
Multi-Objective Reinforcement Learning Using Routed Ensembles and Trajectory Attribution

Zhenshuo Zhang
Northeastern University
zhang.zhens@northeastern.edu

Ruoxuan Xiong
Emory University
ruoxuan.xiong@emory.edu

Hongyang R. Zhang
Northeastern University
ho.zhang@northeastern.edu

Abstract

Multi-objective reinforcement learning (MORL) is a fundamental problem for modern AI, with applications spanning robotics, LLMs, and reasoning. A central challenge is to learn policies that balance competing objectives, such as speed versus energy in robotics, or helpfulness versus honesty in alignment. Optimal policies can vary sharply with the preferred objective weights, and the severity of this variation depends crucially on the degree of misalignment among objectives, making per-preference learning sample-inefficient. To formalize this, we introduce an *objective-conflict coefficient* and use it to derive near-optimal regret bounds for online MORL that quantify the impact of misalignment on learning efficiency. Motivated by this analysis, we propose *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. We further design a scalable, kernel-based attribution algorithm that estimates how feedback signals from one set of trajectories influence policy learning on others. Unlike existing attribution methods, our approach captures nonlinear interactions, such as synergy and antagonism between trajectories, to explain objective conflict. We extensively evaluate our approach across diverse environments and benchmarks: routed ensembles improve average performance by 4.3% over MORL baselines on several environments from the Meta-World and MO-Gymnasium benchmarks, and kernel-based attribution improves correlation with leave-one-out analysis over existing baselines by 50%. When applied to downstream task selection, kernel-based 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 within the framework of multi-agent RL [28] and ubiquitous in modern AI systems. In robotics, agents often need to balance speed and energy efficiency. In reinforcement learning with human feedback and reasoning, fine-tuning large language models requires navigating competing desiderata such as helpfulness, safety, and honesty [20, 37].

A core challenge in MORL is that the optimal policy can vary dramatically with preference weights: small changes in preference can lead to qualitatively different optimal policies [8]. Prior work has focused on approximating the Pareto frontier of optimal objective trade-offs [43, 49, 52], or learning one policy for each objective [45], and it remains unclear *whether or when* transferring data or policies across preferences is effective.

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

under one preference can benefit policy learning for others. When objectives are misaligned or adversarial (e.g., RLHF with corrupted reward components), transfer can become inefficient or even harmful [20]. This issue is especially acute in high-dimensional settings, making it a critical open challenge to identify objective alignment [22].

To tackle this challenge, we introduce a new measure, the *objective conflict coefficient*, based on the χ^2 -divergence, that quantifies the degree of misalignment among objectives. 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 [19, 46, 5, 7] and is the first to formalize how objective misalignment affects learning efficiency across preferences. Motivated by this analysis, we propose *routed ensemble learning*, a new algorithm that learns a small set of basis policies via two components: (i) an online routing mechanism that directs each trajectory to the basis policy most likely to yield the highest expected return, and (ii) clustering of trajectories into disjoint partitions, with a standard policy learning algorithm such as proximal policy optimization applied to each cluster.

We further introduce a scalable *kernel-based trajectory attribution* procedure that estimates how trajectories collected under one preference weight affect policy learning on others [23]. By leveraging kernel features, this procedure captures not only additive effects but also second-order interactions (e.g., synergy and antagonism) between trajectories. These effects are common in high-dimensional settings and often underlie unexpected transfer behavior [22], but they are not captured by first-order approaches such as influence functions [24], surrogate models [26], or gradient-similarity based analysis [50, 15]. Our approach goes beyond these methods by leveraging efficient approximations grounded in policy gradients and random projections.

Lastly, we empirically validate our approach on the Meta-World robotics benchmark and high-dimensional MO-Gymnasium tasks. Our method 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. Additionally, kernel-based attribution provides a 50% improvement in alignment with ground-truth influence scores. When applied to downstream task selection, kernel-based selection yields a 27% higher reward than existing attribution baselines. Our ablation study confirms the theoretical motivation: the coverability metric drops from 1.3 for a single policy to 0.2 for the ensemble, indicating that routing mitigates the exploration bottleneck caused by objective conflicts.

In summary, our contributions are threefold. First, we introduce the objective conflict coefficient and use it to guide the design of routed ensembles. Second, we develop a kernel-based attribution algorithm to interpret inter-trajectory influence, including nonlinear effects. Third, we demonstrate significant gains in sample efficiency and robustness across diverse MORL benchmarks.

2 Preliminaries

We consider an environment modeled as a *multi-objective Markov decision process* (MOMDP), defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \Pr(\cdot | \cdot, \cdot), \mathbf{r}, \Omega, f_\Omega, H)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $\Pr(\cdot | s, a)$ is the transition kernel, $\mathbf{r} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^m$ is a vector-valued reward function with $m \geq 2$ objectives, and H is the horizon. The preference space is $\Omega \subseteq \mathbb{R}_+^m$. 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 $\tau = (s_0, a_0, \mathbf{r}_0, \dots, s_H, a_H, \mathbf{r}_H)$, where $a_h \sim \pi(\cdot | s_h)$ and $s_{h+1} \sim \Pr[\cdot | s_h, a_h]$. More generally, the policy can be time dependent $\pi_1, \dots, \pi_H$ with $a_h \sim \pi_h(\cdot | s_h)$ 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

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

We primarily focus on the linear scalarization operator f_Ω , and thus the policy’s scalarized expected return is $\mathcal{J}(\pi; w) = \langle w, \mathbf{J}(\pi) \rangle$. We consider an episodic online MORL setting with episodes $t = 1, \dots, T$. At the beginning of episode t , the learner has access to the history $\mathcal{D}^{(t-1)} =$

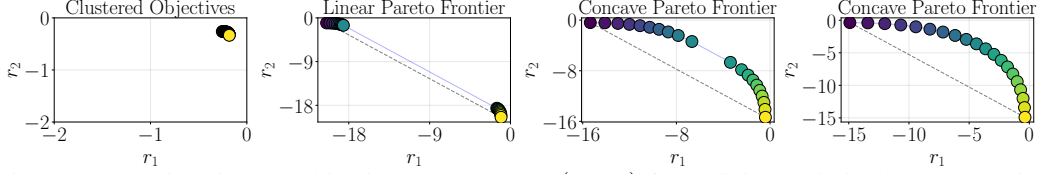


Figure 1: We visualize the objective space returns (r_1, r_2) for policies optimized under varying preference weights $w \in [0, 1]$ in three canonical regimes. Light yellow indicates that w is closer to 1, while dark blue indicates w is closer to 0.

$\{(\tau^{(u)}, w_u)\}_{u=1}^{t-1}$ and observes the current preference vector $w_t \in \Omega$. Based on $(\mathcal{D}^{(t-1)}, w_t)$, the learner applies a (possibly randomized) policy learning algorithm A to select a policy $\pi^{(t)}$, interacts with the MOMDP for horizon H , and observes a trajectory $\tau^{(t)} = (s_0^{(t)}, a_0^{(t)}, \mathbf{r}_0^{(t)}, \dots, s_H^{(t)}, a_H^{(t)}, \mathbf{r}_H^{(t)})$.

After T episodes, the learner has the dataset $\mathcal{D} = \{(\tau^{(t)}, w_t)\}_{t=1}^T$ and aims to learn a preference-conditioned policy $\pi(\cdot; w)$ that attains high utility for target preferences w in the support of $\mathcal{P}$.

