

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 EXPLOITING REFLECTIONAL SYMMETRY IN HETERO- GENEOUS 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. 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 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., 2025a), 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. 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 avail-

able 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.

The complementary components of PRISM synergise as follows. Heterogeneous reward structures cause asymmetric policy learning that violates the agent’s physical symmetry: when dense objectives provide immediate gradients while sparse objectives only signal at the end of an episode, the policy may overfit to the denser objectives in specific states, failing to respect reflectional symmetry. ReSymNet eliminates temporal heterogeneity by aligning objectives to the same frequency, whereas SymReg enforces reflectional symmetry by preventing asymmetric learning dynamics.

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; Ferten et al., 2023), 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 (Rojers et al., 2015; Van Moffaert & Nowé, 2014; Reymond & Nowé, 2019; Lautenbacher et al., 2025); and (3) **meta-policy and single universal policy methods that learn adaptable policies given some preferences** (Chen et al., 2019; 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). **Recent advances utilise statistical decomposition to**

108 address sparsity (Gangwani et al., 2020; Ren et al., 2022), or capture complex reward dependencies using transformers (Tang et al., 2024; 2025b). These approaches improve sample efficiency in 109 single-objective RL, but do not extend naturally to MORL, where heterogeneous sparsity and scale 110 can distort learning dynamics and Pareto-optimal trade-offs.

112 **Reflectional Equivariance.** To incorporate reflectional symmetry, a possible method is data 113 augmentation, which adds mirrored transitions to the replay buffer but doesn't guarantee a symmetric 114 policy and increases data processing costs (Lin et al., 2020). Mondal et al. (2022) propose latent 115 space learning that encourages a symmetric representation through specialised loss functions. **Another 116 line of research focuses on equivariant neural networks (van der Pol et al., 2020; Mondal et al., 117 2020; Wang et al., 2021).** For example, Wang et al. (2022) design a stronger inductive bias via 118 architecture-level symmetry, which hard-codes equivariance into the model for instantaneous 119 generalisation. However, Park et al. (2025) show that strictly equivariant architectures can be too rigid for 120 tasks where symmetries are approximate rather than perfect. Building on this insight, our framework 121 helps overcome the limitations of strictly equivariant architectures through tunable flexibility whilst 122 being model-agnostic.

123

124

3 PRELIMINARIES

125

126

Multi-Objective Markov Decision Process. Formally, we define an MORL problem via the Multi-Objective Markov Decision Process (MOMDP) model, as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathbf{r}, \gamma)$: 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 $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S}' \rightarrow [0, 1]$, also denoted as $P(s'|s, a)$. The agent receives a reward via an L -dimensional vector-valued reward function $\mathbf{r} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^L$, 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 $\pi : \mathcal{S} \rightarrow \mathcal{A}$ that optimises the expected cumulative vector return, defined as $\mathbf{J}(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t \mathbf{r}_t]$. 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} , which describes the probability of observing different possible trajectories under the policy being followed.

138

139

140

141

142

143

144

145

146

147

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 $\mathcal{DC} = \{d_1, d_2, \dots, d_D\}$ 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 $\tau = \{(s_1, a_1), \dots, (s_T, a_T)\}$ with cumulative sparse reward $R^{sp}(\tau)$, the task is to infer $\mathbf{r}^{sp} = [r_1^{sp}, \dots, r_T^{sp}]^\top$, where r_t^{sp} is the sparse reward at timestep t , 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.

148

149

150

151

152

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).

153

154

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 $l_{\infty,1}(f, g) = \max_{x \in \mathcal{X}} (\sum_{i=1}^n |f_i(x) - g_i(x)|)$.*

155

156

157

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.*

158

159

160

161

Definition 3 (Rademacher complexity). *Let \mathcal{F} be a class of real-valued functions on a feature space \mathcal{X} , and let τ_1, \dots, τ_N be i.i.d. samples from a distribution over \mathcal{X} . The empirical Rademacher complexity of \mathcal{F} is $\hat{\mathcal{R}}_N(\mathcal{F}) = \mathbb{E}_\sigma[\sup_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N \sigma_i f(\tau_i)]$, where $\sigma_1, \dots, \sigma_N$ are independent Rademacher random variables taking values ± 1 with equal probability. The Rademacher complexity of \mathcal{F} is the expectation over the sample set.*

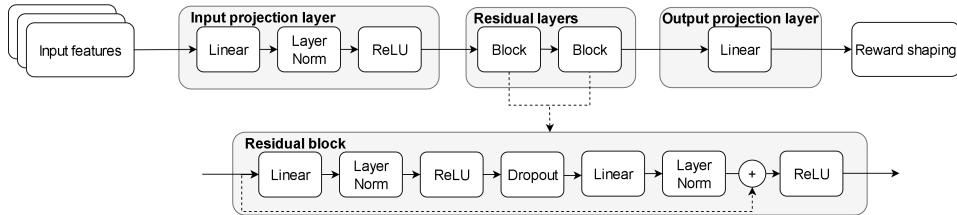


Figure 1: Overview of ReSymNet.

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, $\mathcal{R}_{\text{pred}}$, parametrised by ψ , 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 $\mathbf{h}_t = [s_t, a_t, r_t^{\text{dense}}]$, where s_t is the state, a_t is the action, and $r_{\text{dense},t}$ 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 $\mathcal{R}_{\text{pred}}$ 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:

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

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

However, aligning reward frequencies alone is insufficient, as heterogeneous rewards cause the policy to learn asymmetrically across objectives, violating the agent’s physical symmetry. To address this, we leverage reflectional symmetry as an inductive bias to prevent asymmetric policy learning. 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.

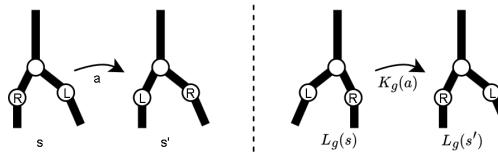


Figure 2: Reflectational symmetry in a two-legged agent. The left panel shows a transition from state s to s' under action a , whereas the right panel shows the reflected transition, where states and actions are transformed by L_g and K_g , respectively.

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, where d_s is the dimension of the state space and d_a of the action space. We define index sets $I_{\text{asym}}^s \subset \{1, \dots, d_s\}$ and $I_{\text{sym}}^s \subset \{1, \dots, d_s\}$ such that $I_{\text{asym}}^s \cap I_{\text{sym}}^s = \emptyset$ and $I_{\text{asym}}^s \cup I_{\text{sym}}^s = \{1, \dots, d_s\}$. This partitions the state vector as $s = (s_{\text{asym}}, s_{\text{sym}})$ where $s_{\text{asym}} = s_{I_{\text{asym}}^s}$ and $s_{\text{sym}} = s_{I_{\text{sym}}^s}$. 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, $L_g : \mathcal{S} \rightarrow \mathcal{S}$, reflects the symmetric part of the state as follows: $L_g(s) = (s_{\text{asym}}, -s_{\text{sym}})$. Similarly, we define index sets I_{asym}^a and I_{sym}^a 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, $K_g : \mathcal{A} \rightarrow \mathcal{A}$, 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:

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

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. Let Π be the full hypothesis space of policies represented by ReSymNet, $R(\pi; \tau)$ is the cumulative return for a single trajectory τ obtained following policy π .

Remark 2. As the backbone of the whole method, the hypothesis complexity and generalisability of ReSymNet contribute significantly to the generalisability of the whole algorithm. Due to space limit, we present Theorem B.5 in the appendices for the covering number of ReSymNet's hypothesis space.

The theory relies on these assumptions:

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

Assumption 2 (Lipschitz-continuous return). There exists $L_R > 0$ such that for all $\pi, \tilde{\pi} \in \Pi$ and any trajectory τ , $|R(\pi; \tau) - R(\tilde{\pi}; \tau)| \leq L_R d(\pi, \tilde{\pi})$, 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

270 in the policy outputs, it follows that returns are Lipschitz in the policies themselves, as stated in
 271 Assumption 2. Assumption 5 ensures that all relevant states are sufficiently sampled under the be-
 272 haviour policy, which is, in practice, reasonable because policy exploration mechanisms prevent the
 273 policy from collapsing onto a subset of states.

275 **5.1 GENERALISABILITY OF REFLECTION-EQUIVARIANT SUBSPACE**
 276

277 Let $G = \mathbb{Z}_2$ act on states and actions via L_g, K_g . An orbit-averaging operator $\mathcal{Q}(\pi)(s) =$
 278 $\frac{1}{2}(\pi(s) + K_g(\pi(L_g(s))))$ maps any policy to a reflection-equivariant subspace (Qin et al., 2022).
 279 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 sub-
 280 space, defined as follows.

281 **Definition 4** (reflection-equivariant subspace). *We define reflection-equivariant subspace as $\Pi_{\text{eq}} :=$*
 282 $\{\pi : \pi(L_g(s)) = K_g(\pi(s))\}$.

283 We prove that \mathcal{Q} is reflectional equivariant, a projection, and that its image coincides with the set
 284 of equivariant policies in Lemmas C.4, C.5, and C.6 in Appendix C.3, respectively. Thus, \mathcal{Q} is
 285 surjective onto Π_{eq} . To prove that the subspace Π_{eq} is less complex, we show that the projection \mathcal{Q}
 286 is non-expansive, which implies its image has a covering number no larger than the original space.

287 **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, $\mathcal{N}_{\infty,1}(\Pi_{\text{eq}}, r) \leq \mathcal{N}_{\infty,1}(\Pi, r)$.*

291 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
 292 distance between their projections, $d(\mathcal{Q}(\pi_\phi), \mathcal{Q}(\pi_\theta))$, is no larger using the fact that K_g is a norm-
 293 preserving isometry, $\|K_g(a)\|_1 = \|a\|_1$, and that L_g is a bijection, which implies that the supremum
 294 over s equals the supremum over $L_g(s)$. Hence \mathcal{Q} is non-expansive, and a non-expansive surjective
 295 map cannot increase the covering number. Following Lemma C.6, $\mathcal{N}(\Pi_{\text{eq}}, r) \leq \mathcal{N}(\Pi, r)$. A detailed
 296 proof can be found in Appendix C.4.

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

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

$$301 \mathbb{E} \left[\sup_{f \in \mathcal{F}} \left| \frac{1}{N} \sum_{i=1}^N (f(\tau_i) - \mathbb{E}[f]) \right| \right] \leq 2\mathbb{E}[\mathfrak{R}_N(\mathcal{F})],$$

304 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 indepen-
 305 dent Rademacher random variables taking values ± 1 with equal probability.

307 This bound transforms the original centred empirical process into a symmetrised version that is
 308 often easier to analyse. We now prove a high-probability uniform generalisation bound over the
 309 reflection-equivariant subspace. A detailed proof can be found in Appendix C.5. We recognise that
 310 PRISM does not necessarily converge to it, which will be discussed in the following subsection.

311 **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
 312 confidence $\delta \in (0, 1)$. Then with probability at least $1 - \delta$,*

$$313 \sup_{\pi \in \Pi_{\text{eq}}} |J(\pi) - \hat{J}_N(\pi)| \leq C \left(\int_r^B \sqrt{\frac{\log \mathcal{N}_{\infty,1}(\mathcal{R}_{\Pi_{\text{eq}}}, \varepsilon)}{N}} d\varepsilon \right) + \frac{8r}{\sqrt{N}} + B \sqrt{\frac{\log(2/\delta)}{2N}},$$

316 where C is an absolute numeric constant, $J(\pi)$ is the population expected return and $\hat{J}_N(\pi) =$
 317 $\frac{1}{N} \sum_{i=1}^N R(\pi; \tau_i)$ is the empirical return on N i.i.d. episodes τ_1, \dots, τ_N .

