Local Policy Manifolds for Efficient Multi-Objective Reinforcement Learning

Qiyue Xia [†]
School of Informatics
University of Edinburgh
qiyue.xia@ed.ac.uk

Tianwei Wang [‡]
School of Informatics
University of Edinburgh
tianwei.wang@ed.ac.uk

J. Michael Herrmann [†§]
School of Informatics
University of Edinburgh
michael.herrmann@ed.ac.uk

Abstract

Multi-objective reinforcement learning (MORL) aims at optimising several, often conflicting goals to improve the flexibility and reliability of RL in practical tasks. This is typically achieved by finding a set of diverse, non-dominated policies that form a Pareto front in the performance space. However, constructing such a policy set remains computationally demanding, as it requires finding not a single optimal policy, but a set of policies to cover a wide range of trade-offs among the objectives. We introduce LLE-MORL, an approach that traces policy manifolds by utilising the local relationship between the high-dimensional policy parameter space and the performance space. This structured representation enables an efficient search within contiguous solution domains via locally linear extrapolation, allowing for the rapid generation of high-quality solutions without extensive retraining. We also provide a theoretical analysis of the policy manifold structure in order to specify the applicability of the method. Experiments across diverse continuous control domains demonstrate that LLE-MORL consistently achieves higher Pareto front quality and efficiency than state-of-the-art approaches.

1 Introduction

Reinforcement Learning (RL) has shown great promise in complex decision-making problems, enabling significant advancements in a wide range [Silver et al., 2016, Levine et al., 2016]. In real-world scenarios, however, problems often feature multiple, often conflicting, objectives. Under this circumstance, multi-objective approaches provide flexibility in practical applications of reinforcement learning by providing a modifiable policy that can be adjusted according to changes of preference among a set of objectives [Roijers et al., 2013, Hayes et al., 2022]. This has fostered the development of the field known as multi-objective reinforcement learning (MORL). Ideally, the modifiable policies developed within MORL allow for efficient adaptation, ensuring that a policy optimal for one set of preferences can be readily transformed to be optimal for a new set when those preferences change. To prepare such a modifiable policy for application, three problems have to be solved: (i) The *learning* problem involves the solution of an RL problem for each combination of preference parameters or at least for a representative subset of preferences. (ii) The *representation* problem requires a parametrisation of the policies, which typically results in either a discrete set of individual policies (common in population-based methods) or a single, continuously adaptable policy (prevalent in deep reinforcement learning approaches). (iii) The *selection* problem is to identify a suitable policy in the application which includes dynamic adjustments to preference drifts and possibly the decision whether a different policy should be invoked or whether further training is required to respond to a temporary detection of suboptimality.

[†] Institute of Perception, Action and Behaviour
[‡] Artificial Intelligence and its Applications Institute
[§] Corresponding author

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

We consider these problems as a coherent task in order to reduce the computational burden of the learning problem and improve the structure of the policy representation. We hypothesise that if a continuous representation of policies can be found where similar preferences correspond to similar policy parameters, then small performance differences might be compensable with brief, targeted retraining. It is also anticipated that such a structured policy representation would benefit the selection problem, though this aspect is not the primary focus of our current study.

While a globally continuous mapping is an ideal, we notice that in non-trivial problems, the relationship between the performance space and parameter space of policies is not a simple, single continuous mapping but can be described by a family of locally continuous components [Xu et al., 2020, Li et al., 2025]. Our findings suggest that effectively exploring just a few of these components can be sufficient to achieve competitive performance in typical benchmark problems. To explore these components, we take a different perspective. Rather than searching for Pareto-optimal policies independently, we aim to trace the Pareto front manifold directly by exploring local structure in the high-dimensional policy parameter space. We formalise the Parameter-Performance Relationship (PPR), which captures the local correspondence between policy parameters and objective performance. This viewpoint enables an efficient strategy for Pareto front (see Sect. 2.2) approximation: starting from a small set of policies, we can expand along directions on the local policy manifold that preserve PPR and generate new candidate solutions via a training-free extension stage, establishing an explicit relationship between the policy parameter space and the reward space.

Based on this perspective, we develop a novel algorithm LLE-MORL (Locally Linear Extrapolation for Multi-Objective Reinforcement Learning). Our experiments demonstrate that the proposed algorithm can achieve high-quality Pareto front approximations with notable sample efficiency compared to existing approaches. By focusing on local structure in the policy parameter space, LLE-MORL reduces the need for exhaustive training while maintaining strong coverage and solution quality. Beyond efficiency, the structured use of the PPR reveals local policy manifold structure, supporting a clearer characterisation of trade-offs along the Pareto front. Our code is publicly available at `https://github.com/AQQQQY/LLE_MORL`.

2 Background

2.1 Multi-Objective Reinforcement Learning

Multi-Objective Reinforcement Learning (MORL) extends the traditional RL framework to scenarios where agents must consider multiple, often conflicting objectives. This extension allows for more sophisticated decision-making models that mirror real-world complexities where trade-offs between competing goals, such as cost versus quality or speed versus safety, are common. To ground this notion formally, we represent a MORL problem as a Multi-Objective Markov Decision Process (MOMDP) which generalises the standard MDP framework to accommodate multiple reward functions, each corresponding to a different objective.

Definition 2.1. *Multi-Objective Markov Decision Process (MOMDP). A MOMDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P(s'|s,a)$ is the state transition probability, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ is a vector-valued reward function with d as the number of objectives, specifying the immediate reward for each of the objectives, γ is the discount factor.*

The crucial difference between MOMDPs and traditional single-objective MDPs is the reward structure. While single-objective MDPs use a scalar reward function, MOMDPs feature a vector-valued reward function that outputs d-dim vector rewards $\mathbf{r} = R(s,a)$, which delivers distinct numeric feedback for each objective. We denote a policy by π. At each timestep t, the agent in state $s_t \in \mathcal{S}$ selects an action $a_t \sim \pi(\cdot \mid s_t)$, transitions to a new state s_{t+1} with probability $P(s_{t+1} \mid s_t, a_t)$, and receives a reward vector $\mathbf{r}_t = \big[(R_1(s_t,a_t),\ R_2(s_t,a_t),\ \ldots,\ R_d(s_t,a_t)\big])$. We define the discounted return vector by $\mathbf{G}_t = \sum_{k=0}^{\infty} \gamma^k\, \mathbf{r}_{t+k}$, and the multi-objective action-value function of a policy π for a given state-action pair (s,a) by $\mathbf{Q}^\pi(s,a) = \mathbb{E}_\pi\big[\mathbf{G}_t \mid s_t = s,\ a_t = a\big]$. The goal of MORL is to find a policy π such that the expected return of each objective can be optimised. In practice, we trade off objectives via a scalarisation function $f_{\boldsymbol{\omega}}(\mathbf{r})$, which produces a scalar utility using preference vector $\boldsymbol{\omega} \in \Omega$, where Ω is the preferences space. The scalarisation function $f_{\boldsymbol{\omega}}(\mathbf{r})$ is used for mapping the multi-objective reward vector $\mathbf{r}(s,a)$ to a single scalar. In this paper, we consider the linear scalarisation function $f_{\boldsymbol{\omega}}(\mathbf{r}(s,a)) = \boldsymbol{\omega}^{\mathbf{T}}\mathbf{r}(s,a)$, where the preference vector $\boldsymbol{\omega}$ lies in the probability

simplex, i.e., $\sum_{i=1}^{d} \omega_i = 1$ and $\omega_i \geq 0$, which is commonly used in MORL literature [Yang et al., 2019, Felten et al., 2024]. When the preference dimension $d = 1$, the MOMDP reduces to a standard single-objective MDP under this linear scalarisation setting.

2.2 Pareto Optimality

In multi-objective optimisation, the concept of optimality differs from the single-objective case. Typically, no single policy simultaneously maximises all objectives due to inherent trade-offs. Without any additional information about the user's preference, there can now be multiple possibly optimal solutions. In the following, we introduce several useful definitions for possibly optimal policies.

Definition 2.2. *Pareto optimality. A policy π is said to* dominate *another policy π' if and only if $\forall i \in \{1, \ldots, d\}$, $V_i^{\pi}(s) \geq V_i^{\pi'}(s)$, and $\exists j, V_j^{\pi}(s) > V_j^{\pi'}(s)$, where $V_i^{\pi}(s) = \mathbb{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t R_i(s_t, a_t) \mid s_0 = s]$ denotes the expected discounted return for objective i under policy π. A policy π^* is* Pareto optimal *if and only if it is not dominated by another policy. The set of all Pareto optimal policies forms the* Pareto set*: $\Pi_P = \{\pi \mid \pi \text{ is Pareto optimal}\}$. The corresponding set of expected returns incurred by policies in the Pareto set is termed* Pareto front*: $\mathcal{P} = \{V^{\pi}(s) \mid \pi \in \Pi_P\}$.*

Since obtaining the true Pareto set is intractable in complex problems, the practical aim of multi-objective optimisation is to construct a finite set of policies that closely approximate the true Pareto front. So that practitioners can select the policy based on their preferred trade-off among objectives.

3 Methods

3.1 Parameter-Performance Relationship

Recent work in MORL has implicitly suggested a relationship between the parameter space of the policy network and the Pareto front in the performance space. [Xu et al., 2020] empirically show for PGMORL that each disjoint policy family occupies a continuous region in parameter space and maps to a contiguous segment of the Pareto front, while MORL/D [Felten et al., 2024] assume that policies with similar parameters should lead to close evaluations. Motivated by these implicit observations, we formalise the *parameter–performance relationship* and proceed to explain and empirically validate this property.

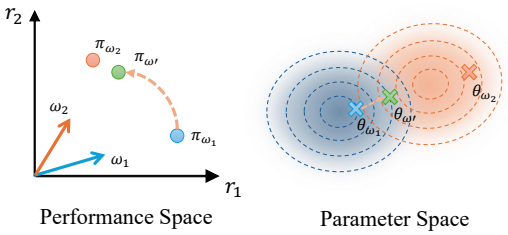


Figure 1: Retraining of a policy can produce policies that are similar in parameter space, but cover other regions in Pareto performance space.

Definition 3.1. *Parameter-Performance Relationship (PPR). Let $\Theta \subseteq \mathbb{R}^n$ be the space of policy parameters, and V be the expected discounted return as a function $V : \Theta \to \mathbb{R}^d$ of the parameter θ. We say that V exhibits a* parameter–performance relationship *on an open region $U \subseteq \Theta$ if $\exists \delta > 0$ and a function $h : U \times B_\delta(0) \to \mathbb{R}^d$,* s.t. *$\forall \theta \in U$ and perturbation $\Delta\theta$ satisfying $\|\Delta\theta\| < \delta$ and $\theta + \Delta\theta \in U$, the difference in the performance space is:*

$$V(\theta + \Delta\theta) \; - \; V(\theta) \; = \; h\big(\theta, \Delta\theta\big). \tag{1}$$

where $B_\delta(0)$ is an open ball with radius δ centred on the origin, i.e. $B_\delta(0) := \{x : \|x\| < \delta\}$.

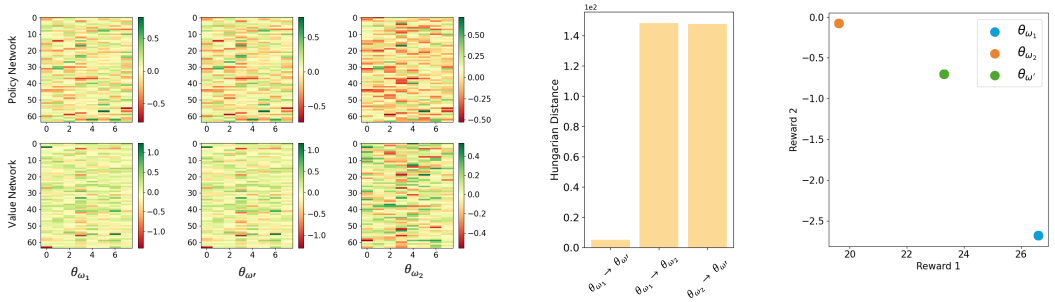
PPR is a local property and need not hold globally across the entire parameter space. In high-dimensional parameter spaces, policies with similar performance may be far apart in parameter space. This raises the question of how to identify policies that satisfy the PPR locally. As a first step, we require a metric for policy closeness in parameter space. We adopt the Hungarian matching distance [Kuhn, 1955, Munkres, 1957] to measure model distance and thereby quantify structural similarity between policies. A formal definition of Hungarian matching distance is provided in Appendix C.3. This metric naturally handles the permutation invariance of hidden units [Goodfellow et al., 2016] and measures the smallest structural change needed to align one model to another: lower Hungarian distance indicates greater model similarity.

As illustrated in Figure 1, the red and blue gradient shadings and contour lines indicate the optimal region under different preference vectors ω_1 and ω_2 in high-dimensional parameter space and the

arrow marks a short retraining update. Although separately trained policies π_{ω_1} and π_{ω_2} under different preference weights ω_1 and ω_2 may be far apart in high-dimensional parameter space, a short retraining using ω_2 starting from π_{ω_1} typically induces a small, structured parameter update together with a directional shift in the expected returns. This observation motivates our use of brief retraining as a practical means of obtaining locally related policies for which the PPR holds.

Sanity Check To obtain an initial empirical understanding of the PPR, we compare independently trained policies with those obtained via brief retraining. We first train two policies to the stable stage using a multi-objective PPO-based [Schulman et al., 2017] algorithm under scalarization vectors ω_1 and ω_2, yielding model parameters θ_{ω_1} and θ_{ω_2}. Starting from θ_{ω_1}, we perform a short retraining step under ω_2 to obtain $\theta_{\omega'}$. We qualitatively evaluated the similarity between these policies using the policy and value network neuron heat maps in Figure 2a, and quantitatively using the Hungarian matching distances between model pairs in Figure 2b. In Figure 2a, the horizontal axis corresponds to network layers, the vertical axis corresponds to neurons within each layer, and colour indicates the parameter values. Their corresponding rewards are visualised in the two-objective performance space for the multi-objective SWIMMER problem in Figure 2c.

We compare three pairs of models: (1) θ_{ω_1} and θ_{ω_2}, representing independently trained policies that differ substantially in both parameter and performance space; (2) θ_{ω_1} and $\theta_{\omega'}$, showing that brief retraining produces a structurally similar policy with a low Hungarian distance while inducing a clear shift in performance; and (3) $\theta_{\omega'}$ and θ_{ω_2}, demonstrating that although their parameters remain distinct, their rewards lie much closer in the performance space. These empirical observations indicate that a short retraining under a new preference produces a small, structured parameter update that directly maps to a predictable shift in performance, validating the PPR. More details of the sanity check procedure are provided in Appendix C.4.



(a) Heatmaps of policy-net and value-net parameters to illustrate the retraining effect, compare Figure 1.

(b) Combined Hungarian model distance.