Our analysis uses *coverability*, a RL complexity measure for a policy class Π [46]:

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

where $d_h^\pi(s, a) = \Pr^\pi(s_h = s, a_h = a)$ 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 use examples to illustrate how objective misalignment affects policy learning. We then formalize this observation through the objective conflict coefficient and show how it interacts with coverability to affect online regret. These theoretical insights motivate our main algorithmic contribution: routed ensemble policy learning, a scalable algorithm that learns a small set of basis policies and routes trajectories to the policies that are most useful for them. To further interpret objective conflict and routing behavior, we develop a kernel-surrogate trajectory attribution method that quantifies how individual trajectories influence policy learning across preference weights.

3.1 Examples and motivation

We are motivated by two common MOMDP use cases.

Example 3.1 (Pareto frontier learning). *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 .*

Importantly, learning in MORL depends on how objectives relate to one another. In the two-objective case ($m = 2$), there are three canonical regimes (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

We study how objective conflict impacts learning efficiency in MORL. Concretely, suppose 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. The severity of conflict is captured by 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, $\bar{d}_h(s, a) := \frac{1}{K} \sum_{k=1}^K d_h^{\pi_{bk}^*}(s, a)$. The objective conflict coefficient is*

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

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:

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

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 $\{\pi_{b1}^*, \dots, \pi_{bK}^*\}$ 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 $d_h^{\pi_w^*}(s, a) = \sum_{k=1}^K \nu_w(k) d_h^{\pi_{bk}^*}(s, a)$ 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 $d_h^{\pi^{(t)}}(s, a) = \sum_{k=1}^K \nu^{(t)}(k) d_h^{\pi_{bk}^*}(s, a)$ for all (s, a) and H . Letting $\bar{\nu}^{(t)}(k) = \frac{1}{t-1} \sum_{i=1}^{t-1} \nu^{(i)}(k)$, we assume there exist $c_0 \in (0, 1]$ and a burn-in time t_0 such that $\min_{k \in [K]} \bar{\nu}^{(t)}(k) \geq c_0 / (C_{\text{cov}} K)$ 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 on learning difficulty. 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 of this algorithm is provided in Appendix C.1. The resulting regret bound is stated below.

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 joint confidence-set radius after a union bound across the K basis classes. If the burn-in time satisfies $t_0 = \mathcal{O}(C_{\text{cov}} K \beta)$, then with probability at least $1 - \delta$, the cumulative regret of the pooled-confidence routed-ensemble algorithm satisfies*

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

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 $\Omega_T(H\sqrt{(1+\Gamma)T})$, establishing the tightness of the dependence on Γ and T up to constant (C_{cov}) and logarithmic factors.

Relaxing the assumptions. 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 Assumption 3.4 is relaxed, the resulting regret bound preserves the same qualitative dependence on K and Γ . The experiments in Section 4 and Appendix B further demonstrate that these insights extend beyond the theoretical setting.

3.3 Learning routed ensemble policies

In Sections 3.1 and 3.2, the basis policies are assumed latent. We now introduce a scalable *routed ensemble policy learning* algorithm to (i) explicitly learn a set of N basis policies, and (ii) learn a preference-conditioned transfer operator $f_w : \Pi^N \rightarrow \Pi$ that maps these basis policies to a composite policy for any given preference w . Here, N may or may not be the same as K .

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 (3) 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 [45]). (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:

$$\max_{\pi_1, \pi_2, \dots, \pi_N} \mathbb{E}_{s_0 \sim \rho, w \sim \mathcal{P}} \left[\left\langle w, \sum_{h=0}^H \mathbf{r}(s_h, f_w) \right\rangle \right], \quad (2)$$

which maximizes the expected scalarized return $\mathcal{J}(\pi; w)$ of the composed policy $f_w(\pi_1, \dots, \pi_N)$ 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 3.7. Fix a set of N policies $\{\pi_1, \dots, \pi_N\}$. Suppose $f_w \in \text{conv}\{\pi_1, \dots, \pi_N\}$ 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 $\hat{f}$ for Equation (2) such that, for every $w \in \Omega$, $\hat{f}_w = \pi_j$, where $j \in \arg \max_{i \in \{1, 2, \dots, N\}} \langle w, J(\pi_i) \rangle$.

Proof. For any w , since f_w lies in the convex hull of $\pi_1, \dots, \pi_N$, 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

$$J_\alpha(f_w) = \sum_h \sum_{a_h, s_h} d_h^{f_w}(s_h, a_h) = \sum_h \sum_{a_h, s_h} \sum_{i=1}^N \alpha_i(w) d_h^{\pi_i}(s_h, a_h) = \sum_{i=1}^N \alpha_i(w) \langle J(\pi_i), w \rangle. \quad (3)$$

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 $\langle J(\pi_i), w \rangle$.

The lemma directly motivates the scoring and routing steps: we compute scores using Equation (3) 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 [45]. Moreover, Prop. C.8 in App. C.3 connects this algorithm to Γ , showing that the intuition underlying Theorem 3.6 carries over to its performance.

Algorithm 1 Routed Ensemble Policies for MOMDP

Input: A MOMDP $\mathcal{M}$; Initial state distribution ρ ; Preference distribution $\mathcal{P}$; Number of policies N

Require: A policy learning algorithm A ; A policy evaluation function J ; Number of iterations T ; Horizon H ; Number of observed trajectories B per iteration

Output: A set of N policies

```
1:  $\pi_1^{(0)}, \pi_2^{(0)}, \dots, \pi_N^{(0)} : \mathcal{S} \rightarrow \Delta(\mathcal{A})$  //  $N$  initializations
2: for  $t = 1, \dots, T$  do
3:    $w_t \sim \mathcal{P}, \mathcal{D}_t \leftarrow \emptyset, S \leftarrow N$  by  $B$  score matrix
4:   for  $i = 1, \dots, B$  do
5:      $j_i \sim$  uniformly from  $\{1, 2, \dots, N\}, s_0 \sim \rho$ 
6:      $\tau_i^{(t)} \leftarrow (s_0, a_0, \dots, s_H, a_H) \sim$  simulate from  $s_0$  for  $H$  steps using  $\pi_{j_i}^{(t-1)}$ 
7:     for  $j = 1, \dots, N$  do
8:        $S_{j,i} \leftarrow \sum_{h=0}^H \left\langle w_t, \sum_{a \in \mathcal{A}} \mathbf{r}(s_h, a) \pi_j^{(t-1)}(s_h, a) \right\rangle$ 
9:     end for
10:     $\mathcal{D}_t \leftarrow \mathcal{D}_t \cup \{(\tau_i^{(t)}, w_t)\}$ 
11:  end for
12:  for  $j = 1, \dots, N$  do
13:     $\mathcal{D}_{t,j} \leftarrow \left\{ (\tau_i^{(t)}, w_t) \in \mathcal{D}_t \mid j \in \arg \max_{1 \leq j' \leq N} S_{j',i} \right\}; \pi_j^{(t)} \leftarrow A(\pi_j^{(t-1)}, \mathcal{D}_{t,j})$ 
14:  end for
15: end for
16: return  $\pi_1^{(T)}, \pi_2^{(T)}, \dots, \pi_N^{(T)}$ 
```

Finally, while our algorithm assumes N is given, in practice it can be selected via cross-validation (Section 4.3). One may adaptively grow N by identifying new clusters when existing policies do not explain incoming trajectories, drawing connections to the clustering literature [6, 27].

