EXPLOITING REFLECTIONAL SYMMETRY IN HETEROGENEOUS MORL

Anonymous authors
Paper under double-blind review

ABSTRACT

This work studies heterogeneous Multi-Objective Reinforcement Learning (MORL), where objectives exhibit considerable discrepancies in, amongst others, sparsity and magnitude. The heterogeneity can cause dense objectives to overshadow sparse but long-term rewards, leading to sample inefficiency. To address this issue, we propose Parallel Reward Integration with reflectional Symmetry for heterogeneous MORL (PRISM), a novel algorithm that aligns reward channels and enforces reflectional symmetry as an inductive bias. We design ReSymNet, a theory-inspired model that aligns time frequency and magnitude across objectives, leveraging residual blocks to gradually learn a 'scaled opportunity value' for accelerating exploration while maintaining the optimal policy. Based on the aligned reward objectives, we then propose SymReg, a reflectional equivariance regulariser to enforce reflectional symmetry in terms of agent mirroring. SymReg constrains the policy search to a reflection-equivariant subspace that is provably of reduced hypothesis complexity, thereby improving generalisability. Across MuJoCo benchmarks, PRISM consistently outperforms the baseline and oracle (with full dense rewards) in both Pareto coverage and distributional balance, achieving hypervolume gains of over 100% against the baseline and even up to 32% against the oracle. The code is at `https://anonymous.4open.science/r/reward_shaping-1CCB`.

1 INTRODUCTION

Reinforcement Learning (RL) has been approaching human-level capabilities in many decision-making tasks, such as playing Go (Silver et al., 2017), autonomous vehicles (Kiran et al., 2021), robotics (Tang et al., 2025), and finance (Hambly et al., 2023). Multi-Objective Reinforcement Learning (MORL) extends this framework to handle multiple reward channels simultaneously, allowing agents to balance competing objectives efficiently (Liu et al., 2014; Hayes et al., 2022). For example, a self-driving car must constantly balance multiple goals, such as minimising travel time while maximising passenger safety and energy efficiency. Prioritising speed would compromise the safety objectives, introducing the need for flexible and robust policies that can optimise across diverse and sometimes conflicting goals.

This paper considers an important, yet premature, setting where reward channels exhibit considerable heterogeneity in facets such as sparsity and magnitude. Dense objectives can overshadow their sparse and long-horizon counterparts, steering policies toward short-term gains, while neglecting the objectives that are harder to optimise but potentially more important. A straightforward approach is to employ reward shaping methods to align the reward channels. However, existing algorithms, such as intrinsic curiosity (Pathak et al., 2017; Aubret et al., 2019) and attention-based exploration (Wei et al., 2025), are developed for single-objective cases and have significant deficiencies: separately shaping individual objectives can distort the Pareto front and structures between objectives. This highlights a critical gap in the literature: MORL requires a reward shaping method that enables efficient integration of the parallel but heterogeneous reward signals, leveraging their intrinsic structure, in order to improve sample efficiency.

To this end, we propose Parallel Reward Integration with reflectional Symmetry for Multi-objective reinforcement learning (PRISM), a method that structurally shapes the reward channels and leverages the reflectional symmetry in agents in heterogeneous MORL problems. We design a Reward

Symmetry Network (ReSymNet) that predicts the reward given the state of the system and any available performance indicators (e.g., dense rewards in this work). The available sparse rewards are used as supervised targets. In ReSymNet, residual blocks are employed to approximate the 'scaled opportunity value', which has been proven to help accelerate training, decrease the approximation error, while maintaining the optimal solution of the native reward signals (Laud, 2004). ReSymNet stacks residual blocks that progressively refine per-step predictions through additive corrections, reconstructing dense reward signals. It aims at maintaining consistent optima with the original sparse objectives while ironing out the heterogeneity and enhancing performance. After proper training, our ReSymNet can be a plug-and-play technique, compatible with any off-the-shelf MORL algorithm in an iterative refinement cycle, where the agent observes the shaped rewards to improve its policy and the reward model observes better trajectories from the updated policy to improve the approximated reward function. To exploit the structural information across reward signals, we design a Symmetry Regulariser (SymReg) to enforce reflectional equivariance of the objectives, which provably reduces the hypothesis complexity. Intuitively, incorporating reflectional symmetry as an inductive bias allows an agent to generalise experience from one situation to its mirrored counterpart.

We prove that PRISM constrains the policy search into a subspace of reflection-equivariant policies. This subspace is a projection of the original policy space, induced by the reflectional symmetry operator, provably of reduced hypothesis complexity, measured by covering number (Zhou, 2002) and Rademacher complexity (Bartlett & Mendelson, 2002). This reduced complexity is further translated to improved generalisation guarantees. In practice, this means that by encouraging policies to respect natural symmetries, the agent effectively searches over a smaller, more structured hypothesis space, reducing overfitting and improving sample efficiency. We further extend this analysis to the approximately reflection-equivariant cases, where PRISM does not necessarily converge to the reflection-equivariant subspace exactly, showing that policies in this more realistic setting inherit similarly improved generalisability.

We conduct extensive experiments on the MuJoCo MORL environments (Todorov et al., 2012), using Concave-Augmented Pareto Q-learning (CAPQL) (Lu et al., 2023) as the backbone for PRISM. Sparse rewards are constructed by releasing cumulative rewards at the end of an episode. PRISM achieves hypervolume gains of over 100% against the baseline operating directly on sparse signals, and even up to 32% over the oracle (full dense rewards), indicating a substantially improved Pareto front coverage. These gains are echoed in distributional metrics, confirming that PRISM learns a set of policies that are also better balanced and more robust. Comprehensive ablation studies further confirm that the design of ReSymNet and the inclusion of SymReg are both critical.

2 Related Work

Multi-Objective Reinforcement Learning. MORL algorithms typically fall into three categories: (1) single-policy methods that optimise user-specified scalarisations (Moffaert et al., 2013; Lu et al., 2023; Hayes et al., 2022); (2) multi-policy methods that approximate the Pareto front by solving multiple scalarisations or training policies in parallel (Roijers et al., 2015; Van Moffaert & Nowé, 2014; Reymond & Nowé, 2019; Lautenbacher et al., 2025); and (3) meta-policy methods that learn adaptable policies conditioned on preferences (Yang et al., 2019; Basaklar et al., 2023; Mu et al., 2025; Liu et al., 2025). While these works have advanced Pareto-optimal learning, less attention has been given to heterogeneity in reward structures.

Reward Shaping. A large volume of literature tackles sparse rewards through reward shaping. Potential-based shaping (Ng et al., 1999) ensures policy invariance but requires hand-crafted potentials. However, this method's reliance on a manually designed potential function proved limiting. Intrinsic motivation methods reward novelty or exploration (Pathak et al., 2017; Burda et al., 2019), while self-supervised methods predict extrinsic returns from trajectories (Memarian et al., 2021; Devidze et al., 2022; Holmes & Chi, 2025). These approaches improve sample efficiency in single-objective RL, but do not extend naturally to MORL, where heterogeneous sparsity and scale can distort learning dynamics and Pareto-optimal trade-offs.

Reflectional Equivariance. To incorporate reflectional symmetry, a possible method is data augmentation, which adds mirrored transitions to the replay buffer but doesn't guarantee a symmetric policy and increases data processing costs (Lin et al., 2020). Mondal et al. (2022) propose latent space learning that encourages a symmetric representation through specialised loss functions. Wang

et al. (2022) design a stronger inductive bias via architecture-level symmetry, which hard-codes equivariance into the model for instantaneous generalisation. However, Park et al. (2025) show that strictly equivariant architectures can be too rigid for tasks where symmetries are approximate rather than perfect. Building on this insight, our framework helps overcome the limitations of strictly equivariant architectures through tunable flexibility whilst being model-agnostic.

3 PRELIMINARIES

Multi-Objective Markov Decision Process. Formally, we define an MORL problem via the Multi-Objective Markov Decision Process (MOMDP) model, as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \boldsymbol{r}, \gamma)$: an agent at state s from a finite or continuous state space $\mathcal{S}$, taking action a from a finite or continuous action space $\mathcal{A}$, moves herself according to a transition probability function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S}' \to [0, 1]$, also denoted as $P(s'|s, a)$. The agent receives a reward via an L-dimensional vector-valued reward function $\boldsymbol{r} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^L$, where L is the reward channel number, which decays by a discount factor $\gamma \in [0, 1)$. The goal in MORL is to find a policy $\pi : \mathcal{S} \to \mathcal{A}$ that optimises the expected cumulative vector return, defined as $\boldsymbol{J}^\pi = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \boldsymbol{r}_t\right]$. This paper addresses episodic tasks, where each interaction sequence has a finite horizon and concludes when the agent reaches a terminal state, at which point the environment is reset. Episodes τ_i are i.i.d. draws from the behaviour distribution $\mathcal{D}$.

Reward Sparsity. Reward sparsity can be modelled as releasing the cumulative reward accumulated since the last non-zero reward with probability p_{rel} at each timestep. When $p_{\text{rel}} = 0$, this reduces to the most extreme case: the agent receives rewards from dense channels $\mathcal{DC} = \{d_1, d_2, \ldots, d_D\}$ with observable rewards $r_t^{d_i}$ at every timestep, but the sparse channel is revealed only once at the end of the episode as $R_T^{sp} = \sum_{t=1}^{T} r_t^{sp}$. The central challenge is to recover instantaneous sparse rewards r_t^{sp} for each (s_t, a_t) using only the cumulative observation R_T^{sp} and correlations with dense channels. Formally, given a trajectory $\tau = \{(s_1, a_1), \ldots, (s_T, a_T)\}$ with cumulative sparse reward $R^{sp}(\tau)$, the task is to infer $\boldsymbol{r}^{sp} = [r_1^{sp}, \ldots, r_T^{sp}]^\top$ such that $\sum_{t=1}^{T} r_t^{sp} \approx R^{sp}(\tau)$. For $p_{\text{rel}} > 0$, an episode decomposes into sub-trajectories where the same formulation applies.

Generalisability and Hypothesis Complexity. A generalisation gap, at the episodic level, characterises the generalisability from a good empirical performance to its expected performance on new data (Wang et al., 2019). It depends on the hypothesis set's complexity, which is measured in this work by covering number (Zhou, 2002) and Rademacher complexity (Bartlett & Mendelson, 2002).

Definition 1 ($l_{\infty,1}$ distance)**.** *Let $\mathcal{X}$ be a feature space and $\mathcal{F}$ a space of functions from $\mathcal{X}$ to $\mathbb{R}^n$. The $l_{\infty,1}$-distance on the space $\mathcal{F}$ is defined as $l_{\infty,1}(f, g) = \max_{x \in \mathcal{X}} \left(\sum_{i=1}^{n} |f_i(x) - g_i(x)|\right)$.*

Definition 2 (covering number)**.** *The covering number, denoted $\mathcal{N}_{\infty,1}(\mathcal{F}, r)$, is the minimum number of balls of radius r required to completely cover the function space $\mathcal{F}$ under the $l_{\infty,1}$-distance.*

Definition 3 (Rademacher complexity)**.** *Let $\mathcal{F}$ be a class of real-valued functions on a feature space $\mathcal{X}$, and let $\tau_1, \ldots, \tau_N$ be i.i.d. samples from a distribution over $\mathcal{X}$. The empirical Rademacher complexity of $\mathcal{F}$ is $\hat{\mathfrak{R}}_N(\mathcal{F}) = \mathbb{E}_\sigma[\sup_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \sigma_i f(\tau_i)]$, where $\sigma_1, \ldots, \sigma_N$ are independent Rademacher random variables taking values ± 1 with equal probability. The Rademacher complexity of $\mathcal{F}$ is the expectation over the sample set.*

4 PARALLEL REWARD INTEGRATION WITH REFLECTIONAL SYMMETRY

This section introduces our algorithm PRISM.