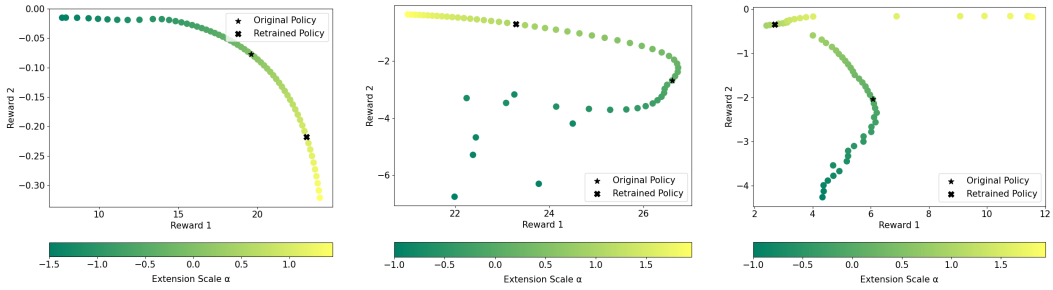
(c) Respective positions in performance space.

Figure 2: Visual comparison of fully trained policies $\theta_{\omega_1}, \theta_{\omega_2}$ and briefly retrained policy $\theta_{\omega'}$.

3.2 Locally Linear Extension

Based on the PPR formalisation, a natural question is whether the parameter-space difference between two policies that satisfy the PPR can serve as a directional update for tracing the Pareto front along a local policy manifold. To explore this, we consider a base policy θ_ω and a retrained policy $\theta_{\omega'}$ that exhibit a PPR. For this directional information to be meaningful for Pareto front exploration, both θ_ω and $\theta_{\omega'}$ should ideally be non-dominated solutions, at least with respect to each other. Given such a pair, we compute the parameter update vector $\Delta\theta = \theta_{\omega'} - \theta_\omega$ and generate intermediate policies by moving from the base policy θ_ω along the parameter displacement $\Delta\theta$ in scaled steps. Concretely, for each scale α, we form $\theta_\alpha = \theta_\omega + \alpha\,\Delta\theta$ and evaluate its reward vectors in preference space.

Figure 3 visualises the resulting trajectory of reward vectors in the two-dimensional objective space: as α grows, the trajectory passes through the region around $\theta_{\omega'}$ and can extend beyond both the base and retrained endpoints, demonstrating how simple parameter-space moves can traverse broad trade-off regions, which offers a cost-effective strategy for efficiently expanding an approximate Pareto front without training each point from scratch.



(a) (b) (c)

Figure 3: Visualisation of the process of applying the parameter difference $\Delta\theta = \theta_{\omega'} - \theta_{\omega}$ between two related policies. The policies are obtained by first training a policy θ_{ω} to a stable stage using a scalarization vector ω and then finding a policy $\theta_{\omega'}$ by a brief additional training period with a different scalarization vector ω'. Iterating the shift $\Delta\theta$ in the policy space induces a sequence of shifts also in the multi-objective reward space. The subfigures show results for different initial preferences: (a) A convex front is found from the two policies. (b) Although the original policy turns out to be Pareto suboptimal, the solution manifold extends into a Pareto optimal component. (c) Retraining can cause the (Pareto-suboptimal) original solution to jump to a different branch so that the corresponding solution consists of two components one of which can be ignored because of Pareto suboptimality.

3.3 The LLE-MORL Algorithm

Motivated by empirical observations, LLE-MORL is built on the hypothesis that there exist local policy manifolds in parameter space that correspond locally to the Pareto front and have the same intrinsic dimension, along which multi-objective performance varies smoothly. Theoretical analysis and formal proofs are provided in Section 3.4 and Appendix B.

Consider a d-objective MORL problem. The full Algorithm 1 (see Appendix A) consists of five stages: **(1) Initialisation:** We train a set of K base policies $\{\theta_{\omega_k}\}_{k=1}^{K}$ to stable using PPO [Schulman et al., 2017]. Each policy is trained under a distinct scalarization weight $\omega_k \in \Omega$, where these weights are chosen to be evenly distributed across the preference space. **(2) Directional Retraining:** For each base policy $\theta_{\omega_k} (k = 1, \ldots, K)$, we estimate a set of $m = d - 1$ local directions on the policy manifold in parameter space by retraining. Specifically, for each $i = 1, \ldots, m$, we continue training from θ_{ω_k} under a nearby preference $\omega_k^{(i)}$ for T_{dir} steps to obtain a directional policy $\theta_{\omega_k^{(i)}}$, where θ_{ω_k} and $\theta_{\omega_k^{(i)}}$ are all non-dominated. We record $\Delta\theta_k^{(i)} = \theta_{\omega_k^{(i)}} - \theta_{\omega_k}$, and $\Delta\omega_k^{(i)} = \omega_k^{(i)} - \omega_k$ as the parameter update vector and the corresponding preference shift. **(3) Locally Linear Extension:** For each base policy θ_{ω_k}, we generate candidate policies by extrapolating along the m local directions using $\{\Delta\theta_k^{(i)}\}_{i=1}^{m}$. Each candidate is constructed as a linear combination of these directions: $\theta_{k,\text{cand}} = \theta_{\omega_k} + \sum_{i=1}^{m} \alpha_i \Delta\theta_k^{(i)}$, where step-scale factors α control the step sizes along each direction. We choose each α_i from a discrete grid defined by α_{start}, α_{end}, and step size $\Delta\alpha$, resulting in M possible values per direction. Iterating over all combinations across the m directions produces M^m candidate policies for each base solution. In parallel, we assign each candidate a matched preference weight $\omega_{k,\text{cand}} = \omega_k + \sum_{i=1}^{m} \alpha_i \, \Delta\omega_k^{(i)}$. **(4) Candidate Selection:** All candidates $\theta_{k,\text{cand}}$ generated in the locally linear extension stage are evaluated to obtain their respective performance vectors. We then select the subset of non-dominated candidates and forward them to the fine-tuning stage. **(5) Preference-Aligned Fine-Tuning:** For each selected candidate θ and its matched weight ω, we perform a short PPO fine-tuning phase of T_{ref} steps under ω to push the policy closer to the true Pareto front.

Together, these stages explore local policy manifold structure in parameter space, enabling efficient tracing of the local Pareto front manifold. This construction yields a structured policy representation: local directions in parameter space correspond to meaningful objective trade-offs, allowing parameter changes to be directly associated with movements along the Pareto front.

3.4 Theoretical Analysis

In this section, we present a theoretical analysis for the effectiveness of LLE-MORL, formally modelling the local manifold structure of Pareto-optimal policies and establishing guarantees for the locally linear extension mechanism used to trace the Pareto front. For clarity, we only provide the theory framework here, while proofs are located in Appendix B. First, the continuity of the reconstructed Pareto front by LLE-MORL is guaranteed by the Lipschitz continuity of PPR. Following Theorem 3.1 and Corollary 3.1 provide the theoretical foundation of these properties.

Theorem 3.1. *(Lipschitz Continuity of PPR function h.) Denote the expected discounted return by $V : U \to \mathbb{R}^d$, where $U \subseteq \Theta \subseteq \mathbb{R}^n$ is an open set. The sufficient conditions for Lipschitz continuity of $h(\theta, \Delta\theta)$ with respect to the second variable $\Delta\theta$ are: $V(\theta)$ is differentiable on U, and the score function is bounded, i.e. $\forall a \in \mathcal{A}, s \in \mathcal{S}$, there is*

$$\sup_{\theta \in U} \left\| \nabla_\theta \log \pi_\theta(a|s) \right\| \leq \mathcal{G} < \infty. \tag{2}$$

The inequality Equation (2) states that the score function $\nabla_\theta \log \pi_\theta(a|s)$ introduced in Baxter and Bartlett [2001] is bounded by a constant $\mathcal{G}$. This is a common regularity assumption in related RL research, which can be found in Aydin and Eksin [2023], Shi et al. [2024].

Corollary 3.1. *(Surface continuity guaranteed by Lipschitz PPR) Algorithm LLE can construct a continuous surface in the performance space, as an approximation to the Pareto front, given appropriate regularity conditions and a range of α.*

Further, we aim to evaluate the error between the reconstructed result by LLE-MORL and the Pareto front $\mathcal{P}$. Assume $\mathcal{P}$ is a differentiable manifold of dimension m embedded in the Euclidean space of dimension d, i.e. $\mathcal{P} \in \mathcal{M}^m(\mathbb{R}^d)$, where $\dim\mathcal{P}$ is $m = d - 1$.

Suppose the initialisation returns $\theta^{(0)}$, assuming it corresponds to a non-dominated performance, where $v := V\big(\theta^{(0)}\big) \in \mathcal{P}$. Let $P_v \subset \mathcal{P}$ be an open neighbourhood of v on the manifold $\mathcal{P}$, where $P_v \in \mathcal{M}^m(\mathbb{R}^d)$. Define the neighbour of v to be P_v, assume there exists a parameter set $\Theta_v \subset \Theta$ bijectively mapping to P_v, denoted by $V|_{\Theta_v} : \Theta_v \to P_v$.

Theorem 3.2. *(Local lifting manifold) If V is differentiable on U and $(V|_{\Theta_v})^{-1} : P_v \to \mathbb{R}^n$ is C^k, then Θ_v is a differentiable manifold embedded in $\mathbb{R}^n$ of the same dimension as P_v, i.e. $\Theta_v \in \mathcal{M}^m(\mathbb{R}^n)$.*

Therefore, the relation between the Pareto front and corresponding parameters can be understood as the diffeomorphism between two differentiable manifolds of the same dimension m. Next, we define the expression of linear extension.

Condition 3.1. *(Local extension directions) Suppose the directional retraining in LLE-MORL returns $\theta^{(i)} \in \Theta_v$, $i = 1, \ldots, m$. Define*

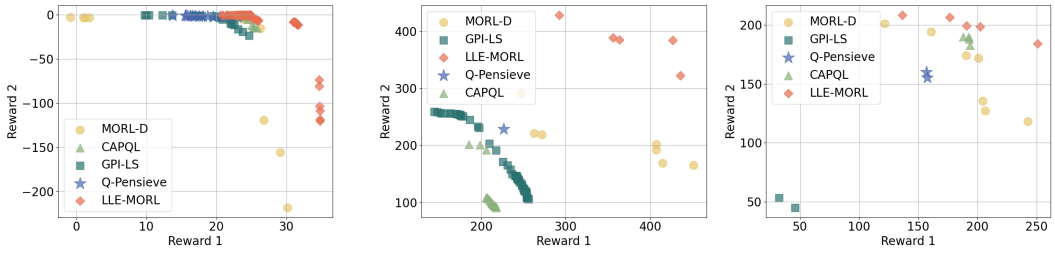
$$\mathcal{D} := \left[\theta^{(1)} - \theta^{(0)},\ \ldots,\ \theta^{(m)} - \theta^{(0)}\right] \in \mathbb{R}^{n \times m} \tag{3}$$

which has full column rank, i.e. $\operatorname{rank}(\mathcal{D}) = m$. Further, $\operatorname{span}(\mathcal{D})$ provides an approximation to the tangent space of Θ_v at $\theta^{(0)}$, i.e. $T_{\theta^{(0)}}\Theta_v$.

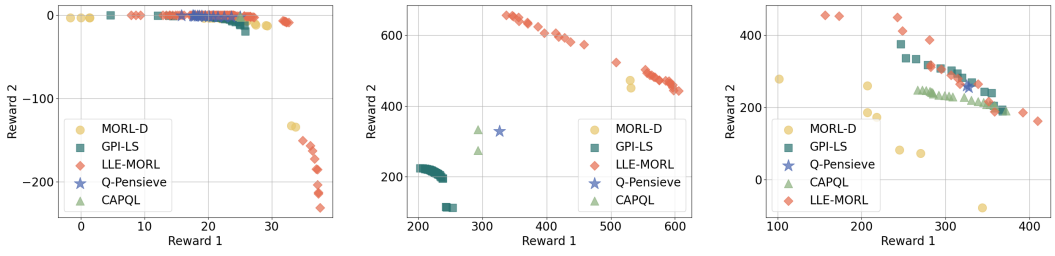
Theorem 3.3. *(Local Pareto error of LLE) Denote the local linear extrapolation by $\tilde{\theta}(\alpha) := \theta^{(0)} + \mathcal{D}\alpha$. Then, for a suitable range of $\alpha \in \mathbb{R}^m$, the error of extrapolation to the Pareto front is bounded, with first-order convergence as $\|\alpha\| \to 0$ and second-order increase as $\|\alpha\|$ grows. The error of exploration to the actual Pareto front is controlled by:*

$$\operatorname{dist}\big(V(\tilde{\theta}(\alpha)),\, \mathcal{P}_v\big) \leq \frac{L_V \eta}{2} \max_{1 \leq i \leq m} \|x^{(i)}\|^2 \Big(\sqrt{m}\,\|\alpha\| + m\,\|\alpha\|^2\Big). \tag{4}$$

The detailed proof of this error bound can be found in Theorem B.3 in Appendix B. Theorem 3.3 supports the effectiveness of LLE-MORL in constructing the Pareto front. And the error is controlled by an appropriate range of α, corresponding to the distance of linear extension. The empirical results and discussion are stated in Appendix D.2 for the feasibility of the choice of α.



(a) MO-Swimmer (sample-efficient) (b) MO-Hopper-2d (sample-efficient) (c) MO-Ant-2d (sample-efficient)



(d) MO-Swimmer (standard-training) (e) MO-Hopper-2d (standard-training) (f) MO-Ant-2d (standard-training)

Figure 4: Pareto fronts from 2 different settings, comparing our LLE-MORL method with baselines on three 2d continuous-control benchmarks. LLE-MORL consistently achieves wider coverage and closer proximity to the true Pareto front.

4 Experiments

4.1 Experiment Setup

In this section, we evaluate the LLE-MORL algorithm using popular continuous MORL benchmark problems from the MO-Gymnasium-v5 [Felten et al., 2023]. Our benchmark problems include three two-objective continuous environments: **MO-Swimmer**, **MO-Hopper-2d**, **MO-Ant-2d**, and two three-objective continuous environments: **MO-Hopper-3d**, **MO-Ant-3d**. We evaluate the quality of the approximate Pareto front using three standard metrics: **Hypervolume (HV)**, **Expected Utility (EU)**, **Sparsity (SP)**, following the formalism in [Zitzler and Thiele, 1999, Zintgraf et al., 2015, Hayes et al., 2022]. Higher HV and EU values indicate better overall front quality, while SP measures the distribution of solutions along the front. Since SP is scale-dependent and less directly related to decision quality, we report it mainly as a complementary metric to assess coverage diversity. More details can be found in Appendix C.

We compare our LLE-MORL against the following state-of-the-art MORL algorithms: (i) **GPI-LS** [Alegre et al., 2023] applies Generalised Policy Improvement over a discretised set of preference weights and uses linear scalarization to construct a diverse Pareto set. (ii) Concave-augmented Pareto Q-learning (**CAPQL**) [Lu et al., 2023]: learns an ensemble of Q-functions under different preferences and selects actions via conservative aggregation to improve front coverage. (iii) **Q-Pensieve** [Hung et al., 2023] boosts the sample efficiency of MORL by storing past Q-function snapshots in a replay buffer, enabling explicit policy-level knowledge sharing across training iterations. (iv) **MORL/D** [Felten et al., 2024] is a deep-RL analogue of decomposition-based multi-objective optimisation that trains subpolicies under scalarised objectives and recombines them via weight decompositions to approximate the Pareto front. We additionally report LLE-MORL-0, which disables the fine-tuning stage of LLE-MORL, as an ablation variant; the corresponding analysis is provided in Section 4.4.

