A PROOFS OF TECHNICAL ANALYSIS

In this section, we provide formal proofs of our technical analysis in detail. For better legibility, we first recall the equations and results that we need for our proofs.

$$\forall \boldsymbol{\omega} \in \Omega, \quad \max_{\pi \in \Pi} \ \phi(V(\pi)), \tag{4}$$

where Ω is the set of valid preference weights sorted in descending order, $\boldsymbol{V}(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t \boldsymbol{r}_t]$ is the expected discounted return, and $\phi(\boldsymbol{J}) = \sum_{i=1}^{D} w_i \boldsymbol{V}_{(i)}$ with $\boldsymbol{V}_{(1)} \leq \cdots \leq \boldsymbol{V}_{(n)}$.

Lemma A.1. *For any MOMDP with linear preferences over objectives, the CCS contains an optimal policy for any linear combination of the objectives.*

Proof. Let $\mathcal{S}$ be the state space, $\mathcal{A}$ be the action space, and $\boldsymbol{r} : \mathcal{S} \times \mathcal{A} \to \boldsymbol{r}^D$ be the vector-valued reward function, where D is the number of objectives. Consider a linear preference vector $\boldsymbol{\omega} \in \Omega$, where $\Omega = \{\boldsymbol{\omega} \in \boldsymbol{r}^D : \sum_{i=1}^{D} w_i = 1, w_i \geq 0\}$. For any policy π, the expected return under a preference $\boldsymbol{\omega}$ is given by $\boldsymbol{\omega}\big(\mathbb{E}_\pi\left[\sum_{t=1}^{\infty} \gamma^{t-1} \boldsymbol{r}(s_t, a_t) \mid s_0 = s\right]\big)$. Thus, the optimal policy $\pi^*_{\boldsymbol{\omega}}$ for preference $\boldsymbol{\omega}$ satisfies

$$\pi^*_{\boldsymbol{\omega}} = \operatorname*{argmax}_{\pi} \boldsymbol{\omega}^T \boldsymbol{V}^{\pi}(s), \ \forall s \in \mathcal{S}.$$

By the definition of the CCS, for any $\boldsymbol{\omega} \in \Omega$, there exists a policy $\pi_{\text{CCS}} \in \text{CCS}$ such that

$$\boldsymbol{\omega}^T \boldsymbol{V}^{\pi_{\text{CCS}}}(s) \geq \boldsymbol{\omega}^T \boldsymbol{V}^{\pi}(s), \ \forall \pi \in \Pi, \forall s \in \mathcal{S}.$$

To prove the proposition, let's recall the Convex Hull Value Iteration (CHVI) algorithm [5]. Note that the CHVI algorithm iteratively updates the value function for each state by considering the convex hull of the achievable rewards via

$$\boldsymbol{V}(s) = \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} \boldsymbol{P}(s' \mid s, a) \text{CH}\left(\boldsymbol{r}(s, a) + \gamma \boldsymbol{V}(s')\right),$$

where $\text{CH}(\cdot)$ denotes the convex hull operation. This update rule ensures that the value function $\boldsymbol{V}(s)$ lies within the convex hull of the achievable rewards and the $\text{CH}(\cdot)$ achievable value functions $\boldsymbol{V}^{\pi}(s) \mid \pi \in \Pi$ forms the CCS. Therefore, for any linear preference vector $\boldsymbol{\omega}$, there must exist at least a policy π_{CSS} such that

$$\boldsymbol{\omega}^T \boldsymbol{V}^{\pi_{\text{CCS}}}(s) = \max_{\pi \in \Pi} \boldsymbol{\omega}^T \boldsymbol{V}^{\pi}(s), \ \forall s \in \mathcal{S}.$$

The resulting policies form the CSS, which are sufficient to cover all linear preferences $\boldsymbol{\omega} \in \Omega$. Thus, for any linear combination of objectives, the optimal policy can be found within the CSS, confirming its sufficiency and optimality. □

While GGF introduces non-linear fairness objectives, its piecewise linearity and concavity allow representation as a maximum of linear functions, which ensures that solutions lie within the CCS. The following proposition establishes the sufficiency of the CCS in representing optimal policies for $\phi_{\boldsymbol{\omega}}$ preference weights.

