2015, pp. 2323–2331.

[6] E. Chen, N. Dullerud, T. Niedermayr, E. Kidd, R. Senanayake, P. W. Koh, S. Koyejo, and C. Guestrin. "MoSH: Modeling Multi-Objective Tradeoffs with Soft and Hard Bounds". In: *arXiv preprint arXiv:2412.06154* (2024).

[7] F. Chen, Z. Jia, A. Rakhlin, and T. Xie. "Outcome-Based Online Reinforcement Learning: Algorithms and Fundamental Limits". In: *arXiv preprint arXiv:2505.20268* (2025).

[8] J. Chen, E. Sadeqi Azer, and Q. Zhang. "A practical algorithm for distributed clustering and outlier detection". In: *Advances in Neural Information Processing Systems* 31 (2018).

[9] Y. Chi, Y. Chen, and Y. Wei. "Statistical and algorithmic foundations of reinforcement learning". In: *Tutorials in Operations Research: Advances in Analytics and Operations Research: Improving Decisions to Secure the Future*. INFORMS, 2025, pp. 104–144.

[10] G. Cui, L. Yuan, N. Ding, G. Yao, B. He, W. Zhu, Y. Ni, G. Xie, R. Xie, Y. Lin, Z. Liu, and M. Sun. "Ultrafeedback: Boosting language models with scaled AI feedback". In: *arXiv preprint arXiv:2310.01377* (2023).

[11] S. V. Deshmukh, A. Dasgupta, B. Krishnamurthy, N. Jiang, C. Agarwal, G. Theocharous, and J. Subramanian. "Explaining RL Decisions with Trajectories". In: *The Eleventh International Conference on Learning Representations*. 2023.

[12] F. Felten, L. N. Alegre, A. Nowé, A. L. C. Bazzan, E. G. Talbi, G. Danoy, and B. C. d. Silva. "A Toolkit for Reliable Benchmarking and Research in Multi-Objective Reinforcement Learning". In: *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*. 2023.

[13] D. J. Foster and A. Rakhlin. "Foundations of reinforcement learning and interactive decision making". In: *arXiv preprint arXiv:2312.16730* (2023).

[14] Z. Gábor, Z. Kalmár, and C. Szepesvári. "Multi-criteria reinforcement learning." In: *ICML*. Vol. 98. 1998, pp. 197–205.

[15] N. Golowich and A. Moitra. "Linear bellman completeness suffices for efficient online reinforcement learning with few actions". In: *The Thirty Seventh Annual Conference on Learning Theory*. PMLR. 2024, pp. 1939–1981.

[16] N. Golowich and A. Moitra. "The role of inherent bellman error in offline reinforcement learning with linear function approximation". In: *Reinforcement Learning Conference*. 2024.

[17] Y. Hu, F. Wu, H. Ye, D. Forsyth, J. Zou, N. Jiang, J. W. Ma, and H. Zhao. "A Snapshot of Influence: A Local Data Attribution Framework for Online Reinforcement Learning". In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. 2025.

[18] W. Hung, B.-K. Huang, P.-C. Hsieh, and X. Liu. "Q-Pensieve: Boosting Sample Efficiency of Multi-Objective RL Through Memory Sharing of Q-Snapshots". In: *arXiv preprint arXiv:2212.03117* (2022).

[19] C. Jin, Q. Liu, and S. Miryoosefi. "Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms". In: *Advances in neural information processing systems* 34 (2021), pp. 13406–13418.

[20] M. Joglekar, J. Chen, G. Wu, J. Yosinski, J. Wang, B. Barak, and A. Glaese. "Training LLMs for Honesty via Confessions". In: *arXiv preprint arXiv:2512.08093* (2025).

[21] V. Joshi, Z. Xu, B. Liu, P. Stone, and A. Zhang. "Benchmarking Massively Parallelized Multi-Task Reinforcement Learning for Robotics Tasks". In: *Reinforcement Learning Conference*. 2025.

[22] S. M. Kakade. "A natural policy gradient". In: *Advances in neural information processing systems* 14 (2001).

[23] P. W. Koh and P. Liang. "Understanding black-box predictions via influence functions". In: *International conference on machine learning*. PMLR. 2017, pp. 1885–1894.

[24] G. Lan. "Policy mirror descent for reinforcement learning: linear convergence, new sampling complexity, and generalized problem classes: G. Lan". In: *Mathematical programming* 198.1 (2023), pp. 1059–1106.

[25] T. Lattimore and C. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

[26] D. Li, H. Nguyen, and H. R. Zhang. "Identification of Negative Transfers in Multitask Learning Using Surrogate Models". In: *Transactions on Machine Learning Research* (2023).

[27] D. Li, Z. Zhang, L. Wang, and H. R. Zhang. "MoPLEx: Estimating Plackett-Luce Mixture Models for Multi-Objective Alignment". In: *arXiv preprint arXiv:2608.25200* (2026).

[28] Y. Li, Y. Song, and Q. Zhang. "Learning to cluster via same-cluster queries". In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2021, pp. 978–987.

[29] M. L. Littman. "Markov games as a framework for multi-agent reinforcement learning". In: *Machine learning proceedings 1994*. Elsevier, 1994, pp. 157–163.

[30] E. Liu, Y.-C. Wu, X. Huang, C. Gao, R.-J. Wang, K. Xue, and C. Qian. "Pareto set learning for multi-objective reinforcement learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 39. 18. 2025, pp. 18789–18797.

[31] H. Lu, D. Herman, and Y. Yu. "Multi-objective reinforcement learning: Convexity, stationarity and pareto optimality". In: *The Eleventh International Conference on Learning Representations*. 2023.

[32] S. Mannor and N. Shimkin. "A geometric approach to multi-criterion reinforcement learning". In: *Journal of machine learning research* 5.Apr (2004), pp. 325–360.

[33] K. Miettinen and M. M. Mäkelä. "On scalarizing functions in multiobjective optimization". In: *OR spectrum* 24.2 (2002), pp. 193–213.

[34] C. H. Papadimitriou and M. Yannakakis. "On the approximability of trade-offs and optimal access of web sources". In: *Proceedings 41st annual symposium on foundations of computer science*. IEEE. 2000, pp. 86–92.

[35] G. Pruthi, F. Liu, S. Kale, and M. Sundararajan. "Estimating training data influence by tracing gradient descent". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 19920–19930.

[36] I. Rubinstein and S. B. Hopkins. "On the Accuracy of Newton Step and Influence Function Data Attributions". In: *arXiv preprint arXiv:2512.12572* (2025).

[37] S. Sengupta, G. Liu, O. Gottesman, J. W. Durham, M. Kearns, A. Roth, and M. Caldara. "Multi-Objective Reinforcement Learning for Large-Scale Tote Allocation in Human-Robot Collaborative Fulfillment Centers". In: *arXiv preprint arXiv:2602.24182* (2026).

[38] S. Sodhani, A. Zhang, and J. Pineau. "Multi-task reinforcement learning with context-based representations". In: *International Conference on Machine Learning*. 2021, pp. 9767–9779.

[39] L. Sun, H. Zhang, W. Xu, and M. Tomizuka. "Paco: Parameter-compositional multi-task reinforcement learning". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 21495–21507.

[40] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. Second. Adaptive Computation and Machine Learning series. MIT Press, 2018.

[41] M. Towers, A. Kwiatkowski, J. Terry, J. U. Balis, G. De Cola, T. Deleu, M. Goulão, A. Kallinteris, M. Krimmel, A. KG, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, H. Tan, and O. G. Younis. "Gymnasium: A standard interface for reinforcement learning environments". In: *NeurIPS Datasets and Benchmarks* (2025).

[42] K. Van Moffaert and A. Nowé. "Multi-objective reinforcement learning using sets of pareto dominating policies". In: *The Journal of Machine Learning Research* (2014), pp. 3483–3512.

[43] Z. Wang, Y. Dong, O. Delalleau, J. Zeng, G. Shen, D. Egert, J. J. Zhang, M. N. Sreedhar, and O. Kuchaiev. "Helpsteer 2: Open-source dataset for training top-performing reward models". In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 1474–1501.

[44] J. Wu, V. Braverman, and L. Yang. "Accommodating picky customers: Regret bound and exploration complexity for multi-objective reinforcement learning". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 13112–13124.

[45] S. Wu, H. R. Zhang, and C. Ré. "Understanding and improving information transfer in multi-task learning". In: *ICLR* (2020).

[46] T. Xie, D. J. Foster, Y. Bai, N. Jiang, and S. M. Kakade. "The Role of Coverage in Online Reinforcement Learning". In: *The Eleventh International Conference on Learning Representations*. 2023.

[47] R. Xiong. "Automated experimental design with optimization from historical data simulations". In: *Available at SSRN 5126080* (2025).

[48] R. Xiong, A. Chin, and S. J. Taylor. "Data-driven switchback experiments: Theoretical tradeoffs and empirical bayes designs". In: *arXiv preprint arXiv:2406.06768* (2024).

[49] J. Xu, Y. Tian, P. Ma, D. Rus, S. Sueda, and W. Matusik. "Prediction-guided multi-objective reinforcement learning for continuous robot control". In: *International conference on machine learning*. PMLR. 2020, pp. 10607–10616.

[50] F. Yang, H. R. Zhang, S. Wu, C. Ré, and W. J. Su. "Precise high-dimensional asymptotics for quantifying heterogeneous transfers". In: *Journal of Machine Learning Research* 26.113 (2025), pp. 1–88.

[51] R. Yang, H. Xu, Y. Wu, and X. Wang. "Multi-task reinforcement learning with soft modularization". In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 4767–4777.

[52] R. Yang, X. Sun, and K. Narasimhan. "A generalized algorithm for multi-objective reinforcement learning and policy adaptation". In: *NeurIPS* 32 (2019).

[53] T. Yang, S. Cen, Y. Wei, Y. Chen, and Y. Chi. "Federated natural policy gradient and actor critic methods for multi-task reinforcement learning". In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 121304–121375.

[54] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning". In: *Conference on Robot Learning*. 2020, pp. 1094–1100.

[55] Z. Zhang, M. Duan, Y. Ye, and H. R. Zhang. "Scalable multi-objective and meta reinforcement learning via gradient estimation". In: *arXiv preprint arXiv:2511.12779* (2025).

[56] Z. Zhang, M. Duan, and H. Zhang. "Efficient estimation of kernel surrogate models for task attribution". In: *International Conference on Learning Representations*. 2026, pp. 61822–61848.

A Related Work

A reinforcement learning environment is typically formalized by means of a Markov decision process [40, 2]. Existing approaches in the literature on multi-objective reinforcement learning (MORL) generally fall into two categories. The first reduces MORL to single-objective RL through scalarization, optimizing a fixed or sampled preference vector to learn a single policy [14, 33, 32, 38]. The second, which includes our work, examines the multi-objective nature of the problem by learning multiple policies (e.g., to approximate the Pareto frontier) to address sharp variation in optimal policies across preferences [42, 44, 30]. Recent work [55] proposes a convex relaxation algorithm for partitioning multiple objectives into disjoint, related groups. *We build on this line of work by providing a theoretical characterization of objective conflicts in MORL, introducing a routed ensemble algorithm, and developing trajectory-level attribution to interpret model behavior*. One interesting future direction is to examine the policy learning dynamics in multi-objective Markov decision processes [22, 24]. Lastly, sequential decision-making problems also occur in experimental design, where one often needs to roll out an experiment through staggered adoption [48, 47].

