

# Navigating Trade-Offs: Policy Summarization for Multi-Objective Reinforcement Learning

Zuzanna Osika<sup>a,\*</sup>, Jazmin Zatarain-Salazar<sup>a</sup>, Frans A. Oliehoek<sup>a</sup> and Pradeep K. Murukannaiah<sup>a</sup>

<sup>a</sup>Delft University of Technology

ORCID (Zuzanna Osika): <https://orcid.org/0000-0002-0602-2812>, ORCID (Jazmin Zatarain-Salazar): <https://orcid.org/0000-0003-2248-2242>, ORCID (Frans A. Oliehoek): <https://orcid.org/0000-0003-4372-5055>, ORCID (Pradeep K. Murukannaiah): <https://orcid.org/0000-0002-1261-6908>

**Abstract.** Multi-objective reinforcement learning (MORL) is used to solve problems involving multiple objectives. An MORL agent must make decisions based on the diverse signals provided by distinct reward functions. Training an MORL agent yields a set of solutions (policies), each presenting distinct trade-offs among the objectives (expected returns). MORL enhances explainability by enabling fine-grained comparisons of policies in the solution set based on their trade-offs as opposed to having a single policy. However, the solution set is typically large and multi-dimensional, where each policy (e.g., a neural network) is represented by its objective values.

We propose an approach for clustering the solution set generated by MORL. By considering both policy behavior and objective values, our clustering method can reveal the relationship between policy behaviors and regions in the objective space. This approach can enable decision makers (DMs) to identify overarching trends and insights in the solution set rather than examining each policy individually. We tested our method in four multi-objective environments and found it outperformed traditional k-medoids clustering. Additionally, we include a case study that demonstrates its real-world application.

## 1 Introduction

Multi-objective reinforcement learning (MORL) is a branch of reinforcement learning (RL) that aims to optimize multiple, often conflicting, objectives simultaneously. MORL has been applied in various domains, including water management ([6, 9]), autonomous driving ([16]), power allocation ([18, 33]), drone navigation ([32]), and medical treatment ([13, 17]). Unlike single objective RL that focuses on maximizing a single reward function, MORL requires balancing trade-offs among multiple goals. As a result, the output of an MORL algorithm is not a single policy, but a set of policies. Depending on the assumptions and the information available, the solution set can be a Pareto set, a coverage set, or a convex hull [10].

A significant advantage of MORL is that it is inherently more explainable than single-objective RL. MORL outputs a rich set of policies (solutions), each offering distinct trade-offs with regard to the objectives. This multiplicity of solutions not only provides a comprehensive overview of the possible actions and their consequences but also facilitates a deeper understanding of how objectives interact and influence decision-making. Further, MORL avoids the need to collapse multiple goals into a single metric. Thus, decision makers

(DMs) could be equipped with a nuanced and detailed understanding of the outcomes, thereby enabling them to make informed choices. This transparency in policy generation enhances the trustworthiness of decisions made using MORL in real-world problems.

Although MORL makes trade-offs explicit, the DMs must still exert considerable cognitive effort to navigate and evaluate the options in the solution set. This process becomes more challenging if the DMs do not have clear preferences on the objectives. Further, as the number of objectives increases, the size of the solution set expands drastically, which makes comprehending the entire set of solutions impractical. Thus, although MORL holds considerable promise in enhancing the explainability of decisions, its practical application is hindered by the associated complexities. To fully leverage MORL's potential, it is important to develop methodologies and tools to support the DM in exploring the policies in the solution set.

The main focus of current MORL research is on developing algorithms and benchmarking their performance (e.g., [2, 8, 34]). However, there is a dearth of tools that allow users (DMs) to comprehend the high-dimensional solution sets the MORL algorithms produce. Such tools facilitate the practical deployment of MORL systems by enabling DMs to analyze policy behavior, understand the trade-offs, and differentiate between policies. Without these decision-support tools, there is a gap in the practical applicability of MORL, hindering the translation of algorithmic advancements into real-world applications. To address this gap, we propose approaches for the post-analysis of trade-offs in the MORL solution sets.

In one-shot multi-objective optimization (MOO), e.g., engine design, a common method to convey a high-dimensional solution set involves clustering the solutions based on their trade-offs in the objective space (see [19] for a review on decision support approaches for traditional one-shot MOO). Clustering of solutions [5] in the objective space can streamline the decision-making process by reducing the dimensionality of the solution set, enabling DMs to focus on clusters of solutions rather than evaluating all solutions. Ulrich [29] extends the standard clustering approach by proposing Pareto-Set Analysis, an approach that identifies compact and well-separated clusters in both decision and objective spaces. This approach aids in identifying trends and key turning points in the data and it reveals the relationships between decision variables and objectives.

We extend the idea of Pareto-Set Analysis to MORL solution sets. Whereas clustering in the objective space is similar to one-shot MOO, finding clusters in the decision variable space is nontrivial for

\* Corresponding Author. Email: z.osika@tudelft.nl

MORL. In MORL, a solution, policy, is described as a mapping from states to a probability distribution over actions. For problems with large (sometimes infinite) state or action spaces, the solution can be a neural network, making post-analysis in both spaces challenging.