Proposition A.1. *For any $s \in \mathcal{S}$ in an MOMDP and a piecewise-linear concave welfare function $\phi_{\boldsymbol{\omega}}$ (e.g., GGF) that can be represented as, $\phi_{\boldsymbol{\omega}}(\boldsymbol{V}^{\pi}(s)) = \min_{\sigma \in \mathbb{S}_D} \{\boldsymbol{\omega}_\sigma^\top \boldsymbol{V}^{\pi}(s)\}$, there exists a policy $\pi^* \in$ CCS such that:*

$$\phi_{\boldsymbol{\omega}}(\boldsymbol{V}^{\pi^*}(s)) \geq \phi_{\boldsymbol{\omega}}(\boldsymbol{V}^{\pi}(s)) \quad \forall \pi \in \Pi.$$

Proof. Consider an arbitrary permutation $\sigma_A \in \mathbb{S}_D$. Since $\phi_{\boldsymbol{\omega}}$ is a piecewise-linear and concave function, under a fixed permutation σ_A it becomes:

$$\phi_{\boldsymbol{\omega}}(\boldsymbol{V}^{\pi}(s)) = \boldsymbol{\omega}_{\sigma_A}^\top \boldsymbol{V}^{\pi}(s).$$

Let $\pi_A \in \Pi$ be the policy that maximizes this linear scalarization:

$$\pi_A = \operatorname*{argmax}_{\pi \in \Pi} \omega_{\sigma_A}^\top \boldsymbol{V}^{\pi}(s).$$

By the definition of CCS and the result from Lemma A.1, there exist a $\pi^* \in$ CCS such that

$$\phi_{\boldsymbol{\omega}_{\sigma_A}}(\boldsymbol{V}^{\pi^*}(s)) \geq \phi_{\boldsymbol{\omega}_{\sigma_A}}(\boldsymbol{V}^{\pi_A}(s)).$$

Thus,

$$\phi_{\boldsymbol{\omega}_{\sigma_A}}(\boldsymbol{V}^{\pi^*}(s)) \geq \phi_{\boldsymbol{\omega}_{\sigma_A}}(\boldsymbol{V}^{\pi}(s)) \quad \forall \pi \in \Pi$$

Because this holds for any permutation $\sigma \in \mathbb{S}_D$, we cam conclude that for any policy $\pi \in \Pi$, there exists a corresponding $\pi^* \in$ CCS such that

$$\forall \pi \in \Pi, \quad \exists \pi^* \in \mathit{CCS}, \quad \phi_{\boldsymbol{\omega}}(\boldsymbol{V}^{\pi^*}(s)) \geq \phi_{\boldsymbol{\omega}}(\boldsymbol{V}^{\pi}(s)).$$

□

Fairness of Non-Stationary Policies. In fair MORL, learning non-stationary policies can be particularly beneficial, as they leverage historical information to make more informed decisions and adapt over time (see Example 4.2).

Proposition A.2. *Let the reward $\boldsymbol{r}$ be nonnegative, and Π_S and Π_{NS} be the sets of stationary and non-stationary policies, respectively. For any $s \in \mathcal{S}$ in an MOMDP and a given $\phi_{\boldsymbol{\omega}}$, there exists a non-stationary policy $\pi_{NS} \in \Pi_{NS}$ that achieves a higher welfare score than any stationary policy $\pi_S \in \Pi_S$, i.e.,*

$$\exists \pi_{NS} \in \Pi_{NS} : \phi_{\boldsymbol{\omega}}(\boldsymbol{V}^{\pi_{NS}}(s)) \geq \max_{\pi_S \in \Pi_S} \phi_{\boldsymbol{\omega}}(\boldsymbol{V}^{\pi_S}(s))$$

Proof. Let the state value function be defined by:

$$\boldsymbol{V}(s) = \mathbb{E}\left[\boldsymbol{G}_t \middle| s_t = s\right]$$

where the return $\boldsymbol{G}_t$ is given by:

$$\boldsymbol{G}_t = \sum_{k=0}^{\infty} \gamma^k \, \boldsymbol{r}_{t+k+1}.$$

Suppose an episode begins at time t and terminates at time T_{end}. For any intermediate time T with $t \leq T < T_{\text{end}}$, we can decompose the return into two parts:

$$\boldsymbol{G}_t = \underbrace{\boldsymbol{r}_{t+1} + \gamma \boldsymbol{r}_{t+2} + \cdots + \gamma^{T-t-1} \boldsymbol{r}_T}_{\boldsymbol{G}_t^{(1)}} + \underbrace{\gamma^{T-t} \left(\boldsymbol{r}_{T+1} + \gamma \boldsymbol{r}_{T+2} + \ldots\right)}_{\boldsymbol{G}_t^{(2)}}.$$