Recent work studies both offline and online RL with linear function approximation [15, 16]. It would be interesting to understand the techniques therein and examine whether it would be possible to extend to learning optimal policies under multiple objectives under linear Bellman completeness. More broadly, we refer the reader to a recent monograph by Foster and Rakhlin [13] for references.

Influence estimation techniques [23, 35] have recently been extended from supervised learning to reinforcement learning to trace model behaviors. In the context of offline RL, [11] proposes a framework to attribute specific policy decisions to past trajectories. A fundamental challenge in the online RL setting is the circular dependency where current policy updates influence future data distribution. Recent work [17] addresses this by employing local data attribution via gradient similarity to identify and filter harmful transitions.

B Policy Gradient Estimation

In Algorithm 2, we compute the kernel by minimizing the following objective over the training set:

$$\min_{g_\alpha \in \mathcal{H}} \sum_{i=1}^{n} \big(v(S_i) - g_\alpha(S_i)\big)^2 + \lambda \|g_\alpha\|_\Psi^2, \tag{5}$$

where $\lambda > 0$ is a regularization parameter and Ψ is an $n \times n$ kernel matrix with entries $\Psi_{i,j} = \psi(S_i, S_j)$. Since S is binary, consider the radial basis function (RBF) kernel: $\psi(S_i, S_j) = \exp\big(-\gamma \|S_i - S_j\|^2\big)$. By the representer theorem of kernel methods, the minimizer is:

$$g_\alpha(S) = \sum_{i=1}^{n} \alpha_i \psi\big(S_i, S\big)\,,$$

where the coefficients α are obtained via $(\Psi + \lambda\,\mathrm{Id})^{-1} V$ with $V = [v(S_1), \ldots, v(S_n)]^\top$, and Id is the identity matrix. Below we describe how we obtain $v(S_i)$.

First-order approximation. We approximate the PPO surrogate objective (See Appendix D.4 for a detailed review of PPO) using a first-order Taylor expansion. Let $\varphi_t = \nabla_\theta \log \pi_\theta(a_t|s_t)|_{\theta=\theta^*}$ be the gradient feature vector. The policy objective is approximated as:

$$J(\theta) \approx \mathbb{E}_t \left[\widehat{A}_t \cdot (1 + \varphi_t^\top \Delta\theta) \right],$$

where $\Delta\theta = \theta - \theta^*$ and $\widehat{A}_t$ is the advantage function.

Directly maximizing this linear approximation is unbounded. To obtain a robust estimator, we reformulate this optimization as a weighted binary classification problem. Intuitively, we seek an update $\Delta\theta$ that aligns the policy change $\varphi_t^\top \Delta\theta$ with the sign of the advantage $\widehat{A}_t$. By defining a binary label $y_t = \mathrm{sign}(\widehat{A}_t)$ and a sample weight $\eta_t = |\widehat{A}_t|$, we maximize the margin $y_t \varphi_t^\top \Delta\theta$. This is effectively approximated by minimizing the weighted logistic loss, which provides a bounded

Table 2: We empirically find that the approximation error of ϵ is negligible for several interactive RL environments. We report the mean and standard deviation of the approximation error across 10 random subsets. We defer a detailed description of the setup to the experiments section.

Distance	MT10	CartPole	Highway	Lunarlander
0.1%	$0.01_{\pm 0.01}\%$	$0.12_{\pm 0.14}\%$	$0.02_{\pm 0.02}\%$	$0.06_{\pm 0.01}\%$
0.5%	$0.43_{\pm 0.73}\%$	$0.73_{\pm 0.10}\%$	$0.11_{\pm 0.09}\%$	$0.03_{\pm 0.02}\%$
1.0%	$0.32_{\pm 0.56}\%$	$0.98_{\pm 0.65}\%$	$2.04_{\pm 0.58}\%$	$0.48_{\pm 0.01}\%$

surrogate for the sign-agreement objective:

$$v(S_i) := \min_{\Delta\theta} L_{S_j}(\theta^* + \Delta\theta) \qquad \text{where } L_{S_j}(\theta^* + \Delta\theta) = \frac{1}{|\mathcal{D}_{S_i}|} \sum_{t \in \mathcal{D}_{S_i}} \eta_t \cdot \log\left(1 + \exp\left(-y_t \tilde{\varphi}_t^\top \Delta\theta\right)\right).$$

This formulation enables rapid estimation of the policy performance for any subset S_i without full retraining. As shown in Table 2, the linearization error remains low ($< 2\%$) within the trust region. Another challenge is the high dimensionality of gradient vectors. We address this by applying random Gaussian projections from φ_t to a low-dimensional space. By the Johnson–Lindenstrauss Lemma, this projection provably preserves the pairwise similarity between pairs of gradients. Solving the regression problem in the reduced space then takes only a few seconds.

Specifically, we compute the relative residual sum of squares (RSS) error across several RL environments: MT10 from MetaWorld [54], CartPole, LunarLander, and Highway from Gymnasium [41]. In each case, a four-layer MLP policy is trained, and gradients are estimated from 2,048 steps per task. The results are averaged over 10 randomly sampled subsets of size 5. As shown in Table 2, the RSS error remains below 2% when $\|\theta - \theta^*\|$ is small with a proper initialization $\theta^\star$, indicating high accuracy in the linear approximation. The error gradually increases to around 10% for perturbations up to 5%. These results validate the surrogate model's ability to approximate outcomes near the meta-policy.

Random projection. We now describe the implementation of the random projection step in our attribution procedure. We sample a random projection matrix $P \in \mathbb{R}^{p \times d}$, where d is smaller than p and every entry of P is independently drawn from a Gaussian distribution $\mathcal{N}(0, d^{-1})$. We project the high-dimensional gradients to a d-dimensional space as

$$\tilde{\varphi}_{i,t} = P^\top \varphi_{i,t}.$$

Then, we solve a logistic regression problem in the d-dimensional space with $\theta_d \in \mathbb{R}^d$:

$$\widehat{\theta}_d \leftarrow \arg\min_{\theta_d \in \mathbb{R}^d} \frac{1}{|\mathcal{D}_{S_j}|} \sum_{(\varphi, y, w) \in \mathcal{D}_{S_j}} \ell(\tilde{\varphi}, y, w; \theta_d), \tag{6}$$

for any $j = 1, 2, \ldots, m$. Finally, we map the solution from dimension d back to the original dimension p as

$$\widehat{\theta}^{(j)} = \theta^\star + P\widehat{\theta}_d.$$

Based on the Johnson-Lindenstrauss lemma, when $d = O\left(\frac{\log p}{\epsilon^2}\right)$, for any φ_i with $\|\varphi_i\| \le G$ and any θ, θ_0 with ℓ_2-norm at most D, we have

$$|\langle g_i, \theta - \theta_0\rangle - \langle Pg_i, P(\theta - \theta_0)\rangle| \le \epsilon |\langle \varphi_i, \theta - \theta_0\rangle| \le 2GD\epsilon.$$

This ensures that the random projection step will preserve the accuracy of the gradient estimation procedure. See Algorithm 3 for the complete description.

Correlation between attribution scores and LOO analysis. The goal of this experiment is to evaluate whether an attribution method can predict which trajectory subsets are more useful for policy improvement. For each task, we first collect a fixed pool of trajectories and sample multiple trajectory subsets from that pool. Each subset is used under the same update protocol to estimate its true downstream effect on the policy. The resulting subset performance can be measured by hypervolume.

Algorithm 3 Scalable Kernel Surrogate Estimation using Policy Gradient Estimation (POLICY-GRADEX)

Input: n subsets $S_1, \ldots, S_n$ of $\{1, \ldots, m\}$; Initial meta-policy $\pi_{\theta^\star}$
Require: Number of episodes T; Projection dimension d

$\pi_{\theta^\star} \leftarrow$ Set up the meta-policy for all objectives in $\{1, \ldots, m\}$
$P \leftarrow$ A p by d isotropic Gaussian random projection matrix with mean zero and variance $\frac{1}{d}$
for $t = 1, \ldots, T$ **do**
  for $i \in \{1, \ldots, m\}$ **do**
    $\widetilde{\varphi}_{i,t} \leftarrow P^\top \nabla_\theta \log \pi_\theta(a_{i,t} \mid s_{i,t})|_{\theta=\theta^\star}$
    $\eta_{i,t} \leftarrow |\widehat{A}_{i,t}|$
    $y_{i,t} \leftarrow \operatorname{sign}(\widehat{A}_{i,t})$
  end for
end for
for $j = 1, \ldots, n$ **do**
  $\Delta\widehat{\theta}_d^{(j)} \leftarrow \arg\min_{\Delta\theta_d \in \mathbb{R}^d} \widehat{L}_{S_j}(\theta^\star + P\Delta\theta_d)$
  $\widehat{\theta}^{(j)} \leftarrow \theta^\star + P\Delta\widehat{\theta}_d^{(j)}$
  $\widehat{v}(S_j) \leftarrow -\widehat{L}_{S_j}(\widehat{\theta}^{(j)})$
end for
return $\widehat{v}(S_1), \ldots, \widehat{v}(S_n)$

Given these true subset effects, each attribution method assigns a predicted score to every subset. We then compare the ranking induced by the predicted scores with that induced by the true subset effects. The LDS score is the Spearman rank correlation between these two rankings. A higher LDS means the attribution method better identifies which trajectory subsets are actually more beneficial.

In Figure 3, we illustrate the correlation between the kernel surrogate attribution scores and leave-one-out analysis results. The results show that these two are strongly correlated.

Remark B.1. *We describe several natural extensions of our approach. First, our algorithms naturally extend to online multitask RL [55], where the state-action space varies across tasks. We leave further exploration of these benchmarks to future work.*

Second, kernel surrogates can also be used to measure the influence of an individual trajectory on the overall model behavior. This can be accomplished by sampling subsets of trajectories instead of objectives in the above algorithm.

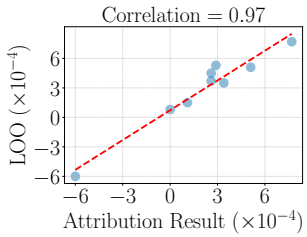
Lastly, while we primarily focus on probabilistic policies, similar ideas can be extended to deterministic policies.

Remark B.2. *Our problem formulation relates to multi-objective optimization in prior literature [34, 6], and we extend this line of work to the online learning setting. Further exploring the connection between ensemble learning and online boosting [5] is another direction for future work.*

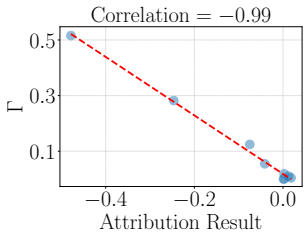
Our work connects to the literature on influence estimation and mechanistic interpretability, which aims to trace model behavior back to influential training data. Prior work include influence functions [23], TracIn [35] and linear surrogate models [26]. In reinforcement learning, recent efforts have adapted these techniques to explain policy decisions, including trajectory-level attribution in offline RL [11] and snapshot influence functions for filtering harmful data in online RL [17]. One of the open questions in this area is to develop better theoretical characterizations of attribution methods. Our kernel-based objective attribution thus offers a principled approach and accurately estimates second-order interactions across preferences. This connection seems particularly relevant in light of recent advances in RL reasoning [4].