3.4 Trajectory and objective-level attribution analysis

We next quantify how individual training trajectories contribute to policy performance. This complements our analysis by providing a data-driven measure of which trajectories help or hinder learning and by improving the interpretability of routed ensemble policies. A natural approach is leave-one-out (LOO) retraining, which measures the change in performance when trajectory t is excluded:

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

where, for any subset indicator $S \in \{0, 1\}^T$, we define $\mathcal{V}(S) := \mathbb{E}_{w \sim \mathcal{P}} [\mathcal{J}(\pi_S; w)]$, $\pi_S := A(\mathcal{D}_S)$, 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 A , $[\mathbf{1}]_T$ 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, 34]. 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 that approximates the subset-performance function $\mathcal{V}(\cdot)$. The method proceeds in three steps. First, we randomly sample n trajectory subsets $S_1, \dots, S_n$, 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 $f(S_i) \approx \mathcal{V}(S_i)$. 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 $F = [\hat{f}(S_1), \dots, \hat{f}(S_n)]^\top$, and compute $\theta = (\Psi + \lambda I_n)^{-1} F$. This yields the predictor $g_\theta(S) = \sum_{i=1}^n \theta_i \psi(S_i, S)$ for $\mathcal{V}(S)$. Because this kernel model $g_\theta(S)$ is nonlinear in the subset indicator, it can capture higher-order interactions such as synergy and antagonism among trajectories.

In online RL, exact retraining for each sampled subset is often impractical because the training data evolve over time. We therefore estimate the subset performance $f(S_i)$ locally around a reference policy θ^* 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.

Algorithm 2 Kernel Surrogates for Trajectory Attribution using Policy Gradient Estimation

Input: m RL objectives

Requires: Subset sampling distribution $\mathcal{D}$, number of subsets n , ℓ_2 -regularization parameter λ

Output: A kernel surrogate model

- 1: $\theta^* \leftarrow$ Train a meta policy with m objectives
 - 2: $S_1, \dots, S_n \leftarrow$ Sample n subsets from $\mathcal{D}$
 - 3: $\hat{f}(S_1), \hat{f}(S_2), \dots, \hat{f}(S_n) \leftarrow$ Apply POLICYGRADEX with $(S_1, S_2, \dots, S_n; \theta^*)$
 - 4: $\Psi_{i,j} \leftarrow \exp(-\gamma \|S_i - S_j\|^2)$, for all $1 \leq i, j \leq n$; $F \leftarrow [\hat{f}(S_1), \dots, \hat{f}(S_n)]^\top$
 - 5: $\theta \leftarrow (\Psi + \lambda \cdot \text{Id}_n)^{-1} F$
 - 6: **return** $g_\theta(S) = \sum_{i=1}^n \theta_i \psi(S_i, S)$
-

The experimental results support this approximation. Figure 3 shows that the fitted kernel model $g_\theta([\mathbf{1}]_T - e_t)$ is strongly positively correlated with the LOO scores $\mathcal{I}_t^{\text{LOO}}$ 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 A.2.

4 Experiments

Our experiments serve two goals: to verify the theoretical insights from the regret analysis and to evaluate the empirical performance of our algorithms across multi-objective RL settings, including synthetic environments, continuous robotic control, and RLHF benchmarks. 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://anonymous.4open.science/r/RoutedEnsemble/>.

4.1 Evaluation in synthetic environments

We validate our theoretical insights in the following two environments (detailed in Appendix B.4):

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.

For Pareto frontier learning, Figure 2a shows that routed ensemble learning substantially reduces cumulative regret relative to the single-policy baseline MO-UCBVI [45]. 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 tasks and LLMs

We evaluate on several robotic and LLM 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 [51]; (ii) *explicit Pareto*

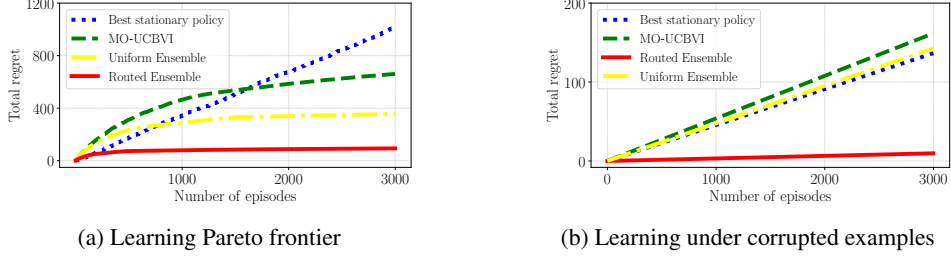


Figure 2: We compare the cumulative regret of routed ensembles (red) against MO-UCBVI (green dashed), uniform ensembles (yellow), and the best stationary policy (blue dotted).

trade-offs on the continuous MO-HalfCheetah/Hopper/Ant and discrete Deep-Sea-Treasure/Fruit-Tree environments of MO-Gymnasium [10]; (iii) *preference-induced conflict* on HelpSteer2 [44], 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 (Appendix B.4). In all regimes, routing is applied as a drop-in module on top of the published base learner.

Baselines. We compare against standard baselines in each regime: (i) for multi-task RL, Soft Modularization [48], PaCo [39], CARE [38]; (ii) For continuous multi-objective RL, PGMORL [47], Q-Pensieve [17], and CAPQL [30]; for discrete multi-objective RL, PQL [43], MPMOQL [43], and Envelop [49]; (iii) For RLHF, a single-policy model; (iv) For trajectory attribution, Influence Functions [24], TracIn [35] adapted to online RL setting [15], and TRAK [34]. For the first three regimes, we also include *Uniform Assignment*, which randomly assigns trajectories to basis policies.

Evaluation. We use standard metrics for each regime: (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) induced-frontier HV for RLHF; (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 by each method.

Results. Our routing algorithm (Algorithm 1) consistently outperforms baselines in multi-task and multi-objective settings. On MT10, it achieves an average success rate of 85.6%, outperforming the Soft Modularization baseline (82.0%) by a relative margin of 4.3% (Table 1). In continuous MO-Gymnasium environments, routing yields an 8% average improvement over Q-Pensieve. In discrete environments, it yields an average 48% improvement over PQL. 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.

Our routing algorithm also improves RLHF performance on both the synthetic environment and HelpSteer2 (Table 5). It 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. On HelpSteer2, the corresponding improvements are 6% and 15%.

Our trajectory attribution method (Algorithm 2) achieves stronger LDS scores and downstream task-selection performance (Table 2). The kernel surrogate improves LDS by 50% over attribution baselines across Multitask-Minigrid and MT10. 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.

4.3 Ablation Studies

Ensemble policies reduce the coverability coefficient. We measure coverability using the union of state-action occupancies in the synthetic environment. A single policy results in a high coverability metric of 1.3 (measured by $C_{\text{cov}}(1 + \Gamma)$). The routed ensembles distribute exploration, reducing the metric to 0.2. This confirms that routing mitigates the exploration bottleneck.

Performance is robust to the number of basis policies. 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

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 Card on Grid-World.

	Metrics	Soft Modularization	PaCo	CARE	Uniform Assignment	Alg. 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%	85.6 $\pm$ 1.8%
	Metrics	PGMORL	Q-Pensieve	CAPQL	Uniform Assignment	Alg. 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	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	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	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	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	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	1.82 $\pm$ 0.12
	Metrics	PQL	MPMOQL	Envelop	Uniform Assignment	Alg. 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	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	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	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	6.40 $\pm$ 0.00

Table 2: We report the average target reward when the target policy is trained with source tasks selected by each attribution algorithm.

Methods	Multitask Minigrid		MT 10		Synergy MTRL (Synthetic)	
	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
TRAK	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
Alg. 2	0.41 $\pm$ 0.04	0.45 $\pm$ 0.17	0.80 $\pm$ 0.04	18.8 $\pm$ 5.6	0.91 $\pm$ 0.01	5.85 $\pm$ 0.01

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 $N = 5$. See Appendix B.5 for the illustration of these ablation results.