4.2 Results and Analysis on 2D Benchmarks

We now present quantitative results on 2D-objective benchmarks. We conduct experiments under two distinct settings to provide a comprehensive understanding of these algorithms' capabilities: **(1) Sample-Efficient Setting:** All methods, including our LLE-MORL, were trained under an interaction budget for 1.5×10^5 timesteps (including all algorithmic training phases). Given the complexity of continuous control benchmarks, this relatively limited interaction budget serves as a

Table 1: Evaluation of the Pareto front quality on 2D-objective benchmarks under different settings.

Setting	Environment	Metric	Method					
			GPI-LS	CAPQL	Q-Pensieve	MORL/D	LLE-MORL-0	LLE-MORL
Sample-efficient	MO-Swimmer	HV(10^4)	4.80±0.43	5.00±0.09	4.77±0.17	5.72±0.25	6.54±0.10	**6.77±0.17**
		EU(10^1)	0.89±0.14	1.08±0.01	1.06±0.11	1.00±0.01	0.93±0.11	**1.11±0.02**
		SP(10^2)	1.12±1.02	1.68±1.33	**0.82±0.61**	11.03±7.17	1.61±1.21	1.78±1.60
	MO-Hopper-2d	HV(10^5)	1.06±0.12	1.40±0.31	1.00±0.05	1.96±0.19	2.73±0.17	**3.02±0.28**
		EU(10^2)	2.11±0.17	2.68±0.40	2.16±0.08	3.29±0.15	4.15±0.20	**4.36±0.28**
		SP(10^2)	5.35±3.64	1.33±1.21	**0.08±0.04**	48.25±22.90	8.08±6.44	15.13±8.31
	MO-Ant-2d	HV(10^4)	3.24±1.28	8.10±0.64	5.01±1.73	9.73±0.42	10.07±0.63	**10.44±0.56**
		EU(10^2)	0.72±0.30	1.82±0.10	1.18±0.41	2.03±0.08	2.16±0.10	**2.23±0.08**
		SP(10^3)	1.14±1.01	**0.05±0.03**	0.12±0.10	1.33±1.02	1.51±0.36	1.22±0.22
Standard-training	MO-Swimmer	HV(10^4)	5.46±0.17	5.12±0.32	5.20±1.08	7.05±0.39	7.74±0.22	**7.82±0.24**
		EU(10^1)	1.06±0.03	1.10±0.05	1.13±0.15	1.13±0.03	1.03±0.07	**1.16±0.09**
		SP(10^2)	**0.04±0.02**	0.12±0.09	0.05±0.02	4.36±1.64	1.11±0.56	1.15±0.63
	MO-Hopper-2d	HV(10^5)	1.15±0.04	2.11±0.91	1.62±0.20	3.75±0.25	4.76±0.08	**4.87±0.09**
		EU(10^2)	2.26±0.07	3.46±1.07	2.76±0.84	4.98±0.17	5.67±0.10	**5.74±0.07**
		SP(10^2)	0.95±0.73	15.37±10.67	**0.02±0.01**	10.60±6.31	5.09±1.77	4.53±1.79
	MO-Ant-2d	HV(10^5)	1.72±0.93	1.63±0.72	1.78±0.34	1.84±0.31	2.58±0.22	**2.68±0.22**
		EU(10^2)	2.68±1.19	2.81±0.80	2.18±0.41	3.09±0.30	3.81±0.27	**3.93±0.26**
		SP(10^3)	0.57±0.50	0.28±0.24	**0.10±0.09**	3.92±2.81	2.23±0.97	1.59±0.40

critical testbed for evaluating how rapidly different MORL strategies can discover effective Pareto front approximations. **(2) Standard-Training Setting:** To assess performance under more common training conditions, all methods, including our LLE-MORL, were trained under an interaction budget for 1×10^6 timesteps. This setting aligns with common practices for benchmarking in continuous control and allows us to assess the final quality of the Pareto fronts achieved by each algorithm after a more thorough learning process. In all experiments, we fix hyperparameters according to a common setting shared across methods, without assuming any task-specific parameter tuning. Full implementation details are provided in Appendix C.

In the sample-efficient setting, shown in Figure 4 and Table 1 top part, LLE-MORL achieves the highest HV and EU in all benchmarks, demonstrating strong capabilities in rapidly achieving high-quality Pareto fronts. Regarding SP, while LLE-MORL does not consistently achieve the leading scores on this metric, its performance generally reflects a good and effective distribution of solutions along the high-quality Pareto fronts it identifies. It should be noted that SP results can be confounded by a fragmentary recovery of the Pareto front. For instance, if Q-Pensieve discovers only two close points of the Pareto front for the MO-Ant problem, the sparsity rating is nearly perfect. Meanwhile, a low SP might also arise from solutions being overly clustered in a small region, as potentially seen with GPI-LS and Q-Pensieve in MO-Swimmer shown in Figure 4a.

In the standard-training setting, shown in Figure 4 and Table 1 bottom part, across all benchmarks, LLE-MORL achieves the highest HV and highly competitive EU, which suggests a more extensive and higher-quality approximation of the Pareto front than the baselines. Pareto front visualisation in Figure 4d further supports this result: on MO-Swimmer, LLE-MORL identifies solutions in the lower-right region that are missed by baselines such as GPI-LS, Q-Pensieve, and CAPQL. Notably, when comparing the GPI-LS, CAPQL and Q-Pensieve performance to those in the sample-efficient setting for the MO-Swimmer environment, these particular baselines appear to remain constrained by suboptimal solutions in this challenging region, indicating that simply extending training duration did not resolve their exploration deficiencies here. While LLE-MORL's thorough exploration to achieve this broader coverage means its SP may not be the numerically lowest, the result could be well-justified by the extensive nature of the front.

Overall, LLE-MORL consistently achieves high-quality Pareto fronts across different settings. Its innovative training-free extension step traces the local policy manifold structure to efficiently generate diverse solutions under limited budgets and consistently improved results with extended training. Runtime comparisons are reported in Appendix C.

4.3 Performance and Scalability in Higher Dimensional Objective Spaces

To evaluate the robustness and discuss the scalability of LLE-MORL beyond standard 2D-objective trade-offs, we extended our experimental evaluation to higher-dimensional scenarios using the MO-Hopper-3d and MO-Ant-3d benchmarks. In these environments, the Pareto front is no longer a

Table 2: Evaluation of the quality of the Pareto front on 3D-objective benchmarks.

Environment	Metric	Method					
		GPI-LS	CAPQL	Q-Pensieve	MORL/D	LLE-MORL-0	LLE-MORL
MO-Hopper-3d	HV(10^7)	5.78±5.49	4.53±2.64	5.23±3.89	8.97±0.09	10.20±0.36	**10.63±0.40**
	EU(10^2)	1.63±0.11	2.51±0.79	2.49±0.88	3.66±0.21	3.58±0.35	**3.69±0.23**
	SP(10^2)	**0.23±0.09**	29.04±16.92	48.44±37.93	11.64±3.24	1.33±0.98	1.35±1.14
MO-Ant-3d	HV(10^7)	1.10±0.37	2.49±0.14	0.71±0.36	2.69±0.01	3.38±0.38	**3.61±0.31**
	EU(10^2)	1.31±0.38	1.90±0.03	1.09±0.25	1.95±0.04	1.89±0.14	**1.96±0.15**
	SP(10^2)	27.79±22.62	3.72±3.70	51.57±51.73	10.38±15.85	0.65±0.28	**0.54±0.15**

one-dimensional curve but a two-dimensional manifold embedded in a three-dimensional performance space. All methods, including our LLE-MORL approach, were trained under an interaction budget for 2×10^6 timesteps (including all algorithmic training phases).

The experimental results, reported in Table 2, are highly consistent with the observations from our 2D benchmarks, showing that LLE-MORL maintains an advantage in HV and EU over baselines. The robust performance in higher dimensions is a direct consequence of the algorithm's design, which generalises to d-objective tasks without requiring additional algorithmic modifications. As established in our theoretical analysis (Section 3.4), the $d-1$ directional vectors obtained during the retraining phase are sufficient to explore and span the local Pareto manifold. This allows the locally linear extension mechanism to fill in contiguous patches of the Pareto surface effectively.

4.4 Ablation Study

The LLE-MORL integrates a locally linear extension process with a subsequent fine-tuning stage. To understand the distinct contributions of these components to the overall performance, our ablation study separates them. We first evaluate LLE-MORL-0, which solely employs the extension process without fine-tuning. As detailed in Table 1 and Table 2, LLE-MORL-0 itself demonstrates competitiveness, achieving strong HV and EU scores that are often competitive with or superior to baselines. This emphasises the efficacy of our extension mechanism in rapidly discovering a high-quality approximation of the Pareto front.

Subsequently, we assess the improvement of the fine-tuning stage by comparing LLE-MORL (which includes fine-tuning) to LLE-MORL-0. This comparison reveals that the inclusion of fine-tuning yields further improvements in HV and EU across all settings. The impact on SP is less uniform, which is an expected outcome, as refining solutions towards a more optimal Pareto front can alter their relative spacing. Nevertheless, the consistent enhancements in HV and EU prove the value of fine-tuning for improving the quality of the approximate Pareto front and its coverage by diverse solutions. This demonstrates that the extension process provides a strong foundation for the fine-tuning stage that enables LLE-MORL to outperform other algorithms.

5 Related work

Prior work in MORL offers various strategies for handling conflicting objectives. These can be broadly grouped into single-policy methods and multi-policy methods for approximating the Pareto front. Single-policy approaches typically convert the multi-objective problem into a single-objective task using a predefined preference or weighting scheme. A common instance of such a weighting scheme is linear scalarization [Van Moffaert et al., 2013]. Limitations of linear scalarization, particularly in capturing non-convex Pareto fronts, have been addressed by more advanced scalarization functions such as Chebyshev methods [Van Moffaert et al., 2013], hypervolume-based approaches [Zhang and Golovin, 2020] and the addition of concave terms to rewards [Lu et al., 2023]. Concurrently, significant efforts have developed generalised single-policy models conditioned on preference inputs to achieve adaptability across diverse objectives [Teh et al., 2017, Yang et al., 2019, Basaklar et al., 2023, Parisi et al., 2016], with subsequent extensions into offline learning contexts [Zhu et al., 2023, Lin et al., 2024] and methods to improve sample efficiency in these settings [Hung et al., 2023].

Multi-policy MORL strategies approximate the Pareto front by learning a diverse collection of policies. One direction is developing single models conditioned on preferences [Abels et al., 2019, Friedman and Fontaine, 2018, Kyriakis et al., 2022]. Other approaches explicitly learn a diverse set of policies or their value functions, including direct value-based methods like Pareto Q-learning [Van Moffaert

and Nowé, 2014], and evolutionary algorithms often guided by prediction models to discover a dense Pareto front [Xu et al., 2020]. Further techniques for generating policy sets involve Generalised Policy Improvement (GPI) for sample-efficient learning [Alegre et al., 2023] or the development of transferable policy components using representations like successor features [Alegre et al., 2022]. The use of constrained optimisation to efficiently complete and refine the Pareto front is also explored in [Liu et al., 2025, He et al., 2024]. Furthermore, the principles of decomposition-based strategies, which find a set of solutions by solving multiple interrelated scalarised sub-problems, have been a significant focus [Felten et al., 2024, Röpke et al., 2025].

While these single-policy and multi-policy paradigms have significantly advanced MORL, the characterisation and systematic exploitation of the structural relationship between the learned policies' underlying parameter space and their resultant performance on the Pareto front remain largely underexplored. Existing studies have explored several related directions. In multi-objective optimisation, the Pareto-navigation method [Ye and Liu, 2022] uses objective-gradient information to search for preferred solutions within a Pareto set, while model-merging approaches [Rame et al., 2023, Chen and Kwok, 2025] combine models optimised for different objectives to realise different trade-offs. Within MORL, prior studies have touched on Pareto-set or policy-manifold learning, which aims to learn a global continuous representation of policies across preferences. This includes directly optimising a continuous policy manifold [Parisi et al., 2016] and training a hypernetwork to map preferences to policy parameters [Shu et al., 2024]. Related work also discusses parameter-space regularities [Xu et al., 2020]. Further studies characterise the exact geometry of Pareto fronts by analysing neighbouring deterministic policies [Li et al., 2025], but only in known multi-objective MDPs that assume finite state and action spaces and known transition and reward models. Other studies investigate preference control [Yuan et al., 2025, Yang et al., 2025].

Although these directions provide related insights, they offer limited guidance on how to use local parameter-performance information for Pareto front reconstruction. More specifically, Pareto navigation, Pareto-set and policy-manifold learning remain primarily learning- or optimisation-based, treating the solution set as an object to be learned, represented, or optimised rather than exploiting its local parameter-performance geometry to guide reconstruction. Model-merging methods can combine discrete solutions, but this alone does not explicitly characterise the local Pareto-front manifold between them. Models independently optimised for different objectives do not by themselves establish how small parameter changes affect multi-objective performance, which is crucial for accurately tracing this manifold. In contrast, LLE-MORL explicitly characterises the local parameter-performance relationship and reuses this structure around a sparse set of trained policies. Brief preference-based retraining provides local parameter-space directions that guide training-free policy-manifold tracing and Pareto-front reconstruction.

6 Conclusion

We have discussed LLE-MORL, an algorithm that identifies solution components in MORL by directly tracing local policy manifolds along the Pareto front. The main benefit of LLE-MORL is the structured representation it provides in policy parameter space, maintaining a direct relation to multi-objective performance, and thereby enabling increased efficiency in Pareto front approximation. We have shown, through both theoretical analysis and empirical evaluation, that this framework is sufficient to reconstruct a high-quality Pareto front, and that LLE-MORL consistently outperforms recent MORL algorithms across a range of benchmark settings. Furthermore, as the LLE stage is a modular plug-in component, it can be integrated into existing deep RL algorithms as a training-free and computationally lightweight module, offering a simple path for future extensions.

Acknowledgments and Disclosure of Funding

This work was supported by the United Kingdom Research and Innovation (grant EP/S023208/1), UKRI Centre for Doctoral Training in Robotics and Autonomous Systems at the University of Edinburgh, School of Informatics. The authors sincerely thank Peize Li for providing additional computational resources to support the experiments, Haocheng Xiao for helpful discussions on figure presentation during the early stages of this work, and the anonymous reviewers for their insightful comments and suggestions.

References

Axel Abels, Diederik Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. Dynamic weights in multi-objective deep reinforcement learning. In *International Conference on Machine Learning*, pages 11–20. PMLR, 2019.

Lucas N Alegre, Ana LC Bazzan, Diederik M Roijers, Ann Nowé, and Bruno C da Silva. Sample-efficient multi-objective learning via generalized policy improvement prioritization. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 2003–2012, 2023.