With above decomposition, We define value function as two parts:

$$\text{Early-period value function: } \boldsymbol{V}_1(s) = \mathbb{E}\left[\boldsymbol{G}_t^{(1)} \middle| s_t = s\right]$$

$$\text{Late-period value function: } \boldsymbol{V}_2(s) = \mathbb{E}\left[\boldsymbol{G}_t^{(2)} \middle| s_T = s\right]$$

so that

$$\boldsymbol{V}(s) = \boldsymbol{V}_1(s) + \gamma^{T-t} \boldsymbol{V}_2(s)$$

At time T, stationary policy π_S selects action solely based on late period value function $\boldsymbol{V}_2(s)$, while non-stationary policy has access

to both early $\boldsymbol{V}_1(s)$ and late period value function $\boldsymbol{V}_2(s)$ and can condition its action selection on the combined information given by two value functions.
Under a stationary policy, The total value can be presented as:

$$\boldsymbol{V}^{\pi_S}(s) = \boldsymbol{V}_1(s) + \gamma^{T-t} \underset{\boldsymbol{V}_2(s)}{\operatorname{argmax}}\{\phi_{\boldsymbol{\omega}}[\boldsymbol{V}_2(s)]\}$$

In contrast, under a non-stationary policy the total value is given by

$$\boldsymbol{V}^{\pi_{NS}}(s) = \underset{\boldsymbol{V}_1(s),\boldsymbol{V}_2(s)}{\operatorname{argmax}} \{\phi_{\boldsymbol{\omega}}[\boldsymbol{V}_1(s) + \gamma^{T-t}\boldsymbol{V}_2(s)]\}$$

therefore:

$$\exists \pi_{\text{NS}} \in \Pi_{\text{NS}} : \phi_{\boldsymbol{\omega}}(\boldsymbol{V}^{\pi_{\text{NS}}}(s)) \geq \max_{\pi_{\text{S}} \in \Pi_{\text{S}}} \phi_{\boldsymbol{\omega}}(\boldsymbol{V}^{\pi_{\text{S}}}(s))$$

This completes the proof. □

Optimality of Stochastic Policies for Fairness. Unlike the single-objective scenario, in MORL, a deterministic policy may not be optimal. A fairer solution can often be achieved through randomization.

Proposition A.3. *Let Π_{ST} be the set of stochastic policies and Π_D be the set of deterministic policies. For an MOMDP $\mathcal{M}$ and a concave welfare function such as $\phi_{\boldsymbol{\omega}}$, there exists a stochastic policy $\pi_{ST} \in \Pi_{ST}$ such that:*

$$\phi_{\boldsymbol{\omega}}(\boldsymbol{V}^{\pi_{ST}}) \geq \max_{\pi_D \in \Pi_D} \phi_{\boldsymbol{\omega}}(\boldsymbol{V}^{\pi_D}).$$

Proof. The key idea here is that a stochastic policy can represent a convex combination of deterministic policies for any concave welfare function $\phi_{\boldsymbol{\omega}}$ [9]. Hence, stochastic policies can achieve outcomes in the objective space that are unattainable by deterministic policies. Specifically, for $\phi_{\boldsymbol{\omega}}$, a deterministic policy π_{D} yields a fixed utility vector $\boldsymbol{V}^{\pi_{\text{D}}}$ while a stochastic policy π_{ST} can yield a distribution over utility vectors. Thanks to concavity of $\phi_{\boldsymbol{\omega}}$, which makes our problem in 3 convex optimization and Jensen's inequality [25], we obtain

$$\phi_{\boldsymbol{\omega}}\left(\mathbb{E}_{\tau\sim\pi}[\boldsymbol{V}^{\pi_{\text{st}}}]\right) \geq \mathbb{E}_{\tau\sim\pi}\left[\phi_{\boldsymbol{\omega}}(\boldsymbol{V}^{\pi_{\text{st}}})\right]. \tag{5}$$

Since $\phi_{\boldsymbol{\omega}}$ is a piecewise linear concave function, there exists a stochastic policy π_{st} that is a convex combination of deterministic policies such that

$$\mathbb{E}_{\tau\sim\pi}[\phi(\boldsymbol{V}^{\pi_{\text{st}}})] \geq \max_{\pi_{\text{d}} \in \Pi_{\text{D}}} \phi(\boldsymbol{V}^{\pi_{\text{d}}}). \tag{6}$$