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 (4) 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.

5 Related Work

A reinforcement learning environment is typically formalized by means of a Markov decision process [40, 1]. 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 [11, 32, 31, 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, 43, 45, 29]. Recent work [52] propose 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.*

Recent work studies both offline and online RL with linear function approximation [12, 13]. 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.

6 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 objective-attribution method that captures widespread nonlinear interactions and serves as a data-driven approximation of objective conflict. The experiment results demonstrate that our approach achieves superior sample efficiency and attribution accuracy compared to state-of-the-art baselines across diverse benchmarks.

References

- [1] 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.
- [2] M. Bateni, V. Cohen-Addad, Y. Gu, S. Lattanzi, S. Meierhans, and C. Mohri. “Algorithmic Thinking Theory”. In: *arXiv preprint arXiv:2512.04923* (2025).
- [3] 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.
- [4] 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).
- [5] 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).
- [6] 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).
- [7] 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.
- [8] I. Das and J. E. Dennis. “A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems”. In: *Structural optimization* 14.1 (1997), pp. 63–69.
- [9] S. V. Deshmukh, A. Dasgupta, B. Krishnamurthy, N. Jiang, C. Agarwal, G. Theodorou, and J. Subramanian. “Explaining RL Decisions with Trajectories”. In: *The Eleventh International Conference on Learning Representations*. 2023.
- [10] 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.
- [11] Z. Gábor, Z. Kalmár, and C. Szepesvári. “Multi-criteria reinforcement learning.” In: *ICML*. Vol. 98. 1998, pp. 197–205.
- [12] 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.
- [13] N. Golowich and A. Moitra. “The role of inherent bellman error in offline reinforcement learning with linear function approximation”. In: *Reinforcement Learning Conference*. 2024.
- [14] J. Grigsby, J. Sasek, S. Parajuli, D. Adeb, A. Zhang, and Y. Zhu. “Amago-2: Breaking the multi-task barrier in meta-reinforcement learning with transformers”. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 87473–87508.
- [15] 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.
- [16] Y. Huang and L. F. Yang. “Winning gold at imo 2025 with a model-agnostic verification-and-refinement pipeline”. In: *arXiv preprint arXiv:2507.15855* (2025).
- [17] 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).
- [18] A. Ilyas, S. M. Park, L. Engstrom, G. Leclerc, and A. Madry. “Datamodels: Predicting Predictions from Training Data”. In: *Proceedings of the 39th International Conference on Machine Learning*. 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] W. B. Johnson. “Extensions of Lipschitz mapping into Hilbert space”. In: *Conference modern analysis and probability, 1984*. 1984, pp. 189–206.
- [22] 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.
- [23] S. M. Kakade. “A natural policy gradient”. In: *Advances in neural information processing systems* 14 (2001).
- [24] P. W. Koh and P. Liang. “Understanding black-box predictions via influence functions”. In: *International conference on machine learning*. PMLR. 2017, pp. 1885–1894.
- [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] 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.
- [28] M. L. Littman. “Markov games as a framework for multi-agent reinforcement learning”. In: *Machine learning proceedings 1994*. Elsevier, 1994, pp. 157–163.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] K. Miettinen and M. M. Mäkelä. “On scalarizing functions in multiobjective optimization”. In: *OR spectrum* 24.2 (2002), pp. 193–213.
- [33] 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.
- [34] S. M. Park, K. Georgiev, A. Ilyas, G. Leclerc, and A. Madry. “TRAK: Attributing Model Behavior at Scale”. In: *ICML*. 2023.
- [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] B. Schölkopf, R. Herbrich, and A. J. Smola. “A generalized representer theorem”. In: *International conference on computational learning theory*. Springer. 2001, pp. 416–426.
- [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] S. Sun, H. Shi, Z. Zhao, X. Wang, and J. Zhou. “PaCo: Parameter-compositional multi-task reinforcement learning”. In: *International Conference on Learning Representations*. 2022.
- [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] P. Vamplew, J. Yearwood, R. Dazeley, and A. Berry. “On the limitations of scalarisation for multi-objective reinforcement learning of pareto fronts”. In: *Australasian joint conference on artificial intelligence*. Springer. 2008, pp. 372–378.
- [43] 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.
- [44] 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.
- [45] 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.
- [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] 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.
- [48] 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.
- [49] R. Yang, X. Sun, and K. Narasimhan. “A generalized algorithm for multi-objective reinforcement learning and policy adaptation”. In: *Advances in neural information processing systems* 32 (2019).
- [50] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn. “Gradient surgery for multi-task learning”. In: *Advances in neural information processing systems* 33 (2020), pp. 5824–5836.
- [51] 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.
- [52] Z. Zhang, M. Duan, Y. Ye, and H. R. Zhang. “Scalable Multi-Objective and Meta Reinforcement Learning via Gradient Estimation”. In: *AAAI* (2026).

A Omitted Algorithm Description

A.1 Proximal policy optimization

In practice, proximal policy optimization (PPO) does not have access to the exact A^π and instead uses an advantage estimator $\hat{A}_t$ 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

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

and the advantage estimate is

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

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

Training proceeds in rounds $k = 0, 1, 2, \dots$. 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

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

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 $\hat{A}_i$. These records are the atomic units used in PPO updates and form the basic entities for local attribution.

Given a buffer $B^{(k)}$, 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 $z = (s, a, r, \dots)$, let $\ell_{\text{PPO}}(\theta; z)$ denote the per-record loss corresponding to the standard clipped PPO objective (possibly combined with value and entropy terms). The empirical PPO loss over the buffer is then

$$\hat{L}^{(k)}(\theta) = \frac{1}{n_k} \sum_{i=1}^{n_k} \ell_{\text{PPO}}(\theta; z_i^{(k)}),$$

and PPO updates θ to approximately minimize $\hat{L}^{(k)}(\theta)$, starting from $\theta^{(k)}$.

A.2 Policy gradient estimation

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

$$\min_{g_\theta \in \mathcal{H}} \sum_{i=1}^n (f(S_i) - g_\theta(S_i))^2 + \lambda \|g_\theta\|_\Psi^2, \quad (4)$$

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: $\psi(S_i, S_j) = \exp(-\gamma \|S_i - S_j\|^2)$. By the representer theorem [36], the minimizer is:

$$g_\theta(S) = \sum_{i=1}^n \theta_i \psi(S_i, S),$$

where the coefficients θ are obtained via $(\Psi + \lambda \text{Id})^{-1} F$ with $F = [f(S_1), \dots, f(S_n)]^\top$, and Id is the identity matrix.

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

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

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

Table 3: 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%

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 $g_t^\top \Delta\theta$ with the sign of the advantage $\hat{A}_t$. By defining a binary label $y_t = \text{sign}(\hat{A}_t)$ and a sample weight $\eta_t = |\hat{A}_t|$, we maximize the margin $y_t g_t^\top \Delta\theta$. This is effectively approximated by minimizing the weighted logistic loss, which provides a bounded surrogate for the sign-agreement objective:

$$f(S_i) := \min_{\Delta\theta} \frac{1}{|\mathcal{D}_{S_i}|} \sum_{t \in \mathcal{D}_{S_i}} \eta_t \cdot \log(1 + \exp(-y_t \tilde{g}_t^\top \Delta\theta)).$$

This formulation enables rapid estimation of the policy performance for any subset S_i without full retraining. As shown in Table 3 (Appendix A.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 g_t to a low-dimensional space. By the Johnson–Lindenstrauss Lemma [21], 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. The estimation procedure is described in Algorithm 3 (Appendix A.2), and the attribution procedure is outlined in Algorithm 2.