4.1 RESYMNET: REWARD SYMMETRY NETWORK

To address the challenge of heterogeneous reward objectives, PRISM first transforms sparse rewards into dense, per-step signals. We frame this as a supervised learning problem, inspired by but distinct from inverse reinforcement learning, as we do not assume access to expert demonstrations (Ng & Russell, 2000; Arora & Doshi, 2021). The goal is to train a reward model, $\mathcal{R}_{\text{pred}}$, that learns to map state-action pairs to individual extrinsic rewards.

We hope to train the reward shaping model on a dataset collected by executing a purely random policy, ensuring broad state-space coverage. For each timestep t, we construct a feature vector

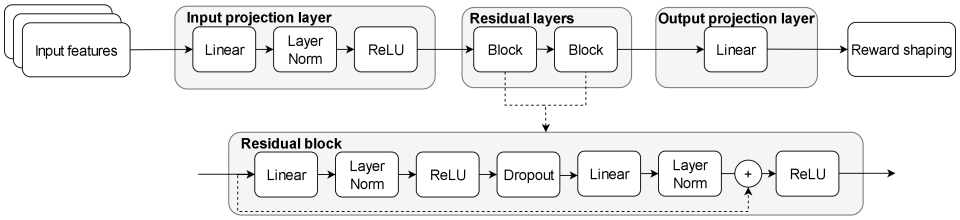


Figure 1: Overview of ReSymNet.

$\boldsymbol{h}_t = [s_t, a_t, \boldsymbol{r}_{\text{dense},t}]$, where s_t is the state, a_t is the action, and $\boldsymbol{r}_{\text{dense},t}$ are the dense rewards obtained from taking action a_t at state s_t, which crucially leverages the information from already-dense objectives to help predict the sparse ones. Figure 1 visualises the ResNet-like architecture.

Remark 1. *Residual connections in $\mathcal{R}_{pred}$ are inspired by the theory of scaled opportunity value (Laud, 2004), whose additive corrections preserve optimal policies, shorten the effective reward horizon, and improve local value approximation (see Appendix B).*

The network is optimised by minimising the mean squared error between the sum of its per-step predictions over a trajectory and the true cumulative sparse reward observed for that trajectory:

$$\mathcal{L}(\psi) = \sum_{\tau \in \mathcal{D}} \left(\sum_{t \in \tau} \mathcal{R}_{\text{pred}}(\boldsymbol{h}_t; \psi) - R^{sp}(\tau) \right)^2 . \tag{1}$$

To ensure the learned reward function is robust and adapts to the agent's improving policy, we incorporate two techniques: (1) we train an ensemble of reward models to reduce variance and produce a more stable shaping signal, and (2) we employ iterative refinement: the reward model is periodically updated using new, on-policy data collected by the agent. This allows the reward model to correct for the initial distribution shift and remain accurate as the agent's behaviour evolves from random exploration to expert execution, as outlined in Algorithm 1 in Appendix B.

4.2 SymReg: Enforcing Reflectional Equivariance

Many RL tasks exhibit natural reflectional symmetry. For example, for legged agents, flexing a leg is essentially the mirror image of extending it. Standard policies must learn both motions separately, wasting data. By encoding symmetry as an inductive bias, experience from one motion can be reused for its mirror, improving sample efficiency and robustness.

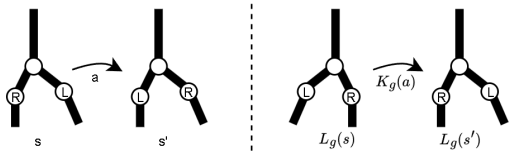


Figure 2: Reflectional symmetry in a two-legged agent. The left panel shows a transition from state s to s' under action a, whereas the right panel shows the reflected transition, where states and actions are transformed by L_g and K_g, respectively.

We formalise this physical intuition using group theory, specifically the reflection group $G = \mathbb{Z}_2$. This group consists of two transformations: the identity and a negation/reflection operator, g. Let $\mathcal{S} \subseteq \mathbb{R}^{d_s}$ and $\mathcal{A} \subseteq \mathbb{R}^{d_a}$ denote the state and action spaces respectively. We define index sets $I^s_{\text{asym}} \subset \{1, \ldots, d_s\}$ and $I^s_{\text{sym}} \subset \{1, \ldots, d_s\}$ such that $I^s_{\text{asym}} \cap I^s_{\text{sym}} = \emptyset$ and $I^s_{\text{asym}} \cup I^s_{\text{sym}} = \{1, \ldots, d_s\}$. This partitions the state vector as $s = (s_{\text{asym}}, s_{\text{sym}})$ where $s_{\text{asym}} = s_{I^s_{\text{asym}}}$ and $s_{\text{sym}} = s_{I^s_{\text{sym}}}$. We first partition the state vector s into an asymmetric part, s_{asym} (e.g., the torso's position), and a symmetric part, s_{sym} (e.g., the leg's relative joint angles and velocities in Figure 2). The state transformation operator, $L_g : \mathcal{S} \to \mathcal{S}$, reflects the symmetric part of the state as follows: $L_g(s) = (s_{\text{asym}}, -s_{\text{sym}})$. Similarly, we define index sets I^a_{asym} and I^a_{sym} for the action space, and the action space is split up

into an asymmetric part, a_{asym}, and a symmetric part, a_{sym}. The action transformation operator, $K_g : \mathcal{A} \to \mathcal{A}$, reflects the symmetric part of the action (e.g., the leg torques): $K_g(a) = (a_{\text{asym}}, -a_{\text{sym}})$.

The goal is to learn a policy, π, that is equivariant in terms of the aforementioned transformation. A policy π is reflectional-equivariant if it satisfies the following condition for all states $s \in \mathcal{S}$: $\pi(L_g(s)) = K_g(\pi(s))$. This property means that the action for a reflected state is the same as the reflection of the action for the original state. To enforce this, we introduce a Symmetry Regulariser (SymReg) that explicitly penalises deviations from the desired symmetry property. During training, for each observation s, we compute both the standard policy output $\pi(a|s;\phi)$, parameterised by ϕ, and the policy output for the reflected state $\pi(a|L_g(s);\phi)$. The equivariance loss is then defined as:

$$\mathcal{L}_{\text{eq}} = \mathbb{E}_{s\sim\mathcal{D},a\sim\pi_\phi}\left[\|\pi(a|L_g(s);\phi) - K_g(\pi(a|s;\phi))\|_1^2\right].$$

SymReg measures the deviation between the policy's actual response to a reflected state and the expected reflected response. The total training objective combines the standard policy gradient loss, $J_\pi(\phi)$, with SymReg: $\mathcal{L}_{\text{total}} = J_\pi(\phi) + \lambda\mathcal{L}_{\text{eq}}$, where λ is a hyperparameter controlling SymReg.

5 Theoretical Analysis

This section presents theoretical guarantees of PRISM's generalisability, relying on these assumptions:

Assumption 1 (bounded returns)**.** *For all policies π and trajectories τ, $0 \le R(\pi;\tau) \le B$.*

Assumption 2 (Lipschitz-continuous return)**.** *There exists $L_R > 0$ such that for all $\pi, \tilde{\pi} \in \Pi$ and any trajectory τ, $|R(\pi;\tau) - R(\tilde{\pi};\tau)| \le L_R d(\pi,\tilde{\pi})$, where $d(\pi,\tilde{\pi}) := \sup_{s\in\mathcal{S}} \|\pi(s) - \tilde{\pi}(s)\|_1$.*

Assumption 3 (compact spaces)**.** *The state space $\mathcal{S}$ and action space $\mathcal{A}$ are compact metric spaces.*

Assumption 4 (bounded policy)**.** *Policies $\pi \in \Pi$ have bounded inputs and weights.*

Assumption 5 (episode sampling)**.** *The behaviour distribution $\mathcal{D}$ has state marginal lower-bounded by $p_{\min} > 0$ on the state support of interest (finite-support or density lower-bound assumption).*

The Assumptions are reasonably mild. Bartlett et al. (2017) prove that feedforward ReLU are Lipschitz functions; since our policies are implemented as ReLU networks, this ensures bounded sensitivity of the policy outputs to perturbations. Assuming further that the return function is Lipschitz in the policy outputs, it follows that returns are Lipschitz in the policies themselves, as stated in Assumption 2. Assumption 5 ensures that all relevant states are sufficiently sampled under the behaviour policy, which is, in practice, reasonable because policy exploration mechanisms prevent the policy from collapsing onto a subset of states.

5.1 Generalisability of Reflection-Equivariant Subspace

Let Π be the full hypothesis space of policies, and $G = \mathbb{Z}_2$ act on states and actions via L_g, K_g. An orbit-averaging operator $\mathcal{Q}(\pi)(s) = \frac{1}{2}\big(\pi(s) + K_g(\pi(L_g(s)))\big)$ maps any policy to a reflection-equivariant subspace (Qin et al., 2022). The regulariser $\mathcal{L}_{\text{eq}} = \mathbb{E}_s\|\pi(L_g(s)) - K_g(\pi(s))\|_1^2$ encourages convergence to the fixed-point subspace, defined as follows.

Definition 4 (reflection-equivariant subspace)**.** *We define reflection-equivariant subspace as* $\Pi_{\text{eq}} := \{\pi : \pi(L_g(s)) = K_g(\pi(s))\}$.

We prove that $\mathcal{Q}$ is reflectional equivariant, a projection, and that its image coincides with the set of equivariant policies in Lemmas C.3, C.4, and C.5 in Appendix C.2, respectively. Thus, $\mathcal{Q}$ is surjective onto Π_{eq}. To prove that the subspace Π_{eq} is less complex, we show that the projection $\mathcal{Q}$ is non-expansive, which implies its image has a covering number no larger than the original space.

Theorem 5.1. *The space Π_{eq} has a covering number less than or equal to that of Π. Let $\mathcal{N}_{\infty,1}(\mathcal{F}, r)$ be the covering number of a function space $\mathcal{F}$ under the $l_{\infty,1}$-distance. Then, $\mathcal{N}_{\infty,1}(\Pi_{eq}, r) \le \mathcal{N}_{\infty,1}(\Pi, r)$.*

The $l_{\infty,1}$-distance between two policies π_ϕ and π_θ is $d(\pi_\phi,\pi_\theta) = \sup_s \|\pi_\phi(s) - \pi_\theta(s)\|_1$. The distance between their projections, $d(\mathcal{Q}(\pi_\phi), \mathcal{Q}(\pi_\theta))$, no larger using the fact that K_g is a norm-preserving isometry, $\|K_g(a)\|_1 = \|a\|_1$, and that L_g is a bijection, which implies that the supremum

over s equals the supremum over $L_g(s)$. Hence $\mathcal{Q}$ is non-expansive, and a non-expansive surjective map cannot increase the covering number. Following Lemma C.5, $\mathcal{N}(\Pi_{\text{eq}}, r) \leq \mathcal{N}(\Pi, r)$. A detailed proof can be found in Appendix C.3.

The symmetrisation technique is fundamental in empirical process theory that reduces the problem of bounding uniform deviations to analysing Rademacher complexity (Bartlett & Mendelson, 2002).

Corollary 5.2. *For any class $\mathcal{F}$ of functions bounded in $[0, B]$, the expected supremum of empirical deviations satisfies:*

$$\mathbb{E}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{N}\sum_{i=1}^{N}(f(\tau_i)-\mathbb{E}[f])\right|\right] \leq 2\mathbb{E}[\mathfrak{R}_N(\mathcal{F})],$$

where $\mathfrak{R}_N(\mathcal{F}) = \mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}}\frac{1}{N}\sum_{i=1}^{N}\sigma_i f(\tau_i)\right]$ is the Rademacher complexity and σ_i are independent Rademacher random variables taking values ± 1 with equal probability.

This bound transforms the original centred empirical process into a symmetrised version that is often easier to analyse. We now prove a high-probability uniform generalisation bound over the reflection-equivariant subspace. A detailed proof can be found in Appendix C.4. We recognise that PRISM does not necessarily converge to it, which will be discussed in the following subsection.