Clustering in the objective space alone is inadequate, as different policies with similar trade-offs can behave differently. For example, two water management policies may yield similar expected returns but differ drastically in terms of disaster risk. Therefore, it’s crucial to summarize and compare policy behaviors relative to their expected returns to understand how they align with or diverge from objectives.

**Contribution.** We address the challenge of decision support in MORL by proposing a method that explains the MORL solution set through clustering, considering both the objective and the behavior spaces. To do so, we show ways to effectively represent the behavior space of the policies by employing techniques from explainable reinforcement learning (XRL) field. Given that representation, we apply Pareto-Set Analysis (PAN) clustering algorithm [29] to find well-defined clusters in both spaces. We apply this method to four diverse MORL environments and show that it improves performance, compared to the traditional k-medoid clustering. We also use one of the environments, the multi-objective highway environment, to demonstrate the importance of considering not just the trade-offs (expected returns) but also policies’ behavior when presenting information to the DMs. To the best of our knowledge, we are the first to tackle the challenge of explaining the solution set of MORL.<sup>1</sup>

## 2 Background and Related Work

We review three lines of work we build on: 1) MORL, 2) Pareto Set Analysis, and 3) policy summarization.

### 2.1 Multi-Objective Reinforcement Learning (MORL)

MORL is used to solve problems where an agent should optimize multiple, potentially conflicting, objectives simultaneously over time. Each objective in MORL is represented by a separate reward function. These problems can be modeled as Multi-objective Markov Decision Processes (MOMDPs) with multiple reward functions.

In MORL, a MOMDP is defined by  $(S, A, p, r, \mu, \gamma)$ , where:

- $S$  represents the state space.
- $A$  is the action space.
- $p(\cdot | s, a)$  is the probability distribution over next states given the current state  $s$  and action  $a$ .
- $r : S \times A \times S \rightarrow \mathbb{R}^m$  is the multi-objective reward function with  $m$  objectives.
- $\mu$  is the initial state distribution.
- $\gamma \in [0, 1)$  is a discount factor.

An MORL policy,  $\pi : S \rightarrow A$ , maps states to actions to optimize outcomes across objectives. The action-value function for a policy  $\pi$  for a given state-action pair  $(s, a)$  is an  $m$ -dimensional vector representing the expected return for each objective [10].

A key concept in MORL is the Pareto frontier—a set of policies whose multi-objective value functions are not dominated by any other policy. A policy’s value profile is said to be Pareto dominant if it is better or equal in all objectives and strictly better in at least one objective compared to another policy’s value function. The Pareto

frontier thus consists of all policies that offer the best possible trade-offs among the objectives, where improving one objective cannot be achieved without worsening at least one other objective [30].

Another key concept is the Convex Hull, specifically in the context of linear utility functions. It defines the region where the weighted sum of objectives (as determined by the utility function) can be maximized. For a set of weights, which represent the relative importance of each objective, the optimal policy is one that maximizes this weighted sum. The Convex Hull, therefore, contains policies that are optimal for different combinations of weights, covering all linear preferences in the multi-objective framework [23].

Hypervolume is a popular metric for measuring the performance of MORL algorithms. It measures the volume (in the objective space) enclosed between the solution set generated by an algorithm and a reference point, which typically represents the worst possible values of objectives [31]. The metric provides a measure of how well a set of solutions covers the objective space, indicating the diversity and quality of the solutions with respect to the multiple objectives being optimized. A larger hypervolume generally indicates a diverse set which is also close to the ideal solution.

**GPI-PD.** In our experiments, we employ the Generalized Policy Improvement-Prioritized Dyna (GPI-PD) algorithm [3] to train our agents. The algorithm leverages Generalized Policy Improvement (GPI) to establish prioritization schemes that enhance the efficiency of learning from samples. Through the utilization of active learning strategies, the algorithm enables the agent to (i) pinpoint the most promising objectives or preferences to focus on at any given time, accelerating the resolution of MORL challenges; and (ii) determine the most pertinent previous experiences to inform the learning of a policy tailored to a specific agent preference, utilizing an innovative approach inspired by Dyna-style MORL methods.

Generalized Policy Improvement (GPI) extends policy improvement by defining a new policy  $\pi$  that improves over a set of policies  $\Pi$ , rather than a single one. For a given weight vector  $\mathbf{w} \in \mathcal{W}$ , the GPI policy is  $\pi^{GPI}(s; \mathbf{w}) \in \arg \max_{a \in \mathcal{A}} \max_{\pi \in \Pi} q_{\pi}^{\mathbf{w}}(s, a)$ . GPI ensures that the selected policy matches or surpasses the performance of any policy  $\pi_i \in \Pi$  across all weight vectors  $\mathbf{w} \in \mathcal{W}$ .

Dyna-style algorithms use a search control mechanism to sample experiences from the learned model for planning. They construct a set of policies  $\Pi$  with value vectors  $V$  approximating the CCS. Each iteration involves selecting a weight vector  $\mathbf{w} \in \mathcal{W}$ , guided by GPI improvements, and learning a new policy  $\pi_{\mathbf{w}}$  optimized for that weight vector. The algorithm employs GPI-based prioritization to efficiently determine relevant experiences for learning the optimal policy  $\pi_{\mathbf{w}}$ . GPI-PD learns a multi-objective dynamics model  $p$  to predict subsequent states and rewards from state-action pairs, facilitating Dyna updates to the action-value functions using simulated experiences.