Numerical results. To assess the accuracy of gradient-based estimation in Algorithm 3, we compute the relative residual sum of squares (RSS) error across several RL environments: MT10 from MetaWorld [51], 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 3 (Appendix A.2), the RSS error remains below 2% when $\|\theta - \theta^*\|$ is small with a proper initialization θ^* , indicating high accuracy in the linear approximation. The error gradually increases to around 10% for perturbations up to 5%. These results validate the surrogate’s ability to model the impact of objective subsets 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 $\mathcal{N}(0, d^{-1})$. We project the high-dimensional gradients to a d -dimensional space as

$$\tilde{g}_{i,t} = P^\top g_{i,t}.$$

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

$$\hat{\theta}_d \leftarrow \arg \min_{\theta_d \in \mathbb{R}^d} \frac{1}{|\mathcal{D}_{S_j}|} \sum_{(g,y,w) \in \mathcal{D}_{S_j}} \ell(\tilde{g}, y, w; \theta_d), \quad (5)$$

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

$$\hat{\theta}^{(j)} = \theta^* + P \hat{\theta}_d.$$

Based on the Johnson-Lindenstrauss lemma [21], when $d = O\left(\frac{\log p}{\epsilon^2}\right)$, for any g_i with $\|g_i\| \leq G$ and any θ, θ_0 with ℓ_2 -norm at most D , we have

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

Algorithm 3 Policy gradient estimation (POLICYGRADEX)

Input: n random subsets from $\{1, 2, \dots, m\}$, $S_1, S_2, \dots, S_n$ **Require:** Initial meta-policy parameter $\theta^* \in \mathbb{R}^p$; Number of episodes T ; Projection dimension d

```
1:  $\pi_{\theta^*} \leftarrow$  Setup the meta-policy for all tasks in  $\{1, 2, \dots, m\}$ 
2:  $S_1, S_2, \dots, S_m \leftarrow m$  random subsets of  $\{1, 2, \dots, n\}$  with size  $\alpha$ 
3:  $P \leftarrow$  A  $p$  by  $d$  isotropic Gaussian random projection matrix
4: for  $t = 1, \dots, T$  do
5:   for  $i \in \{1, 2, \dots, m\}$  do
6:      $g_{i,t} \leftarrow P^\top \nabla \pi_{\theta^*}(a_t | s_t)$ 
7:      $\eta_{i,t} \leftarrow |\hat{A}_t|$ 
8:      $y_{i,t} \leftarrow \text{sign}(\hat{A}_t)$ 
9:   end for
10: end for
11: for  $j = 1, \dots, n$  do
12:    $\Delta \hat{\theta}_d \leftarrow \arg \min_{\Delta \theta \in \mathbb{R}^d} \hat{L}_{S_j}(\theta^* + P \Delta \theta)$ 
13:    $\hat{\theta}^{(j)} \leftarrow \theta^* + P \Delta \hat{\theta}_d$ 
14:    $\hat{f}(S_j) \leftarrow -\hat{L}_{S_j}(\hat{\theta}^{(j)})$ 
15: end for
16: Return  $\hat{f}(S_1), \dots, \hat{f}(S_n)$ 
```

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

Remark A.1 (Extensions). *We describe several natural extensions of our approach. First, our algorithms naturally extend to online multitask RL [14], 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 A.2. *Our problem formulation relates to multi-objective optimization in prior literature [33, 4], and we extend this line of work to the online learning setting. Further exploring the connection between ensemble learning and online boosting [3] is another direction for future work.*

Our work connects to the literature on data attribution, which aims to trace model behavior back to influential training data. Prior work include influence functions [24], TracIn [35] and linear surrogate models [18, 26]. In reinforcement learning, recent efforts have adapted these techniques to explain policy decisions, including trajectory-level attribution in offline RL [9] and snapshot influence functions for filtering harmful data in online RL [15]. 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 [2, 16].

B Omitted Experiments

B.1 Computing resources

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 were performed on a server with an Intel Xeon E5-2623 v3 CPU and a single NVIDIA RTX A6000 GPU. The Meta-World training takes 12 hours, and the RLHF experiment takes 1 hour.

Table 4: Hyper-parameters for the routed ensemble procedure (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

B.2 Metrics

We introduce the definition of the 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

$$\bar{d}^\pi(s) = \frac{1}{H} \sum_{t=0}^H P(s_t = s).$$

Then,

$$\Gamma = \max_k \sum_{s \in \mathcal{S}} \frac{(\bar{d}^{\pi_k}(s) - \bar{d}^{\text{avg}}(s))^2}{\bar{d}^{\text{avg}}(s)},$$

where π_k is the policy trained on objective k alone, and $\bar{d}^{\text{avg}}$ 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:

$$\text{Attr}(j) = \hat{f}(\mathbf{m}_{\text{target}} \cup \{j\}) - \hat{f}(\mathbf{m}_{\text{target}}),$$

where $\mathbf{m}_{\text{target}}$ is the mask containing only the target objective.

B.3 Hyperparameters

We provide the detailed hyperparameter settings used in our experiments to ensure reproducibility. Table 4 lists the parameters for the Routed Ensemble algorithm across different benchmarks. For the Routed Ensemble, we adapt the base policy-learning algorithm for each environment. We use Soft Modularization [48] for Meta-World, Q-Pensieve [17] for continuous MO-Gymnasium tasks, and PQL [43] 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. The specific settings are summarized below.

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

B.4 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.

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 $j \in \{0, 1, \dots, M-1\}$. 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 $S \subseteq \{0, \dots, M-1\}$, 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 $\kappa_{\text{syn}} = 5.0$ 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 that 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 $T_{\text{ep}} = 60$ 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 $\mathcal{G} = \{g_1, \dots, g_m\} \subset \mathcal{S}$ 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 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.

Here, 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.

Table 5: Comparison results for RLHF evaluation on one synthetic environment and another real-world dataset [44].

	Synthetic	HelpSteer2
Single Policy	0.48	0.51
Uniform Assignment	0.49	0.47
Algorithm 1	0.56	0.54

Table 6: 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 ± 0.51	0.30 ± 0.57
Trak	0.62 ± 0.39	0.71 ± 0.53
TracIn	0.70 ± 0.40	0.75 ± 0.45
Algorithm 2	0.75 ± 0.14	0.81 ± 0.18

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:

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

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 4.3. 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.

B.5 Omitted results

Comparison on RLHF tasks. We evaluate the extension of our approach to RLHF settings on both synthetic data and HelpSteer2 [44]. 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. On HelpSteer2, each prompt is paired with multiple candidate responses annotated along several preference dimensions. We implement the policy as a LoRA adapter on a pretrained language model, Qwen-3-1.7B.

In Table 5, we provide the comparison between Algorithm 1, uniform assignment, and applying just a single policy.

Correlation between attribution scores and LOO analysis. 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 across different tests.

Trajectory attribution results. 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.

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.

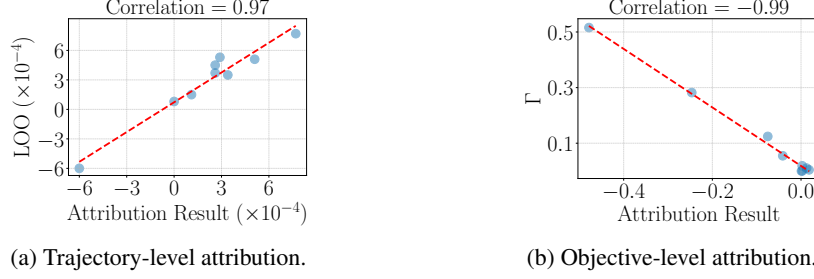


Figure 3: Correlation analysis of attribution scores against LOO scores (left) and the objective conflict coefficient Γ (right).

