Post Hoc Extraction of Pareto Fronts for Continuous Control

Raghav Thakar*
Gaurav Dixit
Kagan Tumer
thakarr@oregonstate.edu
Collaborative Robotics and Intelligent Systems (CoRIS) Institute, Oregon State University
Corvallis, Oregon, USA

ABSTRACT

Agents in the real world must often balance multiple objectives, such as speed, stability, and energy efficiency in continuous control. To account for changing conditions and preferences, an agent must ideally learn a Pareto frontier of policies representing multiple optimal trade-offs. Recent advances in multi-policy multi-objective reinforcement learning (MORL) enable learning a Pareto front directly, but require full multi-objective consideration from the start of training. In practice, multi-objective preferences often arise after a policy has already been trained on a single specialised objective. Existing MORL methods cannot leverage these pre-trained 'specialists' to learn Pareto fronts and avoid incurring the sample costs of retraining. We introduce Mixed Advantage Pareto Extraction (MAPEX), an offline MORL method that constructs a frontier of policies by reusing pre-trained specialist policies, critics, and replay buffers. MAPEX combines evaluations from specialist critics into a mixed advantage signal, and weights a behaviour cloning loss with it to train new policies that balance multiple objectives. MAPEX's post hoc Pareto front extraction preserves the simplicity of single-objective off-policy RL, and avoids retrofitting these algorithms into complex MORL frameworks. We formally describe the MAPEX procedure and evaluate MAPEX on five multi-objective MuJoCo environments. Given the same starting policies, MAPEX produces comparable fronts at 0.001% the sample cost of established baselines.

KEYWORDS

multi-objective reinforcement learning, offline reinforcement learning, continuous control

1 INTRODUCTION

Many real-world continuous control problems require agents to balance multiple, even conflicting, objectives. In legged locomotion, for example, a robot must simultaneously optimise forward speed, gait stability, and energy efficiency. In such settings, there is no single optimal solution, and agents must instead discover a set of non-dominated policies that capture the range of feasible trade-offs. Learning this *Pareto front* enables downstream stakeholders to select behaviours that reflect their current preferences or adapt to changing operational conditions.

Typically, multiple objectives are handled through scalarisation into a single weighted sum and training via standard reinforcement learning (RL). While accessible and tractable, this approach yields only a single, fixed trade-off. Simply repeating this procedure for every trade-off is inefficient and practically infeasible without sophisticated experience and representation sharing.

Multi-Objective RL (MORL), and specifically *multi-policy* MORL algorithms partially address this through various architectural improvements to learn the entire frontier directly. Methods like MORL/D [8] divide the problem into several single-objective problems through scalarisation, and use single-objective RL with buffer sharing to train policies for each scalarisation. Other works like PG-MORL [28] and MOPDERL [25] maintain populations of policies, combining RL with evolutionary principles to jointly improve the policies in the population and cover the objective space. PSL-MORL [16], a notable recent work, trains a hypernetwork to produce a fresh policy for every desired trade-off.

Despite successfully learning Pareto fronts, these methods suffer from a critical limitation: requiring full multi-objective consideration from the outset of training. This rigidity creates a disconnect with practical scenarios, where multi-objective preferences often arise retroactively, typically only after a robust policy for a primary task has already been trained. For instance, a preference for more stability may only emerge after observing a locomotion policy highly optimised for speed. To obtain new trade-offs in these scenarios, practitioners must 1) discard pre-trained policies and incur the sample costs of re-training, and 2) retrofit their algorithm into complex multi-objective learning frameworks. Currently, no method reuses disjoint specialist policies and training data to recover Pareto fronts efficiently and with minimal added algorithmic complexity.

In this work we present Mixed Advantage Pareto Extraction (MAPEX), a novel method that fully leverages prior-trained single-objective policies, critics, and replay buffers to produce Pareto fronts of policies. Our key insight is that agents learn optimal trade-offs by intelligently blending expert behaviour on each objective. MAPEX implements this by blending the single-objective evaluation from each specialist critic into a multi-objective *mixed advantage* value and weighting a behaviour cloning loss with it.

MAPEX bypasses the need to retrofit bespoke or standard off-policy RL into complex multi-objective learning frameworks. It preserves both, the simplicity, and intricacies of these algorithms by providing a realistic pathway to learning multi-objective behaviours from single-objective training data. If training from scratch, MAPEX allows allocating the entire sample budget to training high-performing single-objective specialists, from which high-quality Pareto fronts can later be extracted at a minimal sample cost.

*Corresponding author.

Proc.@ of the Adaptive and Learning Agents Workshop (ALA 2026), Aydeniz, Delgrange, Mohammedalamen, Yang (eds.), May 25 – 26, 2026, Paphos, Cyprus, https://alaworkshop2026.github.io/. 2026.

We provide a detailed view of MAPEX's Pareto extraction procedure, and perform Pareto extraction from specialists trained using both standard and bespoke off-policy RL. Across five multi-objective MuJoCo environments, MAPEX produces fronts comparable to, or better than established baselines from the literature. When extracting Pareto fronts from the same starting policies, other methods consume up to 1000× more samples than MAPEX to produce similar fronts.

2 BACKGROUND

2.1 Actor–Critic Methods and Offline Reinforcement Learning

Actor–critic methods: These methods perform reinforcement learning using an *actor*, which defines the policy, and a *critic*, which estimates the expected return of the actor's behaviour [24]. By relying on learned value estimates rather than Monte Carlo returns alone, actor–critic methods enable efficient and stable policy optimisation, and are commonly used in continuous control. These methods are often implemented in an *off-policy* setting, where experience collected by one or more behaviour policies is stored in a replay buffer and reused for policy updates [9, 10].

Policy improvement is commonly guided by the *advantage* function,

$$A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s), \tag{1}$$

where $Q^\pi(s,a)$ is the critic's action-value function, and $V^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}[Q^\pi(s,a)]$. In practice, particularly in deterministic actor–critic methods, the value function is often approximated by $Q^\pi(s,\pi(s))$, yielding

$$A^\pi(s,a) \approx Q^\pi(s,a) - Q^\pi(s,\pi(s)). \tag{2}$$

Offline Reinforcement Learning: Offline reinforcement learning studies the problem of learning a policy from a fixed dataset of transitions, without any further interaction with the environment during training. A key challenge in this setting is distributional shift: the dataset may contain diverse experiences that are hard to generalise to, leading to unreliable value estimates [13, 18].