### 2.2 Pareto-Set Analysis

Pareto-Set Analysis (PAN) is a clustering technique introduced in the field of MOO [29]. Unlike the traditional clustering approaches that focus solely on the objective space (e.g. k-means clustering), PAN identifies clusters that are compact and well-separated in both decision and objective spaces. Also, PAN does not focus on finding just one best partitioning of the solutions. Instead, it helps the DM to better understand the problem by revealing the relationship between different design choices and their outcomes in the objective space.

PAN is flexible as it does not depend on the identification of specific design variables or feature vectors. The main prerequisite is a

<sup>1</sup> The code is available at <https://github.com/osikazuzanna/Bi-Objective-Clustering>

distance measure in each space. Since the clusters formed in the decision space may not align with those in the objective space, PAN formulates the clustering as a bi-objective optimization challenge (partitioning to perform well in both objective and decision spaces). Thus, the PAN algorithm is a bi-objective evolutionary algorithm tailored for generating a set of effective partitionings.

The essence of PAN is its ability to optimize partitioning in both decision and objective spaces, utilizing two validity indices of the clustering: one for the objective space,  $V(f(C^*), d_O)$ , and another for the decision space,  $V(f(C^*), d_D)$ . These indices evaluate the coherence of solutions within each cluster, guiding the algorithm toward partitionings that optimize the trade-offs between being compact in one space and well-separated in the other.

PAN applies variation and selection processes, iteratively, to evolve a population of partitionings. Each partitioning is assessed based on its clustering quality in both spaces, marking a significant departure from conventional methods that prioritize objectives without considering the structural implications of solutions.

### 2.3 Policy Summarization Techniques in RL

We aim to adapt PAN clustering to MORL policy set. Unlike traditional one-shot MOO tasks that focus on one-time decisions, MORL involves sequential decision-making processes. In one-shot MOO, the decision space typically consists of vectors of decision variables that describe, e.g., a candidate design. In contrast, a MORL solution is a function, that maps states to actions, often represented by a neural network [3]. The objective space in MORL can be constructed by calculating the expected returns of each policy. However, the decision space, comprising sequences of decisions predicted by e.g. a neural network for given states, is complex. Consequently, our initial step is to create the decision space, which we refer to as the Behavior Space as it reflects the behavior of the policy. The goal is to summarize the behavior of each policy, enabling the calculation of similarity (or distance) between these summaries.

Techniques from Explainable RL can be particularly valuable in this context. Specifically, we are using Highlights [4], a policy summarization method, which identifies the most important states for the particular policy and states in the close proximity of those states. These are combined into summary videos of agent behaviors; details can be found in Section 3.1. Other methods developed to summarize agent’s behavior use computational models ([11], [14]) or state abstraction ([26], [27]). We decided not to use them as they produce summaries, which cannot be easily interpreted and analyzed for further computation of clusters.

## 3 Approach

Figure 1 shows an overview of our approach, whose input is the set of policies generated by an MORL agent. Each policy,  $f_i(x)$ , is a mapping function (states to actions), and is represented in two spaces.

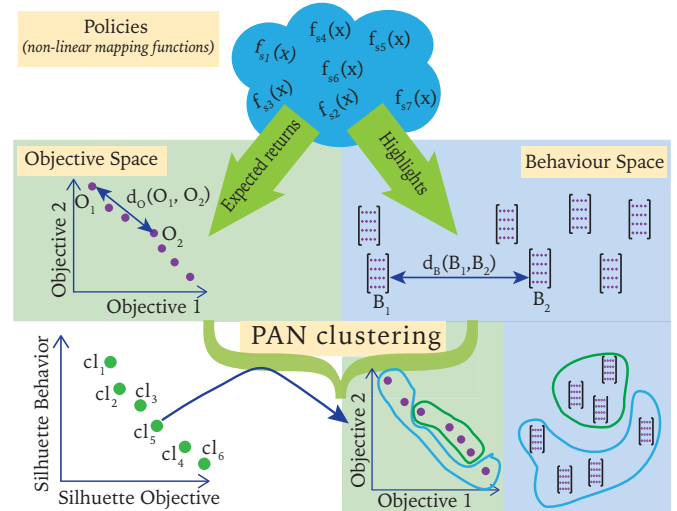
**Objective space:** An objective vector (vector of objective values),  $O_i$ , computed as the expected returns over a number of episodes.

**Behavior space:** A behavior matrix,  $B_i$ , of five Highlights (Section 3.1) states.

We measure the distances between two policies in the objective and behavior spaces,  $d_O$  and  $d_B$ , via Euclidean and Frobenius distances, respectively (Section 3.2). These distances are used by PAN clustering (Section 3.3), a bi-objective clustering algorithm. PAN

outputs a set of partitions (clustering), offering different trade-offs in terms of clustering quality in the two spaces. Each partition,  $C_i$ , represents a clustering (set of clusters) of the policies.

The key advantage of this method is its flexibility—it allows for the customization of partitioning based on different preferences or requirements. For instance, a decision maker may prioritize the analysis of policy behavior over the optimization of objectives, or vice versa. This adaptability ensures that the clustering process aligns with the decision maker’s goals, facilitating a more targeted and insightful exploration of the solution space.