Theorem 5.3. *With $\mathcal{R}_{\Pi_{\text{eq}}} = \{\tau \mapsto R(\pi;\tau) : \pi \in \Pi_{\text{eq}}\}$, fix any accuracy parameter $r \in (0, B)$ and confidence $\delta \in (0, 1)$. Then with probability at least $1-\delta$,*

$$\sup_{\pi\in\Pi_{\text{eq}}}|J(\pi)-\hat{J}_N(\pi)| \leq C\left(\int_r^B\sqrt{\frac{\log\mathcal{N}_{\infty,1}(\mathcal{R}_{\Pi_{\text{eq}}},\varepsilon)}{N}}d\varepsilon\right)+\frac{8r}{\sqrt{N}}+B\sqrt{\frac{\log(2/\delta)}{2N}},$$

where C is an absolute numeric constant, $J(\pi) = \mathbb{E}[R(\pi;\tau)]$ is the population expected return and $\hat{J}_N(\pi) = \frac{1}{N}\sum_{i=1}^{N}R(\pi;\tau_i)$ is the empirical return on N i.i.d. episodes $\tau_1, \ldots, \tau_N$.

Corollary 5.4. *Under the same assumptions as Theorem 5.3, for any $r \in (0, B)$ and $\delta \in (0, 1)$, the upper bound in Theorem 5.3 for Π_{eq} is at most the same bound obtained by replacing Π_{eq} with Π. By Lemma C.7, the return-class covering numbers can be bounded by those of the policy class with radius scaled by $1/L_R$. Mathematically, following Theorem 5.1, for every $\varepsilon > 0$,*

$$\log\mathcal{N}_{\infty,1}\big(\Pi_{eq},\varepsilon/L_R\big) \leq \log\mathcal{N}_{\infty,1}\big(\Pi,\varepsilon/L_R\big), \tag{2}$$

hence the upper bound in Theorem 5.3 is no larger when evaluated on Π_{eq}.

The equivariance regulariser projects policies onto a smaller fixed-point subspace Π_{eq}, which provably has covering numbers no larger than Π. The return class inherits this reduction via the Lipschitz map, so the Dudley entropy integral for Π_{eq} is bounded by that of Π. As a consequence, the upper bound on the generalisation gap is no larger for Π_{eq} compared to Π.

5.2 Generalisability of PRISM

We now study the generalisability of PRISM, which does not necessarily converge to the reflection-equivariant subspace exactly. Rather, PRISM might converge to an approximately reflection-equivariant class. Using the orbit averaging $\mathcal{Q}$, we quantify this effect below.

Definition 5 (approximately reflection-equivariant class)**.** *Approximately reflection-equivariant class is defined as $\Pi_{approx}(\varepsilon_{eq}) := \{\pi \in \Pi : L_{eq}(\pi) \leq \varepsilon_{eq}\}$.*

Theorem 5.5. *Let $\xi := \frac{1}{2}\sqrt{\varepsilon_{eq}/p_{\min}}$. Then for every policy π,*

$$|J(\pi) - J(Q(\pi))| \leq L_R \cdot d(\pi, Q(\pi)) \leq L_R\xi. \tag{3}$$

Then every $\pi \in \Pi_{approx}(\varepsilon_{eq})$ lies in the sup-ball of radius ξ around Π_{eq}. Consequently, for any target covering radius $r > \xi$, we have:

$$\mathcal{N}_{\infty,1}\big(\Pi_{approx}(\varepsilon_{eq}), r\big) \leq \mathcal{N}_{\infty,1}\big(\Pi_{eq}, r-\xi\big). \tag{4}$$

By Lipschitzness of returns, the expected return of a policy and its projection differ by at most $L_R d(\pi, Q(\pi))$. The mismatch Δ_π controls this distance, and Lemma C.9 bounds its supremum by ξ, giving the first inequality. Geometrically, $\Pi_{approx}(\varepsilon_{eq})$ is contained in a ξ-tube around Π_{eq}. Hence any $(r-\xi)$-cover of Π_{eq} yields an r-cover of $\Pi_{approx}(\varepsilon_{eq})$, proving the covering-number relation (see Appendix C.5 for a detailed proof).

Theorem 5.6. *With $\mathcal{R}_{\Pi_{\text{eq}}} = \{\tau \mapsto R(\pi;\tau) : \pi \in \Pi_{\text{eq}}\}$, fix any accuracy parameter $r \in (0, B)$ and confidence $\delta \in (0,1)$. Then with probability at least $1-\delta$,*

$$\sup_{\pi \in \Pi_{approx}(\varepsilon_{eq})} |J(\pi) - \hat{J}_N(\pi)| \leq C\left(\int_r^B \sqrt{\frac{\log \mathcal{N}_{\infty,1}(\mathcal{R}_{\Pi_{\text{eq}}}, \varepsilon)}{N}} d\varepsilon\right) + \frac{8r}{\sqrt{N}} + B\sqrt{\frac{\log(2/\delta)}{2N}} + 2L_R\xi.$$

For $\pi \in \Pi_{approx}(\varepsilon_{eq})$, decompose the generalisation error relative to its projection $Q(\pi) \in \Pi_{eq}$. The difference in population returns $|J(\pi) - J(Q(\pi))|$ and in empirical returns $|\hat{J}_N(\pi) - \hat{J}_N(Q(\pi))|$ are both bounded by $L_R\xi$ (Theorem 5.5). The middle term $|J(Q(\pi)) - \hat{J}_N(Q(\pi))|$ is exactly the generalisation error for an equivariant policy. Taking the supremum, we obtain the equivariant bound (Theorem 5.3) plus $2L_R\xi$. Appendix C.5 provides a detailed proof.

Corollary 5.7. *Under the same assumptions as Theorem 5.6, for any $r \in (0, B)$ and $\delta \in (0,1)$, the upper bound in Theorem 5.6 for $\Pi_{approx}(\varepsilon_{eq})$ is at most the same bound obtained by replacing $\Pi_{approx}(\varepsilon_{eq})$ with Π. By Lemma C.7, the return-class covering numbers can be bounded by those of the policy class with radius scaled by $1/L_R$. Mathematically, following Theorems 5.1 and 5.5, for any target covering radius $r > \xi$:*

$$\log \mathcal{N}_{\infty,1}\big(\Pi_{approx}(\varepsilon_{eq}), r/L_R\big) \leq \log \mathcal{N}_{\infty,1}\big(\Pi_{eq}, (r-\xi)/L_R\big) \leq \log \mathcal{N}_{\infty,1}\big(\Pi, (r-\xi)/L_R\big), \quad (5)$$

hence the upper bound in Theorem 5.6 is no larger when evaluated on Π_{eq}.

The covering relation incurs a slack of size ξ, leading to bounds of the form $N(\Pi_{approx}(\varepsilon_{eq}), r) \leq N(\Pi_{\text{eq}}, r-\xi) \leq N(\Pi, r-\xi)$. By contrast, in Corollary 5.4, this slack disappears. Thus, the exact case guarantees a strict reduction in complexity, whereas the approximate case trades a ξ-shift in the radius for retaining proximity to the equivariant subspace.

6 Experiments

We conduct extensive experiments to verify PRISM. The code is at `https://anonymous.4open.science/r/reward_shaping-1CCB`.

6.1 Experimental Settings

Environments. Four MuJoCo (Todorov et al., 2012) environments are used: mo-hopper-v5, mo-walker2d-v5, mo-halfcheetah-v5, and mo-swimmer-v5. Table 3 in Appendix D displays the environments and their dimensions, highlighting the diversity in space complexity. As a result, a method must be able to find general solutions applicable to various MORL challenges, instead of being just tailored to one specific type of problem. Furthermore, the division of asymmetric and symmetric state and action spaces to model equivariance is detailed in Appendix D.

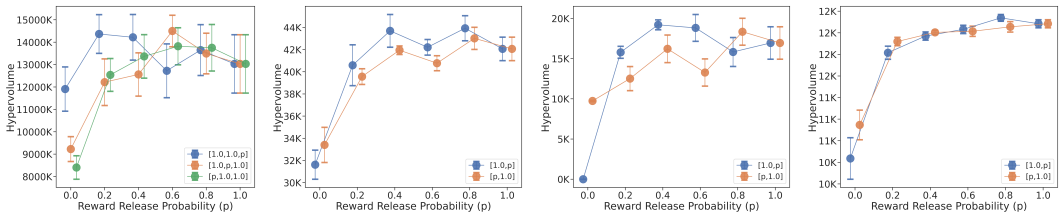
Baselines. PRISM is adaptable to any off-the-shelf MORL algorithm. In this work, CAPQL (Lu et al., 2023) is used as a backbone model, which is a method that trains a single universal network to cover the entire preference space and approximate the Pareto front. We produce (1) **oracle:** instead of artificially setting a reward channel to be sparse, this baseline model can be seen as the gold standard, and (2) **baseline:** instead of utilising the proposed reward shaping model, this method uses CAPQL (Lu et al., 2023) and only observes the sparse rewards.

Evaluation. We use hypervolume (HV), Expected Utility Metric (EUM), and one distributional metric, Variance Objective (VO) (Cai et al., 2023), for evaluation. The used hyperparameters, together with a detailed explanation of evaluation metrics, can be found in Appendix E.

6.2 Empirical Results

Reward Sparsity Sensitivity. Figure 3 illustrates the sensitivity of MORL agents to varying levels of reward sparsity. Across all environments, we observe a sharp decline in HV when one objective is made extremely sparse, with reductions ranging from 20 to 40% relative to the dense setting. For instance, mo-hopper-v5 exhibits a 35% drop in HV under extreme sparsity, while mo-halfcheetah-v5 and mo-walker2d-v5 show declines of 43% and 21%, respectively. These results confirm that sparse objectives worsen policy quality, as agents tend to neglect long-term sparse signals in favour

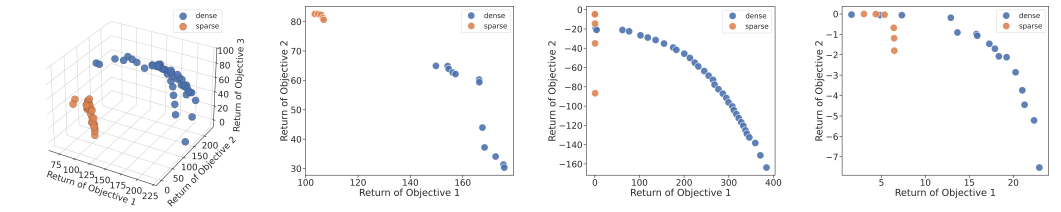
of denser objectives. For the remainder of the paper, we continue with the most difficult setting where extreme sparsity is imposed on the reward objective along the first dimension.



(a) Mo-hopper-v5 (b) Mo-walker2d-v5 (c) Mo-halfcheetah-v5 (d) Mo-swimmer-v5

Figure 3: The obtained hypervolume for various levels of sparsity amongst various dimensions.

Return Distribution of Policy. Figure 4 illustrates the impact of mixed sparsity on MORL across the considered environments. Each subplot compares the approximated Pareto fronts obtained when objective one is dense (blue dots) versus when it is made sparse (orange dots), while keeping all other objectives dense. Extreme sparsity is imposed, where the sparse reward is released at the end of an episode. The results demonstrate a consistent pattern across all environments: when objective one becomes sparse, agents systematically fail to discover high-performing solutions along this dimension, instead concentrating their learning efforts on the remaining dense objectives.



(a) Mo-hopper-v5 (b) Mo-walker2d-v5 (c) Mo-halfcheetah-v5 (d) Mo-swimmer-v5

Figure 4: The approximated Pareto front for dense rewards (blue dots) and sparse rewards (orange dots). Sparsity is imposed on the first reward objective.

The consistent pattern across environments suggests that agents exhibit a systematic bias toward optimising dense reward signals. This overfitting to dense rewards fundamentally distorts the true Pareto front of the problem, leading to a loss of valuable solutions that might represent optimal policies for real-world scenarios where sparse objectives often encode important long-term goals.

Comparison Experiments. Table 1 reports the obtained results for HV, EUM, and VO. The results are averaged over 10 trials, with the standard deviations shown in grey.