Advantage-weighted regression (AWR) [19] provides a simple and stable mechanism for policy improvement in this setting by casting reinforcement learning as a weighted supervised learning problem. Given a critic-derived advantage estimate $A(s,a)$, the AWR policy update solves

$$\pi^+ = \arg\max_\pi \ \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\log\pi(a|s)\ \ \exp\left(\tfrac{1}{\beta}A(s,a)\right)\right], \tag{3}$$

where $\beta > 0$ is a temperature parameter. This objective emphasises actions that are predicted to improve performance while sampled purely from the observed data, making it well suited to off-policy and offline reinforcement learning problems. We use an AWR-inspired regression weighting in MAPEX to learn multi-objective behaviours from static single-objective replay buffers.

2.2 Multi-Objective Sequential Decision-Making

Multi-objective sequential decision-making problems are commonly modelled as Multi-Objective Markov Decision Processes (MOMDPs). A MOMDP is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \gamma, \mathbf{R}\rangle$, containing the state space $\mathcal{S}$, the action space $\mathcal{A}$, the transition dynamics $\mathcal{T} : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to [0,1]$, and a scalar discount factor $\gamma \in [0,1)$. Unlike standard MDPs, the reward function $\mathbf{R} : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to \mathbb{R}^k$ returns a vector of k rewards, each corresponding to a distinct objective.

A policy $\pi : \mathcal{S} \to \mathcal{A}$ induces an expected return for each objective. We characterise policy performance directly through its expected long-term returns. Let $\mathbf{J}(\pi) \in \mathbb{R}^k$ denote the vector of objective returns achieved by policy π, with components

$$J_i(\pi) = \mathbb{E}_{s_0\sim\rho_0,\, a_t\sim\pi(\cdot|s_t)}\left[\sum_{t=0}^{\infty}\gamma^t r_i(s_t,a_t)\right],\ i \in \{1,\dots,k\}. \tag{4}$$

where r_i is the reward function associated with the i-th objective and ρ_0 is the initial state distribution.

From a multi-objective optimisation perspective, learning in a MOMDP can be viewed as maximising a vector-valued objective

$$\max_\pi \ \mathbf{F}(\pi) \ \triangleq\ \max_\pi\ [J_1(\pi), J_2(\pi), \dots, J_k(\pi)]. \tag{5}$$

Pareto Dominance and Optimality. As objectives are often conflicting, it is generally impossible for a single policy to optimise all objectives simultaneously. Thus, optimality in multi-objective settings is defined in terms of *Pareto dominance*. Given two objective return vectors $\mathbf{v}, \mathbf{u} \in \mathbb{R}^k$, $\mathbf{v}$ *Pareto-dominates* $\mathbf{u}$ (denoted $\mathbf{v} \succ_p \mathbf{u}$) if $\mathbf{v}$ is at least as good as $\mathbf{u}$ in all objectives and strictly better in at least one:

$$\mathbf{v} \succ_p \mathbf{u} \iff (\forall i : v_i \geq u_i) \wedge (\exists j : v_j > u_j). \tag{6}$$

If neither vector Pareto-dominates the other, they are said to be *non-dominated*.

A policy π is *Pareto optimal* if its return vector $\mathbf{J}(\pi)$ is not Pareto-dominated by that of any other policy. The set of all Pareto-optimal policies induces the *Pareto front*, which characterises the optimal trade-offs achievable in the objective space.

3 RELATED WORKS

Recent surveys [11, 22] categorise the expanding MORL landscape into single-policy and multi-policy approaches. We adopt this taxonomy herein.

Single-policy methods optimise a fixed scalar utility function [11], using standard RL with linear utilities, and dedicated approaches for non-linear utilities [15, 21, 26]. Conversely, multi-policy approaches approximate the Pareto front, a better-suited approach for unknown or dynamic preferences. Early works like Pareto Q-Learning [17] tracked non-dominated return vectors for each state-action pair, while recent extensions employ preference-conditioned deep networks [1] or convex envelope updates [29] for high-dimensional spaces.

In continuous control, decomposition-based methods divide the problem into several scalarised objectives. Prior work [4] has augmented naively solving each scalar objective with an evolutionary strategy [23] for post-processing. Notably, MORL/D [8] improves efficiency via cooperative buffer sharing and intelligent weight sampling. Another branch of work combines RL with evolutionary operators: PG-MORL [28] iteratively pushes a population of policies toward promising objective space regions, while MOPDERL [25] distills a frontier from subpopulations trained via evolutionary RL [3, 12]. Alternatively, parameter-efficient methods consolidate policies using meta-learning [5], universal preference-conditioned networks [2], or hypernetworks [16].

Despite architectural differences, these methods share a structural limitation: they are designed solely for "from scratch" learning. If training on one or more individual objectives has already been performed, these methods cannot leverage it, effectively requiring pre-trained policies to be discarded and training to be restarted.

A separate line of inquiry focuses on offline MORL, training preference-conditioned agents from massive datasets. Approaches like PEDA [30], DiffMORL [27], and PR-MORL [14] utilise transformers, diffusion, and regularisation to generalise across preferences. While seemingly similar, MAPEX tackles the distinct problem of extracting frontiers from pre-trained specialists rather than large-scale generalisation. Consequently, MAPEX operates with replay buffers two orders of magnitude smaller than the D4MORL benchmarks [30] used in offline RL works (e.g., 1 million vs. 150 million transitions).

Finally, PCN [20] also uses supervised learning from an off-policy dataset, but relies on iterative online data collection to populate the objective space. In contrast, MAPEX addresses the strictly offline extraction of frontiers from fixed, disjoint datasets. PCN is also currently limited to discrete action spaces.

4 METHOD

We now introduce Mixed Advantage Pareto Extraction (MAPEX), an offline algorithm to extract a Pareto front of continuous control policies from a set of disjoint *single-objective specialists*. The core intent of MAPEX is to fully leverage the latent information in prior-trained specialists—which includes their policies, along with their value functions (critics) and replay buffers—to produce new policies that express new trade-off behaviours without requiring additional training interaction with the environment.