**Figure 1:** An outline for our approach which clusters a set of policies, considering clustering quality in both objective and behavior spaces.

### 3.1 Policy Representation using Highlights

The Highlights algorithm [4] is designed to create a video summary of an agent’s behavior through online simulations. It calculates the most significant states encountered by the policy, including those nearby. The algorithm uses a concept of state importance, which is judged based on how critical the choice of action in that state is to future rewards, as indicated by the agent’s Q-values. This method of determining importance has been effective in identifying teaching moments in student-teacher RL, although it does have limitations, such as sensitivity to the number of possible actions.

In the Highlights approach, the state importance was as:

$$I(s) = \max_a Q^\pi(s, a) - \min_a Q^\pi(s, a), \quad (1)$$

where  $Q^\pi_{(s,a)}$  quantifies the expected return of taking action  $a$  in state  $s$ , following the policy  $\pi$  thereafter [28]. The original Highlights algorithm outputs videos of the most important transition for the policy. It also includes transitions proceeding and preceding the important states so that these snapshots can be used to show to the users.

Utilizing the Highlights approach, we identify five most significant states for each policy (as in the original paper). We treat these five states as representative of policy behavior. Subsequently, each policy is represented by five states, which can be represented as a  $n_{\text{feat}} \times 5$  matrix, where  $n_{\text{feat}}$  is the number of features the state is represented with (we assume that it is possible to represent the state numerically). We also create video snapshots for later analysis.

We execute each policy fifty times to determine the set of five most important states for our analysis. During these runs, we assess the importance of each visited state based on Equation 1. We gather states from 50 executions of the policy, ensuring that each state is only recorded once by discarding any duplicates.

### 3.2 Distances between the Policies

Our goal is to cluster policies, considering both objective and behavior spaces. To do so, we define a distance measure for each space.

- For the Objective Space, where policies are represented as objective vectors, we employ the Euclidean distance:

$$d_O(O_i, O_j) = \sqrt{\sum_{x=1}^{n_{\text{obj}}} (O_i^x - O_j^x)^2}, \quad (2)$$

where  $O_i, O_j$  are objective vectors of policies  $i$  and  $j$ , respectively; and  $n_{\text{obj}}$  is the number of objectives in the MORL problem.

- In the behavior space, each policy is represented by a matrix with each column corresponding to a state. To calculate the distance between two matrices, we use the Frobenius distance:

$$d_B(B_i, B_j) = \sqrt{\sum_{x=1}^{n_{\text{feat}}} \sum_{y=1}^{n_{\text{st}}=5} (B_i^{x,y} - B_j^{x,y})^2}, \quad (3)$$

where  $B_i, B_j$  are behavior matrices of policies  $i$  and  $j$ , respectively;  $n_{\text{feat}}$  is the number of features a state is represented with; and  $n_{\text{st}}$  is the number of states the policy is represented with (in our case,  $n_{\text{st}} = 5$ ).

### 3.3 Bi-Objective Clustering using PAN

In our case PAN considers both the objective values of a policy and its behavior, represented by the five Highlights states.

PAN clustering requires a cluster *validity index*, which measures how good the clusters are in a space. We employ the well-known Silhouette index [24]. Given a clustering  $C$  of  $N$  data points, the Silhouette index is:

$$S(C) = \frac{1}{N} \sum_{i=1}^N s(i) \quad (4)$$

where  $s(i)$  is the Silhouette index for a single data point  $i$  and is calculated as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (5)$$

where  $a(i)$  is the average distance from the  $i$  to the other data points in the same cluster, and  $b(i)$  is the minimum average distance from the  $i$  to data points in a different cluster, minimized over all clusters.  $S(C)$  ranges from -1 to 1, where a high value indicates that the clusters are cohesive and well-separated.

The PAN algorithm seeks to minimize two objectives:

$$\text{minimize} \quad -(-S(C_O) - 1) \quad (6)$$

$$\text{minimize} \quad -(-S(C_B) - 1), \quad (7)$$

where  $S(C_O)$  and  $S(C_B)$  are Silhouette indices, measuring the clustering performance in objective and behavior spaces, respectively.

We represent a partitioning of the solution set in the PAN algorithm by a (variable length) list of clusters, where each cluster, in turn, is a list of indices of policies. Similar to the original paper, we also restrict that each partition contains at least two clusters and each cluster contains at least two policies.

The evolutionary algorithm starts with a random population of partitionings. Then, variation and selection operations are applied to the

generated population as well as local optimization of the population in each generation (see [29]). The selection is conducted based on the hypervolume values, with reference point set to  $[2, 2]^2$

As PAN operates as a standard evolutionary algorithm, it requires the configuration of parameters such as the number of generations, population size, and probabilities for various mutation operations. For every environment we tested the algorithm on, we did a parameter search over different combinations to achieve satisfactory performance. The parameter configuration as well as the convergence plots for each environment can be found in the appendix C [20].