C Complete Proofs

In this section, we present the proof of Theorem 3.6 in Appendix C.1. We introduce a lower bound in Appendix C.2 that matches the upper bound. Next, we analyze the magnitude of Γ in Algorithm C.3. Finally, in Appendix C.4, we present a relaxation when only an approximate low-rank basis assumption holds.



(a) Trajectory-level attribution.



(b) Objective-level attribution.

Figure 3: Correlation analysis of attribution scores against LOO scores (left) and objective conflict coefficient Γ (right).

C.1 Proof of Theorem 3.6

This section gives a complete proof of Theorem 3.6. The proof has four key steps. Lemma C.1 reduces cumulative regret to a sum of on-policy Bellman residuals. $\bar{d}_h$ splits each residual into a *conflict factor* (the χ^2 deviation of the deployed occupancy from $\bar{d}_h$) and an *in-sample factor* (the squared Bellman residual measured under $\bar{d}_h$). Lemma C.2 uses Γ, and Lemma C.3 uses the schedule regularity in Assumption 3.5. Lemma C.4 converts the witness-side error into the confidence radius β. Steps 3 and 4 give a $\sqrt{T}$ stable-phase rate, while the burn-in phase contributes $\mathcal{O}(Ht_0)$ with $t_0 = \mathcal{O}(C_{\mathrm{cov}} K\beta)$.

Notations. We analyze the pooled-confidence version of the routed optimistic algorithm. At the beginning of episode t, each basis class $\mathcal{F}_k$ has a confidence set $\mathcal{F}_k^{(t-1)}(\beta)$ constructed from the pooled history $\mathcal{D}^{(t-1)} = \{(\tau^{(i)}, w_i) : i < t\}$, meaning that the regression residual is computed against *every* past trajectory, not only the trajectories routed to basis k. Routing determines which optimistic basis is executed; the confidence guarantee used below controls the Bellman residual of the selected function relative to the cumulative occupancy across all previous episodes. After observing w_t, the learner chooses

$$(k_t,\ f^{(t)}) \in \arg \max_{k\in[K],\, f\in\mathcal{F}_k^{(t-1)}(\beta)} V_1^f(s_0; w_t),\ V_h^f(s; w) := \max_a f_h(s, a; w). \tag{7}$$

The deployed policy $\pi^{(t)}$ is greedy with respect to $f^{(t)}$ under w_t, with state-action occupancy $d_h^{(t)} := d_h^{\pi^{(t)}}$. Write the scalarized Bellman residual of the selected optimistic function as

$$\delta_h^{(t)}(s, a) = f_h^{(t)}(s, a; w_t) - (\mathcal{T}_h^{w_t} f_{h+1}^{(t)})(s, a),$$

with the convention $f_{H+1}^{(t)} \equiv 0$. Finally, define the basis-mixture witness $\bar{d}_h := K^{-1}\sum_{k=1}^K d_h^{\pi_{\mathrm{bk}}^\star}$ and the un-normalized cumulative empirical occupancy $\bar{d}_h^{(t)} := \sum_{i=1}^{t-1} d_h^{(i)}$.

Lastly, recall that the χ^2-divergence between two distributions $P(x)$ and $Q(x)$ is defined as

$$\chi^2(P\|Q) = \sum_{x:Q(x)>0} \frac{(P(x) - Q(x))^2}{Q(x)}.$$

Proof structure. The proof relies on two events holding simultaneously, each with probability at least $1 - \delta/2$ (so that their intersection holds with probability at least $1 - \delta$). For E_{opt}, every realizable $Q^{\pi_{\mathrm{bk}}^\star}(\cdot; w)$ remains in its confidence set $\mathcal{F}_k^{(t-1)}(\beta)$ for all relevant t, k, w. This is the standard optimism event for confidence-set value iteration, union-bounded over the K basis classes. As for E_{reg}, for every selected $f^{(t)}$, the cumulative squared Bellman residual on past trajectories is bounded by β. This is the standard in-sample regression event, again, union-bounded over the K basis classes. Next, we work on the intersection of E_{opt} and E_{reg}.

Lemma C.1. *On the event E_{opt}, for every episode t,*

$$V_1^{f^{(t)}}(s_0; w_t) \geq \mathcal{J}(\pi_{w_t}^\star; w_t).$$

Proof. Assumption 3.4 expresses the occupancy of $\pi^\star_w$ as a convex combination of the basis occupancies. Since the scalarized value is linear in the occupancy measure,

$$\mathcal{J}(\pi^\star_w; w) = \sum_k \nu_w(k)\mathcal{J}(\pi^\star_{\mathsf{b}k}; w).$$

By Lemma 3.8, there exists $k^\star(w_t) \in [K]$ such that the optimal MORL policy and the optimal basis policy at index $k^\star(w_t)$ have the same scalarized return: $\mathcal{J}(\pi^\star_{w_t}; w_t) = \mathcal{J}(\pi^\star_{\mathsf{b}k^\star(w_t)}; w_t)$.

By event E_{opt}, the corresponding true Q-function $Q^{\pi^\star_{\mathsf{b}k^\star(w_t)}}(\cdot; w_t)$ lies in $\mathcal{F}^{(t-1)}_{k^\star(w_t)}(\beta)$. Restricting the joint argmax in (7) to $k = k^\star(w_t)$ then yields

$$\begin{aligned} V_1^{f^{(t)}}(s_0; w_t) \;&\geq\; \max_{f \in \mathcal{F}^{(t-1)}_{k^\star(w_t)}(\beta)} V_1^f(s_0; w_t) \;\geq\; V_1^{Q^{\pi^\star_{\mathsf{b}k^\star(w_t)}}}(s_0; w_t) \\ &=\; \mathcal{J}(\pi^\star_{\mathsf{b}k^\star(w_t)}; w_t) \;=\; \mathcal{J}(\pi^\star_{w_t}; w_t). \end{aligned}$$

Lemma C.1 allows us to upper bound regret by the gap between the optimistic value and the value actually obtained by the greedy policy. The usual telescoping identity for finite-horizon Bellman residuals gives

$$\begin{aligned} \mathsf{Reg}(T) &\leq \sum_{t=1}^{T} \left[V_1^{f^{(t)}}(s_0; w_t) - \mathcal{J}(\pi^{(t)}; w_t) \right] \\ &= \sum_{t=1}^{T} \sum_{h=1}^{H} \mathbb{E}_{(s,a)\sim d_h^{(t)}} \left[\delta_h^{(t)}(s,a) \right]. \end{aligned} \tag{8}$$

The last equality follows from the standard layerwise telescoping along the trajectory of $\pi^{(t)}$. Because $\pi^{(t)}$ is greedy with respect to $f^{(t)}$, at each step h, $V_h^{f^{(t)}}(s_h; w_t) = f_h^{(t)}(s_h, a_h; w_t)$. Subtracting the scalarized reward plus the next optimistic value at each layer telescopes from $V_1^{f^{(t)}}(s_0; w_t)$ down to the realized scalarized return, leaving exactly the expected sum of $\delta_h^{(t)}$.

Let us analyze the term $\mathbb{E}_{(s,a)\sim d_h^{(t)}} \left[\delta_h^{(t)}(s,a) \right]$ in the regret upper bound in equation (8). For each t and h, multiply and divide by $\sqrt{\bar{d}_h(s,a)}$ inside the inner product:

$$\mathbb{E}_{d_h^{(t)}}[\delta_h^{(t)}] = \sum_{s,a} \frac{d_h^{(t)}(s,a)}{\sqrt{\bar{d}_h(s,a)}} \cdot \sqrt{\bar{d}_h(s,a)}\delta_h^{(t)}(s,a) \leq \sqrt{\phi_h(d_h^{(t)})} \cdot \sqrt{\mathbb{E}_{\bar{d}_h}\left[(\delta_h^{(t)})^2\right]}. \tag{9}$$

Notably the conflicting factor $\phi_h(d)$ is connected with the χ^2-divergence:

$$\phi_h(d) := \sum_{s,a} \frac{d(s,a)^2}{\bar{d}_h(s,a)} = 1 + \chi^2(d \| \bar{d}_h).$$

The two factors in equation (9) are controlled separately: the conflict factor by Assumption 3.5 and the in-sample factor by combining the witness-comparison bound with the pooled-regression guarantee.

Lemma C.2. *Under Assumption 3.5, for every t and h,*

$$\phi_h(d_h^{(t)}) := \sum_{s,a} \frac{d_h^{(t)}(s,a)^2}{\bar{d}_h(s,a)} = 1 + \chi^2(d_h^{(t)} \| \bar{d}_h) \leq 1 + \Gamma.$$

Proof. By Assumption 3.5, $d_h^{(t)} = \sum_k \nu^{(t)}(k) d_h^{\pi^\star_{\mathsf{b}k}}$. The map

$$d \mapsto \phi_h(d) = \sum_{s,a} \frac{d(s,a)^2}{\bar{d}_h(s,a)}$$

is a squared norm in the measure weighted by $\bar{d}_h^{-1}$, hence convex. Jensen's inequality gives

$$\phi_h(d_h^{(t)}) \le \sum_k \nu^{(t)}(k)\phi_h(d_h^{\pi^\star_{\mathsf{bk}}}) \le 1 + \Gamma,$$

where the last step is Definition 3.3.

Lemma C.3. *Under Assumption 3.5 (with constants $c_0 \in (0,1]$ and burn-in time $t_0 \in \mathbb{N}$), for every $t \ge t_0$, $h \in [H]$, and (s,a),*

$$\bar{d}_h^{(t)}(s,a) \;\ge\; \frac{c_0\,(t-1)}{C_{\mathrm{cov}}}\,\bar{d}_h(s,a).$$

Proof. The cumulative empirical occupancy can be expanded in terms of the basis occupancies:

$$\frac{\bar{d}_h^{(t)}(s,a)}{t-1} = \sum_{k=1}^K \overline{\nu}^{(t)}(k) d_h^{\pi^\star_{\mathsf{bk}}}(s,a) \ge \frac{c_0}{C_{\mathrm{cov}} K} \sum_{k=1}^K d_h^{\pi^\star_{\mathsf{bk}}}(s,a) = \frac{c_0}{C_{\mathrm{cov}}} \bar{d}_h(s,a).$$

The inequality is exactly the stable-phase lower bound in Assumption 3.5. Thus after burn-in, the data distribution accumulated by the algorithm pointwise dominates the uniform basis mixture up to the factor C_{cov}/c_0.

