Benchmark Details

To evaluate the performance of RPSL, we conduct experiments on five problems from the multi-objective robot control benchmark suite MO-MuJoCo, as summarized in Table 4. MO-MuJoCo is a widely used MORL benchmark built upon the MuJoCo physics engine. It is noteworthy that we did not utilize the original benchmark; rather, we built upon it by introducing changes in the environment on one hand and incorporating observational disturbances on the other. The addition of these uncertain factors aligns with the approach discussed in the proposed method section.

Problem	m	State Space	Action Space
MO-Swimmer-v2	2	$\mathcal{S} \in \mathbb{R}^8$	$\mathcal{A} \in \mathbb{R}^2$
MO-HalfCheetah-v2	2	$\mathcal{S} \in \mathbb{R}^{17}$	$\mathcal{A} \in \mathbb{R}^6$
MO-Walker2d-v2	2	$\mathcal{S} \in \mathbb{R}^{17}$	$\mathcal{A} \in \mathbb{R}^6$
MO-Hopper-v2	2	$\mathcal{S} \in \mathbb{R}^{11}$	$\mathcal{A} \in \mathbb{R}^3$
MO-Hopper-v3	3	$\mathcal{S} \in \mathbb{R}^{11}$	$\mathcal{A} \in \mathbb{R}^3$

Table 4: Dimension Information of the five test problems in a multi-objective robot control benchmark suite MO-MuJoCo.

- **MO-Swimmer-v2**: The agent is a simplified three-link swimming robot that operates in a viscous fluid environment, with $S \subseteq \mathbb{R}^8, A \subseteq \mathbb{R}^2$. It is designed to maximize both forward displacement and energy efficiency. The preference vector $\boldsymbol{\omega}$ modulates the trade-off between propulsion power and energy consumption.
- **MO-HalfCheetah-v2**: The agent is a bidimensional robotic entity resembling a cheetah, tasked with optimizing two goals: maximization of forward momentum and conservation of energy. The environments for state and action are defined as $S \subseteq \mathbb{R}^{17}, A \subseteq \mathbb{R}^6$. The aim is to adjust the torque on the limb joints according to a preference vector $\boldsymbol{\omega}$ to achieve efficient forward motion.
- **MO-Walker2d-v2**: The agent is a planar bipedal robot with state and action spaces $S \subseteq \mathbb{R}^{17}, A \subseteq \mathbb{R}^6$. Its objective is to move forward efficiently while maintaining balance. The reward combines forward velocity and stability, controlled by the preference vector $\boldsymbol{\omega}$ that determines the emphasis between speed and stability.
- **MO-Hopper-v2**: With the state and action spaces described as $S \subseteq \mathbb{R}^{11}, A \subseteq \mathbb{R}^3$, the agent takes the form of a bidimensional mono-legged robot. It focuses on two main objectives: the acceleration of forward movement and the maximization of jump height. Adjusting the torque on its hinge according to a preference vector $\boldsymbol{\omega}$ is key to its forward hopping motion.
- **MO-Hopper-v3**: The agent is a bidimensional mono-legged robot with the state and action spaces defined as $S \subseteq \mathbb{R}^{11}, A \subseteq \mathbb{R}^3$. It aims to balance two conflicting objectives: achieving high forward velocity and maintaining energy efficiency during hopping. The preference vector $\boldsymbol{\omega}$ regulates the trade-off between jumping height, running velocity, and energy consumption, enabling the agent to adaptively adjust torque for efficient motion.

Training Details

Table 5 summarizes the training details of the proposed RPSL framework on the MO-MuJoCo benchmark tasks under perturbed environments. To ensure stable learning and robustness against environmental noise, several hyperparameters were carefully adjusted compared to standard training settings. Specifically, the learning rates were slightly reduced to improve numerical stability, the batch size and replay buffer were enlarged to mitigate distributional shifts caused by perturbations, and the exploration noise was increased to encourage diverse policy behaviors. These configurations collectively enhance the adaptability of RPSL, ensuring consistent convergence performance even in dynamically perturbed environments

Results on non-perturbed MO-MuJoCo

Table 6 reports the performance comparison among RPSL and four state-of-the-art multi-objective reinforcement learning methods on the non-perturbed MO-MuJoCo benchmark. Overall, RPSL achieves comparable but not dominant performance across the tested tasks.

In MO-HalfCheetah-v2, PSL-MORL obtains the highest HV and the lowest SP, suggesting that it converges slightly better and produces a more uniformly distributed Pareto set. RPSL achieves a similar HV but a moderately higher SP, showing that its learned policies are competitive in terms of convergence yet slightly less uniform. For MO-Walker2d-v2, RPSL attains the highest HV, marginally surpassing PSL-MORL and Hyper-MORL. However, its SP is larger than that of PSL-MORL, indicating that while RPSL achieves better convergence, its Pareto solutions are less evenly distributed. In MO-Hopper-v3, RPSL achieves the best HV and the lowest SP, outperforming all baselines in both convergence and uniformity. This suggests that RPSL is particularly effective in environments characterized by highly nonlinear dynamics and sensitive control signals.

It is worth noting that this may be attributed to the fact that, in non-perturbed environments, RPSL cannot effectively capture preference drift due to the absence of environmental perturbations. The two-level robust mechanism designed in RPSL may introduce excessive behavioral regularization in static conditions, occasionally leading to partial behavior collapse. Although we incorporated $\mathcal{L}_{co}$ to mitigate this effect, the mechanism inherently requires environmental perturbations to reach a balanced state between stability and adaptability. Consequently, RPSL is expected to demonstrate its full potential under perturbed environments, where the dynamic interplay between the mechanism and the environment can be properly activated.