Lucas Nunes Alegre, Ana Bazzan, and Bruno C Da Silva. Optimistic linear support and successor features as a basis for optimal policy transfer. In *International Conference on Machine Learning*, pages 394–413. PMLR, 2022.

Sarper Aydin and Ceyhun Eksin. Policy gradient play with networked agents in markov potential games. In Nikolai Matni, Manfred Morari, and George J. Pappas, editors, *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*, volume 211 of *Proceedings of Machine Learning Research*, pages 184–195. PMLR, 15–16 Jun 2023. URL `https://proceedings.mlr.press/v211/aydin23a.html`.

Toygun Basaklar, Suat Gumussoy, and Umit Ogras. PD-MORL: Preference-driven multi-objective reinforcement learning algorithm. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=zS9sRyaPFlJ`.

Jonathan Baxter and Peter L Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

Weiyu Chen and James Kwok. Pareto merging: Multi-objective optimization for preference-aware model merging. In *Forty-second International Conference on Machine Learning*, 2025. URL `https://openreview.net/forum?id=D7qRwx6BOS`.

Florian Felten, Lucas N. Alegre, Ann Nowé, Ana L. C. Bazzan, El Ghazali Talbi, Grégoire Danoy, and Bruno C. da Silva. A toolkit for reliable benchmarking and research in multi-objective reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 23671–23700, 2023.

Florian Felten, El-Ghazali Talbi, and Grégoire Danoy. Multi-objective reinforcement learning based on decomposition: A taxonomy and framework. *Journal of Artificial Intelligence Research*, 79: 679–723, 2024.

Eli Friedman and Fred Fontaine. Generalizing across multi-objective reward functions in deep reinforcement learning. *arXiv preprint arXiv:1809.06364*, 2018.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, Cambridge, 2016.

Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, et al. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.

Xiangkun He, Zhongxu Hu, Haohan Yang, and Chen Lv. Personalized robotic control via constrained multi-objective reinforcement learning. *Neurocomputing*, 565:126986, 2024.

Wei Hung, Bo Kai Huang, Ping-Chun Hsieh, and Xi Liu. Q-pensieve: Boosting sample efficiency of multi-objective RL through memory sharing of q-snapshots. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=AwWaBXLIJE`.

Harold W Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

Panagiotis Kyriakis, Jyotirmoy Deshmukh, and Paul Bogdan. Pareto policy adaptation. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=wfZGut6e09`.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

Yining Li, Peizhong Ju, and Ness Shroff. How to find the exact Pareto front for multi-objective MDPs? In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 99956–99986, 2025.

Qian Lin, Chao Yu, Zongkai Liu, and Zifan Wu. Policy-regularized offline multi-objective reinforcement learning. In *AAMAS*, pages 1201–1209, 2024. URL `https://dl.acm.org/doi/10.5555/3635637.3662977`.

Ruohong Liu, Yuxin Pan, Linjie Xu, Lei Song, Pengcheng You, Yize Chen, and Jiang Bian. Efficient discovery of Pareto front for multi-objective reinforcement learning. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 69652–69678, 2025.

Haoye Lu, Daniel Herman, and Yaoliang Yu. Multi-objective reinforcement learning: Convexity, stationarity and Pareto optimality. In *The Eleventh International Conference on Learning Representations*, 2023.

James Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

Simone Parisi, Matteo Pirotta, and Marcello Restelli. Multi-objective reinforcement learning through continuous Pareto manifold approximation. *Journal of Artificial Intelligence Research*, 57:187–227, 2016.

Antonin Raffin, Ashley Hill, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, and Noah Dormann. PyTorch version of stable baselines. `https://github.com/DLR-RM/stable-baselines3`, 2019.

Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. Rewarded soups: towards Pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 71095–71134. Curran Associates, Inc., 2023.

Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

Willem Röpke, Mathieu Reymond, Patrick Mannion, Diederik M Roijers, Ann Nowé, and Roxana Rădulescu. Divide and conquer: Provably unveiling the Pareto front with multi-objective reinforcement learning. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '25, page 1774–1783, Richland, SC, 2025. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9798400714269.

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

Junru Shi, Xin Wang, Mingchuan Zhang, Muhua Liu, Junlong Zhu, and Qingtao Wu. A distributed adaptive policy gradient method based on momentum for multi-agent reinforcement learning. *Complex & Intelligent Systems*, 10(5):7297–7310, 2024.

Tianye Shu, Ke Shang, Cheng Gong, Yang Nan, and Hisao Ishibuchi. Learning pareto set for multi-objective continuous robot control. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24, 2024. ISBN 978-1-956792-04-1. doi: 10.24963/ijcai.2024/544. URL `https://doi.org/10.24963/ijcai.2024/544`.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

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

Peter Vamplew, John Yearwood, Richard Dazeley, and Adam Berry. On the limitations of scalarisation for multi-objective reinforcement learning of Pareto fronts. In *AI 2008: Advances in Artificial Intelligence: 21st Australasian Joint Conference on Artificial Intelligence Auckland, New Zealand, December 1-5, 2008. Proceedings 21*, pages 372–378. Springer, 2008.

Kristof Van Moffaert and Ann Nowé. Multi-objective reinforcement learning using sets of Pareto dominating policies. *The Journal of Machine Learning Research*, 15(107):3663–3692, 2014.

Kristof Van Moffaert, Madalina M Drugan, and Ann Nowé. Scalarized multi-objective reinforcement learning: Novel design techniques. In *2013 IEEE symposium on adaptive dynamic programming and reinforcement learning (ADPRL)*, pages 191–199. IEEE, 2013.

Jie Xu, Yunsheng Tian, Pingchuan Ma, Daniela Rus, Shinjiro Sueda, and Wojciech Matusik. Prediction-guided multi-objective reinforcement learning for continuous robot control. In *International Conference on Machine Learning*, pages 10607–10616. PMLR, 2020.

Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. *Advances in Neural Information Processing Systems*, 32, 2019.

Yucheng Yang, Tianyi Zhou, Mykola Pechenizkiy, and Meng Fang. Preference controllable reinforcement learning with advanced multi-objective optimization. In *Forty-second International Conference on Machine Learning*, 2025. URL `https://openreview.net/forum?id=49g4c8MWHy`.

Mao Ye and Qiang Liu. Pareto navigation gradient descent: A first-order algorithm for optimization in Pareto set. In *Uncertainty in Artificial Intelligence*, pages 2246–2255. PMLR, 2022.

Yifu Yuan, Zhenrui Zheng, Zibin Dong, and Jianye HAO. MODULI: Unlocking preference generalization via diffusion models for offline multi-objective reinforcement learning. In *Forty-second International Conference on Machine Learning*, 2025. URL `https://openreview.net/forum?id=V364YjXm3T`.

Richard Zhang and Daniel Golovin. Random hypervolume scalarizations for provable multi-objective black box optimization. In *International Conference on Machine Learning*, pages 11096–11105. PMLR, 2020.

Baiting Zhu, Meihua Dang, and Aditya Grover. Scaling Pareto-efficient decision making via offline multi-objective RL. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=Ki4ocDm364`.

Luisa M Zintgraf, Timon V Kanters, Diederik M Roijers, Frans Oliehoek, and Philipp Beau. Quality assessment of MORL algorithms: A utility-based approach. In *Benelearn 2015: Proceedings of the 24th Annual Machine Learning Conference of Belgium and the Netherlands*, 2015.

Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999. doi: 10.1109/4235.797969.

A Algorithm

Algorithm 1 LLE-MORL (General d-Objective Case)

Require: Initial scalarization weights $\{\omega_k\}_{k=1}^{K}$ evenly spanning the preference space, Directional retraining weights $\{\omega_k^{(i)}\}_{k=1,i=1}^{K,m}$, Initialization training length T_{init}, Directional retraining length T_{dir}, Fine-tuning length T_{ref}, Step-scale factors $\alpha_{\text{start}}, \alpha_{\text{end}}, \Delta\alpha$

Ensure: Approximate Pareto-optimal policy set Π

$\Pi \leftarrow \emptyset$
$m \leftarrow d - 1$
$M \leftarrow \left\lfloor \frac{\alpha_{\text{end}} - \alpha_{\text{start}}}{\Delta\alpha} \right\rfloor + 1$

Initialization:
for $k = 1$ to K **do**
    Train base policy θ_{ω_k} with PPO under weight ω_k for T_{init} steps
end for

Directional Retraining:
for $k = 1$ to K **do**
    for $i = 1$ to m **do**
        $\theta_k^{(i)} \leftarrow$ continue training θ_{ω_k} under $\omega_k^{(i)}$ for T_{dir} steps
        $\Delta\theta_k^{(i)} \leftarrow \theta_k^{(i)} - \theta_{\omega_k}$
        $\Delta\omega_k^{(i)} \leftarrow \omega_k^{(i)} - \omega_k$
    end for
end for

Locally Linear Extension:
$\mathcal{C} \leftarrow \emptyset$
for $k = 1$ to K **do**
    for all combination of α_i on the grid $[\alpha_{\text{start}}, \alpha_{\text{end}}, \Delta\alpha]$ **do**
        $\theta_{k,\text{cand}} \leftarrow \theta_{\omega_k} + \sum_{i=1}^{m} \alpha_i \, \Delta\theta_k^{(i)}$
        $\omega_{k,\text{cand}} \leftarrow \omega_k + \sum_{i=1}^{m} \alpha_i \, \Delta\omega_k^{(i)}$
        Evaluate multi-objective performance $V(\theta_{k,\text{cand}})$ under weight $\omega_{k,\text{cand}}$
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{(\theta_{k,\text{cand}}, \omega_{k,\text{cand}})\}$
    end for
end for

Candidate Selection:
$\mathcal{N} \leftarrow$ non-dominated subset of $\mathcal{C}$

Preference-Aligned Fine-Tuning:
$\mathcal{F} \leftarrow \emptyset$
for all $(\theta, \omega) \in \mathcal{N}$ **do**
    fine-tune θ for T_{ref} steps under ω, yielding θ'
    add (θ', ω) to $\mathcal{F}$
end for
$\mathcal{C}_{\text{all}} \leftarrow \mathcal{N} \cup \mathcal{F}$
$\mathcal{N}_{\text{final}} \leftarrow$ non-dominated subset of $\mathcal{C}_{\text{all}}$
$\Pi \leftarrow \Pi \cup \{\theta \mid (\theta, \cdot) \in \mathcal{N}_{\text{final}}\}$

return Π

In this section, we present a complete description of LLE-MORL, an efficient procedure for tracing an approximate Pareto front in a d-objective setting. Algorithm 1 details this process.

The core pipeline involves several stages. First, K base policies $\{\theta_{\omega_k}\}$ are trained, each under its respective initial weight ω_k for T_{init} steps. Next, for each base policy θ_{ω_k}, we perform directional retraining to obtain $m = d - 1$ local directions: continuing training under nearby preferences

$\omega_k^{(i)}$ for T_{dir} steps yields directional policies $\theta_k^{(i)}$, from which we compute the parameter updates $\Delta\theta_k^{(i)} = \theta_k^{(i)} - \theta_{\omega_k}$ and preference shifts $\Delta\omega_k^{(i)} = \omega_k^{(i)} - \omega_k$. In the Locally Linear Extension stage, we generate a candidate set $\mathcal{C}$ by combining these directions with step-scale factors $\{\alpha_i\}$ chosen from a grid $[\alpha_{\text{start}}, \alpha_{\text{end}}]$ with step size $\Delta\alpha$, producing up to M^m candidates per base policy together with matched weights $\omega_{k,\text{cand}}$. We then select the non-dominated subset $\mathcal{N} \subset \mathcal{C}$ and apply Preference-Aligned Fine-Tuning for T_{ref} steps to obtain refined policies $\mathcal{F}$. Finally, LLE-MORL returns Π, the non-dominated set extracted from $\mathcal{N} \cup \mathcal{F}$, which forms the final approximate Pareto front.

B Theoretical Analysis

Throughout, $\|\cdot\|$ denotes the Euclidean norm on vectors by default, other norms on matrices are defined explicitly in the context.

Definition B.1. *Parameter-Performance Relationship (PPR). Let $\Theta \subseteq \mathbb{R}^n$ be the space of policy parameters, and V be the expected discounted return as a function $V : \Theta \to \mathbb{R}^d$ of the parameter θ. We say that V exhibits a* parameter–performance relationship *on an open region $U \subseteq \Theta$ if $\exists \delta > 0$ and a function $h : U \times B_\delta(0) \to \mathbb{R}^d$,* s.t. *$\forall \theta \in U$ and perturbation $\Delta\theta$ satisfying $\|\Delta\theta\| < \delta$ and $\theta + \Delta\theta \in U$, the difference in the performance space is:*

$$V(\theta + \Delta\theta) \; - \; V(\theta) \; = \; h\big(\theta, \Delta\theta\big). \tag{5}$$

To clarify when LLE-MORL can reliably reconstruct a Pareto front, we present the theoretical analysis here. We consider here the Pareto fronts P that are differentiable manifolds, meaning that they are locally similar to Euclidean space and derivatives can be defined. As a slight generalisation, we can assume that the Pareto front is a non-connected manifold or a set of manifolds, including zero-dimensional manifolds (discrete points) or sets of manifolds of different dimensions. We will not further discuss this generalisation apart from mentioning the fact that it is possible to reach all components and to produce sufficiently many charts of each of the components of the Pareto front. For the theoretical framework, we propose the following assumptions:

Assumption B.1. *(Manifold Assumption) The Pareto front P is a set of differentiable manifold branches embedded in the performance space, which consists of all non-dominated points.*

Assumption B.2. *(Continuity of PPR function h) The function h in definition 3.1 is Lipschitz continuous with respect to the second variable $\Delta\theta$.*

Remark on Assumption B.2: This assumption can be reduced to more basic conditions on the expected discount return V, where V is associated with the training phase through the PPO algorithm. Such a reduction is implemented by Theorem B.1.

Assumption B.3. *(Initialization) The initialisation in Algorithm 1 stated in Section 3.3 identifies K different points on the optimal Pareto front. Thus, we have at least one accurate initial point for each branch of the Pareto front.*

Theorem B.1. *(Lipschitz Continuity of PPR function h, corresponding to Theorem 3.1) Denote the expected discounted return by $V : U \to \mathbb{R}^d$ where $U \subseteq \Theta \subseteq \mathbb{R}^n$ is an open set. The sufficient conditions for Lipschitz continuity of $h(\theta, \Delta\theta)$ with respect to the second variable $\Delta\theta$ are:*

(i) $V(\theta)$ is differentiable on U.
(ii) The score function is bounded, i.e. $\forall a \in \mathcal{A}, s \in \mathcal{S}$

$$\sup_{\theta \in U} \big\| \nabla_\theta \log \pi_\theta(a|s) \big\| \leq \mathcal{G} < \infty. \tag{6}$$

The inequality (ii) states that the score function Baxter and Bartlett [2001] is bounded by a constant $\mathcal{G}$. This is a normal regularity assumption in related research Aydin and Eksin [2023], Shi et al. [2024].