Lemma C.4. *On event E_{reg}, for every $t \ge t_0$ and $h \in [H]$,*

$$\mathbb{E}_{\bar{d}_h}\left[(\delta_h^{(t)})^2\right] \le \frac{C_{\mathrm{cov}}\,\beta}{c_0(t-1)}.$$

Proof. The standard regression guarantee for optimistic confidence sets, union bounded over $k \in [K]$, gives for the selected $f^{(t)}$ (i.e., on the event E_{reg}),

$$\sum_{i=1}^{t-1} \mathbb{E}_{d_h^{(i)}}\left[(\delta_h^{(t)})^2\right] = \mathbb{E}_{\bar{d}_h^{(t)}}\left[(\delta_h^{(t)})^2\right] \le \beta. \tag{10}$$

This is exactly where the proof uses pooled confidence updates: the measure on the left is the cumulative occupancy across all previous episodes, not just those routed to k_t. Lemma C.3 gives

$$\bar{d}_h(s,a) \le \frac{C_{\mathrm{cov}}}{c_0(t-1)} \bar{d}_h^{(t)}(s,a), \text{ for } t \ge t_0.$$

Multiplying this pointwise inequality by $(\delta_h^{(t)}(s,a))^2$ and summing over (s,a), then applying (10), yields

$$\mathbb{E}_{\bar{d}_h}\left[(\delta_h^{(t)})^2\right] \le \frac{C_{\mathrm{cov}}}{c_0(t-1)} \mathbb{E}_{\bar{d}_h^{(t)}}\left[(\delta_h^{(t)})^2\right] \le \frac{C_{\mathrm{cov}}\beta}{c_0(t-1)}.$$

Proof of Theorem 3.6. Combining equation (9) with Lemmas C.2 and C.4 gives, for every $t \ge t_0$ and $h \in [H]$,

$$\mathbb{E}_{d_h^{(t)}}[\delta_h^{(t)}] \le \sqrt{1+\Gamma} \cdot \sqrt{\frac{C_{\mathrm{cov}}\beta}{c_0(t-1)}} = \sqrt{\frac{C_{\mathrm{cov}}\,(1+\Gamma)\beta}{c_0(t-1)}}. \tag{11}$$

We split the regret (8) at the burn-in time t_0:

$$\mathsf{Reg}(T) \;\le\; \underbrace{\sum_{t=1}^{t_0-1}\sum_{h=1}^H \mathbb{E}_{d_h^{(t)}}[\delta_h^{(t)}]}_{\text{(I)}} \;+\; \underbrace{\sum_{t=t_0}^{T}\sum_{h=1}^H \mathbb{E}_{d_h^{(t)}}[\delta_h^{(t)}]}_{\text{(II)}}.$$

During burn-in, for each $t < t_0$, $\sum_h \mathbb{E}_{d_h^{(t)}}[\delta_h^{(t)}] = V_1^{f^{(t)}}(s_0;w_t) - \mathcal{J}(\pi^{(t)};w_t) \le H$ since both $V_1^{f^{(t)}} \in [0,H]$ and $\mathcal{J}(\pi^{(t)};w_t) \in [0,H]$. Hence (I) $\le H\,t_0$.

During the stable phase, substituting (11) and summing over h and t, II is at most

$$H \sum_{t=t_0}^{T} \sqrt{\frac{C_{\mathrm{cov}}\,(1+\Gamma)\,\beta}{c_0\,(t-1)}} = H \sqrt{\frac{C_{\mathrm{cov}}\,(1+\Gamma)\,\beta}{c_0}} \sum_{t=t_0}^{T} \frac{1}{\sqrt{t-1}} \le 2H \sqrt{\frac{C_{\mathrm{cov}}\,(1+\Gamma)\,\beta\,T}{c_0}},$$

using $\sum_{t=t_0}^{T}(t-1)^{-1/2} \le \sum_{u=1}^{T-1} u^{-1/2} \le 2\sqrt{T}$.

Adding (I) and (II) and absorbing the universal constant $1/c_0$ into $\mathcal{O}(\cdot)$,

$$\mathsf{Reg}(T) \le H t_0 + 2H \sqrt{\frac{C_{\mathrm{cov}}(1+\Gamma)\beta T}{c_0}} = \mathcal{O}\Big(H\sqrt{C_{\mathrm{cov}}(1+\Gamma)\beta T}\Big) + \mathcal{O}(H t_0).$$

The preference process satisfies the schedule regularity of Assumption 3.5 after $t_0 = \mathcal{O}(C_{\mathrm{cov}} K \beta)$ episodes (a standard Hoeffding's bound on the visitation counts of the K basis components, with the $\log$ factor absorbed into β). Substituting yields

$$\mathsf{Reg}(T) \le \mathcal{O}\Big(H\sqrt{C_{\mathrm{cov}}\,(1+\Gamma)\beta T} + H C_{\mathrm{cov}} K \beta\Big)\,,$$

which is equation (2). The corresponding worst-case scaling follows by applying $1+\Gamma \le K$ from Lemma C.5.

For completeness, we record the structural lemma used above; it shows that the conflict coefficient is at most $K-1$ for any basis, with equality (up to lower-order terms) in the disjoint regime that powers the lower bound.

Lemma C.5. *For any collection of K basis occupancies $\{d_h^{\pi^\star_{\mathrm{b}k}}\}_{k\in[K]}$ and every $h \in [H]$,*

$$\Gamma = \max_{h\in[H]} \max_{k\in[K]} \chi^2\Big(d_h^{\pi^\star_{\mathrm{b}k}} \big\| \bar{d}_h\Big) \le K-1.$$

Equality holds (up to lower-order terms) when the basis occupancies have pairwise disjoint supports.

Proof. Fix h and k. Pointwise, $\bar{d}_h(s,a) \ge \frac{1}{K}\, d_h^{\pi^\star_{\mathrm{b}k}}(s,a)$ on the support of $d_h^{\pi^\star_{\mathrm{b}k}}$. Hence

$$\chi^2\Big(d_h^{\pi^\star_{\mathrm{b}k}} \,\big\|\, \bar{d}_h\Big) = \sum_{s,a} \frac{(d_h^{\pi^\star_{\mathrm{b}k}}(s,a))^2}{\bar{d}_h(s,a)} - 1 \le \sum_{s,a} \frac{(d_h^{\pi^\star_{\mathrm{b}k}}(s,a))^2}{(1/K)\, d_h^{\pi^\star_{\mathrm{b}k}}(s,a)} - 1 = K-1.$$

The disjoint-support case attains equality because then $\bar{d}_h(s,a) = \frac{1}{K}\, d_h^{\pi^\star_{\mathrm{b}k}}(s,a)$ on the support.

C.2 Proof of the regret lower bound

Theorem C.6. *There exists a family of MOMDPs with K basis policies, $C_{\mathrm{cov}} \le K$, and $\Gamma = K-1$, such that for any (possibly randomized) learning algorithm A and any $T \ge K$, a preference sequence that cycles uniformly through $\{e_1, \dots, e_K\}$ satisfies*

$$\mathbb{E}\big[\mathsf{Reg}(T)\big] \ge \Omega\Big(H\sqrt{KT}\Big) = \Omega\Big(H\sqrt{(1+\Gamma)T}\Big)\,.$$

Together, Theorems 3.6 and C.6 pin down the dependence on K and Γ up to factors of $\sqrt{C_{\mathrm{cov}}\,\log|\mathcal{F}|}$.

In the following proof, we prove the lower bound by embedding a standard Bernoulli two-armed bandit problem into a simple subclass of MOMDPs. This is sufficient because a lower bound for one valid MOMDP family is automatically a lower bound for the larger MOMDP class. The constructed family has K pairwise-disjoint sub-chains, one per objective, each containing one Bernoulli two-armed bandit. Here, a sub-chain means a branch of the environment state space, not a policy: the k-th optimal basis policy is the policy that routes into sub-chain $\mathcal{S}_k$ and then plays the better binary action.

Under a cyclic preference sequence, regret has two sources: 1) a routing term for episodes whose first action does not match the active preference; 2) K independent bandit terms for wrong binary choices inside correctly routed sub-chains.

We then show that for each preference e_k, the learner faces a two-armed bandit on the episodes with active preference e_k, with mis-routing acting as a no-information mistake that also costs order Δ. The horizon does not create a harder statistical problem in this construction: after the binary decision, the remaining states reveal no additional choices or independent samples. Instead, the same Bernoulli outcome is paid over the remaining $H-1$ stages, so the bandit regret is simply multiplied by $H-1$. Tuning the bandit gap to $\Delta = \frac{1}{8}\sqrt{K/T}$ makes the total unavoidable cost scale as $\Delta T = \Theta(\sqrt{KT})$, and the repeated payoff yields the claimed $\Omega(H\sqrt{KT})$ lower bound.

Hard MOMDP instances. Fix $K \geq 2$, $H \geq 2$, and $\Delta \in (0, 1/4]$. The MOMDP has $m = K$ objectives and a single start state s_0. From s_0, the K deterministic actions $a \in [K]$ route the agent to disjoint state-space branches $\mathcal{S}_1, \ldots, \mathcal{S}_K$, which we call sub-chains. Inside sub-chain $\mathcal{S}_k$, the agent makes one binary decision $u \in \{0, 1\}$ at the second step; thereafter, the chain is uninformative. There is an unknown bit $b_k \in \{0, 1\}$, and the k-th reward coordinate at every step inside $\mathcal{S}_k$ equals a single Bernoulli outcome drawn at the moment of the binary decision, with mean

$$\tfrac{1}{2} + \Delta \quad \text{if } u = b_k, \tfrac{1}{2} \quad \text{if } u \neq b_k.$$

All other reward coordinates are zero. Thus, each episode yields only one statistically informative bandit observation, even though it has a horizon of H. A correctly routed episode where the wrong binary action is chosen contributes regret $(H-1)\Delta$, and an incorrectly routed episode contributes regret at least $(H-1)\Delta$ as well.

Preference sequences. The preference sequence cycles through $e_1, e_2, \ldots, e_K, e_1, e_2, \ldots$, so each preference e_k appears between $\lfloor T/K \rfloor$ and $\lceil T/K \rceil$ times. The optimal basis policy $\pi^\star_{\mathsf{b}k}$ routes to sub-chain $\mathcal{S}_k$ and plays b_k; the K basis occupancies are supported on disjoint sub-chains, so by Lemma C.5, $\Gamma = K - 1$, and the basis-mixture witness $\bar{d}_h$ gives $C_{\mathrm{cov}} \leq K$. We use a lower bound result from the work of Lattimore and Szepesvári [25].

Lemma C.7 (Bandit lower bound [25])**.** *For any randomized algorithm interacting with a Bernoulli two-armed bandit of gap $\Delta \in (0, 1/4]$, the expected cumulative regret after n pulls is at least $c_{\mathrm{LB}} \min(\Delta n,\, 1/\Delta)$ for a universal constant $c_{\mathrm{LB}} > 0$, in the worst case over the unknown optimal arm.*

Proof of Theorem C.6. We proceed in three steps. First, for a fixed interaction history, partition the T episodes into two disjoint sets: M episodes where the learner *mis-routes*, meaning that its first action does not equal the index k_t of the active preference e_{k_t}. Each such episode loses at least $(H-1)\Delta$ relative to the optimal policy for e_{k_t}. For each $k \in [K]$, n_k episodes where the learner correctly routes to $\mathcal{S}_k$ under preference e_k. Conditional on these episodes, the learner observes exactly n_k pulls from the two-armed Bernoulli bandit associated with unknown bit b_k.