MAPEX achieves this via a Pareto front extraction procedure which iteratively fills gaps in the Pareto front estimate. First, MAPEX analyses the available policies in the objective space to identify sparse regions in the Pareto front estimate. Once a gap is identified, MAPEX derives a vector of 'target weights' that encodes a weighting over the objectives that would optimally fill this gap. To train a policy for this new trade-off, MAPEX constructs a static *hybrid buffer* by sampling from the specialists' buffers in proportion to these target weights. Crucially, the algorithm then calculates a *mixed advantage* for every transition in this dataset by querying each specialist critic, and mixing the individual advantage estimates in the ratio of the target weights. This mixed advantage captures the value of a transition in demonstrating target trade-off behaviour. Finally, a fresh policy is trained using a mechanism inspired by Advantage Weighted Regression (AWR) [19], wherein we regress the policy onto actions from the hybrid buffer, weighted by an exponential of their calculated mixed advantage. Algorithm 1 provides a more rigorous look at this procedure.

Starting Information and Notation

We assume that from prior training we have a policy set Π with at least N policies for an N-objective problem, a critic set $\mathcal{Q}$ with N critics, each specialising on evaluating on one of the problem's objectives, and a replay buffer set $\mathcal{D}$ with N buffers, each containing experiences from training on a single objective.

4.1 Step 1: Gap Identification and Parent Selection

At the start of each iteration, we evaluate the current policy set Π to obtain the performance vector $J(\pi) \in \mathbb{R}^N$ of each policy. We then identify the non-dominated set (the current Pareto front approximation). We then search for the largest N-dimensional 'gap' on the frontier—a sparse region in the objective space. For $N = 2$, we define this as the edge with the maximal Euclidean span. In practice, to avoid repeatedly focusing on the same gap, we select the gap using roulette-wheel sampling based on edge length. The N policies corresponding to the vertices of this gap are selected as the *parent policies*, denoted by $\{\pi_{p_1}, \ldots, \pi_{p_N}\}$.

To guide the offspring into this sparse region, we compute the centroid of the parents' performance vectors in the objective space:

$$J_{mid} = \frac{1}{N} \sum_{i=1}^{N} J(\pi_{p_i}) \tag{7}$$

We then derive a unit vector $\mathbf{w}_{\text{target}}$ of target weights pointing towards this centroid. This vector encodes the linear preference

Algorithm 1: Mixed Advantage Pareto Extraction (MAPEX)

Input: Policy set Π, critic set $\mathcal{Q}$, buffer set $\mathcal{D}$, number of objectives N
Output: Pareto front $\mathcal{P}$ of policies
Function *MAPEX*$(\Pi, \mathcal{Q}, \mathcal{D}, N)$**:**
  while *Required* **do**
    Evaluate(Π)
    $\mathcal{P} \leftarrow$ FindParetoFront(Π)
    $\{\pi_1, \ldots, \pi_N\} \leftarrow$ SelectParents$(\mathcal{P})$
    $\mathbf{v} \leftarrow$ Centroid$(\pi_1, \ldots, \pi_N)$
    // In objective space
    $\mathbf{w}_{\text{target}} \leftarrow \frac{\mathbf{v}}{\|\mathbf{v}\|_2}$ *// Target weight vector*
    $\mathcal{D}_{\text{hybrid}} \leftarrow \bigcup_{k=1}^{N}$ Sample$(\mathcal{D}_k, \propto w_k)$
    $\pi_{\text{new}} \leftarrow$ InitPolicy$(\emptyset)$
    for $\epsilon \leftarrow 1$ **to** E **do**
      $(s, a) \sim \mathcal{D}_{\text{hybrid}}$
      $\mathbf{A} \leftarrow \left[\left(Q_i(s, a) - Q_i(s, \pi_{\text{new}}(s)) \right) \right]_{i=1}^{N}$
      $A_{\text{mixed}} \leftarrow \mathbf{w}_{\text{target}}^{\top} \mathbf{A}$ *// Mixed advantage*
      $\omega(s, a) = \min\left(\exp\left(\frac{A_{\text{mixed}}(s,a)}{\beta} \right), \omega_{max} \right)$
      $\mathcal{L}_{\text{MAPEX}} \leftarrow \mathbb{E}\left[\omega(s, a) \cdot \|a - \pi_{\text{new}}(s)\|_2^2 \right]$
      Update$(\pi_{\text{new}}, \mathcal{L}_{\text{MAPEX}})$
    $\Pi \leftarrow \Pi \cup \{\pi_{\text{new}}\}$
  $\mathcal{P} \leftarrow$ FindParetoFront(Π)
  return $\mathcal{P}$

required to interpolate the gap and guides the subsequent hybrid buffer creation and advantage mixing steps.

4.2 Step 2: Hybrid Buffer Creation and Advantage Mixing

After deriving the target weight vector $\mathbf{w}_{\text{target}}$, we assemble a training distribution that reflects this desired trade-off. We construct a fixed-size *hybrid buffer*, $\mathcal{D}_{\text{hybrid}}$, by sampling transitions from each specialist's buffer $\mathcal{D}_k$ in direct proportion to the corresponding weight $\mathbf{w}_{\text{target},k}$. This creates a dataset that is structurally biased to each objective in the desired proportion.

We then initialise a random policy network π_{new} that will be optimised to achieve the target trade-off behaviour. For this optimisation, we iteratively sample transitions from $\mathcal{D}_{hybrid}$ to compute the mixed advantage training signal. For each transition (s, a), we compute a vector of advantages $\mathbf{A}(s, a) \in \mathbb{R}^N$. The k^{th} element of this advantage vector is the advantage on the k^{th} objective associated with that transition, and is computed using the k^{th} specialist critic Q_k:

$$A_k(s, a) = Q_k(s, a) - Q_k(s, \pi_{\text{new}}(s)) \tag{8}$$

This formulation leverages the specific value estimation expertise of each critic for their respective objective.

Finally, we scalarise these vector-valued advantages into a single training signal. We compute the *mixed advantage* as the dot product of the advantage vector and the target weights derived in Step 1:

$$A_{\text{mixed}}(s, a) = \mathbf{w}_{\text{target}}^{\top} \cdot \mathbf{A}(s, a) \tag{9}$$

This scalar value A_{mixed} represents the quality of the state-action (s, a) specifically regarding the desired trade-off $\mathbf{w}_{\text{target}}$.

4.3 Step 3: Mixed Advantage Weighted Regression

To train the offspring policy π_{new}, we employ a supervised regression objective weighted by the scalarised signal derived in Equation 9. Our goal is to selectively clone actions that contribute positively to the specific target trade-off $\mathbf{w}_{\text{target}}$.