Proof: For a fixed $\theta \in U$, define $h_\theta(\Delta) := V(\theta + \Delta) - V(\theta)$. Suppose the objective return is:

$$V(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t \boldsymbol{r}(s_t, a_t)\right] \in \mathbb{R}^d, \tag{7}$$

with bounded rewards $\|\boldsymbol{r}(s,a)\| \leq \mathcal{R}$, and $\pi_\theta(a|s)$ is a differentiable stochastic policy such that the score function $\nabla_\theta \log \pi_\theta(a|s)$ Baxter and Bartlett [2001] is locally bounded on a neighbourhood

U of θ, i.e. Equation 6. Denote by $\tau_t = (s_0, a_0, \ldots, s_t, a_t)$ to be the trajectory up to time t, with density $p_\theta(\tau_t)$ on all possible trajectories $\mathcal{T}_t = (\mathcal{S} \times \mathcal{A})^{t+1}$.

Take the gradient of V about θ and make transformations:

$$\nabla_\theta V(\theta) = \nabla_\theta \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t \boldsymbol{r}(s_t, a_t)\right] \tag{8}$$

$$= \sum_{t=0}^{\infty} \gamma^t \nabla_\theta \mathbb{E}_{\pi_\theta}[\boldsymbol{r}(s_t, a_t)]. \tag{9}$$

Further, the gradient of the reward expectation in Equation (9) can be represented by

$$\nabla_\theta \mathbb{E}_{\pi_\theta}[\boldsymbol{r}(s_t, a_t)] = \nabla_\theta \int_{\mathcal{T}_t} \boldsymbol{r}(s_t, a_t) p_\theta(\tau_t)\, d\tau_t \tag{10}$$

$$= \int_{\mathcal{T}_t} \boldsymbol{r}(s_t, a_t) \cdot \nabla_\theta p_\theta(\tau_t)\, d\tau_t \tag{11}$$

$$= \int_{\mathcal{T}_t} \boldsymbol{r}(s_t, a_t) p_\theta(\tau_t) \cdot \nabla_\theta \log p_\theta(\tau_t)\, d\tau_t \tag{12}$$

$$= \mathbb{E}_{\pi_\theta}\left[\boldsymbol{r}(s_t, a_t) \cdot \nabla_\theta \log p_\theta(\tau_t)\right]. \tag{13}$$

To handle the score function appears in Equation (13), consider the joint distribution of trajectory τ_t

$$p_\theta(\tau_t) = \rho(s_0) \prod_{k=0}^{t} \pi_\theta(a_k \mid s_k) \prod_{k=0}^{t-1} P(s_{k+1} \mid a_k, s_k). \tag{14}$$

Taking the logarithm of both sides of Equation (14) gives

$$\nabla_\theta \log p_\theta(\tau_t) = \nabla_\theta \log \rho(s_0) + \sum_{k=0}^{t} \nabla_\theta \log \pi_\theta(a_k \,|\, s_k) \tag{15}$$

$$+ \sum_{k=0}^{t-1} \nabla_\theta \log P(s_{k+1} \,|\, a_k, s_k) \tag{16}$$

$$= \sum_{k=0}^{t} \nabla_\theta \log \pi_\theta(a_k \mid s_k). \tag{17}$$

The first term is initial distribution and the third term is state transfer related to environment, they are not dependent on policy, thus has zero gradient.

Combine Equation (9,13,17) together, the gradient of V is

$$\nabla_\theta V(\theta) = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\pi_\theta}\left[\boldsymbol{r}(s_t, a_t) \cdot \nabla_\theta \log p_\theta(\tau_t)\right] \tag{18}$$

$$= \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\pi_\theta}\left[\boldsymbol{r}(s_t, a_t) \cdot \sum_{k=0}^{t} \nabla_\theta \log \pi_\theta(a_k \mid s_k)\right] \tag{19}$$

$$= \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t \boldsymbol{r}(s_t, a_t) \cdot \sum_{k=0}^{t} \nabla_\theta \log \pi_\theta(a_k \mid s_k)\right]. \tag{20}$$

Above boundedness conditions on reward $\|\boldsymbol{r}\| \leq \mathcal{R}$ and score function in Equation (6) implies a uniform bound of $\nabla_\theta V$ by substituting in to Equation (20):

$$\|\nabla_\theta V(\theta)\| \leq \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(t+1)\mathcal{G} = \frac{\mathcal{R}\mathcal{G}}{(1-\gamma)^2}. \tag{21}$$

Hence V is locally Lipschitz on U. Further, let $L_V := \sup_{\theta \in U} \|\nabla_\theta V(\theta)\| \leq \frac{\mathcal{R}\mathcal{G}}{(1-\gamma)^2}$, we have:

$$\|h_\theta(\Delta_1) - h_\theta(\Delta_2)\| = \|V(\theta + \Delta_1) - V(\theta + \Delta_2)\| \leq L_V \|\Delta_1 - \Delta_2\|. \tag{22}$$

Hence $h_\theta(\Delta) = h(\theta, \Delta)$ is Lipschitz on the ball $\Delta \in B_\delta(0)$ with $\theta + B_\delta(0) \subset U$: In particular, Assumption B.2 holds with Lipschitz constant $L_h \leq L_V \leq R_{\max} G/(1-\gamma)^2$ under differentiability of V and bounded score function conditions. □

Theorem B.1 derives the Lipschitz continuity of PPR function h. Hence, small perturbations in the policy parameters only induce controllable small changes in the multi-objective performance vector V. In particular, along any path in the parameter space contained in the neighbour $\theta + B_\delta(0)$ of a fixed θ, the image under $V : \Theta \to \mathbb{R}^d$ varies continuously and is bounded. This property is significant in reconstructing the Pareto surface produced by the LLE algorithm: with a suitable small step size, expanded policy candidates map to a controlled neighbourhood of the performance from initialisation. This guarantee is embodied and verified in the empirical results shown in Figure 3. Theorem B.1 serves as one of the theoretical foundations that guarantees the performance of the central algorithm is stable in approximating and constructing the Pareto front.

Corollary B.1. *(Surface continuity guaranteed by Lipschitz PPR) Algorithm LLE can construct a continuous surface in the performance space, as an approximation to the Pareto front, given appropriate regularity conditions and a range of α.*

Proof: Suppose Theorem B.1 holds at a fixed θ, i.e., $\exists\ \delta, L_V > 0$ such that for all Δ_1, Δ_2 with $\|\Delta_1\|, \|\Delta_2\| \leq \delta$, we have:

$$\|V(\theta + \Delta_1) - V(\theta + \Delta_2)\| \leq L_V \|\Delta_1 - \Delta_2\|.$$

For any two parameters $\theta_1, \theta_2 \in \theta + B_\delta(0)$, if $\|V(\theta_1) - V(\theta_2)\| \geq \varepsilon > 0$, we have $\|\theta_1 - \theta_2\| \geq \varepsilon / L$. In particular, within the PPR neighbourhood $\theta + B_\delta(0)$, solutions that are separated obviously in the performance space must correspond to separated policies set in the parameter space, so LLE with sufficiently small step size cannot induce a spurious large displacement in performance space. □

Corollary B.2. *(Lipschitz continuity of V) If Theorem B.1 holds, then $V(\theta)$ is also Lipschitz continuous for $\theta \in U \subseteq \Theta$.*

Proof: $\forall \theta_1, \theta_2 \in U \subseteq \Theta$, in Equation 22, substitute $\theta = \theta_2$, $\Delta_1 = \theta_1 - \theta_2$, $\Delta_2 = 0$. Then we immediately have:

$$\|V(\theta_1) - V(\theta_2)\| \leq L_V \|\theta_1 - \theta_2\|. \tag{23}$$

Thus, $V(\theta)$ is Lipschitz continuous with constant L_V. □

Condition B.1. *(Chart of Pareto manifold) Based on Assumption B.1, $\mathcal{P} \in \mathcal{M}^m(\mathbb{R}^d)$, where the dimension of $\mathcal{P}$ is $m = d - 1$, and $\forall v \in \mathcal{P}$ is non-dominated. By definition, $\forall v \in \mathcal{P}$, there exists an open neighbourhood $P_v \subset \mathcal{P}$ and a chart:*

$$\varphi_v : P_v \to \Omega_v \subset \mathbb{R}^{d-1} \tag{24}$$

where Ω_v is an open set. Without loss of generality, we choose the chart such that $\varphi_v(v) = 0$. And φ_v is a C^k diffeomorphism ($k \geq 1$), with the inverse mapping is denoted by: $\varphi_v^{-1} : \Omega_v \to P_v \subset \mathbb{R}^d$.

Condition B.2. *(Local lifting of a Pareto neighbourhood) Based on Condition B.1, on the Pareto front, consider a fixed $v \in \mathcal{P}$ and its open neighbourhood $P_v \subset \mathcal{P}$. Suppose a corresponding policy parameter θ_v satisfying $V(\theta_v) = v$. Define an equivalence relation $\sim$ on Θ by $\theta_1 \sim \theta_2$ if $V(\theta_1) = V(\theta_2)$, and denote the quotient space by $\Theta/\sim$. Assume there exist $\Theta_v \subset \Theta$ consisting of the representative elements of $\Theta/\sim$, satisfying the following conditions:*

$$(i)\ \exists \delta > 0,\ s.t.\ \Theta_v \subseteq B_\delta(\theta_v). \tag{25}$$

$$(ii)\ V[\Theta_v] = P_v. \tag{26}$$

$$(iii)\ \forall \theta \in \Theta_v,\ |\Theta_v \cap [\theta]| = 1. \tag{27}$$

where $B_\delta(\theta_v)$ denote an open ball with radius δ centres on θ_v, $[\theta] \in \Theta/\sim$ represent the equivalence class of θ, and $|\cdot|$ denote the cardinality of set. By (ii), $V|_{\Theta_v} : \Theta_v \twoheadrightarrow P_v$ is a surjection. By (iii), we have $V|_{\Theta_v} : \Theta_v \hookrightarrow P_v$ is an injection. Therefore, $V|_{\Theta_v} : \Theta_v \to P_v$ is a bijection. Assume further that $\theta_v \in \Theta_v \subset U$ and $(V|_{\Theta_v})^{-1} : P_v \to \Theta_v \subset \mathbb{R}^n$ is a C^k map ($k \geq 2$).

Condition B.2 serves two purposes. Firstly, the localisation $\Theta_v \subset B_\delta(\theta_v)$ and $V(\Theta_v) = P_v$ ensures that the entire Pareto neighbourhood P_v can be lifted to a region in parameter space, so the algorithm does not need to traverse distant solution components. This is the prerequisite under which local linear extrapolation in parameter space is meaningful for approximating the same Pareto branch. Second,

the representative constraint $|\Theta_v \cap [\theta]| = 1$ removes the multi-to-one ambiguity of V by selecting exactly one representative from each equivalence class, so that notions such as parameter proximity, difference directions, and extrapolation updates are well-defined on the lifted set. More specifically, it formulates an injection from Θ_v to the Pareto neighbourhood of v. Without this condition, a single performance could correspond to many distant parameters and retrained $\Delta\theta$ directions would be unstable and unreliable in exploring the Pareto front.

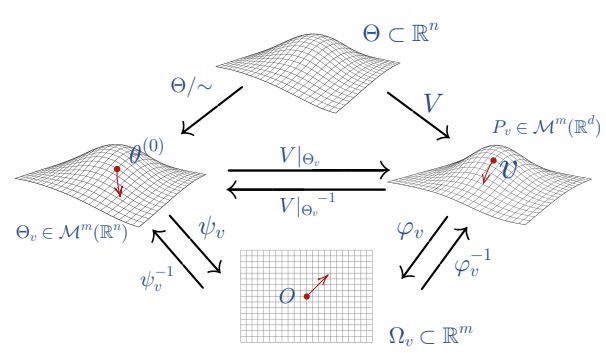


Figure 5: Commutative diagram between Pareto front, parameter manifold and their local lifting. With each initialisation $\theta^{(0)}$, we denote its performance v and the neighbour on the Pareto front manifold to be P_v. Based on our assumptions and conditions, the local lifting, i.e. Θ_v is also a manifold of the same dimension as P_v, located in the neighbourhood of $\theta^{(0)}$. The Linear extrapolation corresponds to the approximation of the tangent space of Θ_v. Each time, LLE-MORL reconstructs the Pareto front branch where v is located. With the appropriate choices of initialization $\theta_k^{(0)}$, it reconstructs all of the branches where the corresponding $v_k = V\left(\theta_k^{(0)}\right)$ is located.

Theorem B.2. *(Local lifting manifold, w.r.t. Theorem 3.2) Let $P_v \subset \mathcal{P}$ be as in Condition B.1, where $P_v \in \mathcal{M}^m(\mathbb{R}^d)$ with a C^k chart $\varphi_v : P_v \to \Omega_v \subset \mathbb{R}^m$. Suppose $V|_{\Theta_v} : \Theta_v \to P_v$ is a bijection, V is differentiable on U and $(V|_{\Theta_v})^{-1} : P_v \to \mathbb{R}^n$ is C^k as in Condition B.2. Then we have Θ_v defined in Condition B.2 is also a differential manifold $\mathcal{M}^m(\mathbb{R}^n)$, with an C^k chart $\psi_v : \Theta_v \to \Omega_v \subset \mathbb{R}^m$.*

Proof: Fix a C^k chart $\varphi_v : P_v \to \Omega_v \subset \mathbb{R}^m$ from Condition B.1. Define $\psi_v := \varphi_v \circ V|_{\Theta_v} : \Theta_v \to \Omega_v$ and $\psi_v^{-1} := (V|_{\Theta_v})^{-1} \circ \varphi_v^{-1} : \Omega_v \to \mathbb{R}^n$. By conditions on $(V|_{\Theta_v})^{-1}$ and φ_v^{-1}, ψ_v^{-1} is also C^k and injective. It follows ψ_v is continuous since $V|_{\Theta_v}$ and φ_v are continuous.

Consider the composition mapping $V|_{\Theta_v} \circ \psi_v^{-1} = \varphi_v^{-1}$, chain rule gives

$$DV\left(\psi_v^{-1}(x)\right) \cdot D\psi_v^{-1}(x) = D\varphi_v^{-1}(x). \tag{28}$$

As φ_v^{-1} is a local parametrisation of an m-dimensional embedded manifold by Condition B.1, it follows that $\operatorname{rank}\big(D\varphi_v^{-1}(x)\big) = m$. By the rank inequality for matrix products,

$$m = \operatorname{rank}\big(D\varphi_v^{-1}(x)\big) \leq \operatorname{rank}\big(D\psi_v^{-1}(x)\big) \leq m, \tag{29}$$

where the last inequality follows from $D\psi_v^{-1}(x) \in \mathbb{R}^{n\times m}$. Therefore, $\operatorname{rank}\big(D\psi_v^{-1}(x)\big) = m$. Hence ψ_v^{-1} is an injective immersion with continuous inverse ψ_v, and therefore a C^k embedding. Consequently, Θ_v is an m-dimensional C^k manifold embedded in $\mathbb{R}^n$, i.e. $\Theta_v \in \mathcal{M}^m(\mathbb{R}^n)$. ψ_v is a C^k diffeomorphism between Θ_v and the open set $\Omega_v \subset \mathbb{R}^m$. □

Condition B.3. *(Local extension directions, corresponding to Condition 3.1) Fix $v \in \mathcal{P}$, an open neighbourhood $P_v \subset \mathcal{P}$ as in Condition B.1, and the corresponding lifted set $\Theta_v \subset \Theta$ from Condition B.2. Suppose the initialization returns $\theta^{(0)} \in \Theta_v$ with $V(\theta^{(0)}) = v$ by Assmption B.3. Applying $m = \dim \mathcal{P}$ distinct nearby preference perturbations and directional retraining from $\theta^{(0)}$, one obtains $\theta^{(i)} \in \Theta_v$, $i = 1, \ldots, m$, such that*

$$\mathcal{D} := \left[\theta^{(1)} - \theta^{(0)},\ \theta^{(2)} - \theta^{(0)},\ \ldots,\ \theta^{(m)} - \theta^{(0)}\right] \in \mathbb{R}^{n\times m} \tag{30}$$

has full column rank, i.e. $\operatorname{rank}(\mathcal{D}) = m$. The approximation of these finite-difference directions by tangent vectors is quantified in Lemma 1. This explains Step (2) in the Algorithm 1 in Section 3.3.