Let M_k be the number of mis-routed episodes whose active preference is e_k, and let $m_k := M_k + n_k$ be the total number of appearances of preference e_k. The cyclic sequence fixes m_k deterministically, with $\lfloor T/K \rfloor \leq m_k \leq \lceil T/K \rceil$, and $M + \sum_{k=1}^K n_k = \sum_{k=1}^K m_k = T$ pathwise. Taking expectation over the learner's randomness and the bandit rewards, the cumulative regret satisfies

$$\mathbb{E}[\mathsf{Reg}(T)] \geq (H-1)\Delta\, \mathbb{E}[M] + (H-1) \sum_{k=1}^K \mathbb{E}[\mathsf{Reg}_k^{\mathrm{bandit}}(n_k)], \tag{12}$$

where $\mathsf{Reg}_k^{\mathrm{bandit}}(n_k)$ denotes the cumulative regret of the induced bandit on sub-chain $\mathcal{S}_k$ (the leading $H-1$ factor accounts for the fact that each Bernoulli outcome is paid out over $H-1$ steps).

Next, set $\Delta = \frac{1}{8}\sqrt{K/T}$, which satisfies $\Delta \leq 1/8$ since $T \geq K$. The cyclic preference sequence ensures $m_k \leq \lceil T/K \rceil \leq 2T/K$, so

$$\Delta m_k \leq \tfrac{1}{8}\sqrt{K/T} \cdot \tfrac{2T}{K} = \tfrac{1}{4}\sqrt{T/K} \leq \tfrac{1}{\Delta}.$$

For a fixed k, restrict attention to the m_k episodes whose active preference is e_k. In these episodes, the learner must either route to $\mathcal{S}_k$ and choose one of the two bandit arms, or mis-route and incur one-stage scalarized regret at least Δ while receiving no observation from the k-th bandit. Thus, the sum $\Delta M_k + \mathsf{Reg}_k^{\mathrm{bandit}}(n_k)$ is lower bounded by the regret of a Bernoulli two-armed bandit problem with m_k opportunities. Lemma C.7 gives

$$\mathbb{E}\Big[\Delta M_k + \mathsf{Reg}_k^{\mathrm{bandit}}(n_k)\Big] \geq c_{\mathrm{LB}} \min(\Delta m_k, 1/\Delta) = c_{\mathrm{LB}} \Delta m_k.$$

Summing over k yields

$$\Delta\mathbb{E}[M] + \sum_{k=1}^{K} \mathbb{E}[\mathsf{Reg}_k^{\mathrm{bandit}}(n_k)] \geq c_{\mathrm{LB}}\Delta \sum_{k=1}^{K} m_k = c_{\mathrm{LB}}\Delta T.$$

Lastly, substituting the preceding bound into equation (12),

$$\mathbb{E}[\mathsf{Reg}(T)] \geq c_{\mathrm{LB}}(H-1)\Delta T,$$

Substituting $\Delta = \frac{1}{8}\sqrt{K/T}$, we have

$$\mathbb{E}[\mathsf{Reg}(T)] \geq \frac{c_{\mathrm{LB}}}{8}(H-1)\sqrt{KT} = \Omega\Big(H\sqrt{KT}\Big).$$

Since the construction has $\Gamma = K-1$, this is equivalent to $\Omega(H\sqrt{(1+\Gamma)T})$. Thus, we have completed the proof of the lower bound.

C.3 Bounds on objective conflict coefficients

Although Algorithm 1 does not require Γ as an input, its off-policy transfer step is governed by the same overlap quantity. At iteration t, sampling a basis policy uniformly yields the routed-ensemble mixture and its associated conflict measure

$$\bar{d}_{h,t}^{\mathrm{RE}} = \frac{1}{K}\sum_{j=1}^{K} d_h^{\pi_j^{(t-1)}}, \ \Gamma_{\mathrm{RE}}^{(t)} = \max_{h\in[H]} \max_{j\in[K]} \chi^2\Big(d_h^{\pi_j^{(t-1)}} \,\Big\|\, \bar{d}_{h,t}^{\mathrm{RE}}\Big). \tag{13}$$

When the learned ensemble recovers the latent basis policies, $\Gamma_{\mathrm{RE}}^{(t)}$ coincides with Γ. The following proposition formalizes that the off-policy estimators used to route and update the ensemble have variance inflated by exactly the factor $1+\Gamma_{\mathrm{RE}}^{(t)}$.

Proposition C.8. *Fix an iteration t of Algorithm 1, a stage h, and a target basis index $j \in [K]$. Suppose B replay samples (s_i, a_i) are drawn i.i.d. from $\bar{d}_{h,t}^{\mathrm{RE}}$, and let $g_h : \mathcal{S}\times\mathcal{A} \to [-G, G]$ be any bounded measurable signal with target value*

$$\theta_{j,h} = \mathbb{E}_{d_h^{\pi_j^{(t-1)}}}[g_h(s,a)].$$

The importance-weighted off-policy estimator

$$\widehat{\theta}_{j,h} := \frac{1}{B}\sum_{i=1}^{B} \frac{d_h^{\pi_j^{(t-1)}}(s_i,a_i)}{\bar{d}_{h,t}^{\mathrm{RE}}(s_i,a_i)} g_h(s_i,a_i)$$

is unbiased and satisfies

$$\mathbb{E}\Big[(\widehat{\theta}_{j,h} - \theta_{j,h})^2\Big] \leq \frac{G^2\big(1+\Gamma_{\mathrm{RE}}^{(t)}\big)}{B}. \tag{14}$$

In the idealized case $\pi_j^{(t-1)} = \pi_{\mathsf{b}j}^\star$ for all j, $\Gamma_{\mathrm{RE}}^{(t)} = \Gamma$.

Proof. Fix iteration t, stage h, and target basis index j. For brevity, write $d_j := d_h^{\pi_j^{(t-1)}}$, $\mu := \bar{d}_{h,t}^{\mathrm{RE}}$, and $\rho_j(s,a) := d_j(s,a)/\mu(s,a)$. Finite $\Gamma_{\mathrm{RE}}^{(t)}$ implies $d_j \ll \mu$, so ρ_j is well defined on $\mathrm{supp}(\mu)$. Since d_j and μ are probability distributions over $\mathcal{S}\times\mathcal{A}$ at stage h, $\sum_{s,a} d_j = \sum_{s,a}\mu = 1$.

Each replay sample (s_i, a_i) is drawn from μ, so

$$\mathbb{E}[\rho_j(s_i,a_i)g_h(s_i,a_i)] = \sum_{s,a}\mu(s,a)\frac{d_j(s,a)}{\mu(s,a)}g_h(s,a) = \sum_{s,a} d_j(s,a)g_h(s,a) = \theta_{j,h}.$$

Averaging over B samples preserves the mean: $\mathbb{E}[\widehat{\theta}_{j,h}] = \theta_{j,h}$.

For a single sample, the second moment of the importance-weighted signal is bounded by

$$\mathbb{E}_\mu\big[\rho_j(s,a)^2 g_h(s,a)^2\big] \leq G^2\sum_{s,a}\frac{d_j(s,a)^2}{\mu(s,a)} = G^2\big(1+\chi^2(d_j\|\mu)\big) \leq G^2\big(1+\Gamma_{\mathrm{RE}}^{(t)}\big), \tag{15}$$

where the equality uses the standard χ^2-identity $\sum_{s,a} d_j^2/\mu = 1 + \chi^2(d_j\|\mu)$ (which holds because d_j and μ are probability distributions), and the last inequality is the definition of $\Gamma_{\mathrm{RE}}^{(t)}$. The B samples are independent, so

$$\mathbb{E}\Big[(\widehat{\theta}_{j,h} - \theta_{j,h})^2\Big] = \mathrm{Var}(\widehat{\theta}_{j,h}) \le \frac{\mathbb{E}_\mu\big[\rho_j^2 g_h^2\big]}{B} \le \frac{G^2\big(1+\Gamma_{\mathrm{RE}}^{(t)}\big)}{B},$$

which leads to equation (14). If $\pi_j^{(t-1)} = \pi_{\mathsf{b}j}^\star$ for every $j \in [K]$, then $\mu = \bar{d}_h$ and $\Gamma_{\mathrm{RE}}^{(t)} = \Gamma$ by Definition 3.3.

C.4 Approximate low-rank conditions

Next, we show how the regret bound changes when Assumption 3.4 only holds approximately. We separate two effects. Relative to the best policy in the chosen K-policy basis, the proof of Theorem 3.6 still gives a sublinear regret bound, with the conflict factor inflated by ε. If the chosen basis does not contain a truly optimal policy for a preference, then true regret also includes the gap between the true optimum and the best basis policy.

Let $\bar{d}_h = K^{-1}\sum_{k=1}^K d_h^{\pi_{\mathsf{b}k}^\star}$ be the same basis-mixture witness used in the proof of Theorem 3.6. We replace Assumption 3.4 by the following approximate version.

Assumption C.9. *There exist latent basis policies $\{\pi_{\mathsf{b}1}^\star, \dots, \pi_{\mathsf{b}K}^\star\}$ and a tolerance $\varepsilon \ge 0$ such that every deployed policy $\pi \in \{\pi^{(t)} : t \le T\}$ admits a single mixing vector $\nu_\pi \in \Delta^{K-1}$ satisfying, for every stage $h \in [H]$,*

$$d_h^\pi = q_{\pi,h} + e_{\pi,h}, q_{\pi,h} := \sum_{k=1}^K \nu_\pi(k) d_h^{\pi_{\mathsf{b}k}^\star}, \|e_{\pi,h}\|_{\bar{d}_h^{-1}}^2 := \sum_{s,a} \frac{e_{\pi,h}(s,a)^2}{\bar{d}_h(s,a)} \le \varepsilon^2.$$

The norm is interpreted with the usual support convention: finite norm requires $e_{\pi,h}(s,a) = 0$ whenever $\bar{d}_h(s,a) = 0$.

The $\bar{d}_h^{-1}$-weighted norm is chosen to match the Cauchy–Schwarz step in Theorem 3.6: the proof controls $\phi_h(d) = \sum_{s,a} d(s,a)^2/\bar{d}_h(s,a) = \|d\|_{\bar{d}_h^{-1}}^2$. Thus, an error bound in this same norm can be carried through the original argument by the triangle inequality.

For this approximate statement, we keep the stable-coverage consequence of Assumption 3.5: for all $t \ge t_0$, $h \in [H]$, and (s,a),

$$\bar{d}_h^{(t)}(s,a) := \sum_{i=1}^{t-1} d_h^{(i)}(s,a) \;\ge\; \frac{c_0(t-1)}{C_{\mathrm{cov}}}\,\bar{d}_h(s,a). \tag{16}$$

This is the only property of coverage needed for the pooled-regression step. It can be assumed directly or derived from an approximate exploration condition that guarantees the accumulated actual occupancies dominate the basis mixture.