For a transition (s, a) we compute a regression weight $\omega(s, a)$ by applying a temperature-scaled exponential to its mixed advantage:

$$\omega(s, a) = \min\left(\exp\left(\frac{A_{\text{mixed}}(s, a)}{\beta}\right), \omega_{max}\right) \tag{10}$$

where $\beta > 0$ is a temperature hyperparameter and ω_{max} is a clipping threshold to ensure numerical stability.

The fresh policy π_{new} is then optimised to minimise the weighted mean squared error between its predicted action and the retrieved buffer action a:

$$\mathcal{L}_{\text{MAPEX}} = \mathbb{E}_{(s,a)}\left[\omega(s, a) \cdot \|\pi_{\text{new}}(s) - a\|_2^2\right] \tag{11}$$

Once the policy has been updated for the desired epochs, it is reinserted into the population Π and the MAPEX procedure is executed for another iteration.

4.4 Mitigating Out-of-Distribution Error

While MAPEX's mixed advantage values are intuitive to compute, they expose two main sources of out-of-distribution (OOD) error: 1) when a transition sampled by one objective's specialist policy is evaluated by another objective's specialist critic, and 2) when a randomly-initialised policy's action is evaluated by specialist critics. We mitigate these issues using the following design choices.

4.4.1 Secondary Critics. When training each specialist policy π_k, we learn not only its *primary* critic for objective k, but also a set of *secondary critics* for the remaining objectives. All critics associated with specialist k are trained on the same replay buffer $\mathcal{D}_k$ generated by π_k, so that every objective can be evaluated on data that is in-distribution for the corresponding critic.

Notation: Let $m \in \{1, \ldots, N\}$ index objectives and $k \in \{1, \ldots, N\}$ index specialists. We denote by $Q_m^{(k)}(s, a)$ the action-value critic that predicts the return for objective m using transitions sampled from $\mathcal{D}_k$ (i.e., collected under π_k). Under this notation, the specialist's *primary critic* is $Q_k^{(k)}$, while the *secondary critics* are $\{Q_m^{(k)}\}_{m \neq k}$.

Procedure: During specialist training, only the primary critic $Q_k^{(k)}$ is used to update the policy π_k. The secondary critics $\{Q_m^{(k)}\}_{m \neq k}$ are trained in parallel on the same buffer $\mathcal{D}_k$, but they do not contribute gradients to the policy update. After training all specialists, we collect the resulting critics into a single family $\boldsymbol{Q} \triangleq \{Q_m^{(k)}\}_{k=1\ldots N,\ m=1\ldots N}$. This construction ensures that when MAPEX samples a transition (s, a) that originates from a particular specialist buffer $\mathcal{D}_k$, the evaluations $\{\boldsymbol{Q}_m^{(k)}(s, a)\}_{m=1}^N$ are produced by critics trained on the same state–action distribution as the sampled data.

Practical note: Secondary critics may be trained alongside primary critics with no change to the policy update rule. If single-objective specialists have already been trained, a secondary critic $\{Q_m^{(k)}\}_{m \neq k}$ can be trained offline using transitions from $\mathcal{D}_k$ and the corresponding objective rewards. Either way, a distribution-matched critic family $\boldsymbol{Q}$ can be learnt for a low cost. We include a comparison of joint-vs. post-hoc-trained secondary critics in Section 5.

4.4.2 Offspring Policy Warm-Up. In step 2 of the MAPEX procedure (Equation 8), computing the mixed advantage requires querying specialist critics with actions proposed by $\pi_{\text{new}}(s)$. If π_{new} is initialised arbitrarily, these actions can be far from the support of the hybrid buffer and cause OOD error. To reduce this effect, we warm up π_{new} by regressing it to the mean of its parents in the action space.

Let $\{\pi_{p_1}, \ldots, \pi_{p_N}\}$ denote the parent policies. The mean parent action at state s is $\bar{a}(s) \triangleq \frac{1}{N}\sum_{j=1}^{N} \pi_{p_j}(s)$, which we use to perform a brief behavioural regression step that minimises

$$L_{\text{init}}(\theta) \triangleq \mathbb{E}_{s \sim \mathcal{D}_{\text{hybrid}}}\left[\|\pi_{\text{new}}(s) - \bar{a}(s)\|_2^2\right], \tag{12}$$

We run this regression for a small number of gradient steps prior to computing the advantage vector in Equation 8. This warm up keeps π_{new}'s predicted actions close to those produced by the parents on states drawn from $\mathcal{D}_{\text{hybrid}}$, more closely matching each critic's training exposure, and improving the reliability of subsequent critic-based updates.