### 3.4 Algorithm Overview

Algorithm 1 shows the psuedo-code of our approach. Its input is a set of policies  $F = \{f_1, \dots, f_i\}$  and PAN parameters—the population size of the partitionings ( $n$ ), number of generations to run the evolution for ( $g$ ), and probabilities of mutation ( $p_m$ ), the union of two randomly chosen clusters ( $p_u$ ), recombination of parent clusters ( $p_r$ ) and split of two randomly chosen clusters ( $p_s$ ).

First, the algorithm computes an objective vector for each policy as its expected returns and a behavior matrix for each policy using Highlights. Given these representations, the algorithm computes distances in objective and behavior spaces. Then, the algorithm performs PAN clustering. The final outputs a set of clusterings. Each clustering represents a set of clusters and the qualities of the clustering in objective and behavior spaces.

---

#### Algorithm 1 Bi-objective clustering of a MORL solution set

---

- 1: **Input:**  $F = \{f_1, \dots, f_m\}$  ▷ Solution set
  - 2: **Input:**  $n, g, p_m, p_u, p_r, p_s$  ▷ PAN parameters
  - 3:  $S_O, S_B \leftarrow []$  ▷ Lists of objective vectors ( $S_O$ ) and behavior matrices ( $S_B$ )
  - 4: **for**  $f_i$  in  $F$  **do**
  - 5:      $S_O.append(\text{ExpectedReturns}(f_i))$
  - 6:      $S_B.append(\text{Highlights}(f_i, 5))$
  - 7: **end for**
  - 8:  $D_O \leftarrow \text{EuclideanDistances}(S_O)$  ▷ Matrix of objective space distances
  - 9:  $D_B \leftarrow \text{FrobeniusDistances}(S_B)$  ▷ Matrix of behavior space distances
  - 10:  $C \leftarrow \text{PAN}(D_O, D_B, n, g, p_m, p_u, p_r, p_s)$
  - 11: **Output:**  $C = \{C_1, \dots, C_n\}$  ▷ Pareto set of clusterings; each  $C_i$  is a set of clusters
- 

## 4 Experimental Setting

We conduct experiments in four multi-objective environments from MO-Gymnasium [1]: MO-Mountaincar (3 objectives), MO-Minecart (3 objectives), MO-Highway (3 objectives), MO-Reacher (4 objectives). Appendix C [20] provides a detailed description of the environments, with convergence plots. For brevity, we deep dive into MO-Highway as a case study to demonstrate the applicability and insights offered by our algorithm. We utilize the MO-Highway environment because it closely mirrors real-world scenarios, making it comprehensible for end-users. Additionally, this environment contains only 12 policies within the solution set, enabling a thorough analysis and understanding of the behavior exhibited.

---

<sup>2</sup> The objectives, derived from the silhouette index, are transformed from a  $[-1, 1]$  range to  $[0, 2]$  for minimization.

In the MO-Highway environment, an ego-vehicle (agent) is driving on a multi-lane highway populated with other vehicles (Figure 2). The agent has three objectives—to reach a high speed, avoiding collisions with other vehicles, and driving on the right side of the road. The agent can move right or left (changing lanes), increase or decrease velocity, or stay idle, i.e., make no change. There is no defined target the agent is required to reach. Instead, the road goes on continuously. This property allows us to observe the agent’s general behavior and preferences instead of focusing on its progression towards reaching the goal. To represent the states, we use KinematicObservation (see [15] for details). It is a  $5 \times 5$  array that describes a list of 5 nearby vehicles by a set of features of size 5.

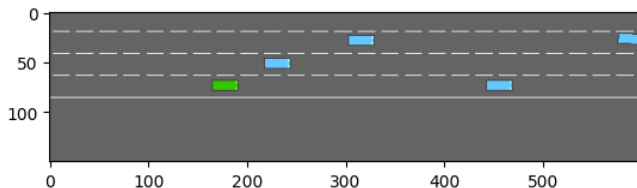


Figure 2: Screenshot from the highway environment

To produce the solution sets for each environment, we sampled 1000 weight combinations, equally spaced in the weight simplex, using the Riesz  $s$ -Energy method [7]. Given the weights and expected returns obtained by executing the policy for five episodes, the policies were compared and filtered based on their Pareto-dominance.

The MORL agents was trained with GPI-PD, using the same parameters and assumptions as in the MORL baselines [8] (default hyper-parameters, trained for 200k steps).

## 5 Results

First, we show the performance, measured via hypervolume of the resulting set of clusterings (or partitionings) for all four environments, followed by a detailed analysis of the MO-Highway environment.

To benchmark the performance of PAN, we compare the clusterings identified by PAN with the clusterings identified via (iterative)  $k$ -medoids clustering [25] (similar to the original paper [29]). The  $k$ -medoids algorithm is similar to the  $k$ -means algorithm. Whereas  $k$ -means requires data points to be represented in an  $n$ -dimensional (typically, Euclidean) space,  $k$ -medoids only requires a matrix of dissimilarities between data points. This is useful for our setting since it allows any type of policy representation to be used.

We apply  $k$ -medoids several times for all possible cluster numbers. Each time, we cluster the solutions twice, once in behavior space and once in objective space, and check whether the optimized partitionings satisfy the constraints (i.e. containing at least two clusters, where each cluster contains at least two solutions). For all partitionings that satisfy the constraints, we calculate the silhouette index in decision space and in objective space. Finally, to compare the resulting population with PAN, we reduce the number of achieved solutions to the population size used with PAN, using PAN’s selection procedure. We do not consider applying  $k$ -medoids to the combined behavior-objective space as the meaning of such a distance would be unclear.