For a preference w, define the best value achievable by the chosen basis as

$$\mathcal{J}_K^\star(w) := \max_{k\in[K]} \mathcal{J}(\pi_{\mathsf{b}k}^\star; w) = \max_{\nu\in\Delta^{K-1}} \sum_{k=1}^K \nu(k)\mathcal{J}(\pi_{\mathsf{b}k}^\star; w).$$

The equality holds because the objective is linear in the mixing weights. We measure basis-relative regret by

$$\mathsf{Reg}_K(T) := \sum_{t=1}^T \Big[\mathcal{J}_K^\star(w_t) - \mathcal{J}(\pi^{(t)}; w_t)\Big].$$

Proposition C.10. *Suppose scalarized rewards are bounded in $[0,1]$ per step, Bellman realizability and completeness hold for each per-basis class $\mathcal{F}_k$, Assumption C.9 holds with tolerance ε, and the stable-coverage condition* (16) *holds after a burn-in time $t_0 = \mathcal{O}(C_{\mathrm{cov}}K\beta)$, where $\beta :=$*

$c_1 \log(K|\mathcal{F}|T/\delta)$. *Then, with probability at least* $1-\delta$*, the pooled-confidence routed-ensemble algorithm satisfies*

$$\mathsf{Reg}_K(T) \;\le\; \mathcal{O}\Big(H\sqrt{C_{\mathrm{cov}}\beta T}\,\big(\sqrt{1+\Gamma}+\varepsilon\big) + HC_{\mathrm{cov}}K\beta\Big). \tag{17}$$

When $\varepsilon = 0$ *and the basis is exact,* $\mathsf{Reg}_K(T) = \mathsf{Reg}(T)$ *and this reduces to Theorem 3.6.*

Proof. The proof follows the same sketch as Theorem 3.6 (Appendix C.1), with two simplifications and one change. First, the comparator is changed from the true optimal value $\mathcal{J}(\pi^\star_w; w)$ to the basis-oracle value $\mathcal{J}^\star_K(w)$, so routed optimism becomes *exact* against this weaker comparator (Lemma C.11 below). Second, the conflict factor $\sqrt{\phi_h(d_h^{(t)})}$ inflates from $\sqrt{1+\Gamma}$ (Lemma C.2) to $\sqrt{1+\Gamma}+\varepsilon$ (Lemma C.12 below), because deployed occupancies are only approximately in the basis hull. Third, the witness-comparison and pooled-regression steps are *unchanged*: they use only the stable coverage condition (16) and Bellman completeness, neither of which is affected by relaxing Assumption 3.4. We carry out the modifications below.

First, we show the following claim on the optimism of the routed basis.

Claim C.11. *On the optimism event* E_{opt} *(defined in Appendix C.1), for every episode* t*,*

$$V_1^{f^{(t)}}(s_0; w_t) \;\ge\; \mathcal{J}^\star_K(w_t).$$

Proof. Let $k^\star_K(w_t) \in \arg\max_{k\in[K]} \mathcal{J}(\pi^\star_{\mathsf{b}k}; w_t)$. By the optimism event E_{opt}, the true Q-function $Q^{\pi^\star_{\mathsf{b}k^\star_K(w_t)}}(\cdot; w_t)$ belongs to the confidence set $\mathcal{F}^{(t-1)}_{k^\star_K(w_t)}(\beta)$. The routed optimistic choice in equation (7) maximizes over all basis indices and all functions in their confidence sets, so we may restrict the maximization to the single feasible candidate $(k^\star_K(w_t), Q^{\pi^\star_{\mathsf{b}k^\star_K(w_t)}})$. This gives

$$V_1^{f^{(t)}}(s_0; w_t) \;\ge\; V_1^{Q^{\pi^\star_{\mathsf{b}k^\star_K(w_t)}}}(s_0; w_t) \;=\; \mathcal{J}(\pi^\star_{\mathsf{b}k^\star_K(w_t)}; w_t) \;=\; \mathcal{J}^\star_K(w_t).$$

Second, we demonstrate regret reduction via a certain identity. This step is identical to the regret reduction in equation (8). Working on the intersection $E_{\mathrm{opt}} \cap E_{\mathrm{reg}}$ (probability at least $1-\delta$), Lemma C.11 together with the standard layerwise telescoping identity gives

$$\mathsf{Reg}_K(T) \;\le\; \sum_{t=1}^{T}\Big[V_1^{f^{(t)}}(s_0; w_t) - \mathcal{J}(\pi^{(t)}; w_t)\Big] \;=\; \sum_{t=1}^{T}\sum_{h=1}^{H}\mathbb{E}_{d_h^{(t)}}\Big[\delta_h^{(t)}\Big]. \tag{18}$$

Below, we control the Bellman residual term on the right-hand side.

Identical to equation (9), for every $t \in [T]$ and $h \in [H]$, we have

$$\mathbb{E}_{d_h^{(t)}}\Big[\delta_h^{(t)}\Big] \le \sqrt{\phi_h(d_h^{(t)})}\cdot\sqrt{\mathbb{E}_{\bar{d}_h}\Big[(\delta_h^{(t)})^2\Big]}. \tag{19}$$

Next, we show the following approximate conflict bound, which is the only step affected by ε.

Claim C.12. *Every policy* π *satisfying the decomposition in Assumption C.9 also satisfies, for every* $h \in [H]$*,*

$$\sqrt{\phi_h(d_h^\pi)} \;\le\; \sqrt{1+\Gamma}+\varepsilon. \tag{20}$$

Proof. Since $\phi_h(d) = \|d\|^2_{\bar{d}_h^{-1}}$, Assumption C.9 and the triangle inequality in ℓ_2 norm give

$$\sqrt{\phi_h(d_h^\pi)} \;=\; \|q_{\pi,h} + e_{\pi,h}\|_{\bar{d}_h^{-1}} \;\le\; \|q_{\pi,h}\|_{\bar{d}_h^{-1}} + \|e_{\pi,h}\|_{\bar{d}_h^{-1}} \;\le\; \sqrt{\phi_h(q_{\pi,h})}+\varepsilon.$$

The first summand is controlled exactly as in Lemma C.2: by convexity of ϕ_h and Definition 3.3,

$$\phi_h(q_{\pi,h}) \;=\; \phi_h\Big(\textstyle\sum_k \nu_\pi(k)\, d_h^{\pi^\star_{\mathsf{b}k}}\Big) \;\le\; \displaystyle\sum_k \nu_\pi(k)\,\phi_h(d_h^{\pi^\star_{\mathsf{b}k}}) \;\le\; 1+\Gamma.$$

Combining this display with the triangle-inequality bound proves equation (20). Applying Lemma C.12 to the deployed policy $\pi^{(t)}$ yields

$$\sqrt{\phi_h(d_h^{(t)})} \;\le\; \sqrt{1+\Gamma}+\varepsilon. \tag{21}$$

The cumulative-occupancy lower bound in equation (16), combined with the pooled-regression event E_{reg}, gives the same in-sample-to-witness bound as Lemma C.4: for every $t \ge t_0$ and $h \in [H]$,

$$\sqrt{\mathbb{E}_{\bar{d}_h}\Big[(\delta_h^{(t)})^2\Big]} \;\le\; \sqrt{\frac{C_{\mathrm{cov}}\,\beta}{c_0(t-1)}}. \tag{22}$$

This step depends only on equation (16), the pooled-regression event E_{reg}, and Bellman completeness. It does not use exact basis decomposition.

Finally, combining equations (19), (21), and (22), for every $t \ge t_0$ and $h \in [H]$,

$$\mathbb{E}_{d_h^{(t)}}\Big[\delta_h^{(t)}\Big] \;\le\; \big(\sqrt{1+\Gamma}+\varepsilon\big)\sqrt{\frac{C_{\mathrm{cov}}\,\beta}{c_0(t-1)}}. \tag{23}$$

We split the regret in equation (18) at the burn-in time t_0 and bound each part.

For each $t < t_0$, $\sum_h \mathbb{E}_{d_h^{(t)}}[\delta_h^{(t)}] = V_1^{f^{(t)}}(s_0;w_t) - \mathcal{J}(\pi^{(t)};w_t) \le H$, since both quantities lie in $[0,H]$. Hence, the burn-in contribution is at most $H\,t_0$.

Substituting equation (23) and using $\sum_{t=t_0}^{T}(t-1)^{-1/2} \le \sum_{u=1}^{T-1} u^{-1/2} \le 2\sqrt{T}$,

$$\sum_{t=t_0}^{T}\sum_{h=1}^{H}\mathbb{E}_{d_h^{(t)}}\Big[\delta_h^{(t)}\Big] \le H\big(\sqrt{1+\Gamma}+\varepsilon\big)\sqrt{\frac{C_{\mathrm{cov}}\,\beta}{c_0}}\sum_{t=t_0}^{T}\frac{1}{\sqrt{t-1}} \quad \le 2H\big(\sqrt{1+\Gamma}+\varepsilon\big)\sqrt{\frac{C_{\mathrm{cov}}\,\beta\,T}{c_0}}.$$

Combined together, substituting $t_0 = \mathcal{O}(C_{\mathrm{cov}}K\beta)$, and treating t_0 as a fixed constant yields

$$\mathsf{Reg}_K(T) \le \mathcal{O}\Big(H\sqrt{C_{\mathrm{cov}}\beta T}\cdot\big(\sqrt{1+\Gamma}+\varepsilon\big) + HC_{\mathrm{cov}}K\beta\Big),$$

which leads to equation (17).

Lastly, we connect basis-relative regret $\mathsf{Reg}_K(T)$ to the cumulative regret $\mathsf{Reg}(T)$ against the true optimal policy via a clean two-term decomposition. For a fixed basis $\{\pi^\star_{\mathsf{b}1},\dots,\pi^\star_{\mathsf{b}K}\}$, define the cumulative basis-approximation error along the realized preference sequence as

$$A_K(T) = \sum_{t=1}^{T}\big[\mathcal{J}(\pi^\star_{w_t};w_t) - \mathcal{J}^\star_K(w_t)\big].$$

Since every basis policy is feasible, $\mathcal{J}^\star_K(w_t) \le \mathcal{J}(\pi^\star_{w_t};w_t)$ for every t, so $A_K(T) \ge 0$. The quantity $A_K(T)$ depends on the basis $\{\pi^\star_{\mathsf{b}k}\}$ and the preference distribution $\mathcal{P}$, but *not* on the algorithm.

First, we present the exact decomposition of regret relative to the true optimal policy. By telescoping the comparator $\mathcal{J}^\star_K(w_t)$ in and out of the regret sum,

$$\mathsf{Reg}(T) = \underbrace{\sum_{t=1}^{T}\big[\mathcal{J}(\pi^\star_{w_t};w_t) - \mathcal{J}^\star_K(w_t)\big]}_{A_K(T)} + \underbrace{\sum_{t=1}^{T}\Big[\mathcal{J}^\star_K(w_t) - \mathcal{J}(\pi^{(t)};w_t)\Big]}_{\mathsf{Reg}_K(T)}. \tag{24}$$

Proposition C.10 controls the second term; the first depends only on the chosen basis. Second, we bound $A_K(T)$ when optimal policies are also approximately in the hull. Following Assumption C.9, we have $d_h^{\pi^\star_w} = q_{w,h} + e_{w,h}$. Since scalarized rewards $r_h^w(s,a)$ are bounded in $[0,1]$, its inner product with approximation error $e_{w,h}$ has

$$\langle e_{w,h}, r_h^w\rangle \le \big|\langle e_{w,h}, r_h^w\rangle\big| \le \|e_{w,h}\|_{\bar{d}_h^{-1}}\sqrt{\mathbb{E}_{\bar{d}_h}[(r_h^w)^2]} \le \varepsilon.$$

Using the same mixing vector ν_w at every stage,

$$\mathcal{J}(\pi_w^\star; w) = \sum_{h=1}^{H} \langle d_h^{\pi_w^\star}, r_h^w \rangle \le \sum_{h=1}^{H} \langle q_{w,h}, r_h^w \rangle + H\varepsilon$$
$$= \sum_{k=1}^{K} \nu_w(k) \mathcal{J}(\pi_{\mathsf{bk}}^\star; w) + H\varepsilon \le \mathcal{J}_K^\star(w) + H\varepsilon.$$

Summing over $t \in [T]$ yields $A_K(T) \le H\varepsilon T$. Lastly, combining equation (24) with the above, and Proposition C.10, we have

$$\mathsf{Reg}(T) \le \mathcal{O}\Big(H\varepsilon T + H\sqrt{C_{\mathrm{cov}}\beta T} \cdot \big(\sqrt{1+\Gamma} + \varepsilon\big) + HC_{\mathrm{cov}} K\beta \Big).$$

D Omitted Experiments and Implementation Details

In this section, we describe several omitted experimental results as well as the implementation details that are left out from Section 4.