By combining (5) and (6), we can obtain

$$\phi(\mathbb{E}_{\tau\sim\pi}[\boldsymbol{V}^{\pi_{\text{st}}}]) \geq \mathbb{E}_{\tau\sim\pi}[\phi(\boldsymbol{V}^{\pi_{\text{st}}})] \geq \max_{\pi_{\text{d}} \in \Pi_{\text{D}}} \phi(\phi^{\pi_{\text{d}}})].$$

This completes the proof. □

The optimality of stochastic policies implies that restricting the search for fair solutions to deterministic policies is insufficient. Stochastic policies offer a broader range of solutions and may better capture the trade-offs among multiple objectives, enhancing the overall fairness of the policy.

B FAIRNESS

In a fair single-policy setting, where the goal is to learn a single policy treating all users equally, three fairness principles, efficiency, equity, and impartiality, are defined below.

Definition B.1. *Efficiency states that among two feasible solutions, if one solution is (weakly or strictly) preferred by all users, then it should be preferred to the other one, e.g., $\boldsymbol{u} \succ \boldsymbol{u'} \Rightarrow \phi(\boldsymbol{u}) > \phi(\boldsymbol{u'})$, where $\phi(\boldsymbol{u})$ is the scalar utility function that specifies the value of a solution.*

Intuitively, the efficiency property specifies that given all else equal, one prefers to increase a user's utility. In the MORL setting, the efficiency property simply means Pareto dominance. More specifically, a solution is considered efficient if it is not dominated by any other solution for all objectives.

Next, we discuss the significance of the *equity* property, which is a stronger property than efficiency and is often associated with distributive justice, as it refers to the fair distribution of resources or opportunities. This property ensures that a fair solution follows the *Pigou-Dalton principle* [34], which states the transferring of rewards from the more advantaged users to the less advantaged users.

Definition B.2. *A solution satisfies the Pigou-Dalton principle if for all $\boldsymbol{u}$, $\boldsymbol{u'}$ equal except for $u_i = u'_i + \delta$ and $u_j = u'_j - \delta$ where $u'_i - u'_j > \delta > 0$, $\phi(\boldsymbol{u}) > \phi(\boldsymbol{u'})$.*

Finally, we discuss the *impartiality* property. This property is rooted in the principle of "equal treatment of equals", which states that individuals sharing similar characteristics should be treated similarly.

Definition B.3. *In a system, individuals with similar characteristics should be treated similarly, i.e., the solution should be independent of the order of its arguments $\phi(\boldsymbol{u}) = \phi(\boldsymbol{u}_\sigma)$, where σ is a permutation and $\boldsymbol{u}_\sigma$ is the vector obtained from vector $\boldsymbol{u}$ permuted by σ.*

B.1 Welfare Function

A welfare function, denoted as $\phi : \mathbb{R}^D \to \mathbb{R}$, aggregates the utilities of all users (or objectives) and offers a metric of the overall desirability of a solution for the entire group, where $\boldsymbol{\omega}$ represents the set of aggregation weights for all objectives. One well-established welfare function used in this paper is the generalized Gini welfare function. The generalized Gini welfare function constitutes a specific instance of the ordered weighted average (OWA)[54]. It is a renowned welfare function employed in multi-objective optimization [19, 46, 47, 52, 56, 57, 60], initially devised to quantify income distribution inequality in economics [53]. The generalized Gini welfare function is defined as follows:

$$G_{\boldsymbol{\omega}}(\boldsymbol{u}) = \sum_{i=1}^{D} \omega_{\sigma(i)} u = \boldsymbol{w}_\sigma^T \boldsymbol{u}, \tag{7}$$

where $\sigma \in \mathbb{S}_D$, which depends on $\boldsymbol{\omega}$, is the permutation that sorts the components of $\boldsymbol{\omega}$ and $\boldsymbol{\omega}_\sigma = (\omega_{\sigma(1)}, \ldots, \omega_{\sigma(D)})$. Equation (7) holds as the weights are rearranged based on the utility vector, assigning the largest weight to the smallest component of $\boldsymbol{u}$, the second-largest weight to the second-smallest component of $\boldsymbol{u}$, and so forth.