PRISM consistently outperforms both the oracle and baseline across environments. For mo-hopper-v5, PRISM improves hypervolume by 21.5% over the oracle (1.58×10^7 compared to 1.30×10^7) and 88% over the baseline. Similar gains are observed for mo-walker2d-v5, where PRISM achieves a 13% HV improvement over oracle and 43% over the baseline. Notably, in mo-halfcheetah-v5, PRISM yields a 32% improvement in HV compared to the oracle (2.25×10^4 against 1.70×10^4) and more than doubles the sparse result. These improvements imply that PRISM not only restores solutions lost under sparsity but also expands the range of trade-offs accessible to the agent. Improvements in EUM follow the same trend, with increases of up to 50% compared to the baseline. The concurrent increase in EUM demonstrates that these additional solutions are not marginal but provide higher expected utility, confirming that PRISM learns policies that are both diverse and practically useful.

On distributional metrics, PRISM delivers more consistent performance than both the oracle and baseline. VO in mo-hopper-v5 increases from 43.36 (baseline) and 59.07 (oracle) to 66.66 under PRISM, and mo-walker2d-v5 shows a 51% gain over the baseline. These gains are crucial because they indicate that PRISM does not simply maximise HV by focusing on extreme solutions, but also produces Pareto fronts that are better balanced, robust, and fair across objectives.

Table 1: Experimental results. We report the average hypervolume (HV), Expected Utility Metric (EUM), and Variance Objective (VO) over 10 trials, with the standard error shown in grey. The largest (best) values are in bold font.

Environment	Metric	Oracle	Baseline	PRISM
Mo-hopper-v5	HV ($\times 10^7$)	1.30 ± 0.13	0.84 ± 0.05	$\mathbf{1.58} \pm 0.05$
	EUM	129.04 ± 7.96	97.64 ± 4.18	$\mathbf{147.43} \pm 2.61$
	VO	59.07 ± 3.45	43.36 ± 1.61	$\mathbf{66.66} \pm 1.40$
Mo-walker2d-v5	HV ($\times 10^4$)	4.21 ± 0.11	3.34 ± 0.16	$\mathbf{4.77} \pm 0.07$
	EUM	107.58 ± 2.86	82.13 ± 4.34	$\mathbf{120.43} \pm 1.64$
	VO	53.22 ± 1.39	39.18 ± 2.49	$\mathbf{59.35} \pm 0.80$
Mo-halfcheetah-v5	HV ($\times 10^4$)	1.70 ± 0.20	0.97 ± 0.00	$\mathbf{2.25} \pm 0.18$
	EUM	81.29 ± 21.85	-1.46 ± 0.27	$\mathbf{89.94} \pm 15.33$
	VO	36.84 ± 10.06	-1.01 ± 0.20	$\mathbf{40.72} \pm 7.02$
Mo-swimmer-v5	HV ($\times 10^4$)	$\mathbf{1.21} \pm 0.00$	1.09 ± 0.02	$\mathbf{1.21} \pm 0.00$
	EUM	9.41 ± 0.12	4.10 ± 0.80	$\mathbf{9.44} \pm 0.14$
	VO	4.22 ± 0.08	1.58 ± 0.40	$\mathbf{4.24} \pm 0.07$

Ablation Study. We analyse the performance of the following ablation models (w/o is the abbreviation for without), which remove several aspects of the reward shaping model or the equivariance loss: (1) **PRISM:** This is the full proposed framework, (2) **w/o residual:** This ablation model removes the two residual blocks from the reward shaping model, (3) **w/o dense rewards:** We remove the dense rewards as input features to the reward model, (4) **w/o ensemble:** We remove the ensemble of reward shaping models, and only employ one, (5) **w/o refinement:** Rather than updating the reward shaping model with expert trajectories, this approach merely trains the reward shaping model using the random trajectories collected at first, and (6) **w/o loss:** We remove the equivariance loss term and merely use the reward shaping model.

The ablation results in Table 10 in Appendix F highlight the contribution of individual components. Removing residual connections reduces HV and EUM across all environments (e.g., mo-hopper-v5 EUM falls from 147.43 to 128.40), showing their importance for scaled opportunity value. Excluding dense reward features or ensembles also lowers performance, but only moderately, suggesting that state–action features already contain substantial signal. Interestingly, removing iterative refinement barely reduces performance; in some cases, such as mo-halfcheetah-v5, HV, and EUM remain comparable or even slightly higher than the full model. This implies that shaping rewards from a broad set of random trajectories is already highly effective. Removing the symmetry loss reduces performance across environments, indicating that the loss term successfully reduces the search space. Similar patterns are observed for VO. The ablation results imply that PRISM's architecture provides multiple overlapping mechanisms for stability, but the symmetry loss and residual structure are the main drivers of consistent performance.

7 Conclusion

This work proposes Parallel Reward Integration with reflectional Symmetry for Multi-objective reinforcement learning (PRISM), a framework designed to tackle sample inefficiency in heterogeneous multi-objective reinforcement learning, particularly in environments with sparse rewards. Our approach is centred around two key contributions: (1) ReSymNet, a theory-inspired reward model that leverages residual blocks to align reward channels by learning a refined 'scaled opportunity value', and (2) SymReg, a novel regulariser that enforces reflectional symmetry as an inductive bias in the policy's action space. We prove that PRISM restricts policy search to a reflection-equivariant subspace, a projection of the original policy space with provably reduced hypothesis complexity; in this way, the generalisability is rigorously improved. Extensive experiments on MuJoCo benchmarks show that PRISM consistently outperforms even a strong oracle with full reward access in terms of a wide range of metrics, including HV, EUM, and VO.

Ethics Statement

We declare no potential conflict of interest nor sponsorship. We are not aware of any issues related to legal compliance, research integrity, or other ethical considerations.

Reproducibility Statement

We have taken several steps to ensure reproducibility. All assumptions underlying our theoretical results are explicitly stated in Section 5. For the empirical results, we use only publicly available environments, described in Section 6, with training details, hyperparameters, and evaluation metrics reported in Appendix E. To further support reproducibility, we released an anonymous code repository containing implementation details at `https://anonymous.4open.science/r/reward_shaping-1CCB`, which will be publicly released.

References

Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500, 2021.

Arthur Aubret, Laetitia Matignon, and Salima Hassas. A survey on intrinsic motivation in reinforcement learning. *arXiv preprint arXiv:1908.06976*, 2019.

Peter L Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In *31st International Conference on Neural Information Processing Systems (NIPS 2017)*, pp. 6241–6250. Curran Associates, 2017.

Toygun Basaklar, Suat Gumussoy, and Ümit Y. Ogras. PD-MORL: Preference-driven multi-objective reinforcement learning algorithm. In *Eleventh International Conference on Learning Representations (ICLR 2023)*. OpenReview.net, 2023.

Yuri Burda, Harrison Edwards, Amos J. Storkey, and Oleg Klimov. Exploration by random network distillation. In *7th International Conference on Learning Representations (ICLR 2019)*. OpenReview.net, 2019.

Xin-Qiang Cai, Pushi Zhang, Li Zhao, Jiang Bian, Masashi Sugiyama, and Ashley Llorens. Distributional Pareto-optimal multi-objective reinforcement learning. In *37th International Conference on Neural Information Processing Systems (NIPS 2023)*, volume 36, pp. 15593–15613. Curran Associates, 2023.

Rati Devidze, Parameswaran Kamalaruban, and Adish Singla. Exploration-guided reward shaping for reinforcement learning under sparse rewards. In *36th International Conference on Neural Information Processing Systems (NIPS 2022)*. Curran Associates, 2022.

Richard M Dudley. The sizes of compact subsets of Hilbert space and continuity of Gaussian processes. *Journal of Functional Analysis*, 1(3):290–330, 1967.

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 *37th International Conference on Neural Information Processing Systems (NIPS 2023)*. Curran Associates, 2023.

Carlos M. Fonseca, Luís Paquete, and Manuel López-Ibáñez. An improved dimension-sweep algorithm for the hypervolume indicator. In *IEEE International Conference on Evolutionary Computation (CEC 2006)*, pp. 1157–1163. IEEE, 2006.

Ben Hambly, Renyuan Xu, and Huining Yang. Recent advances in reinforcement learning in finance. *Mathematical Finance*, 33(3):437–503, 2023.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV 2015)*, pp. 1026–1034. IEEE Computer Society, 2015.

Ian Holmes and Min Chi. Attention-based reward shaping for sparse and delayed rewards. *arXiv preprint arXiv:2505.10802*, 2025.

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE transactions on intelligent transportation systems*, 23(6):4909–4926, 2021.

Adam Daniel Laud. *Theory and application of reward shaping in reinforcement learning*. University of Illinois at Urbana-Champaign, 2004.

Thomas Lautenbacher, Ali Rajaei, Davide Barbieri, Jan Viebahn, and Jochen L Cremer. Multi-objective reinforcement learning for power grid topology control. *arXiv preprint arXiv:2502.00040*, 2025.

Yijiong Lin, Jiancong Huang, Matthieu Zimmer, Yisheng Guan, Juan Rojas, and Paul Weng. Invariant transform experience replay: Data augmentation for deep reinforcement learning. *IEEE Robotics and Automation Letters*, 5(4):6615–6622, 2020.

Chunming Liu, Xin Xu, and Dewen Hu. Multiobjective reinforcement learning: A comprehensive overview. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(3):385–398, 2014.

Erlong Liu, Yu-Chang Wu, Xiaobin Huang, Chengrui Gao, Ren-Jian Wang, Ke Xue, and Chao Qian. Pareto set learning for multi-objective reinforcement learning. In *AAAI Conference on Artificial Intelligence*, pp. 18789–18797. AAAI Press, 2025.

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

Colin McDiarmid et al. On the method of bounded differences. *Surveys in Combinatorics*, 141(1): 148–188, 1989.

Farzan Memarian, Wonjoon Goo, Rudolf Lioutikov, Scott Niekum, and Ufuk Topcu. Self-supervised online reward shaping in sparse-reward environments. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2021)*, pp. 2369–2375. IEEE, 2021.

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 2013)*, pp. 191–199. IEEE, 2013.

Arnab Kumar Mondal, Vineet Jain, Kaleem Siddiqi, and Siamak Ravanbakhsh. Eqr: Equivariant representations for data-efficient reinforcement learning. In *International Conference on Machine Learning (ICML 2022)*, volume 162 of *PMLR*, pp. 15908–15926. PMLR, 2022.

Ni Mu, Yao Luan, and Qing-Shan Jia. Preference-based multi-objective reinforcement learning. *IEEE Transactions on Automation Science and Engineering*, 2025.

Andrew Y. Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *Seventeenth International Conference on Machine Learning (ICML 2000)*, pp. 663–670. Morgan Kaufmann, 2000.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Sixteenth International Conference on Machine Learning (ICML 1999)*, pp. 278–287. Morgan Kaufmann, 1999.

Jung Yeon Park, Sujay Bhatt, Sihan Zeng, Lawson L. S. Wong, Alec Koppel, Sumitra Ganesh, and Robin Walters. Approximate equivariance in reinforcement learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS 2025)*, volume 258 of *PMLR*, pp. 4177–4185. PMLR, 2025.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *34th International Conference on Machine Learning (ICML 2017)*, volume 70 of *PMLR*, pp. 2778–2787. PMLR, 2017.

Tian Qin, Fengxiang He, Dingfeng Shi, Wenbing Huang, and Dacheng Tao. Benefits of permutation-equivariance in auction mechanisms. *36th International Conference on Neural Information Processing Systems (NIPS 2022)*, 35:18131–18142, 2022.

Mathieu Reymond and Ann Nowé. Pareto-DQN: Approximating the Pareto front in complex multi-objective decision problems. In *Adaptive and Learning Agents Workshop (ALA 2019)*, 2019.

Diederik Marijn Roijers, Shimon Whiteson, and Frans A Oliehoek. Computing convex coverage sets for faster multi-objective coordination. *Journal of Artificial Intelligence Research*, 52:399–443, 2015.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

Chen Tang, Ben Abbatematteo, Jiaheng Hu, Rohan Chandra, Roberto Martín-Martín, and Peter Stone. Deep reinforcement learning for robotics: A survey of real-world successes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 28694–28698, 2025.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Kristof Van Moffaert and Ann Nowé. Multi-objective reinforcement learning using sets of Pareto dominating policies. *The Journal of Machine Learning Research*, 15(1):3483–3512, 2014.