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.

Choice of N . In Figure 4, we illustrate the choice of N in the three datasets, including MO-HalfCheetah2d, MO-Hopper, and MO-Ant environments.

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 the relaxation when only an approximate low-rank basis assumption holds.

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 $t_0 = \mathcal{O}(C_{\text{cov}}K\beta)$.

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

$$(k_t, f^{(t)}) \in \arg \max_{k \in [K], f \in \mathcal{F}_k^{(t-1)}(\beta)} V_1^f(s_0; w_t), \quad V_h^f(s; w) := \max_a f_h(s, a; w). \quad (6)$$

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 $\bar{d}_h := K^{-1} \sum_{k=1}^K d_h^{\pi_{bk}^*}$ and the un-normalized cumulative empirical occupancy $\bar{d}_h^{(t)} := \sum_{i=1}^{t-1} d_h^{(i)}$.

Lastly, note that the χ^2 -divergence between two distributions $P(x)$ and $Q(x)$ is defined as $\chi^2(P||Q) = \sum_{x: Q(x)>0} \frac{(P(x)-Q(x))^2}{Q(x)}$.

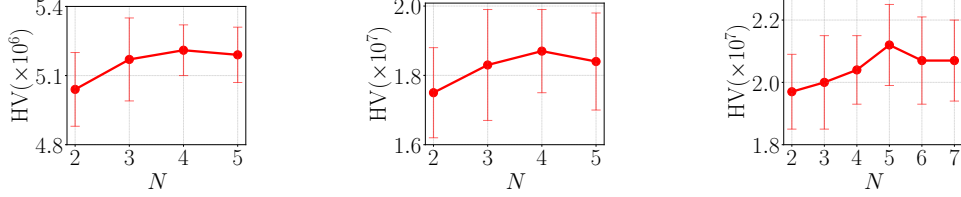


Figure 4: Ablation analysis of the choice of the ensemble size N , measured by hypervolume (HV).

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 $Q^{\pi_{\text{bk}}^*}(\cdot; w)$ 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 ,*

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

Proof. Assumption 3.4 expresses the occupancy of π_w^* as a convex combination of the basis occupancies. Since the scalarized value is linear in the occupancy measure,

$$\mathcal{J}(\pi_w^*; w) = \sum_k \nu_w(k) \mathcal{J}(\pi_{\text{bk}}^*; w).$$

By Lemma 3.7, there exists $k^*(w_t) \in [K]$ such that the optimal MORL policy and the optimal basis policy at index $k^*(w_t)$ have the same scalarized return: $\mathcal{J}(\pi_{w_t}^*; w_t) = \mathcal{J}(\pi_{\text{bk}^*(w_t)}^*; w_t)$.

By event E_{opt} , the corresponding true Q-function $Q^{\pi_{\text{bk}^*(w_t)}^*}(\cdot; w_t)$ lies in $\mathcal{F}_{k^*(w_t)}^{(t-1)}(\beta)$. Restricting the joint argmax in (6) to $k = k^*(w_t)$ then yields

$$\begin{aligned} V_1^{f^{(t)}}(s_0; w_t) &\geq \max_{f \in \mathcal{F}_{k^*(w_t)}^{(t-1)}(\beta)} V_1^f(s_0; w_t) \geq V_1^{Q^{\pi_{\text{bk}^*(w_t)}^*}}(s_0; w_t) \\ &= \mathcal{J}(\pi_{\text{bk}^*(w_t)}^*; w_t) = \mathcal{J}(\pi_{w_t}^*; w_t). \end{aligned}$$

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

$$\begin{aligned} \text{Reg}(T) &\leq \sum_{t=1}^T \left[V_1^{f^{(t)}}(s_0; w_t) - \mathcal{J}(\pi^{(t)}; w_t) \right] \\ &= \sum_{t=1}^T \sum_{h=1}^H \mathbb{E}_{(s,a) \sim d_h^{(t)}} \left[\delta_h^{(t)}(s, a) \right]. \end{aligned} \quad (7)$$

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 $\mathbb{E}_{(s,a) \sim d_h^{(t)}} [\delta_h^{(t)}(s, a)]$ in the regret upper bound in equation (7). For each t and h , multiply and divide by $\sqrt{\bar{d}_h(s, a)}$ inside the inner product:

$$\mathbb{E}_{d_h^{(t)}} [\delta_h^{(t)}] = \sum_{s,a} \frac{d_h^{(t)}(s, a)}{\sqrt{\bar{d}_h(s, a)}} \cdot \sqrt{\bar{d}_h(s, a)} \delta_h^{(t)}(s, a) \leq \sqrt{\phi_h(d_h^{(t)})} \cdot \sqrt{\mathbb{E}_{\bar{d}_h} [(\delta_h^{(t)})^2]}. \quad (8)$$

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 (8) 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, $d_h^{(t)} = \sum_k \nu^{(t)}(k) d_h^{\pi_{bk}^*}$. 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

$$\phi_h(d_h^{(t)}) \leq \sum_k \nu^{(t)}(k) \phi_h(d_h^{\pi_{bk}^*}) \leq 1 + \Gamma,$$

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 $t \geq t_0$, $h \in [H]$, and (s, a) ,*

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

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

$$\frac{\bar{d}_h^{(t)}(s, a)}{t-1} = \sum_{k=1}^K \bar{\nu}^{(t)}(k) d_h^{\pi_{bk}^*}(s, a) \geq \frac{c_0}{C_{\text{cov}} K} \sum_{k=1}^K d_h^{\pi_{bk}^*}(s, a) = \frac{c_0}{C_{\text{cov}}} \bar{d}_h(s, a).$$

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 $t \geq t_0$ and $h \in [H]$,*

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

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}),

$$\sum_{i=1}^{t-1} \mathbb{E}_{d_h^{(i)}} [(\delta_h^{(t)})^2] = \mathbb{E}_{\bar{d}_h^{(t)}} [(\delta_h^{(t)})^2] \leq \beta. \quad (9)$$

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

$$\bar{d}_h(s, a) \leq \frac{C_{\text{cov}}}{c_0(t-1)} \bar{d}_h^{(t)}(s, a) \quad (t \geq t_0).$$

Multiplying this pointwise inequality by $(\delta_h^{(t)}(s, a))^2$ and summing over (s, a) , then applying (9), yields

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

Proof of Theorem 3.6. Combining (8) with Lemmas C.2 and C.4 gives, for every $t \geq t_0$ and $h \in [H]$,

$$\mathbb{E}_{d_h^{(t)}}[\delta_h^{(t)}] \leq \sqrt{1 + \Gamma} \cdot \sqrt{\frac{C_{\text{cov}} \beta}{c_0 (t-1)}} = \sqrt{\frac{C_{\text{cov}} (1 + \Gamma) \beta}{c_0 (t-1)}}. \quad (10)$$

We split the regret (7) at the burn-in time t_0 :

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

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

During the stable phase, substituting (10) and summing over h and t , II is at most

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

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

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

$$\text{Reg}(T) \leq H t_0 + 2H \sqrt{\frac{C_{\text{cov}} (1 + \Gamma) \beta T}{c_0}} = \mathcal{O}\left(H \sqrt{C_{\text{cov}} (1 + \Gamma) \beta T}\right) + \mathcal{O}(H t_0).$$