We conduct the MO-Gymnasium experiments on an AMD Ryzen 7 9700X CPU, with a total training time of approximately 72 hours. Meta-World and RLHF experiments are performed on a server with an Intel Xeon E5-2623 v3 CPU and NVIDIA RTX A6000 GPUs. The Meta-World training takes 12 hours, and the RLHF experiment takes 1 hour.

Next, we define the performance metrics used in our experiments. We compute Γ based on the χ^2-divergence between the state occupancy distributions of single-objective basis policies. To account for the temporal diffusion effect described above, we use the time-averaged state occupancy

$$\bar{d}^\pi(s) = \frac{1}{H}\sum_{t=0}^{H} P(s_t = s), \text{ then } \Gamma = \max_k \sum_{s\in\mathcal{S}} \frac{(\bar{d}^{\pi_k}(s) - \bar{d}^{\mathrm{avg}}(s))^2}{\bar{d}^{\mathrm{avg}}(s)},$$

where π_k is the policy trained on objective k alone, and $\bar{d}^{\mathrm{avg}}$ is the uniform mixture of all basis occupancies.

We train a kernel ridge regression model $\hat{f}$ on the dataset of $\{(\mathbf{m}_i, y_i)\}$. The attribution of a conflicting objective j (where $j \in \{5, \dots, 9\}$) is calculated as the predicted marginal gain $\hat{f}(\mathbf{m}_{\mathrm{target}} \cup \{j\}) - \hat{f}(\mathbf{m}_{\mathrm{target}})$, where $\mathbf{m}_{\mathrm{target}}$ is the mask containing only the target objective.

Lastly, we provide the detailed hyperparameter settings used in our experiments to ensure reproducibility. Table 3 lists the parameters for the routed ensemble algorithm across different benchmarks. For routed ensemble, we adapt the base policy-learning algorithm for each environment. We use Soft Modularization [51] for Meta-World, Q-Pensieve [18] for continuous MO-Gymnasium tasks, and PQL [42] for discrete MO-Gymnasium tasks. For all baselines, we use the hyperparameters from their official implementations/repositories to ensure a fair comparison. Each baseline is trained for 5×10^5 time steps on continuous benchmarks and 10^4 time steps on discrete benchmarks.

We implement kernel-based objective attribution in Algorithm 2 using a RBF kernel defined as $\psi(S_i, S_j) = \exp(-\gamma\|S_i - S_j\|^2)$. We determine the kernel bandwidth $\gamma = 0.1$ and the regularization coefficient $\lambda = 0.1$ via cross-validation on the validation set.

D.1 Synthetic environments

Multi-objective environments. We utilize a 2D parametric optimization environment. The policy is parameterized by a coordinate vector $\theta \in \mathbb{R}^2$. The agent aims to maximize a scalarized return $J(\theta, w) = w \cdot r_1(\theta) + (1-w) \cdot r_2(\theta)$, where $w \in [0, 1]$ represents the preference weight. We optimize θ using the Adam optimizer across a sweep of discrete weights w. All environments include a small regularization penalty $-0.1\|\theta\|_2^2$. We define three distinct regimes:

Case 1 (Aligned objectives). The reward functions are quadratic penalties centered at proximal locations $\mu_1 = (1, 1)$ and $\mu_2 = (1.2, 1.2)$. This configuration simulates tasks where improving one objective implicitly benefits the other.

Table 3: Hyper-parameters for the routed ensemble procedure used in Algorithm 1.

Parameter	**Meta-World**	**MO-Gymnasium**
Number of basis policies	3	3
Base optimizer	Adam	Adam
SAC buffer size	10^6	10^6
Learning rate (Actor)	3×10^{-4}	3×10^{-4}
Learning rate (Critic)	3×10^{-4}	3×10^{-4}
Discount factor	0.99	0.99
Batch size	1280	256
Network architecture	MLP 256×256	MLP 256×256

Case 2 (Conflicting objectives with a linear frontier). We construct a non-convex reward landscape to induce a disjoint set of optimal policies. The targets are separated at $\mu_1 = (-2, 0)$ and $\mu_2 = (2, 0)$. Crucially, we introduce a Gaussian barrier penalty $P(\theta) = -10 \exp(-\theta_x^2)$ centered at the origin. This barrier prevents the optimization from settling in the intermediate region. Consequently, the optimal deterministic policies collapse to the extreme corners, and the Pareto frontier is formed by the linear convex hull of these solutions.

Case 3 (Conflicting objectives with a concave frontier). We remove the central barrier while maintaining the separated targets $\mu_1 = (-2, 0)$ and $\mu_2 = (2, 0)$. The reward functions are standard negative Euclidean distances. This convex landscape allows for smooth interpolation between objectives. It results in a strictly concave Pareto frontier, in which intermediate policies outperform the linear combination of the corner solutions.

Nonlinear synergy. We introduce an environment for conducting synthetic experiments to demonstrate synergy effects. This environment creates a synergy trap in which two objectives are individually detrimental or neutral but highly beneficial when combined.

We define a 2D parametric optimization task with $M = 12$ distinct objectives, indexed by $j \in \{0, 1, \ldots, M-1\}$. The objectives are arranged equidistantly on a unit circle (a clock face). The goal vector for the j-th objective, $v_j \in \mathbb{R}^2$, is defined by the angle $\phi_j = \frac{\pi}{2} - j \cdot \frac{2\pi}{M}$:

$$v_j = [\cos(\phi_j), \sin(\phi_j)]^\top.$$

Here, objective 0 (north) is the designated target with $v_{\text{target}} = [0, 1]^\top$. The environment employs a specialized reward function $R(a, S)$ for an action $a \in \mathbb{R}^2$ given a subset of active objectives $S \subseteq \{0, \ldots, M-1\}$, represented by a binary mask $m \in \{0, 1\}^M$. The reward consists of a standard linear component and a nonlinear synergy component:

$$R(a, S) = a^\top \left(\sum_{j \in S} v_j \right) + \mathbb{I}_{3 \in S \wedge 9 \in S} \cdot \kappa_{\text{syn}} \cdot (a^\top v_{\text{target}}),$$

where $\kappa_{syn} = 5.0$ is the synergy bonus coefficient.

The agent is modeled as a stochastic policy $\pi_\theta(a) = \mathcal{N}(\mu_\theta, \Sigma)$, where $\mu_\theta = \tanh(\theta)$ and $\Sigma = \sigma^2 I$ (with $\sigma = 0.2$). The parameter $\theta \in \mathbb{R}^2$ is optimized via policy gradient, which optimizes the parameters of a stochastic policy by maximizing expected reward through Monte Carlo sampling, using the Adam optimizer with a learning rate of 0.1 for a horizon of $T_{ep} = 60$ steps.

A tabular environment on a grid. To validate the theoretical interactions between the objective conflict coefficient Γ and the coverability coefficient C_{cov}, we implement a controlled tabular environment. This setup allows for the exact calculation of occupancy measures and χ^2-divergences, avoiding the approximation errors inherent in deep RL settings.

We consider a MOMDP modeled as an $N \times N$ grid (default $N = 5$), defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, Pr, r, \gamma)$. Consistent with the conflicting-objectives regime, we construct m objectives based on spatial navigation to distinct goal coordinates. Let $\mathcal{G} = \{g_1, \ldots, g_m\} \subset \mathcal{S}$ be a set of diverse goal states. The vector-valued reward function $r(s, a) \in \mathbb{R}^m$ is defined using the negative Manhattan

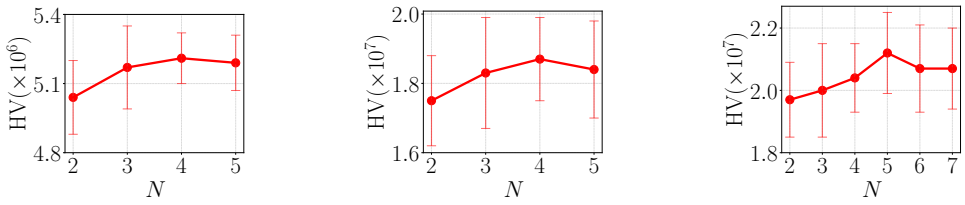


Figure 4: Ablation analysis of the choice of the ensemble size N, measured by hypervolume (HV).

distance (ℓ_1 norm): $r_k(s,a) = -\|\text{coord}(s') - \text{coord}(g_k)\|_1$, where s' is the next state and $\text{coord}(\cdot)$ maps a state index to its (x, y) grid position.

We compute optimal policies exactly via soft value iteration. For a given preference weight $w \in \Delta^{m-1}$, the scalarized reward is $r_w = \langle w, r \rangle$. The policy π_w^* is derived from the soft Bellman operator:

$$V(s) = \tau \log \sum_a \exp\left(\frac{Q(s,a)}{\tau}\right), \quad \pi(a|s) \propto \exp\left(\frac{Q(s,a)}{\tau}\right),$$

with temperature $\tau = 0.5$. The exact state-action occupancy distribution $d^\pi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ is computed by solving the linear system:

$$d^\pi = (I - \gamma P_\pi)^{-1} d_0,$$

where P_π is the policy-induced transition matrix and d_0 is the uniform initial distribution.

We explicitly calculate the conflict coefficient. We define the target mixture distribution $\bar{d}$ as the uniform average of the occupancies of the single-objective experts:

$$\bar{d}(s,a) = \frac{1}{m} \sum_{k=1}^{m} d^{\pi_k^\star}(s,a).$$

The conflict coefficient is then measured as the χ^2-divergence between a candidate policy's occupancy and this target mixture:

$$\Gamma(\pi) = \chi^2(d^\pi \| \bar{d}) = \sum_{s,a} \frac{(d^\pi(s,a) - \bar{d}(s,a))^2}{\bar{d}(s,a) + \epsilon}.$$

We utilize this metric to compare coverability in Section D.2. We find that a single scalarized policy π_{single} exhibits a high χ^2-divergence, indicating poor coverage of the Pareto frontier. In contrast, a routed ensemble with K basis policies significantly reduces this divergence.

D.2 Ablation studies

Ensemble policies reduce coverability. We measure coverability using the union of state-action occupancies in the synthetic environment. A single policy results in a high χ^2-divergence of 1.3. The routed ensembles distribute exploration, reducing the metric to 0.2. This suggests that routing mitigates the exploration bottleneck.