318 **Corollary 5.4.** *Under the same assumptions as Theorem 5.3, for any $r \in (0, B)$ and $\delta \in (0, 1)$, the
 319 upper bound in Theorem 5.3 for Π_{eq} is at most the same bound obtained by replacing Π_{eq} with Π .
 320 By Lemma C.8, the return-class covering numbers can be bounded by those of the policy class with
 321 radius scaled by $1/L_R$. Mathematically, following Theorem 5.1, for every $\varepsilon > 0$,*

$$322 \log \mathcal{N}_{\infty,1}(\Pi_{\text{eq}}, \varepsilon/L_R) \leq \log \mathcal{N}_{\infty,1}(\Pi, \varepsilon/L_R), \tag{2}$$

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

324 The equivariance regulariser projects policies onto a smaller fixed-point subspace Π_{eq} , which prov-
 325 ably has covering numbers no larger than Π . The return class inherits this reduction via the Lipschitz
 326 map, so the Dudley entropy integral for Π_{eq} is bounded by that of Π . As a consequence, the upper
 327 bound on the generalisation gap is no larger for Π_{eq} compared to Π .
 328

329 **5.2 GENERALISABILITY OF PRISM**
 330

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

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

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

$$338 |J(\pi) - J(Q(\pi))| \leq L_R \cdot d(\pi, Q(\pi)) \leq L_R \xi. \quad (3)$$

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

$$342 \mathcal{N}_{\infty,1}(\Pi_{\text{approx}}(\varepsilon_{\text{eq}}), r) \leq \mathcal{N}_{\infty,1}(\Pi_{\text{eq}}, r - \xi). \quad (4)$$

344 By Lipschitzness of returns, the expected return of a policy and its projection differ by at most
 345 $L_R d(\pi, Q(\pi))$. The mismatch Δ_{π} controls this distance, and Lemma C.10 bounds its supremum
 346 by ξ , giving the first inequality. Geometrically, $\Pi_{\text{approx}}(\varepsilon_{\text{eq}})$ is contained in a ξ -tube around Π_{eq} .
 347 Hence any $(r - \xi)$ -cover of Π_{eq} yields an r -cover of $\Pi_{\text{approx}}(\varepsilon_{\text{eq}})$, proving the covering-number
 348 relation (see Appendix C.6 for a detailed proof).

349 **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
 350 confidence $\delta \in (0, 1)$. Then with probability at least $1 - \delta$,*

$$352 \sup_{\pi \in \Pi_{\text{approx}}(\varepsilon_{\text{eq}})} |J(\pi) - \hat{J}_N(\pi)| \leq C \left(\int_r^B \sqrt{\frac{\log \mathcal{N}_{\infty,1}(\mathcal{R}_{\Pi_{\text{eq}}}, \varepsilon)}{N}} d\varepsilon \right) + \frac{8r}{\sqrt{N}} + B \sqrt{\frac{\log(2/\delta)}{2N}} + 2L_R \xi.$$

355 For $\pi \in \Pi_{\text{approx}}(\varepsilon_{\text{eq}})$, decompose the generalisation error relative to its projection $Q(\pi) \in \Pi_{\text{eq}}$.
 356 The difference in population returns $|J(\pi) - J(Q(\pi))|$ and in empirical returns $|\hat{J}_N(\pi) - \hat{J}_N(Q(\pi))|$
 357 are both bounded by $L_R \xi$ (Theorem 5.5). The middle term $|J(Q(\pi)) - \hat{J}_N(Q(\pi))|$ is exactly the
 358 generalisation error for an equivariant policy. Taking the supremum, we obtain the equivariant bound
 359 (Theorem 5.3) plus $2L_R \xi$. Appendix C.6 provides a detailed proof.

361 **Corollary 5.7.** *Under the same assumptions as Theorem 5.6, for any $r \in (0, B)$ and $\delta \in (0, 1)$,
 362 the upper bound in Theorem 5.6 for $\Pi_{\text{approx}}(\varepsilon_{\text{eq}})$ is at most the same bound obtained by replacing
 363 $\Pi_{\text{approx}}(\varepsilon_{\text{eq}})$ with Π . By Lemma C.8, the return-class covering numbers can be bounded by those
 364 of the policy class with radius scaled by $1/L_R$. Mathematically, following Theorems 5.1 and 5.5,
 365 for any target covering radius $r > \xi$:*

$$366 \log \mathcal{N}_{\infty,1}(\Pi_{\text{approx}}(\varepsilon_{\text{eq}}), r/L_R) \leq \log \mathcal{N}_{\infty,1}(\Pi_{\text{eq}}, (r - \xi)/L_R) \leq \log \mathcal{N}_{\infty,1}(\Pi, (r - \xi)/L_R), \quad (5)$$

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

369 The covering relation incurs a slack of size ξ , leading to bounds of the form $N(\Pi_{\text{approx}}(\varepsilon_{\text{eq}}), r) \leq$
 370 $N(\Pi_{\text{eq}}, r - \xi) \leq N(\Pi, r - \xi)$. By contrast, in Corollary 5.4, this slack disappears. Thus, the exact
 371 case guarantees a strict reduction in complexity, whereas the approximate case trades a ξ -shift in the
 372 radius for retaining proximity to the equivariant subspace.

374 **6 EXPERIMENTS**
 375

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

378
379

6.1 EXPERIMENTAL SETTINGS

380
381
382
383
384
385

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.

386
387
388
389
390
391

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.

392
393
394
395

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.

396

6.2 EMPIRICAL RESULTS

397

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.

405

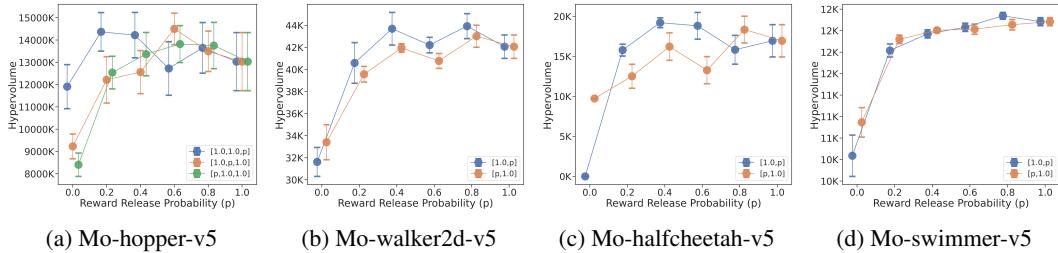
413
414
415

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

416
417
418
419
420
421
422

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.

423
424
425
426

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.

427
428
429

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.

430
431

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

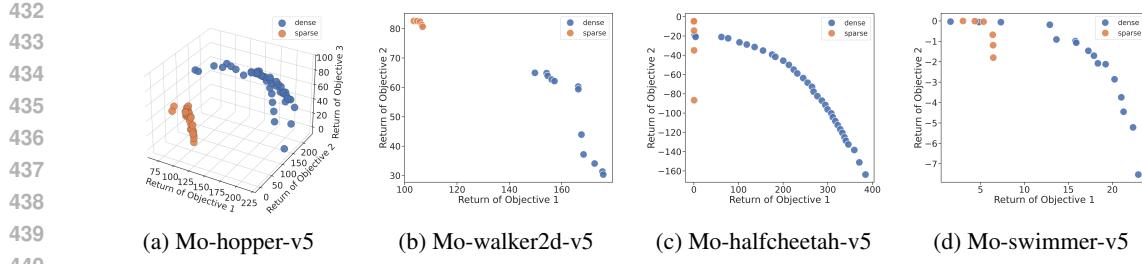


Figure 4: The approximated Pareto front for dense rewards (blue dots) and sparse rewards (orange dots). Sparsity is imposed on the first reward objective.

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

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 solutions 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. [Figure 6 in Appendix F](#), which shows the approximated Pareto fronts, aligns with these results.

We provide two distinct examples to analyse the behaviour of the learned reward signals compared to the oracle for mo-walker2d-v5. Figure 5a illustrates a full 1000-step episode. The shaped reward is highly correlated with the dense reward throughout the entire trajectory. The alignment of peaks and troughs confirms that ReSymNet captures the dynamics of the environment, ensuring accurate credit assignment without temporal drift.

Figure 5b highlights a key theoretical advantage of ReSymNet. In high-performance regions (e.g., steps 250–270), the shaped reward amplifies the signal, exceeding the magnitude of the oracle. By creating steeper gradients for desirable behaviours, the shaped reward can provide more effective guidance than the raw environmental signal, explaining why PRISM is capable of outperforming the oracle.

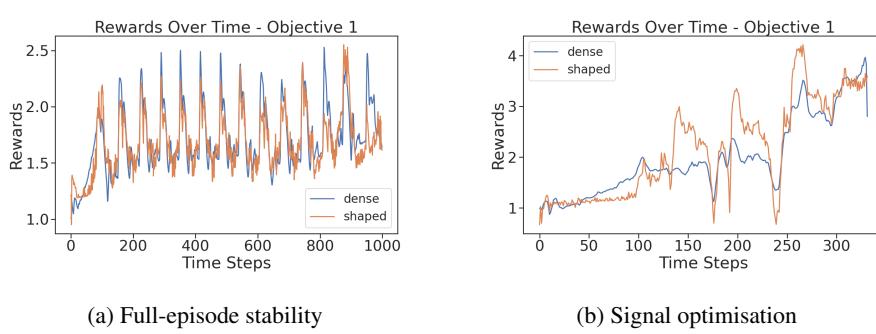


Figure 5: The dense (blue line) and shaped rewards (orange line) over time for mo-walker2d-v5. Sparsity is imposed on the first reward objective.

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. We also include two ablation studies that remove ReSymNet from PRISM and replace the reward shaping model as follows: (7) **uniform**: Distributes the episodic sparse reward $R^{\text{sp}}(\tau)$ equally across all T timesteps, and (8) **random**: Samples random weights $\alpha_t \sim \mathcal{U}(-1, 1)$ for each timestep, normalises to sum to one, and scales by the total reward.

The ablation results in Tables 10 and 11 in Appendix G 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. Considering ReSymNet, uniform achieves moderate performance by providing per-step gradients and leveraging SymReg, while random performs poorly due to noisy, misleading rewards. PRISM consistently outperforms both by learning reward decomposition with ReSymNet and enforcing structural consistency via SymReg, enabling accurate credit assignment in complex multi-objective tasks. 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.