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

$$\text{Reg}(T) \leq \mathcal{O}\left(H \sqrt{C_{\text{cov}} (1 + \Gamma) \beta T} + H C_{\text{cov}} K \beta\right),$$

which is equation (1). The corresponding worst-case scaling follows by applying $1 + \Gamma \leq K$ 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 $\{d_h^{\pi_{\text{bk}}^*}\}_{k \in [K]}$ and every $h \in [H]$,*

$$\Gamma = \max_{h \in [H]} \max_{k \in [K]} \chi^2\left(d_h^{\pi_{\text{bk}}^*} \parallel \bar{d}_h\right) \leq K - 1.$$

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

Proof. Fix h and k . Pointwise, $\bar{d}_h(s, a) \geq \frac{1}{K} d_h^{\pi_{\text{bk}}^*}(s, a)$ on the support of $d_h^{\pi_{\text{bk}}^*}$. Hence

$$\chi^2\left(d_h^{\pi_{\text{bk}}^*} \parallel \bar{d}_h\right) = \sum_{s,a} \frac{(d_h^{\pi_{\text{bk}}^*}(s, a))^2}{\bar{d}_h(s, a)} - 1 \leq \sum_{s,a} \frac{(d_h^{\pi_{\text{bk}}^*}(s, a))^2}{(1/K) d_h^{\pi_{\text{bk}}^*}(s, a)} - 1 = K - 1.$$

The disjoint-support case attains equality because then $\bar{d}_h(s, a) = \frac{1}{K} d_h^{\pi_{\text{bk}}^*}(s, a)$ on the support.

C.2 Proof of the regret lower bound

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

$$\mathbb{E}[\text{Reg}(T)] \geq \Omega\left(H\sqrt{KT}\right) = \Omega\left(H\sqrt{(1+\Gamma)T}\right).$$

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

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 there.

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 $\mathcal{S}_1, \dots, \mathcal{S}_K$, 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

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

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 $e_1, e_2, \dots, e_K, e_1, e_2, \dots$, so each preference e_k appears between $\lfloor T/K \rfloor$ and $\lceil T/K \rceil$ times. The optimal basis policy $\pi_{b_k}^*$ 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 $C_{\text{cov}} \leq K$.

We make use of the lower bound 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 $c_{\text{LB}} \min(\Delta n, 1/\Delta)$ for a universal constant $c_{\text{LB}} > 0$, 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

$$\mathbb{E}[\text{Reg}(T)] \geq (H-1)\Delta \mathbb{E}[M] + (H-1) \sum_{k=1}^K \mathbb{E}[\text{Reg}_k^{\text{bandit}}(n_k)], \quad (11)$$

where $\text{Reg}_k^{\text{bandit}}(n_k)$ 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

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

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 $\Delta M_k + \text{Reg}_k^{\text{bandit}}(n_k)$ is lower bounded by the regret of a Bernoulli two-armed bandit problem with m_k opportunities. Lemma C.7 gives

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

Summing over k yields

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

Lastly, substituting the preceding bound into equation (11),

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

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

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

Since the construction has $\Gamma = K-1$, this is equivalent to $\Omega(H \sqrt{(1+\Gamma)T})$, completing the proof.

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

$$\bar{d}_{h,t}^{\text{RE}} := \frac{1}{K} \sum_{j=1}^K d_h^{\pi_j^{(t-1)}}, \quad \Gamma_{\text{RE}}^{(t)} := \max_{h \in [H]} \max_{j \in [K]} \chi^2 \left(d_h^{\pi_j^{(t-1)}} \parallel \bar{d}_{h,t}^{\text{RE}} \right). \quad (12)$$

When the learned ensemble recovers the latent basis policies, $\Gamma_{\text{RE}}^{(t)}$ 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 $1 + \Gamma_{\text{RE}}^{(t)}$.

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 $\bar{d}_{h,t}^{\text{RE}}$, and let $g_h : \mathcal{S} \times \mathcal{A} \rightarrow [-G, G]$ 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

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

is unbiased and satisfies

$$\mathbb{E}[(\hat{\theta}_{j,h} - \theta_{j,h})^2] \leq \frac{G^2(1 + \Gamma_{\text{RE}}^{(t)})}{B}. \quad (13)$$

In the idealized case $\pi_j^{(t-1)} = \pi_{b_j}^*$ for all j , $\Gamma_{\text{RE}}^{(t)} = \Gamma$.

Proof. Fix iteration t , stage h , and target basis index j . For brevity, write $d_j := d_h^{\pi_j^{(t-1)}}$, $\mu := \bar{d}_{h,t}^{\text{RE}}$, and $\rho_j(s, a) := d_j(s, a)/\mu(s, a)$. Finite $\Gamma_{\text{RE}}^{(t)}$ implies $d_j \ll \mu$, so ρ_j is well defined on $\text{supp}(\mu)$. 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: $\mathbb{E}[\hat{\theta}_{j,h}] = \theta_{j,h}$.

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

$$\mathbb{E}_\mu[\rho_j(s, a)^2 g_h(s, a)^2] \leq G^2 \sum_{s,a} \frac{d_j(s, a)^2}{\mu(s, a)} = G^2(1 + \chi^2(d_j \parallel \mu)) \leq G^2(1 + \Gamma_{\text{RE}}^{(t)}), \quad (14)$$

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

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

which leads to equation (13). If $\pi_j^{(t-1)} = \pi_{b_j}^*$ for every $j \in [K]$, then $\mu = \bar{d}_h$ and $\Gamma_{\text{RE}}^{(t)} = \Gamma$ by Definition 3.3.

C.4 Approximate low-rank conditions

This appendix explains how the regret bound changes when Assumption 3.4 is only approximately true. 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 $\bar{d}_h = K^{-1} \sum_{k=1}^K d_h^{\pi_{b_k}^*}$ 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 $\{\pi_{b_1}^*, \dots, \pi_{b_K}^*\}$ and a tolerance $\varepsilon \geq 0$ such that every deployed policy $\pi \in \{\pi^{(t)} : t \leq T\}$ admits a single mixing vector $\nu_\pi \in \Delta^{K-1}$ satisfying, for every stage $h \in [H]$,*

$$d_h^\pi = q_{\pi,h} + e_{\pi,h}, \quad q_{\pi,h} := \sum_{k=1}^K \nu_\pi(k) d_h^{\pi_{b_k}^*}, \quad \|e_{\pi,h}\|_{\bar{d}_h^{-1}}^2 := \sum_{s,a} \frac{e_{\pi,h}(s, a)^2}{\bar{d}_h(s, a)} \leq \varepsilon^2. \quad (15)$$

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 $t \geq t_0$, $h \in [H]$, and (s, a) ,

$$\bar{d}_h^{(t)}(s, a) := \sum_{i=1}^{t-1} d_h^{(i)}(s, a) \geq \frac{c_0(t-1)}{C_{\text{cov}}} \bar{d}_h(s, a). \quad (16)$$

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

$$\mathcal{J}_K^*(w) := \max_{k \in [K]} \mathcal{J}(\pi_{\text{bk}}^*; w) = \max_{\nu \in \Delta^{K-1}} \sum_{k=1}^K \nu(k) \mathcal{J}(\pi_{\text{bk}}^*; w). \quad (17)$$

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

$$\text{Reg}_K(T) := \sum_{t=1}^T \left[\mathcal{J}_K^*(w_t) - \mathcal{J}(\pi^{(t)}; w_t) \right]. \quad (18)$$

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 $t_0 = \mathcal{O}(C_{\text{cov}}K\beta)$, where $\beta := c_1 \log(K|\mathcal{F}|T/\delta)$. Then, with probability at least $1 - \delta$, the pooled-confidence routed-ensemble algorithm satisfies*

$$\text{Reg}_K(T) \leq \mathcal{O}\left(H\sqrt{C_{\text{cov}}\beta T}(\sqrt{1+\Gamma} + \varepsilon) + HC_{\text{cov}}K\beta\right). \quad (19)$$

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

Proof of Proposition C.10. 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 $\mathcal{J}(\pi_w^*; w)$ to the basis-oracle value $\mathcal{J}_K^*(w)$, 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 that the following lemma on the optimism of the routed basis.