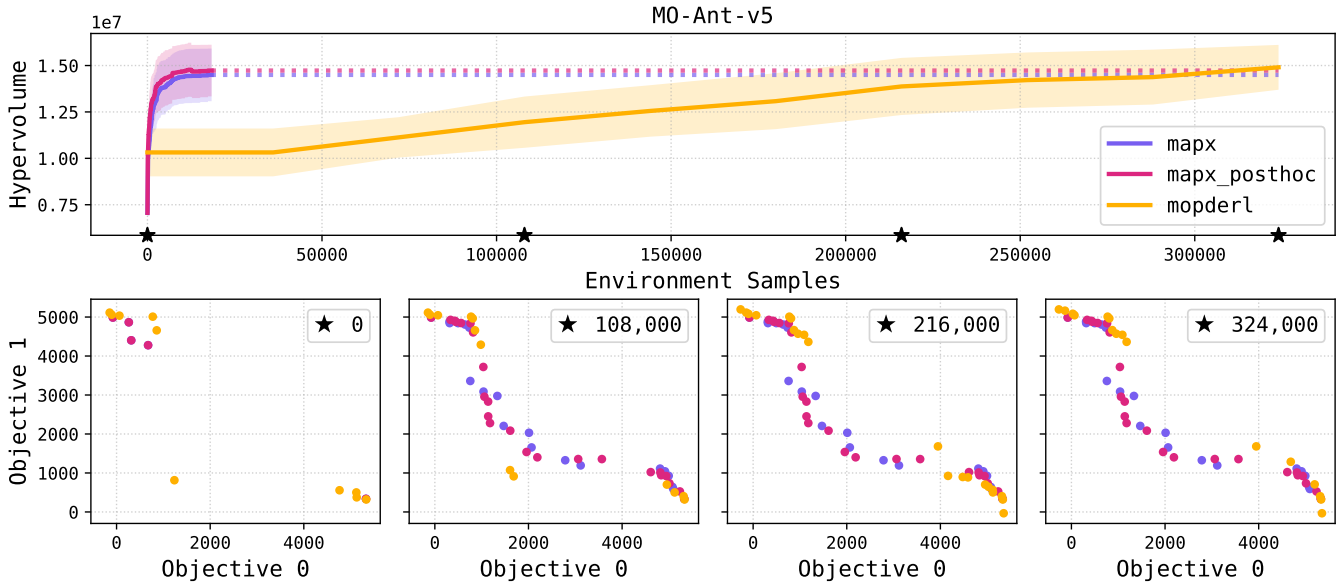

Figure 1: Sample efficiency comparison on MO-Ant-v5. (Top) Mean hypervolume ± SEM vs. cumulative environment samples. MAPEX and MAPEX-PostHoc achieve high hypervolume almost instantaneously, while MOPDERL requires significantly more interaction. (Bottom) Evolution of the Pareto front approximation. MAPEX/MAPEX-PostHoc fill the front immediately, whereas MOPDERL gradually expands coverage over 300,000+ environment samples.

5 EXPERIMENTS

We evaluate MAPEX on three criteria: **sample efficiency** of Pareto extraction; **flexibility** regarding choice of specialist training algorithm and nature of secondary critic training (joint vs. post-hoc); and **general competitiveness** against baselines when training from scratch.

We assume access to secondary critics trained alongside primary critics during specialist training, but also test a *post hoc* variant (MAPEX-PostHoc) where critics are trained offline on static buffers. Unless stated otherwise, specialists are trained using Proximal Distilled Evolutionary RL (PDERL) [3].

5.1 Baselines and Domains

We compare MAPEX against two established MORL methods:

- **MOPDERL** [25]: An evolutionary actor-critic method that first trains on each individual objective using PDERL, and later crosses over solutions across objectives using a multi-objective distilled crossover. It serves as a baseline for both 'from scratch' training and pure Pareto extraction (via its distillation phase).
- **MORL/D** [8]: A decomposition-based approach that uses Soft Actor–Critic [10] on each scalar objective. It uses buffer data sharing and adapts the scalar weights using Pareto Simulated Annealing [6].

We use the `morl-baselines` [7] implementation of MORL/D and the authors' original implementation of MOPDERL with default hyperparameters. Experiments are conducted on five bi-objective continuous control MuJoCo environments from `MO-Gymnasium` [7], with episodes capped at 750 frames. The nature of objectives in each environment are specified in Appendix A table 2.

5.2 Experimental Setup

Sample efficiency: We compare the extraction phase of MAPEX against the distillation phase of MOPDERL starting from identical specialist subpopulations trained with PDERL.

Flexibility: We compare standard MAPEX against MAPEX-PostHoc and MAPEX-TD3 (specialists trained via TD3) to assess robustness to starting policies and nature of critic training.

Competitiveness: We compare the final Pareto fronts of the full MAPEX pipeline against MOPDERL and MORL/D given a fixed sample budget. During specialist training of all MAPEX variants, we use a replay buffer of size 1M. In each test we perform five seeded runs with each method. Exact algorithm-specific and experiment parameters are listed in Appendix B.

6 RESULTS

6.1 Sample Efficiency of Extraction

MAPEX achieves a massive reduction in sample cost compared to baselines. As shown in Figure 1 (MO-Ant-v5), MAPEX and MAPEX-PostHoc extract a high-performing front almost immediately, while MOPDERL requires an additional 300, 000 environment interactions to attain the same performance.

Figure 2 confirms this trend across MO-Hopper-v5 and MO-Walker2d-v5. Particularly in MO-Hopper-v5, MAPEX requires 100 samples to reach hypervolume thresholds that MOPDERL requires $\approx 10^5$ samples to attain—a reduction of three orders of magnitude. This efficiency is driven by MAPEX exploiting the latent representations of multi-objective behaviour in policies and replay buffers. It empirically validates our intuition that following expert behaviour to varying degrees on each objective yields trade-offs

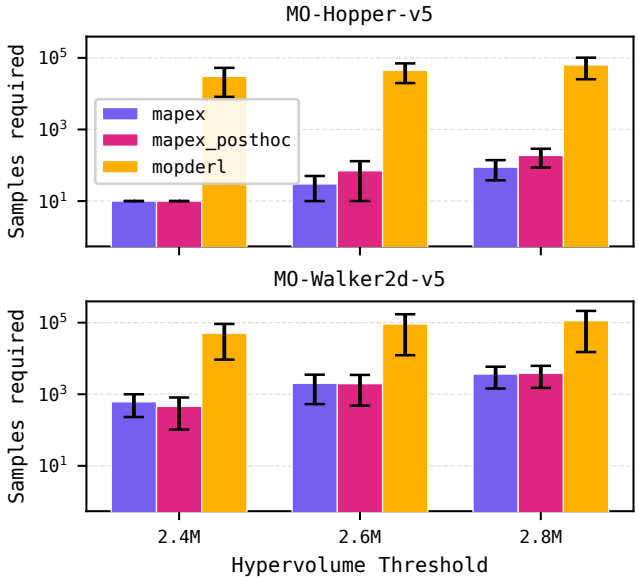

Figure 2: Samples required to attain target hypervolume thresholds. Comparison on MO-Hopper-v5 (top) and MO-Walker2d-v5 (bottom). Note the logarithmic scale on the y-axis. MAPEX and MAPEX-PostHoc require up to three orders of magnitude fewer samples (10^2 vs 10^5) than MOPDERL to reach identical performance levels.