540
541 ETHICS STATEMENT542
543 We declare no potential conflict of interest nor sponsorship. We are not aware of any issues related
544 to legal compliance, research integrity, or other ethical considerations.545
546 REPRODUCIBILITY STATEMENT547
548 We have taken several steps to ensure reproducibility. All assumptions underlying our theoretical
549 results are explicitly stated in Section 5. For the empirical results, we use only publicly available
550 environments, described in Section 6, with training details, hyperparameters, and evaluation met-
551 rics reported in Appendix E. To further support reproducibility, we released an anonymous code
552 repository containing implementation details at https://anonymous.4open.science/r/reward_shaping-1CCB, which will be publicly released.553
554 REFERENCES555
556 Lucas Nunes Alegre, Ana L. C. Bazzan, Diederik M. Roijers, Ann Nowé, and Bruno C. da Silva.
557 Sample-efficient multi-objective learning via generalized policy improvement prioritization. In
558 *2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023)*,
559 pp. 2003–2012. ACM, 2023.
560
561 Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, meth-
562 ods and progress. *Artificial Intelligence*, 297:103500, 2021.
563
564 Arthur Aubret, Laetitia Matignon, and Salima Hassas. A survey on intrinsic motivation in reinforce-
565 ment learning. *arXiv preprint arXiv:1908.06976*, 2019.
566
567 Peter L Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and
568 structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.
569
570 Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for
571 neural networks. *Advances in neural information processing systems*, 30, 2017.
572
573 Toygun Basaklar, Suat Gumussoy, and Ümit Y. Ogras. PD-MORL: Preference-driven multi-
574 objective reinforcement learning algorithm. In *Eleventh International Conference on Learning
575 Representations (ICLR 2023)*. OpenReview.net, 2023.
576
577 Yuri Burda, Harrison Edwards, Amos J. Storkey, and Oleg Klimov. Exploration by random net-
578 work distillation. In *7th International Conference on Learning Representations (ICLR 2019)*.
579 OpenReview.net, 2019.
580
581 Xin-Qiang Cai, Pushi Zhang, Li Zhao, Jiang Bian, Masashi Sugiyama, and Ashley Llorens. Distri-
582 butional Pareto-optimal multi-objective reinforcement learning. In *37th International Conference
583 on Neural Information Processing Systems (NIPS 2023)*, volume 36, pp. 15593–15613. Curran
584 Associates, 2023.
585
586 Xi Chen, Ali Ghadirzadeh, Mårten Björkman, and Patric Jensfelt. Meta-learning for multi-objective
587 reinforcement learning. In *2019 IEEE/RSJ International Conference on Intelligent Robots and
588 Systems (IROS 2019)*, pp. 977–983. IEEE, 2019.
589
590 Rati Devidze, Parameswaran Kamalaruban, and Adish Singla. Exploration-guided reward shaping
591 for reinforcement learning under sparse rewards. In *36th International Conference on Neural
592 Information Processing Systems (NIPS 2022)*. Curran Associates, 2022.
593
594 Richard M Dudley. The sizes of compact subsets of Hilbert space and continuity of Gaussian pro-
595 cesses. *Journal of Functional Analysis*, 1(3):290–330, 1967.
596
597 Florian Felten, Lucas N. Alegre, Ann Nowé, Ana L. C. Bazzan, El-Ghazali Talbi, Grégoire Danoy,
598 and Bruno C. da Silva. A toolkit for reliable benchmarking and research in multi-objective rein-
599 forcement learning. In *37th International Conference on Neural Information Processing Systems
600 (NIPS 2023)*. Curran Associates, 2023.

594 Carlos M. Fonseca, Luís Paquete, and Manuel López-Ibáñez. An improved dimension-sweep algo-
 595 rithm for the hypervolume indicator. In *IEEE International Conference on Evolutionary Compu-*
 596 *tation (CEC 2006)*, pp. 1157–1163. IEEE, 2006.

597

598 Tanmay Gangwani, Yuan Zhou, and Jian Peng. Learning guidance rewards with trajectory-space
 599 smoothing. In *33rd Annual Conference on Neural Information Processing Systems 2020 (NIPS*
 600 *2020)*, 2020.

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

603

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

608 Fengxiang He, Tongliang Liu, and Dacheng Tao. Why resnet works? residuals generalize. *IEEE*
 609 *Transactions on Neural Networks and Learning Systems*, 31(12):5349–5362, 2020.

610

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

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

616

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

620

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

623

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

627

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

630

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

633

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

636

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

639

640 Colin McDiarmid et al. On the method of bounded differences. *Surveys in Combinatorics*, 141(1):
 641 148–188, 1989.

642

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

645

646 Kristof Van Moffaert, Madalina M. Drugan, and Ann Nowé. Scalarized multi-objective reinfor-
 647 cements learning: Novel design techniques. In *2013 IEEE Symposium on Adaptive Dynamic Pro-*
gramming and Reinforcement Learning (ADPRL 2013), pp. 191–199. IEEE, 2013.

648 Arnab Kumar Mondal, Pratheeeksha Nair, and Kaleem Siddiqi. Group equivariant deep reinforce-
 649 ment learning. *arXiv preprint arXiv:2007.03437*, 2020.
 650

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

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

656 Andrew Y. Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *Seventeenth
 657 International Conference on Machine Learning (ICML 2000)*, pp. 663–670. Morgan Kaufmann,
 658 2000.

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

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

666 Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration
 667 by self-supervised prediction. In *34th International Conference on Machine Learning (ICML
 668 2017)*, volume 70 of *PMLR*, pp. 2778–2787. PMLR, 2017.

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

672 Zhizhou Ren, Ruihan Guo, Yuan Zhou, and Jian Peng. Learning long-term reward redistribution via
 673 randomized return decomposition. In *Tenth International Conference on Learning Representa-
 674 tions (ICLR 2022)*. OpenReview.net, 2022.

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

677 Diederik Marijn Roijers, Shimon Whiteson, and Frans A Oliehoek. Computing convex coverage sets
 678 for faster multi-objective coordination. *Journal of Artificial Intelligence Research*, 52:399–443,
 679 2015.

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

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

686 Yuting Tang, Xin-Qiang Cai, Jing-Cheng Pang, Qiyu Wu, Yao-Xiang Ding, and Masashi Sugiyama.
 687 Beyond simple sum of delayed rewards: Non-markovian reward modeling for reinforcement
 688 learning. *arXiv preprint arXiv:2410.20176*, 2024.

689 Yuting Tang, Xin-Qiang Cai, Yao-Xiang Ding, Qiyu Wu, Guoqing Liu, and Masashi Sugiyama. Re-
 690 enforcement learning from bagged reward. *Transactions on Machine Learning Research*, 2025b.

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

694 Elise van der Pol, Daniel E. Worrall, Herke van Hoof, Frans A. Oliehoek, and Max Welling. MDP
 695 homomorphic networks: Group symmetries in reinforcement learning. In *33st Annual Conference
 696 on Neural Information Processing Systems (NIPS 2020)*, 2020.

702 Kristof Van Moffaert and Ann Nowé. Multi-objective reinforcement learning using sets of Pareto
 703 dominating policies. *The Journal of Machine Learning Research*, 15(1):3483–3512, 2014.
 704

705 Dian Wang, Robin Walters, Xupeng Zhu, and Robert Platt Jr. Equivariant Q learning in spatial
 706 action spaces. In *5th Conference on Robot Learning*, volume 164 of *PMLR*, pp. 1713–1723.
 707 PMLR, 2021.

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

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

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

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

719 Ding-Xuan Zhou. The covering number in learning theory. *Journal of Complexity*, 18(3):739–767,
 720 2002.

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

724

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
$\mathbf{r}(s, a) \in \mathbb{R}^L$	Vector-valued reward with L objectives
$\gamma \in [0, 1)$	Discount factor
$\pi : \mathcal{S} \rightarrow \mathcal{A}$	Policy mapping
$\mathbf{J}(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t \mathbf{r}_t]$	Expected cumulative vector return
\mathcal{D}	Behaviour distribution to sample episodes from
$\mathcal{DC} = \{d_1, \dots, d_D\}$	Dense reward channels
$r_t^{d_i}$	Reward from dense channel d_i at timestep t
r_t^{sp}	Sparse reward at timestep t
$\tau = \{(s_1, a_1), \dots, (s_T, a_T)\}$	Trajectory
$R^{sp}(\tau)$	Cumulative sparse reward in episode τ
p_{rel}	Probability of releasing sparse reward
$\mathbf{h}_t = [s_t, a_t, \mathbf{r}_t^{dense}]$	Input feature vector for ReSymNet
\mathcal{R}_{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}	Equivariance regularisation loss
Π	Hypothesis space of policies
$\Pi_{eq} = \{\pi : \pi(L_g(s)) = K_g(\pi(s))\}$	Reflection-equivariant subspace
$\Pi_{approx}(\varepsilon_{eq})$	Approximate equivariant policies with tolerance ε_{eq}

756 **B ADDITIONAL DETAILS AND THEORY OF RESYMNET**
 757

758 We give additional details of ReSymNet as well as the theoretical motivation behind its architecture
 759 in this appendix.
 760

761 **B.1 THEORETICAL MOTIVATION VIA SCALED OPPORTUNITY VALUE**
 762

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

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

769 **Definition 7** (Scaled opportunity value). *For a scale parameter $k > 0$, the scaled opportunity
 770 value shaping function augments the native reward with a scaled opportunity correction:
 771 $\text{OPV}_k(s, a, s') = F_k(s, a, s') = k(\gamma V^M(s') - V^M(s)) + (k - 1)R(s, a)$.*

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

777 Residual blocks mirror the additive structure of scaled opportunity value: each block refines its input
 778 prediction via: $\mathcal{R}_{\text{pred}}^{(i)}(\mathbf{h}_t; \psi) = \mathcal{R}_{\text{pred}}^{(i-1)}(\mathbf{h}_t; \psi) + \Delta_i(\mathbf{h}_t; \psi)$, where Δ_i is a learned correction. A
 779 single block can be viewed as approximating a scaled opportunity-value transformation of its input,
 780 while stacking multiple blocks implements iterative refinement: each stage reduces the residual
 781 error left by the previous one. This residual formulation both stabilises training and aligns with the
 782 principle of scaled opportunity value, gradually shaping per-step predictions into horizon-1 signals
 783 that remain consistent with the sparse episodic return $R^{sp}(\tau)$.
 784

785 **B.2 GENERALISABILITY OF RESYMNET**
 786

787 We extend the theoretical justification of ReSymNet from optimisation to generalisation. Following
 788 the stem–vine decomposition of He et al. (2020), we prove that residual connections do not increase
 789 hypothesis complexity, and derive a high-probability bound.

790 **Notation and Assumptions.** ReSymNet maps feature vectors $\mathbf{h}_t \in \mathbb{R}^{d_0}$ to sparse reward predictions
 791 $r_t^{sp} \in \mathbb{R}$ through a residual network. We decompose the network into:

792

- 793 • **A stem:** the main feedforward pathway consisting of K layers, each with a weight matrix $\mathbf{A}_i \in$
 794 $\mathbb{R}^{d_{i-1} \times d_i}$ and nonlinearity $\sigma_i : \mathbb{R}^{d_i} \rightarrow \mathbb{R}^{d_i}$ for $i = 1, \dots, K$.
- 795 • **A collection of vines:** residual connections (skip connections) indexed by triples (s, t, i) where s
 796 is the source vertex (where the connection starts), t is the target vertex (where it reconnects), and
 797 i distinguishes multiple vines between the same pair of vertices. We denote the set of all vine
 798 indices as \mathcal{I}_V .

799 We denote vertices in the network as $N(t)$, where t indexes the position in the computational graph.
 800 Each vine $\mathcal{V}(s, t, i)$ is itself a small feedforward network with weight matrices $\mathbf{A}_1^{s,t,i}, \dots, \mathbf{A}_{K_{s,t,i}}^{s,t,i}$
 801 and nonlinearities $\sigma_1^{s,t,i}, \dots, \sigma_{K_{s,t,i}}^{s,t,i}$, where $K_{s,t,i}$ is the number of layers in that vine. The output at
 802 vertex $N(t)$ is:
 803

$$F_t(\mathbf{X}) = F_t^S(\mathbf{X}) + \sum_{(s,t,i) \in \mathcal{I}_V} F_{s,t,i}^V(\mathbf{X}),$$

804 where $F_t^S(\mathbf{X})$ is the stem’s output at vertex t and the sum runs over all vines that reconnect at vertex
 805 t .