Dian Wang, Robin Walters, and Robert Platt. SO(2)-equivariant reinforcement learning. In *Tenth International Conference on Learning Representations (ICLR 2022)*. OpenReview.net, 2022.

Huan Wang, Stephan Zheng, Caiming Xiong, and Richard Socher. On the generalization gap in reparameterizable reinforcement learning. In *36th International Conference on Machine Learning (ICML 2019)*, volume 97, pp. 6648–6658. PMLR, 2019.

Wei Wei, Haibin Li, Shiyuan Zhou, Baifeng Li, and Xue Liu. Attention with system entropy for optimizing credit assignment in cooperative multi-agent reinforcement learning. *IEEE Transactions on Automation Science and Engineering*, 22:14775–14787, 2025.

Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. In *33rd International Conference on Neural Information Processing Systems (NIPS 2019)*, pp. 14610–14621. Curran Associates, 2019.

Ding-Xuan Zhou. The covering number in learning theory. *Journal of Complexity*, 18(3):739–767, 2002.

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

A Notation

In this appendix, we provide an overview of the notation used in Table 2.

Table 2: Notation.

Symbol	Meaning
$\mathcal{S}$	State space
$\mathcal{A}$	Action space
$P(s'\|s,a)$	Transition probability
$\boldsymbol{r}(s,a) \in \mathbb{R}^L$	Vector-valued reward with L objectives
$\gamma \in [0,1)$	Discount factor
$\pi : \mathcal{S} \to \mathcal{A}$	Policy mapping
$\boldsymbol{J}^\pi = \mathbb{E}_\pi\big[\sum_{t=0}^{\infty} \gamma^t \boldsymbol{r}_t\big]$	Expected cumulative vector return
$\mathcal{D}$	Behaviour distribution to sample episodes from
$D_C = \{d_1, \ldots, d_D\}$	Dense reward channels
$r_t^{d_i}$	Reward from dense channel d_i at timestep t
r_t^{sp}	Sparse reward at timestep t
$\tau = \{(s_1, a_1), \ldots, (s_T, a_T)\}$	Trajectory
$R^{sp}(\tau)$	Cumulative sparse reward in episode τ
p_{rel}	Probability of releasing sparse reward
$h_t = [s_t, a_t, \boldsymbol{r}_t^{dense}]$	Input feature vector for ReSymNet
$\mathcal{R}_{\text{pred}}$	ReSymNet
r_t^{sh}	Shaped reward at timestep t
L_g, K_g	Reflection operators on states and actions
$\Delta_\pi(s) = \pi(L_g(s)) - K_g(\pi(s))$	Equivariance mismatch
$\mathcal{L}_{eq}(\pi)$	Equivariance regularisation loss
Π	Hypothesis space of policies
$\Pi_{eq} = \{\pi : \pi(L_g(s)) = K_g(\pi(s))\}$	Reflection-equivariant subspace
$\Pi_{\text{approx}}(\varepsilon_{eq})$	Approximate equivariant policies with tolerance ε_{eq}

B Additional Details and Theoretical Motivations of ReSymNet

We give additional details of ReSymNet as well as the theoretical motivation behind its architecture in this appendix.

B.1 Theoretical Motivation

The use of residual connections in $\mathcal{R}_{\text{pred}}$ is motivated by the theory of scaled opportunity value (Laud, 2004).

Definition 6 (Opportunity value)**.** *Let M be an MDP with native reward function R. The opportunity value of a transition (s, a, s') is defined as the difference in the optimal value of successor and current states:* $\mathrm{OPV}(s, a, s') = \gamma V^M(s') - V^M(s)$*, where V^M is the optimal state-value function under MDP M.*

Definition 7 (Scaled opportunity value)**.** *For a scale parameter $k > 0$, the scaled opportunity value shaping function augments the native reward with a scaled opportunity correction:* $\mathrm{OPV}_k(s, a, s') = F_k(s, a, s') = k(\gamma V^M(s') - V^M(s)) + (k-1)R(s, a)$.

Lemma B.1. *Let M be an MDP with reward function R and optimal policy $\pi^\star$. With k sufficiently large, the MDP with shaped reward F_k satisfies: (1) policy invariance, $\pi^\star$ remains optimal under F_k; (2) horizon reduction, the effective reward horizon is reduced to* 1*; and (3) improved local approximation, the additive term increases the separability of local utilities, reducing approximation error in value estimation.*

Residual blocks mirror the additive structure of scaled opportunity value: each block refines its input prediction via: $\mathcal{R}_{\text{pred}}^{(i)}(\boldsymbol{h}_t; \psi) = \mathcal{R}_{\text{pred}}^{(i-1)}(\boldsymbol{h}_t; \psi) + \Delta_i(\boldsymbol{h}_t; \psi)$, where Δ_i is a learned correction. A single block can be viewed as approximating a scaled opportunity-value transformation of its input, while stacking multiple blocks, $\mathcal{R}_{\text{pred}}^{(n)}(\boldsymbol{h}_t; \psi) = f_n \circ \cdots \circ f_1(R(\boldsymbol{h}_t; \psi))$, implements iterative refinement: each stage reduces the residual error left by the previous one. This residual formulation both stabilises training and aligns with the principle of scaled opportunity value, gradually shaping

per-step predictions into horizon-1 signals that remain consistent with the sparse episodic return $R^{sp}(\tau)$.

B.2 Algorithm Chart

Algorithm 1: ReSymNet with any MORL algorithm

Input: Release probability p_{rel}, number of initial episodes N, number of expert episodes E, dense channels $\mathcal{DC}$, any off-the-shelf MORL algorithm, number of timesteps per cycle M, number of ensembles K, number of iterative refinements IR, validation split, patience

Output: Trained reward ensemble $\mathcal{E} = \{R_{\text{pred},\psi_1}, \ldots, R_{\text{pred},\psi_K}\}$, trained MORL policy

```
/* Collecting random experiences */
```

for $i \leftarrow 1$ **to** N **do**

- Execute a random policy to collect trajectory $\tau = \{(s_0, a_0), \ldots, (s_T, a_T)\}$ until episode ends
- Set $l = 0$
- **foreach** $t \in T$ **do**
  - With probability p_{rel}, release cumulative sparse reward $R_t^{sp} = \sum_{s=l}^{t} r_s^{sp}$ at timestep t
  - Set $l = t$ if sparse reward is released
- Segment τ into sub-trajectories $\{\tau_j\}$ based on released rewards
- **foreach** *sub-trajectory* τ_j **do**
  - **foreach** $(s_t, a_t) \in \tau_j$ **do**
    - Compute features: $\boldsymbol{h}_t = [s_t, a_t, \boldsymbol{r}_{\text{dense},t}]$
  - Add datapoint $\big(\{\boldsymbol{h}_t\}_{t\in\tau_j}, R^{sp}(\tau_j)\big)$ to dataset $\mathcal{D}$

```
/* Ensemble training */
```

for $k \leftarrow 1$ **to** K **do**

- Split $\mathcal{D}$ into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{val}$ using the validation split
- Train reward model R_{pred,ψ_k} following Equation 1 using early stopping on the validation loss:

$$\mathcal{L}(\psi_k) = \sum_{\tau\in\mathcal{D}_{\text{train}}} \left(\sum_{t\in\tau} \mathcal{R}_{\text{pred}}(\boldsymbol{h}_t; \psi_k) - R^{sp}(\tau) \right)^2$$

```
/* RL training with iterative refinement */
```

$timestep = 1$

for $cycle \leftarrow 1$ **to** IR **do**

- **for** $t \leftarrow timestep$ **to** $M + timestep$ **do**
  - Observe s_t and a_t following the current policy and compute features $\boldsymbol{h}_t$
  - $r_t^{(k)} \leftarrow \mathcal{R}_{\text{pred}}(\boldsymbol{h}_t; \psi_k)$ for $k = 1, \ldots, K$
  - $r_t^{\text{sh}} \leftarrow \frac{1}{K} \sum_{k=1}^{K} r_t^{(k)}$
  - Use r_t^{sh} with the dense rewards as the reward at timestep t and update RL algorithm
- `/* Iterative refinement */`
- Collect E expert trajectories to obtain $\mathcal{D}_{\text{new}}$ using the new policy
- **foreach** $R_{pred,\psi_k} \in \mathcal{E}$ **do**
  - Update R_{pred,ψ_k} using new data $\mathcal{D}_{\text{new}}$
- $timestep = t$

C Proofs

This appendix collects all proofs omitted from the main text.

C.1 Lemmas

This section introduces the general lemmas used to obtain an upper bound on the generalisation gap.

Dudley Entropy Integral. The Rademacher complexity can be bounded through the metric entropy of the function class using Dudley's entropy integral (Dudley, 1967; Bartlett & Mendelson, 2002).

Lemma C.1 (Dudley Entropy Integral)**.** *For any coarse-scale parameter $r \in (0, B)$, the empirical Rademacher complexity satisfies:*

$$\mathfrak{R}_N(\mathcal{F}) \leq C \left(\int_r^B \sqrt{\frac{\log \mathcal{N}_{\infty,1}(\mathcal{F}, r)}{N}} d\varepsilon \right) + \frac{4r}{\sqrt{N}},$$

where $C > 0$ is an absolute constant, and $\mathcal{N}_{\infty,1}(\mathcal{F}, r)$ is the covering number of $\mathcal{F}$ in ℓ_∞ at scale r with respect to N samples

This inequality connects the probabilistic complexity (Rademacher complexity) to the geometric complexity of the function class and covering numbers.

McDiarmid's Concentration Inequality. To convert expectation bounds into high-probability statements, we employ McDiarmid's bounded difference inequality (McDiarmid et al., 1989).

Lemma C.2 (McDiarmid's Concentration Inequality)**.** *If each trajectory's replacement can change any empirical average by at most B/N, then for any $t > 0$:*

$$\Pr\left(\left| \sup_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} (f(\tau_i) - \mathbb{E}[f]) - \mathbb{E}\left[\sup_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} (f(\tau_i) - \mathbb{E}[f]) \right] \right| \geq t \right) \leq 2 \exp\left(-\frac{2Nt^2}{B^2} \right).$$

This concentration result allows us to bound the deviation between the random supremum and its expectation, completing the pipeline from covering numbers to high-probability uniform generalisation gaps.

C.2 Projection to Reflection-Equivariant Subspace

Let the full hypothesis space of policies be $\Pi = \{\pi_\phi : \phi \in \Phi\}$, where ϕ represents the neural network parameters. The reflection group $G = \mathbb{Z}_2 = \{e, g\}$ acts on the state and action spaces via operators L_g and K_g, respectively.

We can map any policy to its equivariant counterpart using an orbit averaging operator $\mathcal{Q} : \Pi \to \Pi$, defined as:

$$\begin{aligned} \mathcal{Q}(\pi_\phi)(s) &= \frac{1}{|G|} \sum_{h \in G} \rho(h) \pi_\phi(h^{-1} \cdot s) \\ &= \frac{1}{|G|} \sum_{h \in G} K_h\big(\pi_\phi(L_h(s))\big) \\ &= \tfrac{1}{2} \left(\pi_\phi(s) + K_g(\pi_\phi(L_g(s)))\right). \end{aligned} \tag{6}$$

Here, $\rho(h)$ is the abstract representation in the action space, and $h^{-1} \cdot s$ is the abstract action in the state space. In the second line we replace $\rho(h)$ with the action transformation K_h, and $h^{-1} \cdot s$ with the state transformation $L_h(s)$. For the reflection group $G = \mathbb{Z}_2 = \{e, g\}$, since $g = g^{-1}$ we may drop the inverse without ambiguity. This operator averages a policy's output with its reflected-transformed equivalent. The regulariser, $\mathcal{L}_{\text{eq}} = \mathbb{E}_s[\|\pi_\phi(L_g(s)) - K_g(\pi_\phi(s))\|_1^2]$, encourages policies to become fixed points of this operator, thereby learning policies within the subspace of equivariant functions, denoted Π_{eq}.