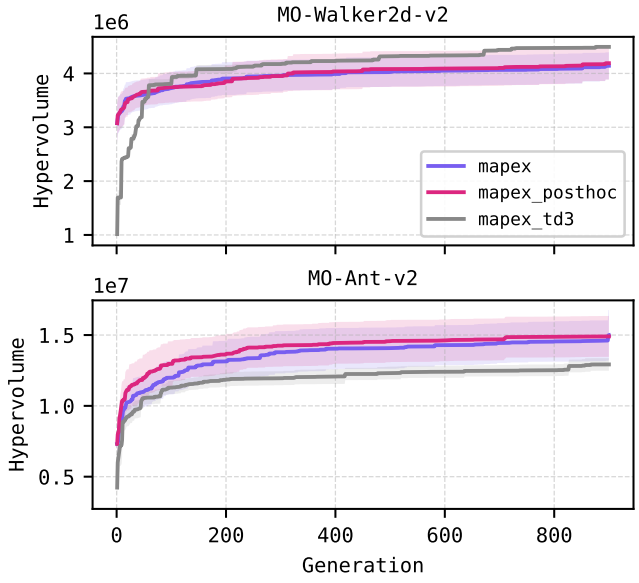

Figure 3: Robustness of MAPEX to specialist type and critic training. Mean hypervolume (± SEM) over generations on MO-Walker2d-v5 and MO-Ant-v5. The similar performance of standard MAPEX, MAPEX-PostHoc (offline critics), and MAPEX-TD3 (off-policy specialists) demonstrates the method's flexibility in effectively extracting fronts from decoupled pre-trained sources.

in the objective space. It also validates our main hypothesis that these behaviours can be learnt by regressing onto target actions, weighed by their *mixed advantage* value.

6.2 Flexibility and Robustness

MAPEX is robust to the source of specialist policies. Figure 3 shows that Pareto extraction for standard MAPEX, MAPEX-PostHoc, and MAPEX-TD3 are largely indistinguishable on MO-Walker2d-v5 and MO-Ant-v5. Crucially, the success of MAPEX-PostHoc confirms that secondary critics can be effectively trained retroactively on static buffers, enabling Pareto extraction from fully decoupled single-objective training. If simplicity is key, then the performance of MAPEX-TD3 shows the potential of elegantly adapting off-policy RL to learn multi-objective behaviours via MAPEX.

While a population of policies (like that produced by PDERL) provides a rich starting point for Pareto extraction, MAPEX-TD3 remains competitive with only one specialist policy per objective (as produced by simple TD3). This is because MAPEX does not simply distil or interpolate between parent policies. Instead, MAPEX draws expertise from the pre-trained replay buffers, which contain a rich and varied set of state-action examples. These experiences span regions of the objective space that an individual policy may only cover sparsely, allowing MAPEX to produce distinct policies for distinct trade-offs. This is also why MAPEX is robust to the exact algorithm used to train specialist policies.

6.3 General Competitiveness

Despite being an offline extraction method, MAPEX produces fronts competitive with MOPDERL and MORL/D, which require full multi-objective consideration from the beginning. Table 1 shows that MAPEX achieves comparable hypervolumes (e.g., 3.34×10^6 vs. MOPDERL's $3.17{\times}10^6$ in MO-Hopper-v5, and $1.78{\times}10^5$ vs. MORL/D's 9.29×10^4 in MO-Swimmer-v5) across five multi-objective MuJoCo environments. While MAPEX fronts are numerically sparser, by inspecting the fronts in Figure 4 visually, it is clear that MAPEX produces even and well-spread Pareto fronts.

7 CONCLUSION

We presented MAPEX, a novel approach to extracting Pareto fronts of policies from prior single-objective training for continuous control. We provided a detailed view of MAPEX's Pareto extraction procedure, and mentioned some practial tips. We tested MAPEX with well-established, dedicated MORL baselines like MOPDERL and MORL/D to empirically validate its mixed advantage approach. Pareto fronts are learnt cheaply with MAPEX when specialists are already trained. If training from scratch, MAPEX integrates easily with off-policy RL methods and still produces fronts that compare well to baselines. Finally, we discuss some limitations and future work.

While MAPEX is highly sample efficient, it makes assumptions inherent to our offline extraction setting. First, MAPEX is strictly bounded by the support of the specialist buffers; it cannot discover novel behaviours or skills that are absent from the specialists' training history. Second, MAPEX relies on the assumption that valid trade-off policies lie on a continuous manifold between specialists. In scenarios where specialists exhibit markedly disjoint behaviours

Table 1: Performance metrics (Hypervolume and Sparsity) across v5 `MO-Gymnasium` MuJoCo environments. Results are Mean ± SEM. Hypervolume is the space between the Pareto front and a dominated reference point and sparsity is the average euclidean distance between neighbouring points.

	Hypervolume (↑)			Sparsity (↓)		
Environment	**MAPX**	**MOPDERL**	**MORL/D**	**MAPEX**	**MOPDERL**	**MORL/D**
Ant-2obj	1.19e7 ± 1.2e6	1.46e7 ± 8.2e5	1.41e7 ± 1.9e4	315.5 ± 36.1	**201.4 ± 13.8**	289.8 ± 8.3
Hopper-2obj	3.34e6 ± 2.9e5	3.17e6 ± 2.8e5	**4.31e6 ± 1.7e5**	209.6 ± 39.8	**64.2 ± 10.3**	115.5 ± 17.7
Swimmer	1.78e5 ± 1.6e4	**2.41e5 ± 2.9e4**	9.29e4 ± 5.9e2	49.5 ± 9.2	30.0 ± 3.2	**2.9 ± 0.3**
Walker2d	3.09e6 ± 2.2e5	3.47e6 ± 3.5e5	**6.52e6 ± 4.2e5**	157.5 ± 7.9	103.2 ± 16.0	96.9 ± 11.2
HalfCheetah	1.10e7 ± 5.3e5	1.55e7 ± 6.8e5	**1.88e7 ± 5.4e4**	337.5 ± 26.7	**101.7 ± 5.9**	136.4 ± 19.6