The generalized Gini welfare function satisfies the three fairness properties. Due to the positive weights, it is monotonically related to Pareto dominance, fulfilling the efficiency property. Moreover, the reordering of the components in the welfare function makes it symmetric with respect to its components, satisfying the impartiality property. Lastly, as the generalized Gini weights are positive and decreasing, it is Schur-concave, meeting the equity property.

Among numerous welfare functions, the generalized Gini welfare function possesses several favorable properties, namely, simplicity as it is a weighted sum in the Lorenz space [12, 38], well-understood properties axiomatized by Weymark [53], and generality. These favorable properties make it a suitable choice for addressing the challenge of finding fair solutions. Moreover, it is notably a concave function, which will make the solution to our problem easier.

To emphasize the versatility of the generalized Gini welfare function, various special cases can be derived by adjusting its weights accordingly. These cases include:

- **Maxmin fairness**: Setting $\omega_1 = 1$ and $\omega_i = 0$ for $i = 2, \cdots, K$ corresponds to the maxmin notion of fairness [41].
- **Regularized maxmin fairness**: Assigning $\omega_1 = 1$ and $\omega_i = \varepsilon$ for $i = 2, \cdots, K$ aligns with the regularized maxmin notion of fairness.
- **Utilitarian approach**: Setting $\omega_i = 1/K$ represents the utilitarian approach.
- **Leximin fairness**: If the ratio ω_j/ω_{j+1} tends toward infinity, it corresponds to the leximin notion of fairness [28, 41].

C DESCRIPTIONS OF ENVIRONMENTS

C.1 Species Conservation

In the field of ecology, the challenge of conserving interdependent endangered species is paramount. The simulation environment focuses on the balance required in the conservation of two such species: the sea otter and the northern abalone, which are currently endangered. The predation relationship between these species, with sea otters feeding on abalones, presents a unique challenge that requires careful consideration of fairness and equity in conservation efforts. Based on the framework in [10], we define the state space as the current population numbers of the sea otters and northern abalones. The action space consists of: introducing sea otters, enforcing antipoaching measures, controlling sea otter populations, implementing a combination of half-antipoaching and half-controlled sea otters, or taking no action. Each action carries significant ecological consequences; for instance, while the reintroduction of sea otters is essential for maintaining the abalone population, it must be carefully managed to prevent the abalone's extinction. Conversely, overlooking other management actions could lead to the demise of either species. The transition function employed in our model accounts for population dynamics, including external threats such as poaching and oil spills. Since our objective is to optimize the population densities of both species, we define the reward function as the densities of both species, i.e., $D = 2$.

C.2 Resource Gathering

In this scenario of resource gathering, we consider a 5×5 grid world domain inspired from [5]. This domain presents a unique challenge centered around the acquisition of three types of resources: gold, gems, and stones, thereby establishing a multi-objective framework with $\mathcal{K} = 3$. The autonomous agent is positioned within this grid world, and resources are distributed randomly across various locations. As a resource is collected by the agent, it is immediately regenerated at a new random location within the grid, ensuring a perpetual availability of resources. In this problem, the state is characterized by the agent's current location on the grid and a cumulative count of each type of resource collected over the course of the agent's trajectory. The agent can navigate the grid through actions aligned with the four cardinal directions: up, down, left, and right, facilitating movement across the grid. To add complexity to the resource management challenge, resources are assigned differing values, reflecting their relative importance. Specifically, gold and gems are attributed a value of 1, underscoring their significance, whereas stones are considered less valuable, with a value of 0.4. This valuation leads to an intentionally uneven distribution of resources within the grid, comprising two stones, one gold, and one gem. This configuration is designed to simulate a scenario where the agent must not only maximize the collection of resources but also achieve a balanced acquisition across the different types of resources. The overarching objective for the agent in this environment is dual: to maximize the total value of resources collected while ensuring an equitable collection across the various resource types. Achieving this balance is crucial for optimizing the agent's resource-gathering strategy, enhancing its overall utility and adaptability within the dynamic grid world. This nuanced approach to resource management in a simulated environment offers insights into the complexities of resource distribution and acquisition strategies, contributing to the broader discourse on multi-objective optimization in dynamic settings.