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

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

Proof. Let $k_K^*(w_t) \in \arg \max_{k \in [K]} \mathcal{J}(\pi_{\text{bk}}^*; w_t)$. By the optimism event E_{opt} , the true Q -function $Q^{\pi_{\text{bk}}^*(w_t)}(\cdot; w_t)$ belongs to the confidence set $\mathcal{F}_{k_K^*(w_t)}^{(t-1)}(\beta)$. The routed optimistic choice in equation (6) maximizes over all basis indices and all functions in their confidence sets, so we may restrict the maximization to the single feasible candidate $(k_K^*(w_t), Q^{\pi_{\text{bk}}^*(w_t)})$. This gives

$$V_1^{f^{(t)}}(s_0; w_t) \geq V_1^{Q^{\pi_{\text{bk}}^*(w_t)}}(s_0; w_t) = \mathcal{J}(\pi_{\text{bk}}^*(w_t); w_t) = \mathcal{J}_K^*(w_t).$$

Second, we demonstrate regret reduction via a certain identity. This step is identical to the regret reduction in equation (7). Working on the intersection $E_{\text{opt}} \cap E_{\text{reg}}$ (probability at least $1 - \delta$), Lemma C.11 together with the standard layerwise telescoping identity gives

$$\text{Reg}_K(T) \leq \sum_{t=1}^T \left[V_1^{f^{(t)}}(s_0; w_t) - \mathcal{J}(\pi^{(t)}; w_t) \right] = \sum_{t=1}^T \sum_{h=1}^H \mathbb{E}_{d_h^{(t)}} \left[\delta_h^{(t)} \right]. \quad (20)$$

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

Identical to equation (8), for every $t \in [T]$ and $h \in [H]$, we have

$$\mathbb{E}_{d_h^{(t)}} \left[\delta_h^{(t)} \right] \leq \sqrt{\phi_h(d_h^{(t)})} \cdot \sqrt{\mathbb{E}_{\bar{d}_h} \left[(\delta_h^{(t)})^2 \right]}. \quad (21)$$

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

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

$$\sqrt{\phi_h(d_h^\pi)} \leq \sqrt{1 + \Gamma} + \varepsilon. \quad (22)$$

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

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

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

$$\phi_h(q_{\pi,h}) = \phi_h\left(\sum_k \nu_\pi(k) d_h^{\pi_{bk}}\right) \leq \sum_k \nu_\pi(k) \phi_h(d_h^{\pi_{bk}}) \leq 1 + \Gamma.$$

Combining this display with the triangle-inequality bound proves equation (22). Applying Lemma C.12 to the deployed policy $\pi^{(t)}$ yields

$$\sqrt{\phi_h(d_h^{(t)})} \leq \sqrt{1 + \Gamma} + \varepsilon. \quad (23)$$

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 $t \geq t_0$ and $h \in [H]$,

$$\sqrt{\mathbb{E}_{\bar{d}_h} \left[(\delta_h^{(t)})^2 \right]} \leq \sqrt{\frac{C_{\text{cov}} \beta}{c_0(t-1)}}. \quad (24)$$

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 (21), (23), and (24), for every $t \geq t_0$ and $h \in [H]$,

$$\mathbb{E}_{d_h^{(t)}} \left[\delta_h^{(t)} \right] \leq (\sqrt{1 + \Gamma} + \varepsilon) \sqrt{\frac{C_{\text{cov}} \beta}{c_0(t-1)}}. \quad (25)$$

We split the regret (20) at the burn-in time t_0 and bound each part.

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

Substituting equation (25) and using $\sum_{t=t_0}^T (t-1)^{-1/2} \leq \sum_{u=1}^{T-1} u^{-1/2} \leq 2\sqrt{T}$,

$$\sum_{t=t_0}^T \sum_{h=1}^H \mathbb{E}_{d_h^{(t)}} \left[\delta_h^{(t)} \right] \leq H(\sqrt{1 + \Gamma} + \varepsilon) \sqrt{\frac{C_{\text{cov}} \beta}{c_0}} \sum_{t=t_0}^T \frac{1}{\sqrt{t-1}} \leq 2H(\sqrt{1 + \Gamma} + \varepsilon) \sqrt{\frac{C_{\text{cov}} \beta T}{c_0}}.$$

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

$$\text{Reg}_K(T) \leq \mathcal{O}\left(H\sqrt{C_{\text{cov}}\beta T} \cdot (\sqrt{1+\Gamma} + \varepsilon) + H C_{\text{cov}} K \beta\right),$$

which leads to equation (19).

Lastly, we connect basis-relative regret $\text{Reg}_K(T)$ to the cumulative regret $\text{Reg}(T)$ against the true optimal policy via a clean two-term decomposition. For a fixed basis $\{\pi_{b1}^*, \dots, \pi_{bK}^*\}$, define the cumulative basis-approximation error along the realized preference sequence as

$$A_K(T) := \sum_{t=1}^T [\mathcal{J}(\pi_{w_t}^*; w_t) - \mathcal{J}_K^*(w_t)]. \quad (26)$$

Since every basis policy is feasible, $\mathcal{J}_K^*(w_t) \leq \mathcal{J}(\pi_{w_t}^*; w_t)$ for every t , so $A_K(T) \geq 0$. The quantity $A_K(T)$ depends on the basis $\{\pi_{b_k}^*\}$ 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 $\mathcal{J}_K^*(w_t)$ in and out of the regret sum,

$$\text{Reg}(T) = \underbrace{\sum_{t=1}^T [\mathcal{J}(\pi_{w_t}^*; w_t) - \mathcal{J}_K^*(w_t)]}_{A_K(T)} + \underbrace{\sum_{t=1}^T [\mathcal{J}_K^*(w_t) - \mathcal{J}(\pi^{(t)}; w_t)]}_{\text{Reg}_K(T)}. \quad (27)$$

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 $d_h^{\pi_w^*} = q_{w,h} + e_{w,h}$. Since scalarized rewards $r_h^w(s, a)$ are bounded in $[0, 1]$, its inner product with approximation error $e_{w,h}$ has

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

Using the same mixing vector ν_w at every stage,

$$\begin{aligned} \mathcal{J}(\pi_w^*; w) &= \sum_{h=1}^H \langle d_h^{\pi_w^*}, r_h^w \rangle \leq \sum_{h=1}^H \langle q_{w,h}, r_h^w \rangle + H\varepsilon \\ &= \sum_{k=1}^K \nu_w(k) \mathcal{J}(\pi_{b_k}^*; w) + H\varepsilon \leq \mathcal{J}_K^*(w) + H\varepsilon. \end{aligned}$$

Summing over $t \in [T]$ yields $A_K(T) \leq H\varepsilon T$. Lastly, combining equation (27) with the above, and Proposition C.10, we have

$$\text{Reg}(T) \leq \mathcal{O}\left(H\varepsilon T + H\sqrt{C_{\text{cov}}\beta T} \cdot (\sqrt{1+\Gamma} + \varepsilon) + H C_{\text{cov}} K \beta\right). \quad (28)$$