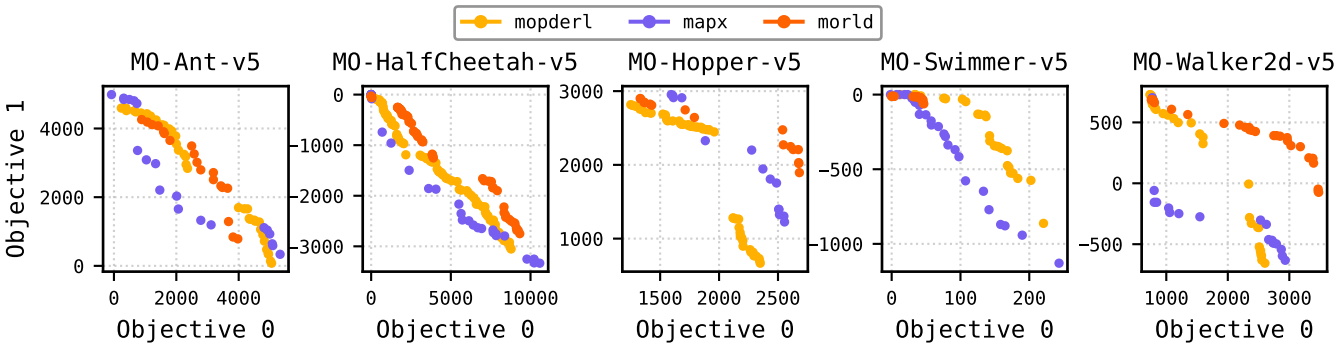

Figure 4: Final Pareto fronts across five MO-MuJoCo benchmarks. Comparison of fronts extracted by MAPEX against fully trained MOPDERL and MORL/D baselines. Despite relying purely on single-objective training data, MAPEX recovers fronts that are dense and competitive with baselines trained from scratch.

(e.g., a humanoid walking on two legs vs. crawling), interpolation may yield low-performance policies. Our empirical evaluation focused on bi-objective domains; scaling MAPEX's gap-identification heuristic to more objectives ($N \geq 3$) remains a subject for future investigation. Finally, we would like to leverage MAPEX with a multiagent reinforcement learning algorithm to extend multi-objective decision-making to the multiagent setting.

REFERENCES

[1] Axel Abels, Diederik M. Roijers, T. Lenaerts, Ann Nowé, and Denis Steckelmacher. 2018. Dynamic Weights in Multi-Objective Deep Reinforcement Learning. *ArXiv* abs/1809.07803 (2018). https://api.semanticscholar.org/CorpusID:52345417

[2] Toygun Basaklar, Suat Gumussoy, and Umit Ogras. 2023. PD-MORL: Preference-Driven Multi-Objective Reinforcement Learning Algorithm. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=zS9sRyaPFlJ

[3] Cristian Bodnar, Ben Day, and Pietro Lio. 2020. Proximal Distilled Evolutionary Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (04 2020), 3283–3290. https://doi.org/10.1609/aaai.v34i04.5728

[4] Diqi Chen, Yizhou Wang, and Wen Gao. 2020. Combining a gradient-based method and an evolution strategy for multi-objective reinforcement learning. *Applied Intelligence* 50, 10 (Oct. 2020), 3301–3317. https://doi.org/10.1007/s10489-020-01702-7

[5] Xi Chen, Ali Ghadirzadeh, Mårten Björkman, and Patric Jensfelt. 2019. Meta-Learning for Multi-objective Reinforcement Learning. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Macau, China). IEEE Press, 977–983. https://doi.org/10.1109/IROS40897.2019.8968092

[6] Piotr Czyżak and Adrezej Jaszkiewicz. 1998. Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-criteria Decision Analysis* 7 (1998), 34–47. https://api.semanticscholar.org/CorpusID:123140619

[7] Florian Felten, Lucas N. Alegre, Ann Nowé, Ana L. C. Bazzan, El Ghazali Talbi, Grégoire Danoy, and Bruno C. da Silva. 2023. A Toolkit for Reliable Benchmarking and Research in Multi-Objective Reinforcement Learning. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*.

[8] Florian Felten, El-Ghazali Talbi, and Grégoire Danoy. 2024. Multi-Objective Reinforcement Learning Based on Decomposition: A Taxonomy and Framework. *J. Artif. Int. Res.* 79 (April 2024), 45. https://doi.org/10.1613/jair.1.15702

[9] Scott Fujimoto, Herke van Hoof, and David Meger. 2018. Addressing Function Approximation Error in Actor-Critic Methods. In *International Conference on Machine Learning*. https://api.semanticscholar.org/CorpusID:3544558

[10] Tuomas Haarnoja, Aurick Zhou, P. Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *ArXiv* abs/1801.01290 (2018). https://api.semanticscholar.org/CorpusID:28202810

[11] Conor F. Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. 2022. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems* 36, 1 (April 2022), 59. https://doi.org/10.1007/s10458-022-09552-y

[12] Shauharda Khadka and Kagan Tumer. 2018. Evolution-guided policy gradient in reinforcement learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Montréal, Canada) *(NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 1196–1208.

[13] Sergey Levine, Aviral Kumar, G. Tucker, and Justin Fu. 2020. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *ArXiv* abs/2005.01643 (2020). https://api.semanticscholar.org/CorpusID:218486979

[14] Qian Lin, Chao Yu, Zongkai Liu, and Zifan Wu. 2024. Policy-regularized Offline Multi-objective Reinforcement Learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems* (Auckland, New Zealand) *(AAMAS '24)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1201–1209.

[15] Xi Lin, Xiaoyuan Zhang, Zhiyuan Yang, Fei Liu, Zhenkun Wang, and Qingfu Zhang. 2024. Smooth Tchebycheff scalarization for multi-objective optimization. In *Proceedings of the 41st International Conference on Machine Learning* (Vienna,