Lemma 1. *(Coordinates of linear extension and error matrix) Let ψ_v be as in Theorem B.2, without loss of generality, define $\psi_v(\theta^{(0)}) = 0$, and denote $x^{(i)} := \psi_v(\theta^{(i)})$ and $X := [x^{(1)}, \dots, x^{(m)}]$. Let $\mathcal{J} := D\psi_v^{-1}(0)$ be the Jacobian at the origin of ψ_v^{-1}, and suppose ψ_v^{-1} is C^2 with $\|D^2(\psi_v^{-1})(x)\| \leq \eta$ on a convex neighbourhood of 0 contained in Ω_v and containing all $x^{(i)}$, $i = 1, \dots, m$.*

Define $\mathcal{D}$ as in Condition B.3. Then there exists error matrix $E \in \mathbb{R}^{n\times m}$ such that: $\mathcal{D} = \mathcal{J}X + E$ and its L_2 norm is controlled by:

$$\|E\|_2 \leq \frac{\eta}{2}\sqrt{m} \max_{1\leq i\leq m} \|x^{(i)}\|^2. \tag{31}$$

Let $\sigma_{\min}(\mathcal{D})$ be the smallest singular value of $\mathcal{D}$. If $\|E\|_2 < \sigma_{\min}(\mathcal{D})$, then X is invertible.

proof: (I). Apply Taylor expansion of ψ_v^{-1} at 0, where $\psi_v^{-1}(0) = \theta^{(0)}$:

$$\psi_v^{-1}(x^{(i)}) = \psi_v^{-1}(0) + \mathcal{J}x^{(i)} + r^{(i)} \tag{32}$$

where $r^{(i)}$ are remainder of the second order. Equivalently,

$$\theta^{(i)} - \theta^{(0)} = \mathcal{J}x^{(i)} + r^{(i)} \tag{33}$$

Arrange all formulas of $1 \leq i \leq m$ in matrix form, we have:

$$\mathcal{D} = \mathcal{J}X + E \tag{34}$$

Matrix $E := \left[r^{(1)}, r^{(2)}, \dots, r^{(m)}\right] \in \mathbb{R}^{n\times m}$ is named the error matrix as it gives the error of the linear approximation. Matrix X collects the chart coordinates of the retraining points $\theta^{(1)}, \dots, \theta^{(m)}$.

Next, we derive the upper bound of E. Denote the components of $\psi_v^{-1} = \left[\psi_1^{-1}, \dots, \psi_n^{-1}\right]$ with $\psi_\ell^{-1} : \Omega_v \subset \mathbb{R}^m \to \mathbb{R}$. For $x \in \Omega_v$ and $u, w \in \mathbb{R}^m$, define the second derivative as a bilinear map

$$D^2(\psi_v^{-1})(x)[u, w] := \left[u^\top \nabla^2\psi_1^{-1}(x)\, w,\ \dots,\ u^\top \nabla^2\psi_n^{-1}(x)\, w\right]^\top \in \mathbb{R}^n, \tag{35}$$

where $\nabla^2\psi_\ell^{-1}(x) = \left[\frac{\partial^2\psi_\ell^{-1}}{\partial x_i \partial x_j}(x)\right]_{i,j=1}^m$ is the Hessian of ψ_ℓ^{-1}. Accordingly, define the operator norm:

$$\|D^2(\psi_v^{-1})(x)\| := \sup_{u,w\neq 0} \frac{\|D^2(\psi_v^{-1})(x)[u, w]\|_2}{\|u\| \cdot \|w\|} \tag{36}$$

By Taylor's theorem, with the notation above, the reminder of ψ_v^{-1}, i.e. $r^{(i)}$ has an integral form:

$$\psi_v^{-1}(x^{(i)}) = \psi_v^{-1}(0) + \mathcal{J}x^{(i)} + \int_0^1 (1-t)D^2(\psi_v^{-1})(tx^{(i)})[x^{(i)}, x^{(i)}]dt. \tag{37}$$

Thus,

$$\|r^{(i)}\| \leq \int_0^1 (1-t)\, \|D^2(\psi_v^{-1})(tx^{(i)})[x^{(i)}, x^{(i)}]\|\, dt \tag{38}$$

$$\leq \int_0^1 (1-t)\, \eta\|x^{(i)}\|^2\, dt \tag{39}$$

$$= \frac{\eta}{2}\|x^{(i)}\|^2. \tag{40}$$

Then we can control the norm of E, i.e. Equation 31, by the upper bound of its Frobenius norm:

$$\|E\|_2 \leq \|E\|_F = \left(\sum_{i=1}^m \|r^{(i)}\|^2\right)^{\frac{1}{2}} \leq \frac{\eta}{2}\left(\sum_{i=1}^m \|x^{(i)}\|^4\right)^{\frac{1}{2}} \leq \frac{\eta}{2}\sqrt{m} \max_{1\leq i\leq m} \|x^{(i)}\|^2. \tag{41}$$

(II). If X were singular, there would exist $u \in \mathbb{R}^m,\ u \neq 0$ with $Xu = 0$, hence $\mathcal{D}u = Eu$. Therefore

$$\|\mathcal{D}u\| \leq \|E\|_2 \cdot \|u\|. \tag{42}$$

On the other hand, by definition of $\sigma_{\min}(\mathcal{D})$,

$$\|\mathcal{D}u\| = \|u\| \cdot \left\|\mathcal{D}\frac{u}{\|u\|}\right\| \geq \|u\| \cdot \inf_{\|x\|=1} \|\mathcal{D}x\|_2 \geq \sigma_{\min}(\mathcal{D})\, \|u\|. \tag{43}$$

Thus $\sigma_{\min}(\mathcal{D}) \le \|E\|_2$, contradicting $\|E\|_2 < \sigma_{\min}(\mathcal{D})$. Therefore X is invertible. Furthermore, for every $u \in \mathbb{R}^m$, there is

$$\|\mathcal{D}u - \mathcal{J}Xu\| = \|Eu\| \le \|E\|_2\|u\| \le \frac{\|E\|_2}{\sigma_{\min}(\mathcal{D})}\|\mathcal{D}u\|.$$

Since $T_{\theta^{(0)}}\Theta_v = \mathrm{range}(\mathcal{J})$, we have $\mathcal{J}Xu \in T_{\theta^{(0)}}\Theta_v$. Thus, the relative approximation error is small when $\|E\|_2/\sigma_{\min}(\mathcal{D})$ is small. □

All above, the preparation work is finished. We now focus on the core theorem supporting the effectiveness of the LLE algorithm. Notations in previous deductions are continuously adopted.

Theorem B.3. *(Local Pareto error of LLE, corresponding to Theorem 3.3) Denote the Linear exploration by $\tilde{\theta}(\alpha) := \theta^{(0)} + \mathcal{D}\alpha$. Then for a suitable range of $\alpha \in \mathbb{R}^m$ with $X\alpha \in \Omega_v$, the error of exploration to the actual Pareto front is controlled by:*

$$\mathrm{dist}\big(V(\tilde{\theta}(\alpha)),\, \mathcal{P}_v\big) \le \frac{L_V\eta}{2} \max_{1\le i\le m} \|x^{(i)}\|^2\Big(\sqrt{m}\,\|\alpha\| + m\,\|\alpha\|^2\Big). \tag{44}$$

Proof: Define the point-to-set distance in performance space by

$$\mathrm{dist}(z, \mathcal{P}_v) := \inf_{p\in\mathcal{P}_v} \|z - p\|, \qquad z \in \mathbb{R}^d. \tag{45}$$

Let $\psi_v : \Theta_v \to \Omega_v$ be the C^2 chart from Theorem B.2 with $\psi_v(\theta^{(0)}) = 0$ and set $\theta^\star(\alpha) := \psi_v^{-1}(X\alpha) \in \Theta_v$, which is the lifted manifold point with chart coordinate $X\alpha$. Since $V[\Theta_v] = \mathcal{P}_v$, we have $V(\theta^\star(\alpha)) \in \mathcal{P}_v$ whenever $X\alpha \in \Omega_v$. Hence,

$$\mathrm{dist}\big(V(\tilde{\theta}(\alpha)), \mathcal{P}_v\big) \le \big\|V(\tilde{\theta}(\alpha)) - V(\theta^\star(\alpha))\big\|_2. \tag{46}$$

By the local Lipschitz continuity of V on the neighbourhood under consideration,

$$\big\|V(\tilde{\theta}(\alpha)) - V(\theta^\star(\alpha))\big\|_2 \le L_V\,\big\|\tilde{\theta}(\alpha) - \theta^\star(\alpha)\big\|_2. \tag{47}$$

It remains to bound $\|\tilde{\theta}(\alpha) - \theta^\star(\alpha)\|_2$. By Lemma 1, for the retraining points $\theta^{(i)} \in \Theta_v$ we have $\mathcal{D} = \mathcal{J}X + E$ with $\mathcal{J} := D(\psi_v^{-1})(0)$. By Taylor's theorem with integral remainder applied to ψ_v^{-1},

$$\psi_v^{-1}(X\alpha) = \psi_v^{-1}(0) + \mathcal{J}(X\alpha) + r(X\alpha) \tag{48}$$

By Equation 40, $\|r(X\alpha)\| \le \frac{\eta}{2}\|X\alpha\|^2$. Using $\psi_v^{-1}(0) = \theta^{(0)}$ and $\tilde{\theta}(\alpha) = \theta^{(0)} + \mathcal{D}\alpha$, we obtain

$$\tilde{\theta}(\alpha) - \psi_v^{-1}(X\alpha) = \Big(\theta^{(0)} + \mathcal{D}\alpha\Big) - \Big(\theta^{(0)} + \mathcal{J}X\alpha + r(X\alpha)\Big) \tag{49}$$

$$= E\alpha - r(X\alpha). \tag{50}$$

Therefore,

$$\big\|\tilde{\theta}(\alpha) - \theta^\star(\alpha)\big\| = \big\|E\alpha - r(X\alpha)\big\| \le \|E\|_2\,\|\alpha\| + \frac{\eta}{2}\|X\alpha\|^2. \tag{51}$$

Combining inequalities of Equation 51, 46, 47, yields:

$$\mathrm{dist}\big(V(\tilde{\theta}(\alpha)), \mathcal{P}_v\big) \le L_V\Big(\|E\|_2 \cdot \|\alpha\| + \frac{\eta}{2}\|X\alpha\|^2\Big) \tag{52}$$

Further, let $\|\cdot\|_F$ be Frobenius norm,

$$\|X\alpha\|^2 \le \|X\|_2^2 \cdot \|\alpha\|^2 \tag{53}$$

$$\le \|X\|_F^2 \cdot \|\alpha\|^2 \tag{54}$$

$$= \|\alpha\|^2 \sum_{i=1}^{m} \|x^{(i)}\|^2 \tag{55}$$

$$\le m\|\alpha\|^2 \max_{1\le i\le m} \|x^{(i)}\|^2 \tag{56}$$

Substitute Equation 31 from Lemma 1, Equation 56 into Equation 52, we have :

$$\mathrm{dist}\big(V(\tilde{\theta}(\alpha)), \mathcal{P}_v\big) \le \frac{L_V\eta}{2} \max_{1\le i\le m} \|x^{(i)}\|^2\Big(\sqrt{m}\,\|\alpha\| + m\,\|\alpha\|^2\Big), \tag{57}$$

which completes the proof. □

Under the conditions of the theorem, we describe the LLE-MORL time complexity as follows.

Proposition B.1. *(Time complexity) Given that the number of objectives is n, the number of base policies is K, the base policy training time is T, the number of locally linear extensions sampled per direction is M, and noting that the locally linear extension is training-free, the expected running time of LLE-MORL is $O(TK + KM^{n-1})$.*

Proof for proposition B.1. In an n-objective reinforcement learning problem, if the Pareto front in the policy parameter space can be represented as a differentiable $(n-1)$-dimensional manifold, LLE-MORL can locally reconstruct this manifold by extending each base policy along $(n-1)$ distinct directions obtained via directional retraining. The locally linear extension then combines these directions to generate a grid of new candidate policies. In a n-objective task, $n-1$ directional retrained directions are used to form a $(n-1)$-dimensional grid: $\theta = \theta_{\text{base}} + \sum \alpha_i(\theta_i - \theta_{\text{base}})$, $i = 1, \ldots, n-1$, allowing dense coverage of the local Pareto surface with or without further training. Training K base policies requires $O(TK)$. Around each base policy, local linear extension samples M^{n-1} candidate policies across the $n-1$-dimensional tangent space. These extensions require no retraining, so their cost is dominated by evaluation, yielding the second term $O(KM^{n-1})$. Thus the overall runtime is $O(TK + KM^{n-1})$.

The first term in the time complexity proposition corresponds to the cost of training K base policies, which is the dominant cost. The second term accounts for generating and evaluating M^{n-1} extended policies per base policy. This step is training-free and its cost is controllable via $M = (\frac{\alpha_{\max} - \alpha_{\min}}{\Delta\alpha})$, allowing the user to balance Pareto front accuracy/density against computational resources. The second term grows exponentially with the number of objectives, but for low- to moderate-dimensional objectives, this cost remains small compared to policy training. For very high-dimensional objective spaces, adaptive sampling or sparse grids can mitigate this cost.

C Experiment Setup Details

C.1 Benchmarks

To evaluate the performance of our proposed LLE-MORL method and compare it against existing baselines, we utilise a suite of continuous control benchmarks from the MO-Gymnasium library [Felten et al., 2023]. These environments are designed to test the ability of an agent to learn policies that effectively balance multiple, often conflicting objectives. The specific environments and their multi-objective reward formulations are detailed below:

MO-SWIMMER-V5. A planar, three-link swimmer operating in a viscous fluid, utilising a 2D continuous action space to control its joint torques. The objectives are to maximise forward velocity along the x-axis and minimise the control cost.