806 **Assumption 6** (Bounded parameters). *Each stem weight matrix satisfies $\|\mathbf{A}_i\|_\sigma \leq s_i$ for $i =$
 807 $1, \dots, K$, where $\|\cdot\|_\sigma$ denotes the spectral norm. Each vine weight matrix satisfies $\|\mathbf{A}_j^{s,t,i}\|_\sigma \leq$*

810 $s_j^{s,t,i}$. All nonlinearities are ρ_i -Lipschitz continuous: for any $\mathbf{x}_1, \mathbf{x}_2$ in the domain,
 811

$$812 \quad \|\sigma_i(\mathbf{x}_1) - \sigma_i(\mathbf{x}_2)\|_2 \leq \rho_i \|\mathbf{x}_1 - \mathbf{x}_2\|_2.$$

813 Input features satisfy $\|\mathbf{h}_t\|_2 \leq B_h$, network per-step outputs satisfy $|R_{\text{pred}}(\mathbf{h}_t; \psi)| \leq B_{\text{pred}}$ for all t ,
 814 and sparse rewards satisfy $|R^{\text{sp}}(\tau)| \leq B_r$ for all trajectories τ . Trajectories have length bounded
 815 by T_{max} .

816 **Lemma B.2.** Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be a data matrix with n samples and d features, satisfying $\|\mathbf{X}\|_2 \leq B$.
 817 Consider the hypothesis space formed by all linear transformations with bounded spectral norm:

$$819 \quad \mathcal{H}_A = \{\mathbf{XA} : \mathbf{A} \in \mathbb{R}^{d \times m}, \|\mathbf{A}\|_{\sigma} \leq s\}.$$

820 Then the ε -covering number satisfies:

$$822 \quad \log \mathcal{N}_{\infty,2}(\mathcal{H}_A, \varepsilon) \leq \left\lceil \frac{s^2 B^2 m^2}{\varepsilon^2} \right\rceil \log(2dm),$$

823 where m is the output dimension.

824 This lemma (Bartlett et al., 2017) shows that the complexity of a single linear layer scales with the
 825 square of its spectral norm and input norm.

826 **Lemma B.3.** For an K -layer feedforward network with hypothesis space \mathcal{H}_{ff} , the covering number
 827 satisfies:

$$828 \quad \mathcal{N}_{\infty,2}(\mathcal{H}_{\text{ff}}, \varepsilon) \leq \prod_{i=1}^K \sup_{\mathbf{A}_1, \dots, \mathbf{A}_{i-1}} \mathcal{N}_i,$$

829 where \mathcal{N}_i is the covering number of layer i (viewed as a function of its input) when the preceding
 830 layers $\mathbf{A}_1, \dots, \mathbf{A}_{i-1}$ are held fixed. The supremum is taken over all choices of the preceding weight
 831 matrices within their respective spectral norm bounds.

832 This result shows that the covering number of a deep network is the product of the covering numbers
 833 of its individual layers. For residual networks, where outputs are sums of stem and vine contribu-
 834 tions, we require:

835 **Lemma B.4.** Let \mathcal{F} and \mathcal{G} be two function classes. If \mathcal{W}_F is an ε_F -cover of \mathcal{F} (meaning every
 836 $f \in \mathcal{F}$ is within distance ε_F of some element in \mathcal{W}_F), and \mathcal{W}_G is an ε_G -cover of \mathcal{G} , then the set

$$837 \quad \mathcal{W}_F + \mathcal{W}_G = \{f + g : f \in \mathcal{W}_F, g \in \mathcal{W}_G\}$$

838 is an $(\varepsilon_F + \varepsilon_G)$ -cover of the sum class $\mathcal{F} + \mathcal{G} = \{f + g : f \in \mathcal{F}, g \in \mathcal{G}\}$, and

$$839 \quad \mathcal{N}_{\infty,2}(\mathcal{F} + \mathcal{G}, \varepsilon_F + \varepsilon_G) \leq \mathcal{N}_{\infty,2}(\mathcal{F}, \varepsilon_F) \mathcal{N}_{\infty,2}(\mathcal{G}, \varepsilon_G).$$

840 *Proof.* For any $f + g \in \mathcal{F} + \mathcal{G}$, there exist $w_f \in \mathcal{W}_F$ and $w_g \in \mathcal{W}_G$ such that $\|f - w_f\|_2 \leq \varepsilon_F$ and
 841 $\|g - w_g\|_2 \leq \varepsilon_G$. By the triangle inequality:

$$842 \quad \|(f + g) - (w_f + w_g)\|_2 \leq \|f - w_f\|_2 + \|g - w_g\|_2 \leq \varepsilon_F + \varepsilon_G.$$

843 The covering number bound follows since there are at most $|\mathcal{W}_F| \cdot |\mathcal{W}_G|$ distinct pairs (w_f, w_g) . \square

844 **Theorem B.5.** Under Assumption 6, let $\{\varepsilon_j\}_{j=1}^K$ be tolerances for each stem layer and
 845 $\{\varepsilon_{s,t,i}\}_{(s,t,i) \in \mathcal{I}_V}$ be tolerances for each vine, satisfying

$$846 \quad \sum_{j=1}^K \varepsilon_j + \sum_{(s,t,i) \in \mathcal{I}_V} \varepsilon_{s,t,i} \leq \varepsilon.$$

847 Then the covering number of ReSymNet's hypothesis space \mathcal{H}_{res} satisfies:

$$848 \quad \mathcal{N}_{\infty,2}(\mathcal{H}_{\text{res}}, \varepsilon) \leq \prod_{j=1}^K \mathcal{N}_{\infty,2}(\mathcal{H}_j, \varepsilon_j) \prod_{(s,t,i) \in \mathcal{I}_V} \mathcal{N}_{\infty,2}(\mathcal{H}_{s,t,i}^V, \varepsilon_{s,t,i}),$$

864 where \mathcal{H}_j is the hypothesis space of stem layer j and $\mathcal{H}_{s,t,i}^V$ is the hypothesis space of vine $\mathcal{V}(s, t, i)$.
 865 Applying Lemma B.2 to each weight matrix, this yields:

$$866 \quad 867 \quad 868 \quad \log \mathcal{N}_{\infty,2}(\mathcal{H}_{\text{res}}, \varepsilon) \leq \frac{\mathcal{R}}{\varepsilon^2},$$

869 where the complexity measure \mathcal{R} is:

$$870 \quad 871 \quad 872 \quad \mathcal{R} = \sum_{i=1}^K \frac{s_i^2 \|F_{i-1}(\mathbf{X})\|_2^2}{\varepsilon_i^2} \log(2d_i^2) + \sum_{(s,t,i) \in \mathcal{I}_V} \frac{(s^{s,t,i})^2 \|F_s(\mathbf{X})\|_2^2}{\varepsilon_{s,t,i}^2} \log(2d_{s,t,i}^2).$$

873 Here, $F_{i-1}(\mathbf{X})$ denotes the output of the network after layer $i-1$ (the input to layer i), and d_i is
 874 the dimension at layer i .

875 *Proof.* We proceed by analysing how residual connections compose with the stem. Consider vertex
 876 $N(t)$ where one or more vines reconnect. The output is:

$$877 \quad 878 \quad F_t(\mathbf{X}) = F_t^S(\mathbf{X}) + \sum_{(s,t,i) \in \mathcal{I}_V} F_{s,t,i}^V(\mathbf{X}).$$

879 Let \mathcal{W}_t be an ε_t -cover of \mathcal{H}_t (all possible stem outputs at vertex t). For each vine $\mathcal{V}(s, t, i)$ that
 880 reconnects at t , let $\mathcal{W}_{s,t,i}^V$ be an $\varepsilon_{s,t,i}$ -cover of $\mathcal{H}_{s,t,i}^V$ (all possible outputs of that vine). By repeated
 881 application of Lemma B.4, the set:

$$882 \quad 883 \quad 884 \quad \mathcal{W}'_t = \left\{ W_S + \sum_{(s,t,i) \in \mathcal{I}_V} W_{s,t,i}^V : W_S \in \mathcal{W}_t, W_{s,t,i}^V \in \mathcal{W}_{s,t,i}^V \right\}$$

885 is an $(\varepsilon_t + \sum_{(s,t,i) \in \mathcal{I}_V} \varepsilon_{s,t,i})$ -cover of \mathcal{H}'_t (the combined outputs at vertex t), with covering number:
 886

$$887 \quad 888 \quad \mathcal{N}_{\infty,2}(\mathcal{H}'_t, \varepsilon'_t) \leq \mathcal{N}_{\infty,2}(\mathcal{H}_t, \varepsilon_t) \cdot \prod_{(s,t,i) \in \mathcal{I}_V} \mathcal{N}_{\infty,2}(\mathcal{H}_{s,t,i}^V, \varepsilon_{s,t,i}),$$

889 where $\varepsilon'_t = \varepsilon_t + \sum_{(s,t,i) \in \mathcal{I}_V} \varepsilon_{s,t,i}$.

890 Each vine $\mathcal{V}(s, t, i)$ is itself a chain-like feedforward network, so Lemma B.3 applies to bound
 891 $\mathcal{N}_{\infty,2}(\mathcal{H}_{s,t,i}^V, \varepsilon_{s,t,i})$. For identity vines (containing no trainable parameters), we have $\mathcal{N}_{s,t,i}^V = 1$
 892 since there is only one function in the class.

893 Propagating this argument through all K stem layers yields:

$$894 \quad 895 \quad 896 \quad \mathcal{N}_{\infty,2}(\mathcal{H}_{\text{res}}, \varepsilon) \leq \prod_{j=1}^K \mathcal{N}_{\infty,2}(\mathcal{H}_j, \varepsilon_j) \prod_{(s,t,i) \in \mathcal{I}_V} \mathcal{N}_{\infty,2}(\mathcal{H}_{s,t,i}^V, \varepsilon_{s,t,i}).$$

901 The bound on \mathcal{R} follows by applying Lemma B.2 to each weight matrix. For the stem, layer i
 902 contributes:

$$903 \quad 904 \quad 905 \quad \log \mathcal{N}_i \leq \frac{s_i^2 \|F_{i-1}(\mathbf{X})\|_2^2 d_i^2}{\varepsilon_i^2} \log(2d_{i-1} d_i) \approx \frac{s_i^2 \|F_{i-1}(\mathbf{X})\|_2^2}{\varepsilon_i^2} \log(2d_i^2),$$

906 where we simplify by assuming similar dimensions. Summing over all stem layers and all vine
 907 layers gives \mathcal{R} . \square

908 **Corollary B.6.** Let \mathcal{H}_{ff} be the hypothesis space of feedforward networks with the same total number
 909 of weight matrices $K_{\text{total}} = K + \sum_{(s,t,i) \in \mathcal{I}_V} K_{s,t,i}$ as ReSymNet. Then for any $\varepsilon > 0$,

$$910 \quad 911 \quad \mathcal{N}_{\infty,2}(\mathcal{H}_{\text{res}}, \varepsilon) \leq \mathcal{N}_{\infty,2}(\mathcal{H}_{\text{ff}}, \varepsilon).$$

912 *Proof.* Both covering numbers have the product form $\prod_{k=1}^{K_{\text{total}}} \mathcal{N}_k$, where each factor \mathcal{N}_k corresponds
 913 to a single weight matrix. By Lemma B.2, each \mathcal{N}_k depends only on the spectral norm s_k of that
 914 weight matrix and the norm of its input $\|F_{k-1}(\mathbf{X})\|_2$, regardless of whether the matrix appears in
 915 the stem or a vine. Therefore, when the total number of weight matrices and their norms are held
 916 fixed, the covering numbers are bounded identically. \square