The operator $\mathcal{Q}$ and the subspace Π_{eq} have several crucial properties, which we state in the following lemmas.

Lemma C.3. *For any $\pi \in \Pi$, the function $\mathcal{Q}(\pi)$ is reflectional equivariant:*

$$\mathcal{Q}(\pi)(L_g(s)) = K_g(\mathcal{Q}(\pi)(s)), \quad \forall s \in \mathcal{S}.$$

Proof. By direct calculation:

$$\begin{aligned} \mathcal{Q}(\pi)(L_g(s)) &= \tfrac{1}{2}\big(\pi(L_g(s)) + K_g(\pi(L_g(L_g(s))))\big) \\ &= \tfrac{1}{2}\big(\pi(L_g(s)) + K_g(\pi(s))\big), \\ K_g\mathcal{Q}(\pi)(s) &= \tfrac{1}{2}\big(K_g(\pi(s)) + K_gK_g(\pi(L_g(s)))\big) \\ &= \tfrac{1}{2}\big(K_g(\pi(s)) + \pi(L_g(s))\big), \end{aligned}$$

since K_g and L_g are involutions. Thus, the two expressions coincide. Therefore $\mathcal{Q}(\pi)$ is equivariant. □

Lemma C.4. *The operator $\mathcal{Q}$ is a projection, meaning it is idempotent: $\mathcal{Q}(\mathcal{Q}(\pi)) = \mathcal{Q}(\pi)$ for any $\pi \in \Pi$.*

Proof. We apply the operator to its own output:

$$\mathcal{Q}(\mathcal{Q}(\pi))(s) = \frac{1}{2}\left(\mathcal{Q}(\pi)(s) + K_g(\mathcal{Q}(\pi)(L_g(s)))\right).$$

First, evaluating the second term, $\mathcal{Q}(\pi)(L_g(s))$:

$$\begin{aligned}\mathcal{Q}(\pi)(L_g(s)) &= \frac{1}{2}\left(\pi(L_g(s)) + K_g(\pi(L_g(L_g(s))))\right) \\ &= \frac{1}{2}\left(\pi(L_g(s)) + K_g(\pi(s))\right).\end{aligned}$$

Substituting this back:

$$\begin{aligned}\mathcal{Q}(\mathcal{Q}(\pi))(s) &= \frac{1}{2}\left(\mathcal{Q}(\pi)(s) + K_g\left[\tfrac{1}{2}(\pi(L_g(s)) + K_g(\pi(s)))\right]\right) \\ &= \frac{1}{2}\mathcal{Q}(\pi)(s) + \frac{1}{4}\left(K_g(\pi(L_g(s))) + K_g(K_g(\pi(s)))\right) \\ &= \frac{1}{2}\mathcal{Q}(\pi)(s) + \frac{1}{4}\left(K_g(\pi(L_g(s))) + \pi(s)\right) \\ &= \frac{1}{2}\mathcal{Q}(\pi)(s) + \frac{1}{2}\left(\tfrac{1}{2}(\pi(s) + K_g(\pi(L_g(s))))\right) \\ &= \frac{1}{2}\mathcal{Q}(\pi)(s) + \frac{1}{2}\mathcal{Q}(\pi)(s) \\ &= \mathcal{Q}(\pi)(s).\end{aligned}$$

Thus $\mathcal{Q}$ is idempotent. □

Lemma C.5. *The image of the operator $\mathcal{Q}$ coincides with the set of equivariant policies:* $\mathrm{Im}(\mathcal{Q}) = \{\mathcal{Q}(\pi) : \pi \in \Pi\} = \Pi_{eq}$.

Proof. We establish set equality by showing inclusion in both directions.

First inclusion ($\mathrm{Im}(\mathcal{Q}) \subseteq \Pi_{\text{eq}}$): By Lemma C.3, for any $\pi \in \Pi$, the output $\mathcal{Q}(\pi)$ is equivariant. Therefore, every element in the image of $\mathcal{Q}$ belongs to Π_{eq}.

Second inclusion ($\Pi_{\text{eq}} \subseteq \mathrm{Im}(\mathcal{Q})$): Let π_{eq} be any equivariant policy, so $\pi_{\text{eq}} \in \Pi_{\text{eq}}$. We need to show that π_{eq} can be expressed as $\mathcal{Q}(\pi)$ for some $\pi \in \Pi$.

Since π_{eq} is equivariant, it satisfies $\pi_{\text{eq}}(L_g(s)) = K_g(\pi_{\text{eq}}(s))$ for all s. Therefore:

$$\begin{aligned}\mathcal{Q}(\pi_{\text{eq}})(s) &= \frac{1}{2}\left(\pi_{\text{eq}}(s) + K_g(\pi_{\text{eq}}(L_g(s)))\right) \\ &= \frac{1}{2}\left(\pi_{\text{eq}}(s) + K_g(K_g(\pi_{\text{eq}}(s)))\right) \quad \text{(by equivariance)} \\ &= \frac{1}{2}\left(\pi_{\text{eq}}(s) + \pi_{\text{eq}}(s)\right) \quad \text{(since } K_g \text{ is an involution)} \\ &= \pi_{\text{eq}}(s).\end{aligned}$$

Therefore, $\pi_{\text{eq}} = \mathcal{Q}(\pi_{\text{eq}}) \in \mathrm{Im}(\mathcal{Q})$. This shows that equivariant policies are fixed points of $\mathcal{Q}$, which is consistent with Lemma C.4. Since every equivariant policy is its own image under $\mathcal{Q}$, we have $\Pi_{\text{eq}} \subseteq \mathrm{Im}(\mathcal{Q})$. Combining both inclusions yields $\mathrm{Im}(\mathcal{Q}) = \Pi_{\text{eq}}$. Therefore $\mathcal{Q}$ is surjective onto Π_{eq}. □

C.3 Reduced Hypothesis Complexity of Reflection-Equivariant Subspace

To prove that the subspace Π_{eq} is less complex, we show that the projection $\mathcal{Q}$ is non-expansive, which implies its image has a covering number no larger than the original space.

Theorem C.6. *The space Π_{eq} has a covering number less than or equal to that of Π. Let $\mathcal{N}_{\infty,1}(\mathcal{F}, r)$ be the covering number of a function space $\mathcal{F}$ under the $l_{\infty,1}$-distance. Then, $\mathcal{N}_{\infty,1}(\Pi_{eq}, r) \leq \mathcal{N}_{\infty,1}(\Pi, r)$.*

Proof. We show that $\mathcal{Q}$ is non-expansive. The $l_{\infty,1}$-distance between two policies π_ϕ and π_θ is

$$d(\pi_\phi, \pi_\theta) = \sup_s \|\pi_\phi(s) - \pi_\theta(s)\|_1.$$

The distance between their projections is:

$$\begin{aligned}
d(\mathcal{Q}(\pi_\phi), \mathcal{Q}(\pi_\theta)) &= \sup_s \left\| \tfrac{1}{2}\big(\pi_\phi(s) + K_g(\pi_\phi(L_g(s)))\big) - \tfrac{1}{2}\big(\pi_\theta(s) + K_g(\pi_\theta(L_g(s)))\big) \right\|_1 \\
&= \tfrac{1}{2} \sup_s \left\| (\pi_\phi(s) - \pi_\theta(s)) + K_g(\pi_\phi(L_g(s)) - \pi_\theta(L_g(s))) \right\|_1 . \\
&\leq \tfrac{1}{2} \sup_s \Big(\|\pi_\phi(s) - \pi_\theta(s)\|_1 + \|K_g(\pi_\phi(L_g(s)) - \pi_\theta(L_g(s)))\|_1 \Big). \\
&\leq \tfrac{1}{2} \Big(\sup_s \|\pi_\phi(s) - \pi_\theta(s)\|_1 + \sup_s \|\pi_\phi(L_g(s)) - \pi_\theta(L_g(s))\|_1 \Big). \\
&= \tfrac{1}{2}\big(d(\pi_\phi, \pi_\theta) + d(\pi_\phi, \pi_\theta)\big) = d(\pi_\phi, \pi_\theta),
\end{aligned}$$

where we use the triangle inequality, the fact that K_g is a norm-preserving isometry, $\|K_g(a)\|_1 = \|a\|_1$, and that L_g is a bijection, which implies that the supremum over s equals the supremum over $L_g(s)$. Hence $\mathcal{Q}$ is non-expansive, and a non-expansive surjective map cannot increase the covering number. Following Lemma C.5, $\mathcal{N}(\Pi_{\text{eq}}, r) \leq \mathcal{N}(\Pi, r)$. □

The following lemma links coverings of the policy class (with metric d) to coverings of the induced return class (supremum over trajectories). This is the deterministic Lipschitz step that makes the entropy of returns comparable to the entropy of the policy class.

Lemma C.7. *For any policy set $\mathcal{P} \subseteq \Pi$ and any $\varepsilon > 0$,*

$$\mathcal{N}_{\infty,1}\big(\{\tau \mapsto R(\pi;\tau) : \pi \in \mathcal{P}\}, \varepsilon\big) \leq \mathcal{N}_{\infty,1}\big(\mathcal{P}, \varepsilon/L_R\big),$$

where the left covering number is with respect to the sup-norm over trajectories and the right is with respect to $d(\cdot,\cdot)$.

Proof. Let $\{\pi_1, \ldots, \pi_M\}$ be an ε/L_R-cover of $\mathcal{P}$ under $d(\cdot,\cdot)$. For any $\pi \in \mathcal{P}$ choose j with $d(\pi, \pi_j) \leq \varepsilon/L_R$. Then for every trajectory τ,

$$|R(\pi;\tau) - R(\pi_j;\tau)| \leq L_R d(\pi, \pi_j) \leq \varepsilon,$$

so the set $\{\tau \mapsto R(\pi_j;\tau)\}_{j=1}^M$ is an ε-cover of the return-class. Thus, the covering inequality holds. □

C.4 Generalisation of Reflection-Equivariant Subspace

We now prove a high-probability uniform bound over the equivariant class.

Theorem C.8. *With $\mathcal{R}_{\Pi_{\text{eq}}} = \{\tau \mapsto R(\pi;\tau) : \pi \in \Pi_{\text{eq}}\}$, fix any accuracy parameter $r \in (0, B)$ and confidence $\delta \in (0,1)$. Then with probability at least $1-\delta$,*

$$\sup_{\pi \in \Pi_{\text{eq}}} |J(\pi) - \hat{J}_N(\pi)| \leq C \left(\int_r^B \sqrt{\frac{\log \mathcal{N}_{\infty,1}(\mathcal{R}_{\Pi_{\text{eq}}}, \varepsilon)}{N}} d\varepsilon \right) + \frac{8r}{\sqrt{N}} + B\sqrt{\frac{\log(2/\delta)}{2N}},$$

where C is an absolute numeric constant, $J(\pi) = \mathbb{E}[R(\pi;\tau)]$ is the population expected return and $\hat{J}_N(\pi) = \frac{1}{N}\sum_{i=1}^N R(\pi;\tau_i)$ is the empirical return on N i.i.d. episodes $\tau_1, \ldots, \tau_N$.