Results on discrete benchmark FTN

Fruit Tree Navigation (FTN): The Fruit Tree Navigation (FTN) environment is a discrete multi-objective reinforcement learning benchmark, in which the agent navigates through a full binary tree of depth d (commonly $d = 5, 6,$ or 7). At each non-terminal node, the agent chooses one of two actions—left or right subtree—until reaching a leaf node. Each leaf node is associated with a six-dimensional

Parameter	MO-Swimmer-v2	MO-Walker2d-v2	MO-HalfCheetah-v2	MO-Hopper-v2	MO-Hopper-v3
Total number of steps	1×10^6	1×10^6	1×10^6	1×10^6	1×10^6
Minibatch size	512	512	512	512	512
Discount factor	0.995	0.995	0.995	0.995	0.995
Soft update coefficient	0.003	0.003	0.003	0.003	0.003
Buffer size	3×10^6	3×10^6	3×10^6	3×10^6	3×10^6
CriticNet Learning rate	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}
CriticNet hidden layers	1	1	1	1	1
CriticNet hidden neurons	400	400	400	400	400
ActorNet Learning rate	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}
ActorNet hidden layers	1	1	1	1	1
ActorNet hidden neurons	400	400	400	400	400
Policy update delay	10	10	10	10	20
Parameter fusion coefficient	0.05	0.05	0.05	0.05	0.05
Exploration noise std.	0.2	0.2	0.2	0.25	0.25
Noise clipping limit	0.7	0.7	0.7	0.7	0.7
Loss coefficient	15	15	15	15	15

Table 5: Hyperparameter Settings for RPSL MO-Mujoco Tasks under Perturbed Environments

Problem	Metrics	Hyper-MORL	PD-MORL	PSL-MORL	PG-MORL	**RPSL**
MO-HalfCheetah-v2	HV($\times 10^6$)	$5.53_{\pm 0.21}$	$5.89_{\pm 0.13}$	$\mathbf{5.92}_{\pm 0.19}$	$5.75_{\pm 0.12}$	$5.78_{\pm 0.10}$
	SP($\times 10^3$)	$0.29_{\pm 0.07}$	$0.47_{\pm 0.10}$	$\mathbf{0.16}_{\pm 0.23}$	$0.44_{\pm 0.18}$	$0.35_{\pm 0.09}$
MO-Walker2d-v2	HV($\times 10^6$)	$5.37_{\pm 0.38}$	$5.08_{\pm 0.42}$	$5.36_{\pm 0.25}$	$4.41_{\pm 0.22}$	$\mathbf{5.43}_{\pm 0.15}$
	SP($\times 10^4$)	$0.07_{\pm 0.29}$	$0.03_{\pm 0.08}$	$\mathbf{0.01}_{\pm 0.05}$	$0.04_{\pm 0.03}$	$0.19_{\pm 0.02}$
MO-Hopper-v3	HV($\times 10^{10}$)	$3.36_{\pm 0.26}$	$2.45_{\pm 0.11}$	$1.57_{\pm 0.29}$	$3.08_{\pm 0.31}$	$\mathbf{3.69}_{\pm 0.33}$
	SP($\times 10^7$)	$2.25_{\pm 0.30}$	$1.79_{\pm 0.27}$	$3.19_{\pm 0.20}$	$2.87_{\pm 0.33}$	$\mathbf{1.29}_{\pm 0.25}$

Table 6: Performance comparison between RPSL and competing algorithms on the **non-perturbed** MO-MuJoCo benchmark, evaluated in terms of Hypervolume (HV) and Spacing (SP). Higher HV and lower SP indicate better performance. The average values and standard deviations over 8 runs are reported. The best results are highlighted in bold.

reward vector $\boldsymbol{r} \in \mathbb{R}^6$, corresponding to nutritional components such as *Protein, Carbohydrates, Fats, Vitamins, Minerals,* and *Water*. These reward vectors are constructed such that each leaf lies on the convex coverage set of the Pareto front: for every leaf, there exists some preference weight $\boldsymbol{\omega}$ under which that leaf's reward is optimal. The objective of the agent, given a preference vector $\boldsymbol{\omega}$, is to select a path from the root to a leaf that maximizes the scalarized utility $\boldsymbol{\omega} \cdot \boldsymbol{r}$. Because this decision process must account for multiple trade-offs between objectives, FTN serves as an effective benchmark for evaluating whether a MORL algorithm can learn and adapt across different preference distributions and recover a diverse set of Pareto-optimal policies.

Method	FTN (d=5)		FTN (d=6)		FTN (d=7)	
	HV	SP	HV	SP	HV	SP
Envelope	6920.58	N/A	8427.51	N/A	6395.27	N/A
PD-MORL	6920.58	N/A	9299.15	N/A	11419.58	N/A
RPSL	**6763.95**	0.12	**9517.08**	0.16	**11277.06**	0.09

Table 7: Performance comparison on the Fruit Tree Navigation (FTN) benchmark with different tree depths.

The current results indicate that RPSL can be deployed on discrete benchmarks, showing strong compatibility across different task formulations. However, since the framework was originally designed for continuous control problems, these experiments mainly validate its adaptability rather than optimized performance. Future work will focus on fine-tuning the model and adjusting its components to achieve optimal performance in discrete decision-making environments.

More Experimental Results

More experimental results are currently being conducted to comprehensively evaluate RPSL under broader conditions, including discrete benchmarks, hyperparameter sensitivity, and perturbation intensities. These additional analyses are expected to provide a more complete understanding of the RPSL's behavior and further substantiate the claims of robustness and adaptability. The extended results will be included in our public repository.