918 B.3 ALGORITHM CHART
919920 **Algorithm 1:** ReSymNet with any MORL algorithm
921

922 **Input:** Release probability p_{rel} , number of initial episodes N , number of expert episodes E , dense
923 channels \mathcal{DC} , any off-the-shelf MORL algorithm, number of timesteps per cycle M , number of
924 ensembles K , number of iterative refinements IR , validation split, patience
925 **Output:** Trained reward ensemble $\mathcal{E} = \{\mathcal{R}_{\text{pred}, \psi_1}, \dots, \mathcal{R}_{\text{pred}, \psi_K}\}$, trained MORL policy
926 /* Collecting random experiences */
927 **for** $i \leftarrow 1$ **to** N **do**
928 Execute a random policy to collect trajectory $\tau = \{(s_0, a_0), \dots, (s_T, a_T)\}$ until episode ends
929 Set $l = 0$
930 **foreach** $t \in T$ **do**
931 With probability p_{rel} , release cumulative sparse reward $R_t^{sp} = \sum_{s=l}^t r_s^{sp}$ at timestep t
932 Set $l = t$ if sparse reward is released
933 Segment τ into sub-trajectories $\{\tau_j\}$ based on released rewards
934 **foreach** sub-trajectory τ_j **do**
935 **foreach** $(s_t, a_t) \in \tau_j$ **do**
936 Compute features: $\mathbf{h}_t = [s_t, a_t, \mathbf{r}_t^{\text{dense}}]$
937 Add datapoint $(\{\mathbf{h}_t\}_{t \in \tau_j}, R^{sp}(\tau_j))$ to dataset \mathcal{D}
938 /* Ensemble training */
939 **for** $k \leftarrow 1$ **to** K **do**
940 Split \mathcal{D} into $\mathcal{D}_{\text{train}}$ and \mathcal{D}_{val} using the validation split
941 Train reward model $\mathcal{R}_{\text{pred}, \psi_k}$ following Equation 1 using early stopping on the validation loss:
942
$$\mathcal{L}(\psi_k) = \sum_{\tau \in \mathcal{D}_{\text{train}}} \left(\sum_{t \in \tau} \mathcal{R}_{\text{pred}}(\mathbf{h}_t; \psi_k) - R^{sp}(\tau) \right)^2$$

943 /* RL training with iterative refinement */
944 $timestep = 1$
945 **for** $cycle \leftarrow 1$ **to** IR **do**
946 **for** $t \leftarrow timestep$ **to** $M + timestep$ **do**
947 Observe s_t and a_t following the current policy and compute features \mathbf{h}_t
948 $r_t^{(k)} \leftarrow \mathcal{R}_{\text{pred}}(\mathbf{h}_t; \psi_k)$ for $k = 1, \dots, K$
949 $r_t^{\text{sh}} \leftarrow \frac{1}{K} \sum_{k=1}^K r_t^{(k)}$
950 Use r_t^{sh} with the dense rewards as the reward at timestep t and update RL algorithm
951 /* Iterative refinement */
952 Collect E expert trajectories to obtain \mathcal{D}_{new} using the new policy
953 **foreach** $\mathcal{R}_{\text{pred}, \psi_k} \in \mathcal{E}$ **do**
954 Update $\mathcal{R}_{\text{pred}, \psi_k}$ using new data \mathcal{D}_{new}
955 $timestep = t$

957 C PROOFS
958960 This appendix collects all proofs omitted from the main text.
961962 C.1 LEMMAS
963964 This section introduces the general lemmas used to obtain an upper bound on the generalisation gap.
965966 **Dudley Entropy Integral.** The Rademacher complexity can be bounded through the metric entropy
967 of the function class using Dudley's entropy integral (Dudley, 1967; Bartlett & Mendelson, 2002).
968969 **Lemma C.1** (Dudley Entropy Integral). *For any coarse-scale parameter $r \in (0, B)$, the empirical
Rademacher complexity satisfies:*

970
$$\hat{\mathfrak{R}}_N(\mathcal{F}) \leq C \left(\int_r^B \sqrt{\frac{\log \mathcal{N}_{\infty, 1}(\mathcal{F}, r)}{N}} d\varepsilon \right) + \frac{4r}{\sqrt{N}},$$

971

972 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
 973 with respect to N samples
 974

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

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

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

$$982 \Pr \left(\left| \sup_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N (f(\tau_i) - \mathbb{E}[f]) - \mathbb{E} \left[\sup_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N (f(\tau_i) - \mathbb{E}[f]) \right] \right| \geq t \right) \leq 2 \exp \left(- \frac{2Nt^2}{B^2} \right).$$

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

989 C.2 GENERALISATION OF SCALARISED RETURNS

991 This section shows that generalisation for an arbitrary scalar return implies guarantees for the
 992 scalarised components of the Pareto front.

993 **Corollary C.3.** *Let Π be a policy class equipped with a metric $d(\cdot, \cdot)$, and let $\mathbf{R}(\pi; \tau) \in \mathbb{R}^L$ denote
 994 the vector-valued return of policy π on trajectory τ . Following Assumption 1:*

$$996 \sup_{\tau} \|\mathbf{R}(\pi; \tau) - \mathbf{R}(\tilde{\pi}; \tau)\|_{\infty} \leq L_R d(\pi, \tilde{\pi}) \quad \text{for all } \pi, \tilde{\pi} \in \Pi.$$

997 *For any weight vector $\omega \in \mathbb{R}^L$ define the scalarised return $R^\omega(\pi; \tau) = \omega^\top \mathbf{R}(\pi; \tau)$ and let \mathcal{R}_Π^ω be
 998 the class of scalarised returns induced by Π . Then for any $\varepsilon > 0$,*

$$1001 \mathcal{N}_{\infty,1}(\mathcal{R}_\Pi^\omega, \varepsilon) \leq \mathcal{N}_{\infty,1}(\Pi, \varepsilon/L_R^\omega), \quad \text{where } L_R^\omega := \|\omega\|_1 L_R.$$

1002 *In particular, when $\|\omega\|_1 = 1$ we have $L_R^\omega = L_R$ and the scalarised return class has covering
 1003 numbers no larger than those of the policy class. Consequently, any complexity reduction obtained
 1004 by projecting Π to an equivariant subspace (e.g. Π_{eq}) is inherited by the scalarised objective class
 1005 \mathcal{R}_Π^ω .*

1006 *Proof.* Fix $\omega \in \mathbb{R}^L$ and let $\pi, \tilde{\pi} \in \Pi$. For any trajectory τ ,

$$1009 |R^\omega(\pi; \tau) - R^\omega(\tilde{\pi}; \tau)| = |\omega^\top (\mathbf{R}(\pi; \tau) - \mathbf{R}(\tilde{\pi}; \tau))| \leq \sum_{j=1}^L |\omega_j| |R_j(\pi; \tau) - R_j(\tilde{\pi}; \tau)|.$$

1013 Using $\max_j |R_j(\pi; \tau) - R_j(\tilde{\pi}; \tau)| = \|\mathbf{R}(\pi; \tau) - \mathbf{R}(\tilde{\pi}; \tau)\|_{\infty}$, we obtain

$$1014 |R^\omega(\pi; \tau) - R^\omega(\tilde{\pi}; \tau)| \leq \|\omega\|_1 \|\mathbf{R}(\pi; \tau) - \mathbf{R}(\tilde{\pi}; \tau)\|_{\infty}.$$

1016 Taking the supremum over trajectories and applying the vector Lipschitz assumption yields

$$1017 \sup_{\tau} |R^\omega(\pi; \tau) - R^\omega(\tilde{\pi}; \tau)| \leq \|\omega\|_1 L_R d(\pi, \tilde{\pi}) = L_R^\omega d(\pi, \tilde{\pi}).$$

1019 Thus the scalarised return map $\pi \mapsto R^\omega(\pi; \cdot)$ is Lipschitz with constant $L_R^\omega = \|\omega\|_1 L_R$. Following
 1020 Lemma C.8, for any $\varepsilon > 0$,

$$1022 \mathcal{N}_{\infty,1}(\mathcal{R}_\Pi^\omega, \varepsilon) \leq \mathcal{N}_{\infty,1}(\Pi, \varepsilon/L_R^\omega).$$

1023 This proves the displayed inequality. The special case $\|\omega\|_1 = 1$ follows immediately. Finally, since
 1024 the inequality holds for any policy class Π , replacing Π by the equivariant subspace Π_{eq} shows
 1025 that any complexity reduction ($\mathcal{N}(\Pi_{eq}, \cdot)$) smaller than is directly inherited by the scalarised return
 1026 class. \square

1026 C.3 PROJECTION TO REFLECTION-EQUIVARIANT SUBSPACE
1027

1028 Let the full hypothesis space of policies be $\Pi = \{\pi_\phi : \phi \in \Phi\}$, where ϕ represents the neural
1029 network parameters [and \$\Phi\$ represents the parameter space](#). The reflection group $G = \mathbb{Z}_2 = \{e, g\}$
1030 acts on the state and action spaces via operators L_g and K_g , respectively.

1031 We can map any policy to its equivariant counterpart using an orbit averaging operator $\mathcal{Q} : \Pi \rightarrow \Pi$,
1032 defined as:

$$\begin{aligned} 1034 \quad \mathcal{Q}(\pi_\phi)(s) &= \frac{1}{|G|} \sum_{h \in G} \rho(h) \pi_\phi(h^{-1} \cdot s) \\ 1035 \\ 1036 \quad &= \frac{1}{|G|} \sum_{h \in G} K_h(\pi_\phi(L_h(s))) \\ 1037 \\ 1038 \quad &= \frac{1}{2} (\pi_\phi(s) + K_g(\pi_\phi(L_g(s)))) . \end{aligned} \quad (6)$$

1041 Here, $\rho(h)$ is the abstract representation in the action space, and $h^{-1} \cdot s$ is the abstract action in
1042 the state space. In the second line we replace $\rho(h)$ with the action transformation K_h , and $h^{-1} \cdot s$
1043 with the state transformation $L_h(s)$. For the reflection group $G = \mathbb{Z}_2 = \{e, g\}$, since $g = g^{-1}$ we
1044 may drop the inverse without ambiguity. This operator averages a policy's output with its reflected-
1045 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
1046 to become fixed points of this operator, thereby learning policies within the subspace of equivariant
1047 functions, denoted Π_{eq} .

1048 The operator \mathcal{Q} and the subspace Π_{eq} have several crucial properties, which we state in the following
1049 lemmas.