Proof. Let $\mathcal{F} = \mathcal{R}_{\Pi_{\text{eq}}}$. Following Corollary 5.2, we have:

$$\mathbb{E}\Big[\sup_{f\in\mathcal{R}_{\Pi_{\text{eq}}}}\Big|\tfrac{1}{N}\sum_{i=1}^{N}(f(\tau_i)-\mathbb{E}[f])\Big|\Big] \le 2\mathbb{E}\big[\mathfrak{R}_N(\mathcal{R}_{\Pi_{\text{eq}}})\big].$$

Applying Lemma C.1, for any $r > 0$:

$$\mathbb{E}\Big[\sup_{f\in\mathcal{R}_{\Pi_{\text{eq}}}}\Big|\tfrac{1}{N}\sum_{i=1}^{N}(f(\tau_i)-\mathbb{E}[f])\Big|\Big] \le C\left(\int_r^B \sqrt{\frac{\log \mathcal{N}_{\infty,1}(\mathcal{R}_{\Pi_{\text{eq}}},\varepsilon)}{N}}d\varepsilon\right) + \frac{8r}{\sqrt{N}}. \tag{7}$$

Now apply Lemma C.2 to convert the expectation bound into a high-probability statement, with probability at least $1-\delta$:

$$\sup_{f\in\mathcal{R}_{\Pi_{\text{eq}}}}\Big|\tfrac{1}{N}\sum_{i=1}^{N}(f(\tau_i)-\mathbb{E}[f])\Big| \le \mathbb{E}\Big[\sup_{f\in\mathcal{R}_{\Pi_{\text{eq}}}}\Big|\tfrac{1}{N}\sum_{i=1}^{N}(f(\tau_i)-\mathbb{E}[f])\Big|\Big] + B\sqrt{\frac{\log(2/\delta)}{2N}}. \tag{8}$$

Combining Equations 7 and 8 yields the claimed inequality. □

C.5 Generalisatisability of PRISM

Lemma C.9. *If a policy π satisfies $L_{eq}(\pi) \le \varepsilon_{eq}$, then*

$$\sup_s \|\Delta_\pi(s)\|_1 \le \sqrt{\frac{\varepsilon_{eq}}{p_{\min}}}.$$

Consequently, the sup–ℓ_1 distance between π and its orbit projection $Q(\pi)$ satisfies

$$d(\pi, Q(\pi)) = \sup_s \|\pi(s) - Q(\pi)(s)\|_1 \le \sqrt{\frac{\varepsilon_{eq}}{p_{\min}}}.$$

Proof. Assume the state space has density $\frac{d\mu}{ds}(s) \ge p_{\min}$ on the common support. Let s^* be such that $\|\Delta_\pi(s^*)\|_1 = \sup_s \|\Delta_\pi(s)\|_1$. The expectation is:

$$\varepsilon_{eq} = \mathbb{E}_\mu\big[\|\Delta_\pi(s)\|_1^2\big] = \int \|\Delta_\pi(s)\|_1^2 d\mu(s).$$

For any neighbourhood $B_\delta(s^*)$ of s^*:

$$\varepsilon_{eq} \ge \int_{B_\delta(s^*)} \|\Delta_\pi(s)\|_1^2 d\mu(s).$$

By continuity of $\|\Delta_\pi(\cdot)\|_1$ and the density lower bound:

$$\int_{B_\delta(s^*)} \|\Delta_\pi(s)\|_1^2 d\mu(s) \ge \left(\|\Delta_\pi(s^*)\|_1 - \epsilon\right)^2 \int_{B_\delta(s^*)} d\mu(s) \ge \left(\|\Delta_\pi(s^*)\|_1 - \epsilon\right)^2 p_{\min}\cdot\text{vol}(B_\delta(s^*)),$$

for sufficiently small δ and any $\epsilon > 0$. Taking $\delta \to 0$ and $\epsilon \to 0$:

$$\varepsilon_{eq} \ge p_{\min}\left(\sup_s \|\Delta_\pi(s)\|_1\right)^2.$$

Rearranging gives $\sup_s \|\Delta_\pi(s)\|_1 \le \sqrt{\frac{\varepsilon_{eq}}{p_{\min}}}$. □

We can now translate this approximation to a bound on returns and to a covering-number statement.

Theorem C.10. *Let $\xi := \frac{1}{2}\sqrt{\varepsilon_{eq}/p_{\min}}$. Then for every policy π,*

$$|J(\pi) - J(Q(\pi))| \le L_R \cdot d(\pi, Q(\pi)) \le L_R \xi.$$

Define the approximately reflection-equivariant class $\Pi_{approx}(\varepsilon_{eq}) := \{\pi \in \Pi : L_{eq}(\pi) \le \varepsilon_{eq}\}$. Then every $\pi \in \Pi_{approx}(\varepsilon_{eq})$ lies in the sup-ball of radius ξ around Π_{eq}. Consequently, for any target covering radius $r > \xi$:

$$N_{\infty,1}\big(\Pi_{approx}(\varepsilon_{eq}), r\big) \le N_{\infty,1}\big(\Pi_{eq}, r - \xi\big).$$

Proof. The first claim is that $|J(\pi) - J(Q(\pi))| \le L_R \cdot d(\pi, Q(\pi)) \le L_R \xi$.

First, we establish the L_R-Lipschitz property of the expected return $J(\pi) = \mathbb{E}_\tau[R(\pi;\tau)]$. Using the property from that the return function R is L_R-Lipschitz, we have:

$$\begin{aligned} |J(\pi) - J(Q(\pi))| &= |\mathbb{E}_\tau[R(\pi;\tau) - R(Q(\pi);\tau)]| \\ &\le \mathbb{E}_\tau\big[|R(\pi;\tau) - R(Q(\pi);\tau)|\big] \\ &\le \mathbb{E}_\tau\big[L_R \cdot d(\pi, Q(\pi))\big] = L_R \cdot d(\pi, Q(\pi)). \end{aligned}$$

Next, we bound the distance $d(\pi, Q(\pi))$. Using the definition of the projection $Q(\pi)$, we find the distance from π to its projection:

$$\begin{aligned} d(\pi, Q(\pi)) &= \sup_s \|\pi(s) - Q(\pi)(s)\|_1 \\ &= \sup_s \big\|\pi(s) - \tfrac{1}{2}\left(\pi(s) + K_g(\pi(L_g(s)))\right)\big\|_1 \\ &= \tfrac{1}{2} \sup_s \|\pi(s) - K_g(\pi(L_g(s)))\|_1 . \end{aligned}$$

The term inside the norm is equal to the equivariance mismatch $\Delta_\pi(s') := \pi(L_g(s')) - K_g(\pi(s'))$ evaluated at $s' = L_g(s)$, since L_g is an involution.

$$\Delta_\pi(L_g(s)) = \pi(L_g(L_g(s))) - K_g(\pi(L_g(s))) = \pi(s) - K_g(\pi(L_g(s))).$$

Since L_g is a bijection, $\sup_s \|\Delta_\pi(L_g(s))\|_1 = \sup_{s'} \|\Delta_\pi(s')\|_1$. By Lemma C.9, this supremum is bounded by ξ. Therefore:

$$d(\pi, Q(\pi)) = \frac{1}{2} \sup_{s'} \|\Delta_\pi(s')\|_1 \le \xi.$$

The second claim is that for any radius $r > \xi$, we have $N_{\infty,1}\big(\Pi_{approx}(\varepsilon_{eq}), r\big) \le N_{\infty,1}\big(\Pi_{eq}, r - \xi\big)$. We know that for any $\pi \in \Pi_{approx}(\varepsilon_{eq})$, its projection $Q(\pi) \in \Pi_{eq}$ satisfies $d(\pi, Q(\pi)) \le \xi$. This implies that the set $\Pi_{approx}(\varepsilon_{eq})$ is contained in a ξ-neighbourhood of Π_{eq}. Let $\{\pi_j\}_{j=1}^M$ be a minimal $(r - \xi)$-cover for Π_{eq}, where $M = N_{\infty,1}(\Pi_{eq}, r - \xi)$. Now, consider any policy $\pi \in \Pi_{approx}(\varepsilon_{eq})$. There must exist a centre π_j from our cover such that $d(Q(\pi), \pi_j) \le r - \xi$. By the triangle inequality, we can bound the distance from π to this centre π_j:

$$\begin{aligned} d(\pi, \pi_j) &\le d(\pi, Q(\pi)) + d(Q(\pi), \pi_j) \\ &\le \xi + (r - \xi) = r. \end{aligned}$$

This shows that the set $\{\pi_j\}_{j=1}^M$ is an r-cover for $\Pi_{approx}(\varepsilon_{eq})$. Since we have found a valid cover of size M, the size of the minimal cover must be no larger:

$$N_{\infty,1}\big(\Pi_{approx}(\varepsilon_{eq}), r\big) \le N_{\infty,1}\big(\Pi_{eq}, r - \xi\big).$$

□

Theorem C.11. *With $\mathcal{R}_{\Pi_{\text{eq}}} = \{\tau \mapsto R(\pi;\tau) : \pi \in \Pi_{\text{eq}}\}$, fix any accuracy parameter $r \in (0, B)$ and confidence $\delta \in (0, 1)$. Then with probability at least $1 - \delta$,*

$$\sup_{\pi \in \Pi_{approx}(\varepsilon_{eq})} |J(\pi) - \hat{J}_N(\pi)| \le C\left(\int_r^B \sqrt{\frac{\log \mathcal{N}_{\infty,1}(\mathcal{R}_{\Pi_{\text{eq}}}, \varepsilon)}{N}} d\varepsilon\right) + \frac{8r}{\sqrt{N}} + B\sqrt{\frac{\log(2/\delta)}{2N}} + 2L_R\xi.$$

Proof. For any policy $\pi \in \Pi_{approx}(\varepsilon_{eq})$, we can decompose the generalisation error using the triangle inequality by introducing its exact-equivariant projection $Q(\pi) \in \Pi_{eq}$:

$$|J(\pi) - \hat{J}_N(\pi)| \leq |J(\pi) - J(Q(\pi))| + |J(Q(\pi)) - \hat{J}_N(Q(\pi))| + |\hat{J}_N(Q(\pi)) - \hat{J}_N(\pi)|.$$

We bound each of the three terms on the right-hand side.

From Theorem C.10, we have:

$$|J(\pi) - J(Q(\pi))| \leq L_R \cdot d(\pi, Q(\pi)) \leq L_R \xi.$$

Since the return function $R(\cdot; \tau)$ is L_R-Lipschitz:

$$\begin{aligned} |\hat{J}_N(Q(\pi)) - \hat{J}_N(\pi)| &= \left| \frac{1}{N} \sum_{i=1}^{N} (R(Q(\pi); \tau_i) - R(\pi; \tau_i)) \right| \\ &\leq \frac{1}{N} \sum_{i=1}^{N} |R(Q(\pi); \tau_i) - R(\pi; \tau_i)| \\ &\leq \frac{1}{N} \sum_{i=1}^{N} L_R \cdot d(\pi, Q(\pi)) \leq L_R \xi. \end{aligned}$$

The middle term, $|J(Q(\pi)) - \hat{J}_N(Q(\pi))|$, is the generalisation error for an exactly equivariant policy. Combining the bounds, we get:

$$\sup_{\pi \in \Pi_{approx}(\varepsilon_{eq})} |J(\pi) - \hat{J}_N(\pi)| \leq \sup_{\pi' \in \Pi_{eq}} |J(\pi') - \hat{J}_N(\pi')| + L_R \gamma.$$

Applying the high-probability bound from Theorem C.8 to the supremum over Π_{eq} yields the final result. □

D Additional Details of Environments

This appendix presents the tables on the environments and how the state space is divided into a symmetric and an asymmetric part. First Table 3 highlights the differences between environments in dimension sizes. Tables 4, 5, 6, and 7 show the division for mo-hopper-v5, mo-walker2d-v5, mo-halfcheetah-v5, and mo-swimmer-v5, respectively. The action space is always divided into an empty set for the asymmetric part, and the complete set for the symmetric part.

Table 3: Considered MuJoCo environments.

	State Space	Action Space	Reward Space
Mo-hopper-v5	$\mathcal{S} \in \mathbb{R}^{11}$	$\mathcal{A} \in \mathbb{R}^3$	$\mathcal{R} \in \mathbb{R}^3$
Mo-walker2d-v5	$\mathcal{S} \in \mathbb{R}^{17}$	$\mathcal{A} \in \mathbb{R}^6$	$\mathcal{R} \in \mathbb{R}^2$
Mo-halfcheetah-v5	$\mathcal{S} \in \mathbb{R}^{17}$	$\mathcal{A} \in \mathbb{R}^6$	$\mathcal{R} \in \mathbb{R}^2$
Mo-swimmer-v5	$\mathcal{S} \in \mathbb{R}^8$	$\mathcal{A} \in \mathbb{R}^2$	$\mathcal{R} \in \mathbb{R}^2$

Table 4: Reflectional symmetry partition for mo-hopper-v5 observation space.