Varying the number of basis policies in the ensemble. We vary N in $\{2, 3, 4, 5\}$ to evaluate the sensitivity of Algorithm 1 on three MuJoCo benchmarks. For MO-HalfCheetah-v4, the Hypervolumes are highest at 5.21×10^6 when $N = 3$. For MO-Hopper, it is 1.87×10^7 when $N = 3$. For MO-Ant, it is 2.12×10^7 when $N = 5$.

Nonlinear kernels improve trajectory attribution. We compare linear, quadratic, cubic, and RBF kernels by measuring the fitting loss of the kernel surrogate model in equation (5) on the synthetic dataset. The linear kernel converges to a loss of 0.2, while the quadratic, cubic, and RBF kernels all achieve a lower loss of 0.1. This gap demonstrates that the relationship between objective subsets and policy returns involves high-order interactions that a linear model cannot capture.

From occupancy conflict to regret reduction. Lastly, we evaluate how occupancy conflict relates to regret reduction via routing in Table 5. We vary the separation between two reward goals. As goal separation increases, measured Γ rises from 0 to 1, while regret reduction from routing rises from 0 to 36.03. Across these settings, routing is more beneficial when preference-induced occupancies diverge. This trend is consistent with Theorem 3.6, where larger occupancy conflict increases the conflict-dependent term in the regret bound.

Table 4: Hypervolume comparison with learned-gate MoE and GPI-LS.

Approach	HalfCheetah2d	Hopper	Ant
MoE	$4.40_{\pm 0.09}$	$1.56_{\pm 0.13}$	$1.60_{\pm 0.18}$
GPI-LS	$5.00_{\pm 0.15}$	$1.50_{\pm 0.12}$	$1.85_{\pm 0.24}$
Alg. 1	$\mathbf{5.17}_{\pm 0.16}$	$\mathbf{1.83}_{\pm 0.14}$	$\mathbf{2.03}_{\pm 0.16}$

Table 5: Measuring conflict and regret reduction under varying goal separation levels.

Separation	Γ	Regret reduction
0	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
2	$0.74_{\pm 0.00}$	$5.77_{\pm 0.11}$
8	$1.00_{\pm 0.00}$	$36.03_{\pm 0.57}$

Table 6: We report the average target reward when the target policy is trained with source tasks selected by each attribution algorithm.

Methods	Multitask Minigrid		Meta-World MT10		Synergy MTRL	
	LDS	Reward	LDS	Reward	LDS	Reward
Influence functions	$-0.23_{\pm 0.08}$	$0.07_{\pm 0.13}$	$0.71_{\pm 0.11}$	$16.3_{\pm 1.4}$	$0.04_{\pm 0.16}$	$1.77_{\pm 0.67}$
TracIn	$0.31_{\pm 0.06}$	$0.25_{\pm 0.21}$	$0.45_{\pm 0.15}$	$16.0_{\pm 2.3}$	$0.16_{\pm 0.04}$	$2.65_{\pm 0.03}$
Linear surrogates	$0.37_{\pm 0.12}$	$0.33_{\pm 0.19}$	$0.42_{\pm 0.19}$	$15.8_{\pm 4.5}$	$0.16_{\pm 0.04}$	$2.67_{\pm 0.01}$
Algorithm 2	$\mathbf{0.41}_{\pm 0.04}$	$\mathbf{0.45}_{\pm \mathbf{0.17}}$	$\mathbf{0.80}_{\pm 0.04}$	$\mathbf{18.8}_{\pm \mathbf{5.6}}$	$\mathbf{0.91}_{\pm \mathbf{0.01}}$	$\mathbf{5.85}_{\pm \mathbf{0.01}}$

D.3 Additional empirical results

Empirical evaluation of the basis assumption. We measure the residual ϵ obtained by approximating preference-policy occupancies with convex combinations of basis-policy occupancies. Tabular occupancies are computed exactly, while benchmark occupancies are estimated from visitation distributions. Table 8 reports the mean and maximum residuals over the evaluated preferences.

In the concave tabular setting, the mean residual decreases from 0.186 at three bases to 0.088, 0.024, and 1.45×10^{-4} at five, nine, and 21 bases, respectively. On MO-Hopper, increasing the candidate-selected basis from three to 15 policies decreases the mean residual from 0.443 to 0.231, whereas the three policies learned by Algorithm 1 have mean residual 1.33. These measurements show that a larger candidate basis can approximate the evaluated occupancies more closely, but the learned ensemble need not recover the exact structure assumed in the analysis.

Severe basis under-specification. In a controlled GridWorld with 21 distinct preference-optimal policies, we restrict the learner to two policies for 3000 episodes. The resulting mean and maximum occupancy residuals are 0.475 and 3.474, and the cumulative value excess is 2780.4. This test measures performance when the available ensemble is substantially smaller than the set of preference-optimal policies.

Comparison on RLHF tasks. We evaluate the extension of our approach to RLHF settings on synthetic data, HelpSteer2 [43], and UltraFeedback [10]. For the synthetic experiment, we construct a synthetic preference dataset that captures the helpfulness-safety trade-off in RLHF. Each prompt is paired with candidate textual responses corresponding to different response styles, while rewards are generated from a controlled two-objective preference function. This allows us to study expert routing under interpretable language scenarios while retaining access to an oracle policy. For the real-data experiments, we evaluate Qwen3-1.7B and Llama-3.2-1B on both HelpSteer2 and UltraFeedback, yielding four model–dataset configurations. All four configurations follow the same training protocol, with per-basis LoRA adapters trained using PPO.

D.4 Proximal policy optimization

In proximal policy optimization (PPO), access to the exact A^π is not available. Instead, one must use an advantage estimator $\widehat{A}_t$ computed from rollout trajectories and a learned value function V_ϕ. A common choice is generalized advantage estimation (GAE), where the temporal-difference residual at time t is

$$\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t),$$

Table 7: Summary of trajectory attribution results on two MO-Gymnasium benchmarks, Deep Sea Treasure and Fruit Tree.

	Deep Sea Treasure	Fruit Tree
Influence functions	$0.28_{\pm 0.51}$	$0.30_{\pm 0.57}$
Linear surrogate models	$0.62_{\pm 0.39}$	$0.71_{\pm 0.53}$
TracIn	$0.70_{\pm 0.40}$	$0.75_{\pm 0.45}$
Algorithm 2	$\mathbf{0.75}_{\pm 0.14}$	$\mathbf{0.81}_{\pm 0.18}$

Table 8: Convex-hull occupancy residuals for exact tabular occupancies and empirical benchmark visitation distributions. Mean and maximum residuals are computed over the evaluated preferences.

Setting	Basis construction	Number of policies	Mean ϵ	Max ϵ
Tabular, aligned	Exact	1	$\sim 10^{-18}$	$\sim 10^{-17}$
Tabular, linear	Exact	2	$\sim 10^{-9}$	$\sim 10^{-8}$
Tabular, concave	Exact	3	0.186	0.55
Tabular, concave	Exact	21	1.45×10^{-4}	6.10×10^{-4}
MO-Hopper	Candidate-selected	3	0.443	0.80
MO-Hopper	Candidate-selected	15	0.231	0.51
MO-Hopper	Learned by Alg. 1	3	1.33	1.42
Deep-Sea-Treasure	Learned by Alg. 1	6	0.15	0.75
Fruit-Tree	Oracle	6	0.553	0.85
GridWorld, 21 distinct optima	Restricted learner	2	0.475	3.474

Table 9: Comparison results for RLHF settings between our approach, using a single policy, and uniform assignment to the same number of basis policies.

	Synthetic	Qwen3-1.7B		Llama-3.2-1B	
Method		HelpSteer2	UltraFeedback	HelpSteer2	UltraFeedback
Single Policy	0.48	0.51	0.46	0.34	0.52
Uniform Assignment	0.49	0.47	0.47	0.35	0.54
Algorithm 1	**0.56**	**0.54**	**0.48**	**0.35**	**0.55**

and the advantage estimate is

$$\widehat{A}_t = \sum_{l=0}^{L-1} (\gamma\lambda)^l \delta_{t+l},$$

for some truncation horizon L, the discount factor $\gamma \in [0, 1)$, and GAE parameter $\lambda \in [0, 1]$. More generally, we use $\widehat{A}_t$ to denote any scalar advantage estimate derived from the rollout and a baseline; our theory treats these $\widehat{A}_t$ as given and is independent of the particular estimator.

Training proceeds in rounds $k = 0, 1, 2, \ldots$. In round k, the current policy $\pi_{\theta^{(k)}}$ interacts with the environment and collects a rollout buffer $B^{(k)} = \{z_i^{(k)}\}_{i=1}^{n_k}$, where each record

$$z_i^{(k)} = (s_i, a_i, r_i, \log \pi_{\theta^{(k)}}(a_i \mid s_i), v_i, \widehat{A}_i)$$

contains the transition (s_i, a_i, r_i) together with auxiliary quantities such as the log-probability under the current policy, a discounted value estimate v_i, and an advantage estimate $\widehat{A}_i$. These records are the atomic units used in PPO updates and form the basic entities for local attribution.

Given $B^{(k)}$, PPO performs a number of gradient steps to update θ from $\theta^{(k)}$ to $\theta^{(k+1)}$ by maximizing a clipped surrogate objective. For a single record $z = (s, a, r, \ldots)$, let $\ell_{\mathrm{PPO}}(\theta; z)$ denote the per-record loss corresponding to the standard clipped PPO objective (possibly combined with value and entropy terms). PPO updates θ to approximately minimize $\hat{L}^{(k)}(\theta)$, starting from $\theta^{(k)}$.

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: Section 1 provides a summary of the contributions and the scope of this paper.

   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: Our theoretical result relied on several structural assumptions, particularly the existence of a finite number of basis policies for the mixture policy and sufficient coverage from the observed trajectory data. We note that the first assumption is conceptually similar to the low-rank condition used in tensor factorizations, a reasonable modeling assumption for multi-objective tabular MDPs. It is possible to relax this assumption to account for noise or small errors, which would propagate to the regret bound in Theorem 3.6. This extension has been added to Appendix C.4. The coverage assumption is inherited from prior work [46] and is a necessary assumption for accurately estimating the unknown MDP. Our empirical evaluation is currently limited to simulated benchmarks such as Meta-World and MO-Gymnasium. The reported RLHF results are promising but are still preliminary. Further improving upon these limitations represents an interesting avenue for future work.

   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: Section 3 and Appendix C provide the assumptions and the complete proof.

   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: Section 4 and Appendix D describe the experiments along with the setup.

   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: Section 1 includes the link to the entire code repository.

   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: Section 4 and Appendix D include detailed description of the experimental setup.

   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: Section 4 includes a detailed comparison where error bars are included in the main comparison table and the illustrative figures.

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: Appendix D discusses computational resources used to conduct each experiment.

   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: This paper presents work aimed at advancing the theory and algorithms of multi-objective reinforcement learning. The advance could potentially lead to improvements in downstream applications where multi-objective RL arises.

   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: [Yes]

    Justification: Due to the technical nature of this paper, there are no negative societal consequences or ethical considerations from this work.

    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: [N/A]

    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: [Yes]

Justification: Section 4 includes the citations to the datasets that we have used in 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: [N/A]

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: [N/A]

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: [N/A]

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: [N/A]

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.