1050 **Lemma C.4.** *For any $\pi \in \Pi$, the function $\mathcal{Q}(\pi)$ is reflectional equivariant:*

$$1052 \quad \mathcal{Q}(\pi)(L_g(s)) = K_g(\mathcal{Q}(\pi)(s)), \quad \forall s \in \mathcal{S}.$$

1055 *Proof.* By direct calculation:

$$\begin{aligned} 1057 \quad \mathcal{Q}(\pi)(L_g(s)) &= \frac{1}{2} (\pi(L_g(s)) + K_g(\pi(L_g(s)))) \\ 1058 \\ 1059 \quad &= \frac{1}{2} (\pi(L_g(s)) + K_g(\pi(s))), \\ 1060 \quad K_g \mathcal{Q}(\pi)(s) &= \frac{1}{2} (K_g(\pi(s)) + K_g K_g(\pi(L_g(s)))) \\ 1061 \\ 1062 \quad &= \frac{1}{2} (K_g(\pi(s)) + \pi(L_g(s))), \end{aligned}$$

1063 since K_g and L_g are involutions. Thus, the two expressions coincide. Therefore $\mathcal{Q}(\pi)$ is equivariant. \square

1066 **Lemma C.5.** *The operator \mathcal{Q} is a projection, meaning it is idempotent: $\mathcal{Q}(\mathcal{Q}(\pi)) = \mathcal{Q}(\pi)$ for any
1067 $\pi \in \Pi$.*

1070 *Proof.* We apply the operator to its own output:

$$1072 \quad \mathcal{Q}(\mathcal{Q}(\pi))(s) = \frac{1}{2} (\mathcal{Q}(\pi)(s) + K_g(\mathcal{Q}(\pi)(L_g(s)))) .$$

1075 First, evaluating the second term, $\mathcal{Q}(\pi)(L_g(s))$:

$$\begin{aligned} 1076 \quad \mathcal{Q}(\pi)(L_g(s)) &= \frac{1}{2} (\pi(L_g(s)) + K_g(\pi(L_g(s)))) \\ 1077 \\ 1078 \quad &= \frac{1}{2} (\pi(L_g(s)) + K_g(\pi(s))) . \end{aligned}$$

1080 Substituting this back:
 1081

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

1094 Thus \mathcal{Q} is idempotent. □
 1095

1096 **Lemma C.6.** *The image of the operator \mathcal{Q} coincides with the set of equivariant policies: $\text{Im}(\mathcal{Q}) = \{ \mathcal{Q}(\pi) : \pi \in \Pi \} = \Pi_{\text{eq}}$.*

1100 *Proof.* We establish set equality by showing inclusion in both directions.

1102 First inclusion ($\text{Im}(\mathcal{Q}) \subseteq \Pi_{\text{eq}}$): By Lemma C.4, for any $\pi \in \Pi$, the output $\mathcal{Q}(\pi)$ is equivariant.
 1103 Therefore, every element in the image of \mathcal{Q} belongs to Π_{eq} .

1104 Second inclusion ($\Pi_{\text{eq}} \subseteq \text{Im}(\mathcal{Q})$): Let π_{eq} be any equivariant policy, so $\pi_{\text{eq}} \in \Pi_{\text{eq}}$. We need to show
 1105 that π_{eq} can be expressed as $\mathcal{Q}(\pi)$ for some $\pi \in \Pi$.

1106 Since π_{eq} is equivariant, it satisfies $\pi_{\text{eq}}(L_g(s)) = K_g(\pi_{\text{eq}}(s))$ for all s . Therefore:

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

1116 Therefore, $\pi_{\text{eq}} = \mathcal{Q}(\pi_{\text{eq}}) \in \text{Im}(\mathcal{Q})$. This shows that equivariant policies are fixed points of \mathcal{Q} ,
 1117 which is consistent with Lemma C.5. Since every equivariant policy is its own image under \mathcal{Q} , we
 1118 have $\Pi_{\text{eq}} \subseteq \text{Im}(\mathcal{Q})$. Combining both inclusions yields $\text{Im}(\mathcal{Q}) = \Pi_{\text{eq}}$. Therefore \mathcal{Q} is surjective
 1119 onto Π_{eq} . □

1122 C.4 REDUCED HYPOTHESIS COMPLEXITY OF REFLECTION-EQUIVARIANT SUBSPACE

1124 To prove that the subspace Π_{eq} is less complex, we show that the projection \mathcal{Q} is non-expansive,
 1125 which implies its image has a covering number no larger than the original space.

1126 **Theorem C.7.** *The space Π_{eq} has a covering number less than or equal to that of Π . Let $\mathcal{N}_{\infty,1}(\mathcal{F}, r)$
 1127 be the covering number of a function space \mathcal{F} under the $l_{\infty,1}$ -distance. Then, $\mathcal{N}_{\infty,1}(\Pi_{\text{eq}}, r) \leq$
 1128 $\mathcal{N}_{\infty,1}(\Pi, r)$.*

1131 *Proof.* We show that \mathcal{Q} is non-expansive. The $l_{\infty,1}$ -distance between two policies π_ϕ and π_θ is

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

1134

The distance between their projections is:

$$\begin{aligned}
 1136 \quad d(\mathcal{Q}(\pi_\phi), \mathcal{Q}(\pi_\theta)) &= \sup_s \left\| \frac{1}{2}(\pi_\phi(s) + K_g(\pi_\phi(L_g(s)))) - \frac{1}{2}(\pi_\theta(s) + K_g(\pi_\theta(L_g(s)))) \right\|_1 \\
 1137 \quad &= \frac{1}{2} \sup_s \|(\pi_\phi(s) - \pi_\theta(s)) + K_g(\pi_\phi(L_g(s)) - \pi_\theta(L_g(s)))\|_1. \\
 1138 \quad &\leq \frac{1}{2} \sup_s \left(\|\pi_\phi(s) - \pi_\theta(s)\|_1 + \|K_g(\pi_\phi(L_g(s)) - \pi_\theta(L_g(s)))\|_1 \right). \\
 1139 \quad &\leq \frac{1}{2} \left(\sup_s \|\pi_\phi(s) - \pi_\theta(s)\|_1 + \sup_s \|\pi_\phi(L_g(s)) - \pi_\theta(L_g(s))\|_1 \right). \\
 1140 \quad &\leq \frac{1}{2} (d(\pi_\phi, \pi_\theta) + d(\pi_\phi, \pi_\theta)) = d(\pi_\phi, \pi_\theta),
 \end{aligned}$$

1141 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.6, $\mathcal{N}(\Pi_{\text{eq}}, r) \leq \mathcal{N}(\Pi, r)$. \square

1142

1143 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.

1144 **Lemma C.8.** *For any policy set $\mathcal{P} \subseteq \Pi$ and any $\varepsilon > 0$,*

$$1145 \quad \mathcal{N}_{\infty,1}(\{\tau \mapsto R(\pi; \tau) : \pi \in \mathcal{P}\}, \varepsilon) \leq \mathcal{N}_{\infty,1}(\mathcal{P}, \varepsilon/L_R),$$

1146 where the left covering number is with respect to the sup-norm over trajectories and the right is with respect to $d(\cdot, \cdot)$. \square

1147 *Proof.* Let $\{\pi_1, \dots, \pi_M\}$ 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 ,

$$1148 \quad |R(\pi; \tau) - R(\pi_j; \tau)| \leq L_R d(\pi, \pi_j) \leq \varepsilon,$$

1149 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. \square

1150

C.5 GENERALISATION OF REFLECTION-EQUIVARIANT SUBSPACE

1151

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

1152 **Theorem C.9.** *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 $1 - \delta$,*

$$1153 \quad \sup_{\pi \in \Pi_{\text{eq}}} |J(\pi) - \hat{J}_N(\pi)| \leq C \left(\int_r^B \sqrt{\frac{\log \mathcal{N}_{\infty,1}(\mathcal{R}_{\Pi_{\text{eq}}}, \varepsilon)}{N}} d\varepsilon \right) + \frac{8r}{\sqrt{N}} + B \sqrt{\frac{\log(2/\delta)}{2N}},$$

1154

1155 where C is an absolute numeric constant, $J(\pi)$ 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 τ_1, \dots, τ_N . \square

1156

1157 *Proof.* Let $\mathcal{F} = \mathcal{R}_{\Pi_{\text{eq}}}$. Following Corollary 5.2, we have:

$$1158 \quad \mathbb{E} \left[\sup_{f \in \mathcal{R}_{\Pi_{\text{eq}}}} \left| \frac{1}{N} \sum_{i=1}^N (f(\tau_i) - \mathbb{E}[f]) \right| \right] \leq 2\mathbb{E}[\mathfrak{R}_N(\mathcal{R}_{\Pi_{\text{eq}}})].$$

1159

1160 Applying Lemma C.1, for any $r > 0$:

$$1161 \quad \mathbb{E} \left[\sup_{f \in \mathcal{R}_{\Pi_{\text{eq}}}} \left| \frac{1}{N} \sum_{i=1}^N (f(\tau_i) - \mathbb{E}[f]) \right| \right] \leq C \left(\int_r^B \sqrt{\frac{\log \mathcal{N}_{\infty,1}(\mathcal{R}_{\Pi_{\text{eq}}}, \varepsilon)}{N}} d\varepsilon \right) + \frac{8r}{\sqrt{N}}. \quad (7)$$

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

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

Combining Equations 7 and 8 yields the claimed inequality. \square

C.6 GENERALISATABILITY OF PRISM

Lemma C.10. *If a policy π satisfies $\mathcal{L}_{\text{eq}} \leq \varepsilon_{\text{eq}}$, then*

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

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

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

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

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

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

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

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

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

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

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

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

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

Theorem C.11. *Let $\xi := \frac{1}{2} \sqrt{\varepsilon_{\text{eq}} / p_{\min}}$. Then for every policy π ,*

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

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

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

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

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:

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

1242 Next, we bound the distance $d(\pi, Q(\pi))$. Using the definition of the projection $Q(\pi)$, we find the
 1243 distance from π to its projection:

$$\begin{aligned} 1244 \quad d(\pi, Q(\pi)) &= \sup_s \|\pi(s) - Q(\pi)(s)\|_1 \\ 1245 \quad &= \sup_s \|\pi(s) - \frac{1}{2}(\pi(s) + K_g(\pi(L_g(s))))\|_1 \\ 1246 \quad &= \frac{1}{2} \sup_s \|\pi(s) - K_g(\pi(L_g(s)))\|_1. \\ 1247 \quad & \\ 1248 \quad & \\ 1249 \quad & \end{aligned}$$

1250 The term inside the norm is equal to the equivariance mismatch $\Delta_\pi(s') := \pi(L_g(s')) - K_g(\pi(s'))$
 1251 evaluated at $s' = L_g(s)$, since L_g is an involution.

$$1252 \quad \Delta_\pi(L_g(s)) = \pi(L_g(L_g(s))) - K_g(\pi(L_g(s))) = \pi(s) - K_g(\pi(L_g(s))). \\ 1253$$

1254 Since L_g is a bijection, $\sup_s \|\Delta_\pi(L_g(s))\|_1 = \sup_{s'} \|\Delta_\pi(s')\|_1$. By Lemma C.10, this supremum
 1255 is bounded by ξ . Therefore:

$$1256 \quad d(\pi, Q(\pi)) = \frac{1}{2} \sup_{s'} \|\Delta_\pi(s')\|_1 \leq \xi. \\ 1257$$

1258 The second claim is that for any radius $r > \xi$, we have $N_{\infty,1}(\Pi_{approx}(\varepsilon_{eq}), r) \leq N_{\infty,1}(\Pi_{eq}, r - \xi)$.
 1259 We know that for any $\pi \in \Pi_{approx}(\varepsilon_{eq})$, its projection $Q(\pi) \in \Pi_{eq}$ satisfies $d(\pi, Q(\pi)) \leq \xi$.
 1260 This implies that the set $\Pi_{approx}(\varepsilon_{eq})$ is contained in a ξ -neighbourhood of Π_{eq} . Let $\{\pi_j\}_{j=1}^M$
 1261 be a minimal $(r - \xi)$ -cover for Π_{eq} , where $M = N_{\infty,1}(\Pi_{eq}, r - \xi)$. Now, consider any policy
 1262 $\pi \in \Pi_{approx}(\varepsilon_{eq})$. There must exist a centre π_j from our cover such that $d(Q(\pi), \pi_j) \leq r - \xi$. By
 1263 the triangle inequality, we can bound the distance from π to this centre π_j :

$$1265 \quad d(\pi, \pi_j) \leq d(\pi, Q(\pi)) + d(Q(\pi), \pi_j) \\ 1266 \quad \leq \xi + (r - \xi) = r.$$