Austria) *(ICML'24)*. JMLR.org, Article 1227, 31 pages.
[16] Erlong Liu, Yu-Chang Wu, Xiaobin Huang, Chengrui Gao, Ren-Jian Wang, Ke Xue, and Chao Qian. 2025. Pareto set learning for multi-objective reinforcement learning. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence (AAAI'25/IAAI'25/EAAI'25)*. AAAI Press, Article 2095, 9 pages. https://doi.org/10.1609/aaai.v39i18.34068
[17] Kristof Van Moffaert and Ann Nowé. 2014. Multi-Objective Reinforcement Learning using Sets of Pareto Dominating Policies. *Journal of Machine Learning Research* 15, 107 (2014), 3663–3692. http://jmlr.org/papers/v15/vanmoffaert14a.html
[18] Kishan Panaganti, Zaiyan Xu, Dileep Kalathil, and Mohammad Ghavamzadeh. 2025. Bridging Distributionally Robust Learning and Offline RL: An Approach to Mitigate Distribution Shift and Partial Data Coverage. In *Proceedings of the 7th Annual Learning for Dynamics & Control Conference (Proceedings of Machine Learning Research, Vol. 283)*, Necmiye Ozay, Laura Balzano, Dimitra Panagou, and Alessandro Abate (Eds.). PMLR, 619–634. https://proceedings.mlr.press/v283/panaganti25a.html
[19] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. 2019. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177* (2019).
[20] Mathieu Reymond, Eugenio Bargiacchi, and Ann Nowé. 2022. Pareto Conditioned Networks. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems* (Virtual Event, New Zealand) *(AAMAS '22)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1110–1118.
[21] Mathieu Reymond, Conor Hayes, Denis Steckelmacher, Diederik Roijers, and Ann Nowe. 2023. Actor-critic multi-objective reinforcement learning for non-linear utility functions. *Autonomous Agents and Multi-Agent Systems* 37 (04 2023). https://doi.org/10.1007/s10458-023-09604-x
[22] Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. 2013. A survey of multi-objective sequential decision-making. *J. Artif. Int. Res.* 48, 1 (Oct. 2013), 67–113.
[23] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. 2017. Evolution Strategies as a Scalable Alternative to Reinforcement Learning. arXiv:1703.03864 [stat.ML] https://arxiv.org/abs/1703.03864
[24] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA.
[25] Hai-Long Tran, Long Doan, Ngoc Hoang Luong, and Huynh Thi Thanh Binh. 2023. A Two-Stage Multi-Objective Evolutionary Reinforcement Learning Framework for Continuous Robot Control. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Lisbon, Portugal) *(GECCO '23)*. Association for Computing Machinery, New York, NY, USA, 577–585. https://doi.org/10.1145/3583131.3590441
[26] Kristof Van Moffaert, Madalina Drugan, and Ann Nowe. 2013. Scalarized Multi-Objective Reinforcement Learning: Novel Design Techniques. *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, ADPRL*. https://doi.org/10.1109/ADPRL.2013.6615007
[27] Yuchen Xiao, Lei Yuan, Lihe Li, Ziqian Zhang, Yichen Li, and Yang Yu. 2025. Generalizable Offline Multiobjective Reinforcement Learning via Preference-Conditioned Diffuser. *IEEE Transactions on Neural Networks and Learning Systems* 36, 12 (2025), 20199–20213. https://doi.org/10.1109/TNNLS.2025.3591838
[28] Jie Xu, Yunsheng Tian, Pingchuan Ma, Daniela Rus, Shinjiro Sueda, and Wojciech Matusik. 2020. Prediction-Guided Multi-Objective Reinforcement Learning for Continuous Robot Control. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 10607–10616. https://proceedings.mlr.press/v119/xu20h.html
[29] Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. 2019. *A generalized algorithm for multi-objective reinforcement learning and policy adaptation*. Curran Associates Inc., Red Hook, NY, USA.
[30] Baiting Zhu, Meihua Dang, and Aditya Grover. 2023. Scaling Pareto-Efficient Decision Making via Offline Multi-Objective RL. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=Ki4ocDm364

A ENVIRONMENT DETAILS

Table 2 specifies each objective across five multi-objective MuJoCo environments. Episodes in each environment are capped at 750 environment steps.

Table 2: The v5 `MO-Gymnasium` MuJoCo environments. Objectives include velocity, energy efficiency, jump height, and stability and must all be *maximised*. The action dimensions are d_a, and d_o the observation (state) dimensions.

Env.	d_a	d_o	Objective 1	Objective 2
Ant-2obj	8	105	x-vel	y-vel
Hopper-2obj	3	11	x-vel & survival	Jump (w/ cost)
Swimmer	2	8	x-vel	Energy eff.
Walker2d	6	17	x-vel	Energy eff.
HalfCheetah	6	17	x-vel	Energy eff.

B EXPERIMENTAL HYPERPARAMETERS

We report the hyperparameters used in our experiments. Table 3 lists the settings for the base TD3 algorithm and the PDERL algorithm which uses TD3, which are shared across all environments. Table 4 details the general parameters for MAPEX, and Table 5 lists the environment-specific values for policy warm up and Pareto extraction. Finally Table 6 lists the number of frames each algorithm was run for, and the division of frames for MOPDERL, which contains distinct warm up and Pareto distillation phases.

Table 3: PDERL and TD3 hyperparameters (Shared across environments).

Parameter	Value
PDERL	
Population Size	10
Mini Buffer Size	50,000
TD3 / General	
Start Timesteps	25,000
Discount Factor (γ)	0.99
Target Smoothing (τ)	0.005
Hidden Dimension	256
Actor Learning Rate	3×10^{-4}
Critic Learning Rate	3×10^{-4}
Batch Size	256
Buffer Size	1×10^{6}
Exploration Noise	0.1
Policy Noise	0.2
Noise Clip	0.5
Policy Frequency	2

Table 4: MAPEX hyperparameters shared across environments.

Hyperparameter	Value
Total iterations	1,200
Child buffer size	200,000
Warm-up steps / epoch	1,000
Warm-up learning rate	3×10^{-4}
Warm-up batch size	256
Evaluation episodes / actor	5

Table 5: Environment-specific MAPEX hyperparameters across five v5 `MO-Gymnasium` MuJoCo environments.

Parameter	Environment				
	Ant-2obj	**Hopper-2obj**	**Swimmer**	**Walker2d**	**HalfCheetah**
MAPEX epochs	20	10	10	20	20
AWR β	0.5	0.1	1.0	0.5	1.0
AWR clip ($\omega_{\max}$)	1.0	20.0	20.0	20.0	1.0

Table 6: Environment interaction budgets (frames) per method and environment. For bi-objective tasks (N=2), the "/ obj." column reports frames per objective-specific training process (PDERL for MOPDERL; PDERL / TD3 for MAPEX). MAPEX extraction uses 0 additional environment interaction.

Environment	MOPDERL				MAPEX (specialists)		MORL/D Total
	Warm up	**Warm up / obj.**	**Stage 2**	**Total**	**Specialist / obj.**	**Total**	
MO-Ant-2obj-v5	2.0M	1.0M	2.0M	4.0M	2.0M	4.0M	4.0M
MO-Hopper-2obj-v5	2.25M	1.125M	1.75M	4.0M	2.0M	4.0M	4.0M
MO-Swimmer-v5	1.0M	0.5M	1.0M	2.0M	1.0M	2.0M	2.0M
MO-Walker2d-v5	2.0M	1.0M	2.0M	4.0M	2.0M	4.0M	4.0M
MO-HalfCheetah-v5	2.0M	1.0M	2.0M	4.0M	2.0M	4.0M	4.0M