Table 1 shows that PAN performs comparably to  $k$ -medoids in the MO-Reacher and MO-Minecart environments. This similarity in performance can be attributed to  $k$ -medoids’ focus on either behavior or objective space, potentially leading to better clustering outcomes that enhance the hypervolume by increasing the dispersion of the set of clusterings. In contrast, PAN tends to prioritize a balanced behavior-objectives performance, resulting in more moderate clusterings that

do not significantly spread the set of clusters, thus achieving a lower hypervolume. However, in the other two environments evaluated, our method surpasses  $k$ -medoids, demonstrating its capability to achieve comparable or superior performance in a broader range of scenarios.

Environment	PAN	$k$ -medoids
MO-Highway	1.73	1.66
MO-Reacher	2.46	2.42
MO-Minecart	2.61	2.62
MO-Lunar-lander	2.45	1.75

Table 1: Hypervolume achieved by PAN vs.  $k$ -medoids clustering.

### 5.1 Detailed Analysis of MO-Highway

**Solution Set.** The solution set obtained from the MO-Highway environment consists of 12 policies. Although this is a smaller solution set than what is usually seen in real-world problems [22], it is sufficient to illustrate the challenges associated with exploring potential policies. After analyzing the expected returns of these policies (the trade-off plot can be found in the Appendix A [20]), it was evident that all the policies effectively mitigate collisions (one of the objectives). Thus, in the following analysis, we only focus on the trade-offs between two objectives—speed vs. staying on the right lane.

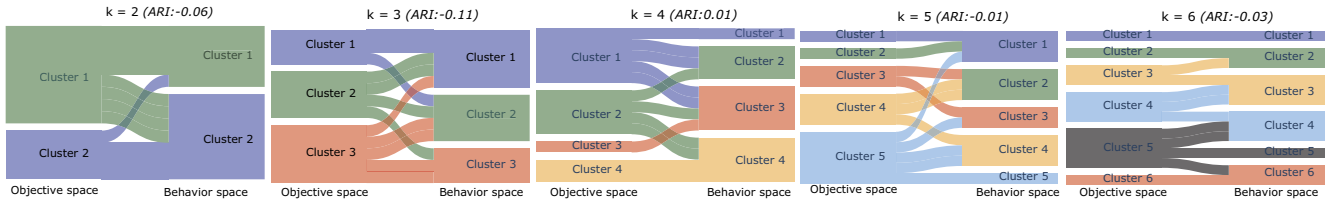
**Objective and Behavior Spaces** We cluster the solution set separately in the objective and behavior space to evaluate whether and to what extent the two sets of clusters differ using  $k$ -medoids clustering. Figure 3 shows Sankey diagrams depicting the overlap between clusters in the two spaces. It is evident from these diagrams that the set of cluster differ substantially between the two spaces.

To quantify the difference between the two sets of clusters, we employ the Adjusted Rand Index (ARI). ARI adjusts the Rand Index, which measures similarity between two data clusterings by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the two clusterings. The values of the Adjusted Rand Index can range from -1 to 1, where 1 indicates perfect agreement between the two clusterings, 0 suggests random or chance agreement between the two clusterings, and a value less than 0 indicates that the observed clustering is less accurate than would be expected by random chance [12]. As Figure 3 shows, most ARI values are close to zero. This demonstrates that the clustering in objective and behavior spaces can lead to very different sets of clusters.

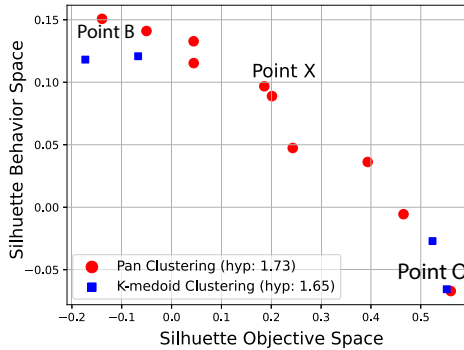
**PAN Clustering** Having shown that the clusters in objective and behavior spaces differ substantially, we report the results of PAN clustering, which considers both spaces. The ten red dots in Figure 4 are the ten clusterings PAN found. Each of these clusterings considers a different trade-off between the clustering qualities (measured via Silhouette index) in the objective and behavior spaces.

As mentioned in the previous section, to benchmark the performance of PAN, we compare the clusterings identified by PAN with the clusterings identified via (iterative)  $k$ -medoids clusterings. Of the ten  $k$ -medoids clusterings we performed (Figure 3), we only consider four clusterings that adhere to the criteria of valid partitioning. The four blue rectangles in Figure 4 represent the four  $k$ -medoids clusterings. First, from the figure, it is evident that the clusterings represented by blue points is either good in the objective space or the behavior space, but there are no compromise clusterings. In contrast, PAN offers a wider range of clusterings covering the full spectrum.

Each point in Figure 4 represents a partitioning of the solution set. Figure 5 shows how three different partitionings look like in the Objective Space. These partitionings represent 1) compromise clustering, 2) clustering with preference for behavior, and 3) clustering with



**Figure 3:** Sankey diagram showing similarity between clusters in objective and behavior space. Each clusters is a node and the links are policies.