1267 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
 1268 of size M , the size of the minimal cover must be no larger:

$$1269 \quad N_{\infty,1}(\Pi_{approx}(\varepsilon_{eq}), r) \leq N_{\infty,1}(\Pi_{eq}, r - \xi).$$

□

1270 **Theorem C.12.** *With $\mathcal{R}_{\Pi_{eq}} = \{\tau \mapsto R(\pi; \tau) : \pi \in \Pi_{eq}\}$, fix any accuracy parameter $r \in (0, B)$
 1271 and confidence $\delta \in (0, 1)$. Then with probability at least $1 - \delta$,*

$$1272 \quad \sup_{\pi \in \Pi_{approx}(\varepsilon_{eq})} |J(\pi) - \hat{J}_N(\pi)| \leq C \left(\int_r^B \sqrt{\frac{\log N_{\infty,1}(\mathcal{R}_{\Pi_{eq}}, \varepsilon)}{N}} d\varepsilon \right) + \frac{8r}{\sqrt{N}} + B \sqrt{\frac{\log(2/\delta)}{2N}} + 2L_R \xi.$$

1273 *Proof.* For any policy $\pi \in \Pi_{approx}(\varepsilon_{eq})$, we can decompose the generalisation error using the
 1274 triangle inequality by introducing its exact-equivariant projection $Q(\pi) \in \Pi_{eq}$:

$$1275 \quad |J(\pi) - \hat{J}_N(\pi)| \leq |J(\pi) - J(Q(\pi))| + |J(Q(\pi)) - \hat{J}_N(Q(\pi))| + |\hat{J}_N(Q(\pi)) - \hat{J}_N(\pi)|.$$

1276 We bound each of the three terms on the right-hand side.

1277 From Theorem C.11, we have:

$$1278 \quad |J(\pi) - J(Q(\pi))| \leq L_R \cdot d(\pi, Q(\pi)) \leq L_R \xi.$$

1279 Since the return function $R(\cdot; \tau)$ is L_R -Lipschitz:

$$\begin{aligned} 1280 \quad |\hat{J}_N(Q(\pi)) - \hat{J}_N(\pi)| &= \left| \frac{1}{N} \sum_{i=1}^N (R(Q(\pi); \tau_i) - R(\pi; \tau_i)) \right| \\ 1281 \quad &\leq \frac{1}{N} \sum_{i=1}^N |R(Q(\pi); \tau_i) - R(\pi; \tau_i)| \\ 1282 \quad &\leq \frac{1}{N} \sum_{i=1}^N L_R \cdot d(\pi, Q(\pi)) \leq L_R \xi. \\ 1283 \quad & \\ 1284 \quad & \\ 1285 \quad & \\ 1286 \quad & \\ 1287 \quad & \\ 1288 \quad & \\ 1289 \quad & \\ 1290 \quad & \\ 1291 \quad & \\ 1292 \quad & \\ 1293 \quad & \\ 1294 \quad & \\ 1295 \quad & \end{aligned}$$

1296 The middle term, $|J(Q(\pi)) - \hat{J}_N(Q(\pi))|$, is the generalisation error for an exactly equivariant
 1297 policy. Combining the bounds, we get:
 1298

$$1299 \sup_{\pi \in \Pi_{approx}(\varepsilon_{eq})} |J(\pi) - \hat{J}_N(\pi)| \leq \sup_{\pi' \in \Pi_{eq}} |J(\pi') - \hat{J}_N(\pi')| + L_R \gamma. \\ 1300$$

1301 Applying the high-probability bound from Theorem C.9 to the supremum over Π_{eq} yields the final
 1302 result. \square
 1303

1304 D ADDITIONAL DETAILS OF ENVIRONMENTS 1305

1306 This appendix presents the tables on the environments and how the state space is divided into a
 1307 symmetric and an asymmetric part. First Table 3 highlights the differences between environments
 1308 in dimension sizes. Tables 4, 5, 6, and 7 show the division for mo-hopper-v5, mo-walker2d-v5,
 1309 mo-halfcheetah-v5, and mo-swimmer-v5, respectively. The action space is always divided into an
 1310 empty set for the asymmetric part, and the complete set for the symmetric part.
 1311

1312 Table 3: Considered MuJoCo environments.
 1313

	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	$\mathcal{S} \in \mathbb{R}^8$	$\mathcal{A} \in \mathbb{R}^2$	$\mathcal{R} \in \mathbb{R}^2$

1320 Table 4: Reflectional symmetry partition for mo-hopper-v5 observation space.
 1321

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

1404 E ADDITIONAL DETAILS OF EXPERIMENTAL SETTINGS

1405
 1406 **Evaluation Measures.** For the approximated Pareto front, we consider three well-known metrics
 1407 that investigate the extent of the approximated front.

1408 First, we consider hypervolume (HV) (Fonseca et al., 2006), which measures the volume of the ob-
 1409 jective space dominated by the approximated Pareto front relative to a reference point. A downside
 1410 of many evaluation measures is that they require domain knowledge about the true underlying Pareto
 1411 front, whereas HV only considers a reference point without any a priori knowledge, making it ideal
 1412 to assess the volume of the front. The reference point is typically set to the nadir point or slightly
 1413 worse, and following Felten et al. (2023), we set it to -100 for all objectives and environments. The
 1414 HV is defined as follows:

$$1415 \quad 1416 \quad 1417 \quad 1418 \quad HV(CS, \mathbf{r}) = \lambda \left(\bigcup_{\mathbf{cs} \in CS} \mathbf{x} \in \mathbb{R}^L : \mathbf{cs} \preceq \mathbf{x} \preceq \mathbf{r} \right),$$

1419 where $CS = \mathbf{cs}_1, \mathbf{cs}_2, \dots, \mathbf{cs}_n$ is the coverage set, or the Pareto front approximation, $\mathbf{r} \in \mathbb{R}^L$ is
 1420 the reference point, $\mathbf{cs} \preceq \mathbf{x}$ means $cs_i \leq x_i$ for all objectives $i = 1, \dots, L$, and $\lambda(\cdot)$ denotes the
 1421 Lebesgue measure. Yet, hypervolume values are difficult to interpret, as they do not have a direct
 1422 link to any notion of value or utility (Hayes et al., 2022).

1423 As such, we also consider the Expected Utility Metric (EUM) (Zintgraf et al., 2015), which com-
 1424 putes the expected maximum utility across different preference weight vectors, and is defined as
 1425 follows:

$$1426 \quad 1427 \quad EUM(CS, \mathcal{W}) = \frac{1}{|\mathcal{W}|} \sum_{\omega \in \mathcal{W}} \max_{\mathbf{cs} \in CS} U(\omega, \mathbf{cs}),$$

1428 where $\mathcal{W} = \{\omega_1, \omega_2, \dots, \omega_k\}$ is a set of weight vectors, $|\mathcal{W}|$ is the cardinality of the weight set,
 1429 $U(\omega, \mathbf{cs})$ is the utility function, which is set to $U(\omega, \mathbf{s}) = \omega \cdot \mathbf{cs} = \sum_{i=1}^L \omega_i \cdot cs_i$.

1430 To specifically assess performance with respect to distributional preferences, we also consider one
 1431 metric designed to evaluate the optimality of the entire return distribution associated with the learned
 1432 policies (Cai et al., 2023).

1433 To be precise, we consider the Variance Objective (VO), which evaluates how well the policy set
 1434 can balance the trade-off between maximising expected returns and minimising their variance. A
 1435 set of M random preference vectors is generated, where each vector specifies a different weighting
 1436 between the expected return and its standard deviation for each objective. The satisfaction score
 1437 $u(p_i, \pi_j)$ for a policy π_j under preference p_i is a weighted sum of the expected return $\mathbb{E}[Z(\pi_j)]$
 1438 and the negative standard deviation $-\sqrt{\text{Var}[Z(\pi_j)]}$. The final metric is the mean score over these
 1439 preferences, rewarding policies that achieve high expected returns with low variance:

$$1441 \quad 1442 \quad 1443 \quad \text{VO}(\Pi, \{p_i\}_{i=1}^M) = \frac{1}{M} \sum_{i=1}^M \max_{\pi_j \in \Pi} u(p_i, \pi_j).$$

1444 **Hyperparameters.** Due to time, computational limitations, and the excessive number of hyper-
 1445 parameters, we do not perform an extensive hyperparameter tuning process. Below are the used
 1446 hyperparameters. All hyperparameters that are not mentioned below are set to their default value.

1447 The probability of releasing sparse rewards p_{rel} is always set to a one-hot vector, where sparsity
 1448 is imposed on the reward dimension related to moving forward. Since the main goal is to move
 1449 forward, imposing sparsity on this channel should make it a more difficult task for the reward shaping
 1450 model. Furthermore, we deal with extreme heterogeneous sparsity, where most channels exhibit
 1451 regular rewards, but one channel only releases a reward at the end of an episode, making it more
 1452 difficult for the model to link certain states and actions to the observed cumulative reward.

1453 The hyperparameters in Table 8 for ReSymNet are identical for each environment. The advantage of
 1454 using the same hyperparameters for each environment is that if one configuration performs well ev-
 1455 erywhere, it could indicate that the proposed method is inherently stable, especially given the noted
 1456 diversity between the considered environments. However, this does come at a cost of potentially
 1457 suboptimal performance per environment.

1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
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 $ \mathcal{E} $	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.

1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
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 PARETO FRONTS

Figure 6 shows the approximated Pareto fronts. The results demonstrate that shaped rewards yield superior performance, covering a wider and more optimal range of the objective space compared to dense and sparse rewards.

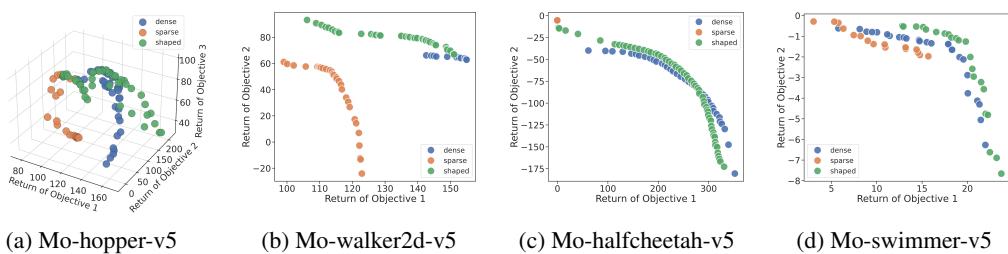


Figure 6: The approximated Pareto front for dense rewards (blue dots), sparse rewards (orange dots), and shaped rewards (green dots). Sparsity is imposed on the first reward objective.

G ABLATION STUDY

Tables 10 and 11 report the obtained values for the ablation study. Results are again averaged over ten trials, similar to the main experiments.