C.3 Multi-Product Web Advertising

We now consider the multi-product web advertising (MWP) problem, where an online store offers D distinct types of products for sale and an intelligent agent makes strategic decisions at each timestep about which advertisement to display: a product-specific advertisement for one of the products $i \in [0, ..., D-1]$, or a general advertisement that is not tailored to any specific product. The effectiveness of an advertisement is contingent upon its relevance to the customer's recent web activity, with appropriate advertisements significantly increasing the likelihood of a purchase, whereas inappropriate ones may deter the customer altogether. The state space of this problem is defined by the number of products available in the store, augmented by the number of visits, purchases, and exits. A visit state indicates a customer's interest in a particular product, a purchase state signifies the completion of a transaction, and an exit state occurs when a customer leaves the website without making a purchase. The action space is expanded to $n + 1$ actions, where actions 0 through n correspond to displaying advertisements for specific products, and action n represents the option to show a general advertisement that does not target any specific product in the inventory. This additional action introduces an additional layer of complexity, as the agent must decide the optimal moment to transition between states. The reward function is designed such that the agent receives a reward of 1 in the i^{th} dimension of the

reward vector if a product of type i is sold after the display of its advertisement. The primary objective of this problem is to maximize the aggregate returns from product sales while striving for an equitable distribution of sales across the different product types. This goal underscores the need for fair solutions that not only optimize overall profitability but also ensure a balanced representation of product sales, thereby addressing the dual challenges of efficiency and equity in this domain.

D HYPERPARAMETERS

To ensure reproducibility, we have meticulously documented all hyperparameters across different environments in Tables 1,2,3, and 4. We utilize the well-known high-quality MORL baselines[1] for implementing baseline algorithms. In these tables, we present the hyperparameters corresponding to Envelope, GPI, PCN, and our proposed algorithms in three distinct environments, namely, species conservation (SC), resource gathering (RC), and multi-web product advertising (MWP).

[1] https://github.com/LucasAlegre/morl-baselines

Table 2: Set of hyperparameters used for training Envelope.

Hyperparameter	SC	RC	MWP
Discount factor (γ)	0.99	0.99	0.99
Learning rate (α)	0.0001	0.0005	0.005
Batch size	64	64	64
Hidden Layers	256 x 256 x 256 x 256	256 x 256 x 256 x 256	256 x 256 x 256 x 256
Buffer Size	50000	50000	50000
Initial Epsilon	1.0	1.0	1.0
Final Epsilon	0.05	0.05	0.05
Epsilon Decay Steps	50000	50000	50000
Learning Starts	100	100	100
Gradient Updates	1	1	5
Max Gradient Norm	1.0	1.0	1.0
Ω Distribution	Gaussian	Gaussian	Gaussian
Tau	0.5	0.5	0.5

Table 3: Set of hyperparameters used for training our proposed methods.

Hyperparameter	SC	RC	MWP
Discount factor (γ)	0.99	0.99	0.99
Learning rate (α)	0.0001	0.0005	0.005
Batch size	64	64	64
Hidden Layers	256 x 256 x 256 x 256	256 x 256 x 256 x 256	256 x 256 x 256 x 256
Buffer Size	50000	50000	50000
Initial Epsilon	1.0	1.0	1.0
Final Epsilon	0.05	0.05	0.05
Epsilon Decay Steps	50000	50000	50000
Learning Starts	100	100	100
Gradient Updates	1	1	5
Max Gradient Norm	1.0	1.0	1.0
Ω Distribution	Gaussian	Gaussian	Gaussian
Tau	0.5	0.5	0.5

Table 4: Set of hyperparameters used for training GPI.

Hyperparameter	SC	RC	MWP
Discount factor (γ)	0.99	0.99	0.99
Learning rate (α)	0.0001	0.0005	0.005
Batch size	128	128	256
Hidden Layers	256 x 256 x 256 x 256	256 x 256 x 256 x 256	256 x 256 x 256 x 256
Num Networks	2	2	2
Buffer Size	50000	50000	50000
Initial Epsilon	1.0	1.0	1.0
Final Epsilon	0.05	0.05	0.05
Epsilon Decay Steps	50000	50000	50000
Learning Starts	100	100	100
Gradient Updates	1	1	5

Table 5: Set of hyperparameters used for training PCN.

Hyperparameter	SC	RC	MWP
Discount factor (γ)	0.99	0.99	0.99
Learning rate (α)	0.0001	0.0001	0.0005
Batch size	128	256	128
Hidden Layers	64 x 64	64 x 64	64 x 64
Desired Return	[1, 1]	[200, 200, 200]	[100, 100, 100, 100, 100]
Buffer Size	500000	500000	1000000
Max Horizon	5000	1000	1000