**Figure 4:** Clusterings obtained by the PAN clustering (red dots) and iterative k-medoids clustering (blue rectangles).

preference for objective values, represented by Points X, B and O, respectively, in Figure 4. We show only two objectives (even though MO-Highway is a three-objective problem) as one of the objectives (Collision) does not offer any trade-offs.

Finally, Figure 6 compares PAN clusterings with behavior-only and objective-only clusterings in three cases:

**Compromise clustering** The top Sankey diagram shows a compromise clustering from PAN (Point X in Figure 4) in the middle, consisting of two clusters. The left and right columns show the clusters resulting from  $k$ -medoids clustering ( $k = 3$ ) in objective and behavior spaces, respectively.

**Clustering with preference for behavior** The bottom-left Sankey diagram shows a clustering from PAN which gives behavior the highest preference (Point B in Figure 4), consisting of two clusters, in the middle. The left and right columns are chosen from  $k$ -medoids clustering ( $k = 2$ ) similar to the above case.

**Clustering with preference for objective values** The bottom-right Sankey diagram shows a clustering from PAN which gives objective the highest preference (Point O in Figure 4), consisting of three clusters, in the middle. The left and right columns are chosen from  $k$ -medoids clustering ( $k = 3$ ) similar to the above cases.

Figure 6 also shows ARI values comparing PAN clustering with behavior space clustering, and PAN clustering with objective space clustering. From the diagrams and ARI values it is evident that choosing an extreme clustering from PAN aligns better with the corresponding space (objective/behavior space) than the other space (behavior/objective space). In contrast, the compromise clustering balances alignment with both spaces. Eventually, which of these clusterings to choose depends on the preference of the decision maker.

## 6 Discussion

We analyzed the snapshot videos produced by the highlights algorithm [4], and observed distinct policy behaviors—e.g., a tendency to decelerate during lane changes, maintaining a short distance between vehicles, and overtaking with marked caution (keeping too much distance between cars in front)—that are not apparent solely from the

trade-offs in the objective space. For comparative analysis, we selected two clusterings to observe variations in cluster configurations, as illustrated in Figures 5a and 5b visualized in the objective space.

Unsurprisingly, Figure 5b shows that the clusters are well-defined in the objective space for the given clustering. When comparing two clusterings (Point X and Point O), we observe differences in solution distribution solely between Cluster 0 and Cluster 1, while the policies in Cluster 2 remain consistent across both clustering options.

All three policies in Cluster 2 are notably fast and exhibit similar objective values. Upon reviewing the highlight videos, it is evident that two of these policies display similar behaviors, characterized by speed and flexibility during overtaking maneuvers. However, the third policy consistently maintains a significant distance from the preceding vehicle, indicating a more cautious approach to overtaking, which in the real-world can be dangerous. As Point X maintains the homogeneity of clusters in both spaces, this is acceptable.

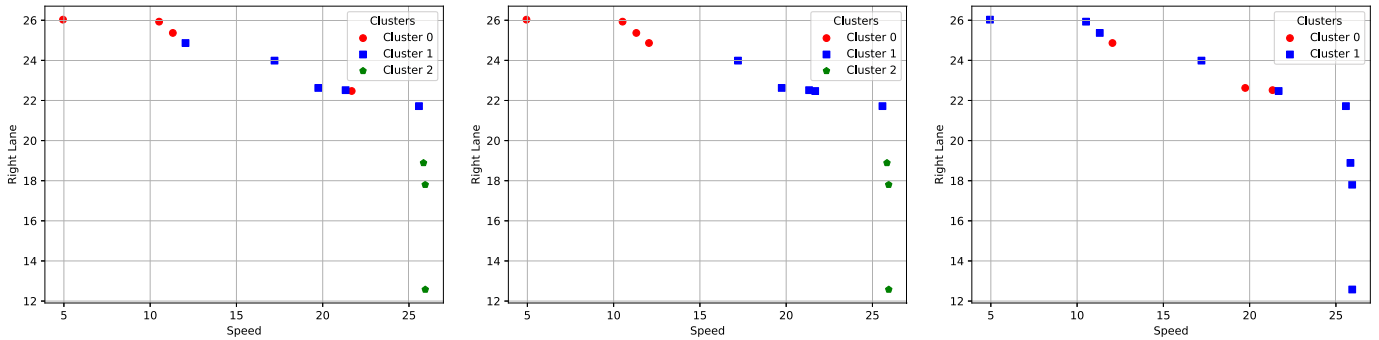
In Clusters 0 and 1, two policies (2 and 0) switch their cluster affiliations, reducing the homogeneity of the clusters in the objective space for Clustering X. Analysis of the highlight videos reveals that, for Clustering X, Cluster 1 comprises policies more inclined to overtake compared to Cluster 0. This suggests that while homogeneity in the objective space for Point X decreases, the categorization of policies based on their behavior improves.

By offering multiple clustering options, bi-objective clustering enables the selection of the most appropriate one for the problem. For instance, selecting a knee point partitioning—a point on the Pareto front balancing competing objectives where gains in one would mean disproportionate losses in another—ensures consideration of both spaces in cluster formation. Alternatively, if the DM has a preference for the number of clusters or one space over another, a suitable clustering can be chosen accordingly while also conveying information about the representation’s efficacy in both spaces. This approach not only enhances the DM’s awareness but also provides comprehensive information to support informed and explainable decision-making.