1512 Table 10: PRISM ablation study results. We report the average hypervolume (HV), Expected Utility Metric (EUM), and Variance
 1513 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
 1514 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 \pm 0.05	1.29 \pm 0.09	1.38 \pm 0.11	1.38 \pm 0.08	1.55 \pm 0.04	1.42 \pm 0.07
	EUM	147.43 \pm 2.61	128.40 \pm 6.06	134.67 \pm 6.89	135.28 \pm 4.91	145.89 \pm 2.73	137.85 \pm 4.22
	VO	66.66 \pm 1.40	58.61 \pm 2.71	61.21 \pm 3.03	61.51 \pm 2.19	66.54 \pm 1.34	62.71 \pm 1.83
Mo-walker2d-v5	HV ($\times 10^4$)	4.77 \pm 0.07	4.65 \pm 0.11	4.66 \pm 0.06	4.60 \pm 0.08	4.60 \pm 0.09	4.58 \pm 0.13
	EUM	120.43 \pm 1.64	114.33 \pm 2.48	116.83 \pm 1.65	113.79 \pm 2.02	114.98 \pm 2.84	112.77 \pm 3.01
	VO	59.35 \pm 0.80	56.46 \pm 1.21	57.67 \pm 0.73	56.19 \pm 0.97	57.03 \pm 1.42	55.59 \pm 1.44
Mo-halfcheetah-v5	HV ($\times 10^4$)	2.25 \pm 0.18	1.95 \pm 0.20	2.08 \pm 0.21	1.91 \pm 0.19	2.23 \pm 0.18	1.90 \pm 0.19
	EUM	89.94 \pm 15.33	73.06 \pm 16.57	82.24 \pm 16.97	81.60 \pm 17.65	92.68 \pm 14.79	71.12 \pm 16.91
	VO	40.72 \pm 7.02	32.99 \pm 7.65	37.31 \pm 7.99	36.76 \pm 8.06	42.28 \pm 6.85	32.12 \pm 7.75
Mo-swimmer-v5	HV ($\times 10^4$)	1.21 \pm 0.00	1.21 \pm 0.00	1.20 \pm 0.00	1.20 \pm 0.00	1.21 \pm 0.00	1.20 \pm 0.00
	EUM	9.44 \pm 0.14	9.39 \pm 0.15	9.07 \pm 0.11	9.25 \pm 0.13	9.46 \pm 0.13	9.35 \pm 0.14
	VO	4.24 \pm 0.07	4.20 \pm 0.08	4.09 \pm 0.05	4.15 \pm 0.08	4.24 \pm 0.07	4.24 \pm 0.07

1525
 1526 Table 11: ReSymNet ablation study results. We report the average hypervolume (HV), Expected Utility Metric (EUM), and
 1527 Variance Objective (VO) over 10 trials, with the standard error shown in grey. w/o is the abbreviation of without.
 1528
 1529

Environment	Metric	uniform	random
Mo-hopper-v5	HV ($\times 10^7$)	1.38 \pm 0.08	0.49 \pm 0.06
	EUM	135.19 \pm 5.30	65.22 \pm 6.63
	VO	63.90 \pm 2.34	29.62 \pm 3.68
Mo-walker2d-v5	HV ($\times 10^4$)	4.67 \pm 0.07	1.18 \pm 0.10
	EUM	116.72 \pm 2.11	16.52 \pm 4.98
	VO	56.22 \pm 1.01	3.77 \pm 2.46
Mo-halfcheetah-v5	HV ($\times 10^4$)	0.98 \pm 0.00	0.78 \pm 0.05
	EUM	-1.34 \pm 0.39	-10.52 \pm 2.67
	VO	-0.85 \pm 0.20	-6.51 \pm 1.48
Mo-swimmer-v5	HV ($\times 10^4$)	1.09 \pm 0.01	1.10 \pm 0.02
	EUM	4.37 \pm 0.69	3.75 \pm 0.87
	VO	1.56 \pm 0.33	1.06 \pm 0.40

H GENERALISABILITY

H.1 SPARSITY ON OTHER OBJECTIVES

1550 We further investigate the robustness of PRISM by inverting the sparsity setting: we maintain the
 1551 forward velocity reward as dense but make the control cost objective sparse. Table 12 shows that,
 1552 without hyperparameter tuning, PRISM handles this problem much better than the baselines.

1553 For mo-hopper-v5, PRISM improves HV by 16% over the oracle (1.51×10^7 compared to 1.30×10^7)
 1554 and 27% over the baseline. Similar gains are observed for mo-walker2d-v5, where PRISM achieves
 1555 a 9% HV improvement over the oracle and 45% over the baseline. Notably, in mo-halfcheetah-v5,
 1556 the baseline suffers a collapse (HV of 0.00), whereas PRISM recovers the performance to exceed
 1557 the oracle (1.72×10^4 against 1.70×10^4). These improvements imply that PRISM effectively
 1558 reconstructs the dense penalty signal, preventing the agent from exploiting the delay to maximise
 1559 velocity at the cost of extreme energy inefficiency.

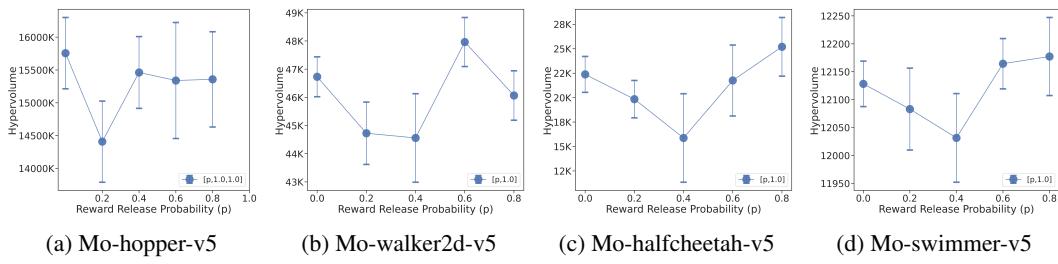
1560 Improvements in EUM follow the same trend, with mo-walker2d-v5 showing an increase of roughly
 1561 33% compared to the baseline (114.62 vs 85.95). On distributional metrics, PRISM delivers more
 1562 consistent performance than the baseline. In mo-swimmer-v5, the baseline's VO drops to -0.61 ,
 1563 indicating high instability, whereas PRISM achieves 3.95, comparable to the oracle (4.22). These
 1564 gains are crucial because they indicate that PRISM produces Pareto fronts that are not only high-
 1565 performing but also balanced and robust, effectively mitigating the high-variance behaviour from
 1566 the baseline.

1566
 1567
 1568
 1569
 1570
 1571 Table 12: Experimental results on the control cost objective. We report the average
 1572 hypervolume (HV), Expected Utility Metric (EUM), and Variance Objective (VO)
 1573 over 10 trials, with the standard error shown in grey. The largest (best) values are
 1574 in bold font.
 1575

Environment	Metric	Oracle	Baseline	PRISM
Mo-hopper-v5	HV ($\times 10^7$)	1.30 ± 0.13	1.19 ± 0.10	1.51 ± 0.11
	EUM	129.04 ± 7.96	124.82 ± 7.21	142.89 ± 7.38
	VO	59.07 ± 3.45	56.21 ± 3.20	67.58 ± 3.31
Mo-walker2d-v5	HV ($\times 10^4$)	4.21 ± 0.11	3.16 ± 0.13	4.59 ± 0.14
	EUM	107.58 ± 2.86	85.95 ± 3.27	114.62 ± 2.80
	VO	53.22 ± 1.39	41.29 ± 1.49	54.84 ± 1.25
Mo-halfcheetah-v5	HV ($\times 10^4$)	1.70 ± 0.20	0.00 ± 0.00	1.72 ± 0.19
	EUM	81.29 ± 21.85	-101.49 ± 3.23	76.50 ± 20.85
	VO	36.84 ± 10.06	-56.26 ± 1.63	31.27 ± 8.68
Mo-swimmer-v5	HV ($\times 10^4$)	1.21 ± 0.00	1.05 ± 0.02	1.21 ± 0.01
	EUM	9.41 ± 0.12	1.50 ± 1.00	9.32 ± 0.19
	VO	4.22 ± 0.08	-0.61 ± 0.68	3.95 ± 0.08

H.2 SENSITIVITY TO SPARSITY

1588
 1589 Figure 7 demonstrates that PRISM maintains robust performance across varying levels of reward
 1590 sparsity. While performance is generally consistent, we observe minor fluctuations at inter-
 1591 mediate values (e.g., $p_{rel} = 0.2$ in mo-hopper-v5 and mo-walker2d-v5). Two key factors explain
 1592 this behaviour: (1) PRISM was hyperparameter-tuned specifically for the extreme sparsity setting
 1593 ($p_{rel} = 0$), which is the most challenging MORL scenario. We utilised a fixed set of hyperpara-
 1594 meters across all experiments to demonstrate method stability rather than optimising for each sparsity
 1595 level, and (2) increasing p_{rel} increases the number of available reward signals (data points) per
 1596 episode. Since ReSymNet was calibrated for the data-scarce sparse setting, the increase of super-
 1597 vision targets at higher p_{rel} levels changes optimisation dynamics, leading to temporary instability.
 1598 Despite these factors, PRISM consistently recovers high performance, proving its capability to han-
 1599 dle heterogeneous reward structures without requiring specific tuning for denser environments.



1608 Figure 7: The obtained hypervolume for various levels of sparsity for PRISM.
 1609
 1610

H.3 SENSITIVITY TO MORL ALGORITHMS

1613 To demonstrate that PRISM is a model-agnostic framework not limited to specific architectures, we
 1614 evaluated its performance using GPI-PD (Generalised Policy Improvement with Linear Dynamics)
 1615 (Alegre et al., 2023) as an alternative backbone to CAPQL. Table 13 confirms that PRISM remains
 1616 highly effective, consistently outperforming the sparse baseline and obtaining near-oracle perfor-
 1617 mance.

1618 In mo-hopper-v5, PRISM achieves an HV of 1.65×10^7 , matching the oracle exactly and far ex-
 1619 ceeding the baseline (0.67×10^7). This trend of near-perfect recovery is consistent across mo-
 walker2d-v5 and mo-swimmer-v5. This indicates that the shaped rewards generated by ReSymNet

1620
1621
1622
1623
1624

Table 13: Experimental results of GPI-PD. 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.

1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640

Environment	Metric	Oracle	Baseline	PRISM
Mo-hopper-v5	HV ($\times 10^7$)	1.65 ± 0.10	0.67 ± 0.04	1.65 ± 0.07
	EUM	151.45 ± 5.87	85.87 ± 3.17	148.19 ± 4.26
	VO	72.26 ± 2.90	41.21 ± 1.44	70.24 ± 2.51
Mo-walker2d-v5	HV ($\times 10^4$)	5.93 ± 0.10	3.20 ± 0.23	5.61 ± 0.10
	EUM	141.88 ± 2.38	76.41 ± 6.47	132.67 ± 2.26
	VO	67.63 ± 1.17	35.64 ± 3.91	63.19 ± 1.75
Mo-halfcheetah-v5	HV ($\times 10^4$)	1.80 ± 0.22	1.00 ± 0.02	2.24 ± 0.16
	EUM	164.75 ± 14.21	-1.31 ± 0.54	99.89 ± 8.06
	VO	73.90 ± 7.05	-1.14 ± 0.31	40.74 ± 5.17
Mo-swimmer-v5	HV ($\times 10^4$)	1.23 ± 0.01	1.12 ± 0.01	1.22 ± 0.00
	EUM	9.68 ± 0.17	5.17 ± 0.58	9.56 ± 0.13
	VO	4.23 ± 0.14	2.18 ± 0.39	4.37 ± 0.18

1641
1642

are robust enough to guide different policy optimisation mechanisms effectively. In mo-halfcheetah-v5, PRISM achieves a significantly higher HV (2.24) compared to the oracle (1.80).

1643
1644
1645
1646
1647

Notably, these results were obtained with minimal hyperparameter tuning due to computational constraints. While this lack of fine-tuning explains the slight gap in EUM/VO metrics for mo-halfcheetah-v5 compared to the oracle, the method’s ability to achieve such strong results with a completely different backbone highlights PRISM’s inherent stability and generalisability.

1648
1649

I DECLARATION ON LARGE LANGUAGE MODELS

1650
1651
1652

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.

1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673