Index	Observation Component	Type	Symmetry
0	z-coordinate of the torso	position	Asymmetric
1	angle of the torso	angle	Asymmetric
2	angle of the thigh joint	angle	Symmetric
3	angle of the leg joint	angle	Symmetric
4	angle of the foot joint	angle	Symmetric
5	velocity of the x-coordinate of the torso	velocity	Asymmetric
6	velocity of the z-coordinate of the torso	velocity	Asymmetric
7	angular velocity of the angle of the torso	angular velocity	Asymmetric
8	angular velocity of the thigh hinge	angular velocity	Symmetric
9	angular velocity of the leg hinge	angular velocity	Symmetric
10	angular velocity of the foot hinge	angular velocity	Symmetric

Table 5: Reflectional symmetry partition for mo-walker2d-v5 observation space.

Index	Observation Component	Type	Symmetry
0	z-coordinate of the torso	position	Asymmetric
1	angle of the torso	angle	Asymmetric
2	angle of the thigh joint	angle	Symmetric
3	angle of the leg joint	angle	Symmetric
4	angle of the foot joint	angle	Symmetric
5	angle of the left thigh joint	angle	Symmetric
6	angle of the left leg joint	angle	Symmetric
7	angle of the left foot joint	angle	Symmetric
8	velocity of the x-coordinate of the torso	velocity	Asymmetric
9	velocity of the z-coordinate of the torso	velocity	Asymmetric
10	angular velocity of the angle of the torso	angular velocity	Asymmetric
11	angular velocity of the thigh hinge	angular velocity	Symmetric
12	angular velocity of the leg hinge	angular velocity	Symmetric
13	angular velocity of the foot hinge	angular velocity	Symmetric
14	angular velocity of the left thigh hinge	angular velocity	Symmetric
15	angular velocity of the left leg hinge	angular velocity	Symmetric
16	angular velocity of the left foot hinge	angular velocity	Symmetric

Table 6: Reflectional symmetry partition for mo-halfcheetah-v5 observation space.

Index	Observation Component	Type	Symmetry
0	z-coordinate of the front tip	position	Asymmetric
1	angle of the front tip	angle	Asymmetric
2	angle of the back thigh	angle	Symmetric
3	angle of the back shin	angle	Symmetric
4	angle of the back foot	angle	Symmetric
5	angle of the front thigh	angle	Symmetric
6	angle of the front shin	angle	Symmetric
7	angle of the front foot	angle	Symmetric
8	velocity of the x-coordinate of front tip	velocity	Asymmetric
9	velocity of the z-coordinate of front tip	velocity	Asymmetric
10	angular velocity of the front tip	angular velocity	Asymmetric
11	angular velocity of the back thigh	angular velocity	Symmetric
12	angular velocity of the back shin	angular velocity	Symmetric
13	angular velocity of the back foot	angular velocity	Symmetric
14	angular velocity of the front thigh	angular velocity	Symmetric
15	angular velocity of the front shin	angular velocity	Symmetric
16	angular velocity of the front foot	angular velocity	Symmetric

Table 7: Reflectional symmetry partition for mo-swimmer-v5 observation space.

Index	Observation Component	Type	Symmetry
0	angle of the front tip	angle	Asymmetric
1	angle of the first rotor	angle	Symmetric
2	angle of the second rotor	angle	Symmetric
3	velocity of the tip along the x-axis	velocity	Asymmetric
4	velocity of the tip along the y-axis	velocity	Symmetric
5	angular velocity of the front tip	angular velocity	Asymmetric
6	angular velocity of first rotor	angular velocity	Symmetric
7	angular velocity of second rotor	angular velocity	Symmetric

E Additional Details of Experimental Settings

Evaluation Measures. For the approximated Pareto front, we consider three well-known metrics that investigate the extent of the approximated front.

First, we consider hypervolume (HV) (Fonseca et al., 2006), which measures the volume of the objective space dominated by the approximated Pareto front relative to a reference point. A downside of many evaluation measures is that they require domain knowledge about the true underlying Pareto front, whereas HV only considers a reference point without any a priori knowledge, making it ideal to assess the volume of the front. The reference point is typically set to the nadir point or slightly worse, and following Felten et al. (2023), we set it to -100 for all objectives and environments. The HV is defined as follows:

$$HV(CS, \boldsymbol{r}) = \lambda \left(\bigcup_{\boldsymbol{cs} \in CS} \boldsymbol{x} \in \mathbb{R}^L : \boldsymbol{cs} \preceq \boldsymbol{x} \preceq \boldsymbol{r} \right),$$

where $CS = \boldsymbol{cs}_1, \boldsymbol{cs}_2, \ldots, \boldsymbol{cs}_n$ is the coverage set, or the Pareto front approximation, $\boldsymbol{r} \in \mathbb{R}^L$ is the reference point, $\boldsymbol{cs} \preceq \boldsymbol{x}$ means $cs_i \leq x_i$ for all objectives $i = 1, \ldots, L$, and $\lambda(\cdot)$ denotes the Lebesgue measure. Yet, hypervolume values are difficult to interpret, as they do not have a direct link to any notion of value or utility (Hayes et al., 2022).

As such, we also consider the Expected Utility Metric (EUM) (Zintgraf et al., 2015), which computes the expected maximum utility across different preference weight vectors, and is defined as follows:

$$EUM(CS, \mathcal{W}) = \frac{1}{|\mathcal{W}|} \sum_{\boldsymbol{\omega} \in \mathcal{W}} \max_{\boldsymbol{cs} \in CS} U(\boldsymbol{\omega}, \boldsymbol{cs}),$$

where $\mathcal{W} = \{\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_k\}$ is a set of weight vectors, $|\mathcal{W}|$ is the cardinality of the weight set, $U(\boldsymbol{\omega}, \boldsymbol{cs})$ is the utility function, which is set to $U(\boldsymbol{\omega}, \boldsymbol{s}) = \boldsymbol{\omega} \cdot \boldsymbol{cs} = \sum_{i=1}^{L} \omega_i \cdot cs_i$.

To specifically assess performance with respect to distributional preferences, we also consider one metric designed to evaluate the optimality of the entire return distribution associated with the learned policies (Cai et al., 2023).

To be precise, we consider the Variance Objective (VO), which evaluates how well the policy set can balance the trade-off between maximising expected returns and minimising their variance. A set of M random preference vectors is generated, where each vector specifies a different weighting between the expected return and its standard deviation for each objective. The satisfaction score $u(p_i, \pi_j)$ for a policy π_j under preference p_i is a weighted sum of the expected return $\mathbb{E}[Z(\pi_j)]$ and the negative standard deviation $-\sqrt{\text{Var}[Z(\pi_j)]}$. The final metric is the mean score over these preferences, rewarding policies that achieve high expected returns with low variance:

$$\text{VO}(\Pi, \{p_i\}_{i=1}^{M}) = \frac{1}{M} \sum_{i=1}^{M} \max_{\pi_j \in \Pi} u(p_i, \pi_j).$$

Hyperparameters. Due to time, computational limitations, and the excessive number of hyperparameters, we do not perform an extensive hyperparameter tuning process. Below are the used hyperparameters. All hyperparameters that are not mentioned below are set to their default value.

The probability of releasing sparse rewards p_{rel} is always set to a one-hot vector, where sparsity is imposed on the reward dimension related to moving forward. Since the main goal is to move forward, imposing sparsity on this channel should make it a more difficult task for the reward shaping model. Furthermore, we deal with extreme heterogeneous sparsity, where most channels exhibit regular rewards, but one channel only releases a reward at the end of an episode, making it more difficult for the model to link certain states and actions to the observed cumulative reward.

The hyperparameters in Table 8 for ReSymNet are identical for each environment. The advantage of using the same hyperparameters for each environment is that if one configuration performs well everywhere, it could indicate that the proposed method is inherently stable, especially given the noted diversity between the considered environments. However, this does come at a cost of potentially suboptimal performance per environment.

Table 8: Hyperparameters for ReSymNet.

	PRISM
Initial collection N	1000
Expert collection E	1000
Number of refinements IR	2
Timesteps per cycle M	100,000
Epochs	1000
Learning rate	0.005
Learning rate scheduler	Exponential
Learning rate decay	0.99
Ensemble size $\lvert\mathcal{E}\rvert$	3
Hidden dimension	256
Dropout	0.3
Initialisation	Kaiman (He et al., 2015)
Validation split	0.2
Patience	20
Batch size	32

The hyperparameter controlling the symmetry loss differs per environment, since some environments require strict equivariance, whereas others require a more flexible approach. Table 9 shows the used values.

Table 9: SymReg hyperparameter.

	Mo-hopper-v5	Mo-walker2d-v5	Mo-halfcheetah-v5	Mo-swimmer-v5
λ	0.01	1	0.01	0.005

F Ablation Study

Table 10 reports the obtained values for the ablation study. Results are again averaged over ten trials, similar to the main experiments.

Table 10: PRISM ablation study results. We report the average hypervolume (HV), Expected Utility Metric (EUM), and Variance Objective (VO) over 10 trials, with the standard error shown in grey. w/o is the abbreviation of without. The largest values are in bold font.

Environment	Metric	PRISM	w/o residual	w/o dense rewards	w/o ensemble	w/o refinement	w/o loss
Mo-hopper-v5	HV ($\times 10^7$)	**1.58** ± 0.05	1.29 ± 0.09	1.38 ± 0.11	1.38 ± 0.08	1.55 ± 0.04	1.42 ± 0.07
	EUM	**147.43** ± 2.61	128.40 ± 6.06	134.67 ± 6.89	135.28 ± 4.91	145.89 ± 2.73	137.85 ± 4.22
	VO	**66.66** ± 1.40	58.61 ± 2.71	61.21 ± 3.03	61.51 ± 2.19	66.54 ± 1.34	62.71 ± 1.83
Mo-walker2d-v5	HV ($\times 10^4$)	**4.77** ± 0.07	4.65 ± 0.11	4.66 ± 0.06	4.60 ± 0.08	4.60 ± 0.09	4.58 ± 0.13
	EUM	**120.43** ± 1.64	114.33 ± 2.48	116.83 ± 1.65	113.79 ± 2.02	114.98 ± 2.84	112.77 ± 3.01
	VO	**59.35** ± 0.80	56.46 ± 1.21	57.67 ± 0.73	56.19 ± 0.97	57.03 ± 1.42	55.59 ± 1.44
Mo-halfcheetah-v5	HV ($\times 10^4$)	**2.25** ± 0.18	1.95 ± 0.20	2.08 ± 0.21	1.91 ± 0.19	2.23 ± 0.18	1.90 ± 0.19
	EUM	89.94 ± 15.33	73.06 ± 16.57	82.24 ± 16.97	81.60 ± 17.65	**92.68** ± 14.79	71.12 ± 16.91
	VO	40.72 ± 7.02	32.99 ± 7.65	37.31 ± 7.99	36.76 ± 8.06	**42.28** ± 6.85	32.12 ± 7.75
Mo-swimmer-v5	HV ($\times 10^4$)	**1.21** ± 0.00	**1.21** ± 0.00	1.20 ± 0.00	1.20 ± 0.00	**1.21** ± 0.00	1.20 ± 0.00
	EUM	9.44 ± 0.14	9.39 ± 0.15	9.07 ± 0.11	9.25 ± 0.13	**9.46** ± 0.13	9.35 ± 0.14
	VO	**4.24** ± 0.07	4.20 ± 0.08	4.09 ± 0.05	4.15 ± 0.08	**4.24** ± 0.07	**4.24** ± 0.07

G Declaration on Large Language Models

Large Language Models (LLMs) were used for (1) polishing the wording of the manuscript for clarity and readability, (2) brainstorming about algorithm names and their abbreviations, and (3) searching for algorithms for consideration in the preliminary stage.