The observation space $\mathcal{S} \subset \mathbb{R}^8$ includes joint angles and velocities, and the action space $\mathcal{A} \subset \mathbb{R}^2$ represents joint torques in $[-1, 1]$. Let x_{before} and x_{after} be the x-coordinates of the centre of mass of the swimmer before and after an action, Δt be the time step, and a_j be the j-th component of the action vector.

The first objective is the forward speed

$$R_1 = \frac{x_{\text{after}} - x_{\text{before}}}{\Delta t},$$

and the second objective is the energy efficiency (negative control cost):

$$R_2 = -\sum_j a_j^2$$

MO-HOPPER-2OBJ-V5. This environment features a 2D one-legged hopper with a 3-dimensional continuous action space controlling torques for its thigh, leg, and foot joints. Originally a 3-objective task (forward speed, jump height, control cost), we use the 2-objective variant, in which the separate control-cost objective is added to other objectives.

The observation space $\mathcal{S} \subset \mathbb{R}^{11}$ includes joint states and torso position, and the action space $\mathcal{A} \subset \mathbb{R}^3$ represents joint torques in $[-1, 1]$. Let $v_x = (x_{\text{after}} - x_{\text{before}})/\Delta t$ be the forward velocity of the agent along the x-axis, where x_{after} and x_{before} are x-positions of the torso. Let $h_{\text{jump}} = 10 \times (z_{\text{after}} - z_{\text{init}})$

be a measure of jumping height, where z_{after} is the current z-position of the torso and z_{init} is its initial z-position. Let c_{ctrl} be the positive control cost, computed as $w_{\text{env_ctrl}} \sum_j (a_j)^2$, where $w_{\text{env_ctrl}}$ is the environment control cost weight (typically 0.001). Let r_{healthy} be the health reward (typically $+1$ if the agent has not fallen). The reward vector $\mathbf{R} = [R_1, R_2]$ is defined as:

- R_1 (Adjusted Forward Performance):
$$R_1 = v_x + r_{\text{healthy}} - c_{\text{ctrl}}$$
- R_2 (Adjusted Height Performance):
$$R_2 = h_{\text{jump}} + r_{\text{healthy}} - c_{\text{ctrl}}$$

MO-ANT-2OBJ-V5. A quadrupedal "ant" robot in 2D with an eight-dimensional action space for joint torques. By default, the environment emits a three-dimensional reward vector: (1) x-velocity, (2) y-velocity, and (3) control cost. Here, we use the two-objective variant in which the separate control-cost objective is added to other objectives.

The observation space $\mathcal{S} \subset \mathbb{R}^{27}$ includes joint states, torso position, and contact forces, and the action space $\mathcal{A} \subset \mathbb{R}^8$ represents joint torques in $[-1, 1]$. Let $v_x = (x_{\text{after}} - x_{\text{before}})/\Delta t$ be the forward velocity of the agent along the x-axis, where x_{after} and x_{before} are x-positions of the torso. Let $v_y = (y_{\text{after}} - y_{\text{before}})/\Delta t$ be the forward velocity of the agent along the y-axis, where y_{after} and y_{before} are y-positions of the torso. Let c_{ctrl} be the positive control cost, computed as $w_{\text{env_ctrl}} \sum_j (a_j)^2$, where $w_{\text{env_ctrl}}$ is the environment control cost weight (typically 0.05). Let r_{healthy} be the health reward (typically $+1$ if the Ant is healthy). Let p_{contact} be the positive contact penalty, which is used for penalising the Ant if the external contact forces are too large, computed as $w_{\text{env_contact}} \sum_k (\text{force}_k)^2$, where $w_{\text{env_contact}}$ is the environment contact cost weight (typically 5×10^{-4}). The reward vector $\mathbf{R} = [R_1, R_2]$ is defined as:

- R_1 (Adjusted x-Velocity Performance):
$$R_1 = v_x + r_{\text{healthy}} - c_{\text{ctrl}} - p_{\text{contact}}$$
- R_2 (Adjusted y-Velocity Performance):
$$R_2 = v_y + r_{\text{healthy}} - c_{\text{ctrl}} - p_{\text{contact}}$$

MO-HOPPER-3OBJ-V5. This environment is identical to MO-HOPPER-2OBJ-V5 in terms of state and action spaces, but retains all three original objectives instead of merging the control cost into the other rewards. As a result, the task features three objectives: forward speed, jump height, and control cost. The reward vector $\mathbf{R} = [R_1, R_2, R_3]$ is defined as:

- R_1 (x-Velocity Performance):
$$R_1 = v_x + r_{\text{healthy}}$$
- R_2 (y-Velocity Performance):
$$R_2 = h_{\text{jump}} + r_{\text{healthy}}$$
- R_3 (Control Cost):
$$R_3 = -c_{\text{ctrl}} + r_{\text{healthy}}$$

MO-ANT-3OBJ-V5. This environment is identical to MO-ANT-2OBJ-V5 in terms of state and action spaces, but retains all three original objectives instead of merging the control cost into the other rewards. As a result, the task features three objectives: x-velocity, y-velocity, and control cost. The reward vector $\mathbf{R} = [R_1, R_2, R_3]$ is defined as:

- R_1 (x-Velocity Performance):
$$R_1 = v_x + r_{\text{healthy}} - p_{\text{contact}}$$
- R_2 (y-Velocity Performance):
$$R_2 = v_y + r_{\text{healthy}} - p_{\text{contact}}$$
- R_3 (Control Cost):
$$R_3 = -c_{\text{ctrl}} + r_{\text{healthy}} - p_{\text{contact}}$$

C.2 Evaluation Metrics

We evaluate the quality of the approximate Pareto front using three standard metrics, following the formalism in [Zitzler and Thiele, 1999, Zintgraf et al., 2015, Hayes et al., 2022].

Hypervolume (HV). Let P be an approximate Pareto front and G_0 a reference point dominated by all $p \in P$. The hypervolume is $\mathcal{H}(P) = \int_{\mathbb{R}^d} \mathbb{1}_{H(P)}(z)\, dz$, where $H(P) = \{\, z \in \mathbb{R}^n \mid \exists\, j,\ 1 \leq j \leq |P| \ :\ G_0 \preceq z \preceq P(j)\}$. Here, $P(j)$ is the j^{th} solution in P, the symbol $\preceq$ denotes objective dominance, and $\mathbb{1}_{H(P)}$ is an indicator function that equals 1 if $z \in H(P)$ and 0 otherwise. A higher hypervolume implies a front closer to and more extensive with respect to the true Pareto front.

Expected Utility (EU). Let P be an approximate Pareto front and Π be the corresponding policy set. The expected utility metric is $\mathcal{U}(P) = \mathbb{E}_{\omega\sim\Omega}\big[\max_{\pi\in\Pi}\ \omega^\top G_\omega^\pi\big]$. A higher EU denotes better average performance over preferences.

Sparsity (SP). Let P be an approximate Pareto front in a d-dimensional objective space. The sparsity metric is $S(P) = \frac{1}{|P|-1}\sum_{i=1}^{d}\sum_{k=1}^{|P|-1}\big(\tilde{G}_i(k) - \tilde{G}_i(k+1)\big)^2$, where $\tilde{G}_i$ is the sorted list of the i^{th} objective values in P, and $\tilde{G}_i(k)$ is the k^{th} entry in this sorted list. Lower sparsity indicates a more uniform distribution of solutions along each objective.

C.3 Hungarian Matching Distance

To measure structural similarity between policies, we use the Hungarian matching distance [Kuhn, 1955, Munkres, 1957].

Definition C.1. *Hungarian matching distance. For a given layer l with neuron sets $A^{(l)}$ and $B^{(l)}$, let $w_i^{(l)}$ and $w_j^{(l)}$ denote the incoming weight vectors of neurons $i \in A^{(l)}$ and $j \in B^{(l)}$, respectively. The minimum-cost perfect matching $M^{(l)}$ between $A^{(l)}$ and $B^{(l)}$ is obtained by using the Hungarian algorithm:*

$$M^{(l)} = \underset{\textit{matching}}{\arg\min} \sum_{(i,j)\in \textit{matching}} \left\| w_i^{(l)} - w_j^{(l)} \right\|_2 \tag{58}$$

The Hungarian matching distance between the two networks is then defined as

$$d_{\textit{Hungarian}}(A, B) = \sum_{l=1}^{L} \sum_{(i,j)\in M^{(l)}} \left\| w_i^{(l)} - w_j^{(l)} \right\|_2 \tag{59}$$

where L is the total number of layers.

C.4 Sanity Check Details

In Section 3.1, the sanity check was designed as a qualitative validation of the parameter–performance relationship (PPR), rather than a full experiment.

To construct policy pairs, we trained 11 base policies θ_{ω_1} using the scalarization vector ω_1 that were equally distributed in the preference space, i.e. $\omega_1 = [1, 0], [0.9, 0.1], ..., [0, 1]$.

For each base policy, we formed a paired scalarization vector ω_2 by shifting ω_1 with a retraining shift parameter δ_s:

$$\omega_2 = \begin{cases} [\omega_{11} - \delta_s, \omega_{12} + \delta_s], \text{ if } \omega_{11} - \delta_s \in [0, 1] \\ [\omega_{11} + \delta_s, \omega_{12} - \delta_s], \text{ otherwise} \end{cases} \tag{60}$$

where $\delta_s \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. This procedure yielded multiple pairs of (ω_1, ω_2) across the preference space. For each pair, we performed short retraining from θ_{ω_1} under ω_2 and compared the original θ_{ω_1} and resulting model $\theta_{\omega'}$ with the independently trained θ_{ω_2}. This allowed us to consistently observe that retrained policies stay close in parameter space (low Hungarian distance), while having directional movement in performance space, supporting the local PPR hypothesis. Figure 2 in the main paper presents a representative example from these trials.

C.5 Training and Evaluation Details

All learning phases within our LLE-MORL algorithm, including the initial training of base policies, the directional retraining, and the final preference-aligned fine-tuning, utilise the Proximal Policy Optimization (PPO) algorithm [Schulman et al., 2017]. We employed a standard PPO implementation from the Stable Baselines3 library [Raffin et al., 2019]. The PPO parameters used across all training stages and benchmarks are detailed in Table 3.

The specific parameters for the LLE-MORL pipeline include:

- **Number of Initial Base Policies (K)**: The total count of base policies θ_{w_k}, trained in the initialization stage. The corresponding K initial scalarization weights $\{w_k\}_{k=1}^{K}$ are generated by evenly distributing them across the preference space (e.g., for 2D objectives, from $[1, 0]$ to $[0, 1]$ in K steps).
- **Initialisation Training Timesteps (T_{init})**: The number of environment interaction steps for which the initial base policy θ_{w_k} is trained under its weight w_k.
- **Retraining Preference Shift Strategy (controlled by shift magnitude δ_s)**: Target scalarization weights $\{w_k^{(i)}\}$ for directional retraining are generated by shifting each initial weight w_k to a nearby, distinct point on the preference space. The extent of this shift is controlled by a hyperparameter δ_s.
- **Directional Retraining Timesteps (T_{dir})**: The number of environment interaction steps for which the base policy θ_{w_k} is retrained under its target weight w_k^i to produce $\theta_{w_k^{(i)}}$.
- **Step-Scale Factor Generation** ($\alpha_{\text{start}}, \alpha_{\text{end}}, \Delta\alpha$): The set of step-scale factors $\{\alpha_i\}$ used in Locally Linear Extension is generated based on a starting value (α_{start}), an ending value (α_{end}), and either a step increment ($\Delta\alpha$).
- **Fine-tuning Timesteps (T_{ref})**: The number of environment interaction steps for which the selected candidate policy from the extension phase is fine-tuned under its matched preference weight $w_{k,\text{cand}}$.

Table 3: PPO hyperparameters for benchmarks.

Parameter Name	MO-Swimmer	MO-Hopper-2d	MO-Ant-2d	MO-Hopper-3d	MO-Ant-3d
steps per actor batch	512	512	512	512	512
learning rate ($\times 10^{-4}$)	3	3	3	3	3
learning rate decay ratio	1	1	1	1	1
γ	0.995	0.995	0.995	0.995	0.995
GAE lambda	0.95	0.95	0.95	0.95	0.95
number of mini batches	32	32	32	32	32
PPO epochs	10	10	10	10	10
entropy coefficient	0.0	0.0	0.0	0.0	0.0
value loss coefficient	0.5	0.5	0.5	0.5	0.5
maximum gradient norm	0.5	0.5	0.5	0.5	0.5
clip parameter	0.2	0.2	0.2	0.2	0.2

In all experiments reported in the main results, we fix the number of initial base policies to $K = 6$, the retraining preference shift magnitude to $\delta_s = 0.1$, and the step-scale factor grid to $\alpha \in [-1.5, 1.5]$ with step size $\Delta\alpha = 0.05$. We clarify that K and δ_s are chosen for simplicity and are not intended to be optimal. We have an additional discussion about the choice of δ_s in Appendix D.1. The step-scale factor α controls the Locally Linear Extension stage and has a meaningful but limited influence on performance; its impact and selection are discussed in Appendix D.2.

The number of base policies K controls the initial coverage of the preference space. Too few base policies may leave some Pareto-front components unexplored, whereas increasing K provides more starting points but also increases training costs. Under a fixed total interaction budget, a larger K also reduces the training budget available to each base policy, so increasing K does not necessarily improve performance monotonically. We use the same moderate setting $K = 6$ across all two- and three-objective benchmarks, without environment-specific tuning, and consistently obtain strong Pareto-front coverage. This suggests that LLE-MORL is not highly sensitive to the precise value of K within a reasonable range; additional base policies primarily improve coverage density with diminishing returns relative to their cost.

Table 4: Approximate execution times (hours) for each method and benchmark under different settings.

(a) 2D sample-efficient setting

Method	MO-Swimmer	MO-Hopper-2d	MO-Ant-2d
GPI-LS	3	3	3
CAPQL	3	3	3
Q-Pensieve	3	3	3
MORL/D	1	1	1
LLE-MORL	1	1	1

(b) 2D standard-training setting

Method	MO-Swimmer	MO-Hopper-2d	MO-Ant-2d
GPI-LS	22	23	19
CAPQL	11	11	12
Q-Pensieve	17	15	18
MORL/D	3	3	3
LLE-MORL	3	3	3

(c) 3D setting

Method	MO-Hopper-3d	MO-Ant-3d
GPI-LS	44	43
CAPQL	13	18
Q-Pensieve	24	23
MORL/D	5	6
LLE-MORL	5	5

For baseline methods, since the original papers do not evaluate performance on the same benchmark problems considered in this work, we adopt the implementations and hyperparameter settings provided by morl-baselines [Felten et al., 2023]. As LLE-MORL is a PPO-based algorithm, we employ PPO hyperparameter configurations commonly used in PPO-based MORL methods, such as PGMORL [Xu et al., 2020], which serves as a standard and widely adopted reference for PPO settings in MORL tasks.

In all experiments reported in the main results, we strictly control the total training interaction budget for all methods to ensure a fair comparison. For LLE-MORL, this fixed budget is automatically partitioned across the three learning phases using a simple $3:1:1$ ratio for initial base policy training, directional retraining, and preference-aligned fine-tuning, respectively. When the total training budget is sufficiently large, allocating additional interaction steps to the base policy training stage can further improve performance, as directional retraining and fine-tuning require fewer updates. The optimal budget allocation depends on the target application and can be adjusted on a case-by-case basis.

Approximate execution times for all methods and benchmarks are reported separately in Table 4. The computational resources used are described in Appendix C.6.

Evaluation cost. Interactions with the environment for evaluation of the algorithm are usually not considered towards sample efficiency because they are not used for policy improvement. Nevertheless, because evaluation plays a particularly visible role in selecting LLE candidates, we understand the concern and report the evaluation cost separately here. Under our default configuration, $\alpha \in [-1.5, 1.5]$ at intervals of 0.05, LLE-MORL evaluates $M = 61$ candidates for 5 episodes per base policy, corresponding to a total $K \times M \times$ evaluation episodes $= 61 \times 6 \times 5 = 1830$ evaluation episodes in the LLE stage under 2d environment. For the full algorithm, we also have base-policy evaluation, directional-retraining diagnostics, and fine-tuned policy evaluations. As the final candidate number is not fixed, the total evaluation will be approximate around 3000 episodes. Evaluation outside the training-step budget is also standard in the compared implementations. For baselines: (1) CAPQL evaluates every 10,000 training steps. At each evaluation it uses 100 preference weights $\times$ 5 episodes $= 500$ evaluation episodes. Under a standard 10^6 training budget, CAPQL has approximately 50,000 episodes of evaluation cost. (2) GPI-LS has several separate evaluation components and has at least 10,000 episodes of evaluation cost under a 10^6 training budget. (3) MORL/D evaluates the whole population once before training and again after every 10,000 training steps. Under a 10^6 training budget, it has 15,300 evaluation episodes. (4) Q-Pensieve evaluates every 50,000 training steps, and it has 3,600 evaluation episodes under a 10^6 training budget. These evaluation roll-outs do not update policy parameters and are separate from the common training-interaction budget. Overall, the evaluation cost remains reasonable in the low- to moderate-dimensional settings considered here. The end-to-end wall-clock times in Table 4 include both training and evaluation.

C.6 Computational Resources

All experiments were run on a workstation equipped with an AMD Ryzen Threadripper PRO 5975WX (32 cores), an NVIDIA GeForce RTX 3090 GPU (24 GB GDDR6X), and 256 GiB of RAM, running Ubuntu 24.04 LTS. The software stack included CUDA Toolkit 12.0 and the corresponding NVIDIA drivers.

D Additional Results

D.1 Effect of Directional Retraining Shift

This ablation study investigates the influence of the directional retraining shift δ_s, on the Locally Linear Extension process. We analyse the performance of the LLE-MORL-0 variant (which excludes fine-tuning) under the standard-training setting. We vary δ_s over the set $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, keeping other parameters at their default values, and report Hypervolume (HV), Expected Utility (EU), and Sparsity (SP) for each environment in Table 5.

The results indicate that δ_s influences the resulting Pareto front approximation, as different shift values generate distinct extension trajectories that directly shape the front. While optimal performance varies across environments, a moderate retraining shift (e.g., δ_s in the range of 0.2 to 0.3, depending on the specific benchmark characteristics evident in Table 5) generally appears to strike an effective balance between front coverage (HV, EU) and solution diversity (SP). Our default configuration utilised $\delta_s = 0.1$, chosen for its simplicity and promising results in preliminary tests. However, this ablation demonstrates that this typical shift is not universally optimal; selecting an appropriately tuned moderate δ_s can further enhance the quality and coverage of the extended manifold, thereby improving the final Pareto front approximation achieved by LLE-MORL-0.

Table 5: Impact of retraining shift δ_s on LLE-MORL-0 performance (HV, EU, SP) under standard-training setting.

Environment	Metric	$\delta_s = 0.1$	$\delta_s = 0.2$	$\delta_s = 0.3$	$\delta_s = 0.4$	$\delta_s = 0.5$
MO-Swimmer	HV(10^4)	7.37	7.38	7.38	7.36	7.37
	EU(10^1)	1.03	1.29	1.41	0.99	1.60
	SP(10^2)	2.10	0.83	3.05	2.39	2.12
MO-Hopper-2d	HV(10^5)	4.88	4.84	4.79	4.86	4.80
	EU(10^2)	5.65	5.56	5.49	5.54	5.58
	SP(10^2)	7.20	4.49	3.33	2.54	3.33
MO-Ant-2d	HV(10^5)	2.43	2.66	2.65	2.72	2.44
	EU(10^2)	3.44	3.94	3.69	3.95	3.66
	SP(10^3)	1.76	2.64	2.23	8.98	2.41

D.2 Effect of Step-Scale in Locally Linear Extension

We next examine how the choice of step-scale factors $\{\alpha_j\}$ — defined by their start α_{start}, end α_{end}, and increment $\Delta\alpha$ — influences the performance of LLE-MORL-0. For each benchmark, we compare different ranges and resolutions of $\{\alpha_j\}$ and report HV, EU, and SP in Table 6.

The choice of $\Delta\alpha$ significantly impacts Sparsity (SP), as shown in Table 6. Finer increments (smaller $\Delta\alpha$) lead to substantially lower SP values, indicating denser Pareto front approximations, whereas coarser steps result in sparser solutions. This confirms that LLE offers a training-free mechanism to control the density of the approximated front simply by adjusting $\Delta\alpha$.

Regarding the range of α_j (controlled by α_{start} and α_{end}), expanding it generally leads to improvements in both HV and EU, as the extension manifold can reach further from the initial policy along the identified directional vector (visualized conceptually in Figure 3). However, Table 6 suggests that an extremely wide range does not always yield proportional gains in coverage. This indicates that beyond a certain point, the linearity assumption underpinning LLE may become less effective for extending the front into novel, high-quality regions using a single directional vector, or that the most valuable regions reachable by the current $\Delta\theta$ vectors are already sufficiently captured.

Table 6: Influence of Step-Scale α on LLE-MORL-0 performance under standard-training setting.

Environment	Metric	$(\alpha_{\text{start}}, \alpha_{\text{end}})$	(-1,1)				(-1.5,1.5)				(-2,2)				(-3,3)			
		$\Delta\alpha$	0.01	0.05	0.1	0.5	0.01	0.05	0.1	0.5	0.01	0.05	0.1	0.5	0.01	0.05	0.1	0.5
MO-Swimmer	HV(10^4)		7.37	7.37	7.37	7.33	7.38	7.37	7.37	7.33	7.40	7.40	7.39	7.34	7.40	7.40	7.40	7.34
	EU(10^1)		1.08	1.27	1.32	1.33	1.08	1.03	1.42	1.65	1.13	1.41	1.53	1.70	1.43	1.75	1.73	2.37
	SP(10^2)		0.24	1.17	2.54	25.55	1.13	2.10	1.46	25.55	0.11	0.53	1.18	11.48	0.11	0.51	1.18	11.48
MO-Hopper-2d	HV(10^5)		4.95	4.88	4.85	4.80	4.96	4.88	4.90	4.80	4.95	4.88	4.88	4.81	4.96	4.88	4.88	4.81
	EU(10^2)		5.82	5.74	5.73	5.75	5.76	5.65	5.72	5.75	5.75	5.69	5.69	5.67	5.75	5.69	5.69	5.68
	SP(10^2)		5.63	9.43	11.58	60.35	5.01	7.20	13.06	60.35	3.02	5.42	10.05	29.66	3.62	5.41	10.05	29.66
MO-Ant-2d	HV(10^5)		2.57	2.43	2.40	2.23	2.60	2.43	2.23	2.14	2.60	2.47	2.25	2.14	2.60	2.47	2.25	2.14
	EU(10^2)		3.61	3.47	3.62	3.66	3.66	3.44	3.44	3.31	3.67	3.59	3.43	3.37	3.67	3.59	3.43	3.37
	SP(10^3)		0.54	2.03	3.11	7.92	1.85	1.76	1.79	8.36	1.85	1.55	1.79	8.36	1.85	1.55	1.79	8.36

E Additional Discussion

E.1 Discussion of Reliability and Robustness

In the LLE-MORL algorithmic procedure, the underlying Reinforcement Learning method (PPO) used in the training and retraining phases is subject to inherent stochasticity arising from factors such as random seed initialisation for neural network weights and data sampling during training. This means that while the overarching behaviour and effectiveness of LLE-MORL are generally reproducible, the exact set of policies discovered on the approximated Pareto front, their specific parameter values, or their ordering might exhibit some variation across independent runs started with different random seeds. Consequently, direct averaging of entire Pareto fronts or performing straightforward statistical tests on the precise composition of these policy sets can be non-trivial and may not always be the most informative way to capture the consistent ability of the algorithm to find high-quality solution regions.

Our demonstration of reliability and robustness relies on: (1) the consistent observation of the superior performance of LLE-MORL in generating high-quality Pareto fronts compared to baselines; (2) sensitivity analyses of key hyperparameters (detailed in Appendix D), which show consistent outcome patterns within certain parameter ranges; (3) the qualitative consistency in the shape, extent, and dominance characteristics of the visualized Pareto fronts (e.g. Figure 4); and (4) the quantitative results reported in Table 1 and Table 2 across multiple seeds, further reinforcing the statistical reliability of our findings. These combined observations support the robustness of our conclusions and the effectiveness of LLE-MORL.

E.2 Discussion of Assumptions

The assumption of diffeomorphic relationships or of Lipschitz conditions in the local manifold analysis provides sufficient conditions for the theoretical error bound and complexity. Importantly, LLE-MORL only requires these properties *locally*, around a base policy and a single Pareto-front branch. In general deep RL problems, solutions are approached by gradient-like methods which assume that the loss function is locally sufficiently well approximated by a tangential space, which is a hint that diffeomorphic relationships for successfully solvable deep RL problems can be seen as natural, although much more work and additional (albeit less restrictive) assumptions would be needed to prove this. Also, this would not include multi-objective application cases where an optimal solution is not required nor available, but the extensions can still provide good suboptimal solutions. Therefore, we believe that the present assumptions provide a reasonable compromise between the generality of the currently possible propositions and the coverage of practically relevant situations.

More specifically, the analysis separates several conditions. First, Theorem 3.1 derives local Lipschitz regularity of the parameter–performance map from differentiability of the expected return and a bounded policy score function, conditions commonly used in RL analysis. Second, the diffeomorphism in Theorem 3.2 concerns only a local representative parameter subset Θ_v and a neighbourhood $\mathcal{P}_v$ of one Pareto-front branch; it does not assume a global bijection mapping between all neural-network parameters and performance. Third, exact non-dominated initialisation and tangent-space recovery are theoretical conditions to express approximation error relative to the true Pareto front.

The practical algorithm is less restrictive: the true Pareto front is unknown and the base policies are finite-budget, approximately non-dominated solutions, and brief directional retraining provides an approximation to local manifold directions rather than exact tangent vectors. Across the five continuous-

control benchmarks considered, including two three-objective tasks, these local approximations consistently produced useful extension directions and high-quality Pareto-front reconstructions.

F Limitations

Limitations of our approach are implied by inherent challenges in multi-objective optimisation, but we also note some limitations that are specific to our algorithm and require further study.

- A larger number of objectives is a problem for most of the existing MORL algorithms. Although usually some of the objectives are of different importance and can be lexicographically ranked, so that the complexity does not necessarily increase exponentially with the number of objectives. The high-dimensional case is nevertheless challenging, but our approach can be seen as promising: In higher dimensions, the number of solutions that are in the same local quasi-linear patch increases dramatically, so that the efficiency of the proposed local search will be even more beneficial. This benefit could be reduced by the potentially increasing complexity of the topological relation between the performance space and the parameter space which could be a fascinating subject for future work.
- We are assuming that the Pareto front consists of a limited number of connectivity components which have a manifold structure. While there is no theoretical bound to the complexity of the Pareto front, the idea of MORL implies that the objectives are at least in some sense comparable. For Pareto fronts that are fractal or of high genus, the result of multi-objective optimisation lacks robustness, although it will neither be possible to fix any limits for the complexity of the Pareto front. However, as long as there are only a limited number of manifold-like connectivity components, our algorithm will be applicable.
- Widely different scales and elasticities of the objectives can lead to problems as in optimisation in anisotropic error landscapes. Step size control that helps in gradient methods in optimisation will also be useful here, but has not been studied yet, as the typical (benchmark) problems are sufficiently isotropic.
- The density of the identified solutions on the Pareto front is clearly a challenge, which may be solved by step size control as mentioned in the previous point. This concerns higher-dimensional cases as well as extended one-dimensional trails as visible in the top trail in Figure 3c also a simple reduction of the parameter $\Delta\alpha$ at the observation of large steps in the performance space could have solved this issue already so that a more uniform covering of the Pareto from is not difficult to achieve in the present approach. See also Appendix D.2. In contrast to other approaches, locally smooth regions of the Pareto front can trivially be tracked by LLE-MORL. Concave regions connected to the Pareto front will be followed through without problem, but will need to be removed in a single postprocessing step as they are dominated by other solutions. Even full patches of solutions may turn out to be Pareto sub-optimal and require a similar treatment.
- We have made use of scalarisation to seed the solution domains, whereas the reconstruction of the Pareto front is done by a lateral process that does not depend on preference weights. It is in principle possible that a solution patch is not reachable by any scalarisation-based seeding attempt, see also the early discussion in [Vamplew et al., 2008]. In this case our approach might not find this patch, although it is still possible that it is found by retraining from a different solution domain as shown in Figure 3c.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract includes the contribution and the scope of developing an algorithm for efficient multi-objective reinforcement learning.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitations of the approach are discussed in Appendix F.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: The paper states the main theoretical assumptions and results in Section 3.4, while complete proofs and supporting propositions are provided in Appendix B.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The algorithm is described in detail in Section 3.3 and Algorithm 1, and the benchmark settings, hyperparameters, training details, and computational resources are provided in Section 4 and Appendix C.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: Our code is publicly available at `https://github.com/AQQQQY/LLE_MORL`. The simulator (MO-Gymnasium) used in the experiments has been made publicly available [Felten et al., 2023].

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: The benchmark settings, hyperparameters, training details, and computational resources are provided in Section 4 and Appendix C.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: The main result tables report mean $\pm$ standard deviation values across runs, and Appendix D discusses robustness and sensitivity analyses of key hyperparameters.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: All information is given in Appendix C.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research in this paper follows the NeurIPS Code of Ethics. The project did not work with human data or human subjects. We have considered possible misuse and have not identified any concerns for the method in its current form.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: Without further subsequent work, we do not expect the algorithm described in the paper to be deployed in society in the near future, and thus do not discuss societal concerns at this stage.

Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper does not pose such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The external software and benchmark environments used in this work are properly cited in the paper. MO-Gymnasium, MORL-Baselines, and Stable-Baselines3 are used in accordance with their respective open-source licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

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

13. **New assets**

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

    Answer: [Yes]

    Justification: We release the implementation of LLE-MORL at `https://github.com/AQQQQY/LLE_MORL`, together with documentation and instructions for running the experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs have been used only for writing as provided via Overleaf.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.