In our proposed approach, DMs are presented with clusters derived from compromise clustering. Each cluster is characterized by the average trade-offs and behavior patterns of the solutions it contains. These patterns can be illustrated through highlight videos of each cluster’s medoid or by identifying significant features within the highlights of the solutions in each cluster.

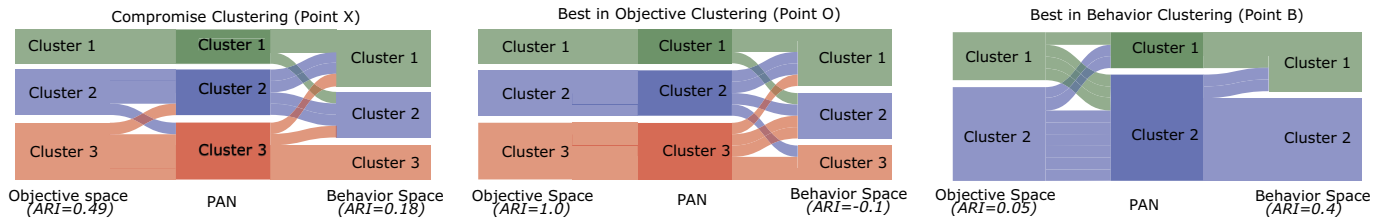
For the MO-Highway, we have analyzed these patterns through video examination; however, a more automated method is feasible through the assessment of feature importance derived from highlight’s state representations. We elucidate the distinct behaviors and effectiveness of separation for each cluster, empowering DMs with a comprehensive understanding of each cluster’s characteristics.

Depending on the role of the DM—non-technical DM, technical DM (analyst), or developer—this method can be used for different purposes. For non-technical decision makers, we suggest displaying only representative policies from each cluster, like the cluster’s



(a) Clusters of the policies in the objective space based on the middle clustering (Point X). (b) Clusters of the policies in the objective space based on the clustering with preference for objective values (Point O). (c) Clusters of the policies in the objective space based on the clustering with preference for behavior (Point B).

**Figure 5:** Clusters visualised in the objective space for chosen point in the Figure 4



**Figure 6:** Sankey diagrams illustrating the correspondence between clusters generated by PAN and those identified through  $k$ -medoids clustering for three chosen PAN clusterings. Three points chosen are: a compromise clustering (point X from Figure 4) and two clusterings representing the extreme values—best clustering in the objective space and the best clustering in the behavior space (Point O and Point B).

medoid, to allow straightforward selections based on their preferences and avoid overwhelming them. They can then implement these directly or as a stochastic mixture of the cluster’s policies [10]. For the technical DM, who is able to analyse solution set and understands how it was produced, it can help to identify trends within the clusters by analysing both spaces and the clusters. Additionally, regarding developers, this method holds the potential for enhancing interactive MORL algorithms, particularly during the preference-learning phase. By frequently soliciting DMs’ preferences in an interactive manner, the algorithm can more accurately tailor its focus to areas of interest, thereby optimizing the learning process and outcomes.

### 7 Conclusions and Future Directions

We address post-analysis of MORL-generated multi-dimensional solution sets to elucidate trade-offs and policy behaviors for decision makers tasked with policy analysis and implementation. We employ clustering as a means of reducing the complexity of a solution set by developing a partitioning strategy effective in both behavior and objective spaces, optimizing for two indices related to these spaces.

We demonstrate that effective partitioning in one space does not always correspond to similar results in the other, especially in MORL contexts where policies with different behaviors produce similar outcomes. We address the bi-objective challenge of partitioning across behavior and objective spaces by using PAN clustering to optimize two validity indices without merging them into a single measure, allowing decision makers to visualize and prioritize trade-offs between the two spaces for informed strategy selection.

We identify multiple avenues for future work. A critical next step is conducting user studies to validate the efficacy of the proposed methods as well as the best approach to identify representative clustering. These studies can investigate the utility of different cluster-

ing techniques by comparing their performance and user perception. Specifically, it is important to examine how well users can make informed decisions with the aid of additional information like policy behavior and trade-off highlights. In practice, this means presenting participants with problem descriptions and policies characterized by trade-offs, enriched with clustering and behavioral information, to evaluate which format improves decision-making effectiveness. This user-centric approach is pivotal in ensuring that the theoretical advancements in MORL translate into practical tools that enhance decision-making processes in real-world scenarios. Additional to that, we acknowledge that parameter choices for Highlights can be tuned specifically to each environment, which in turn, can yield better PAN clusterings. Another important direction is automating the analysis of highlight videos by detecting key features in highlighted states to enhance cluster descriptions, particularly as the number of policies grows. Finally, new methods for capturing policy behavior should be studied further. In this paper we focused on the clustering as a mean to reduce the solution set. However, there exist other approaches for reducing the size of the solution set, such as pruning methods [21], which should be explored further.

## References

- [1] L. N. Alegre, F. Felten, E.-G. Talbi, G. Danoy, A. Nowé, A. L. C. Bazzan, and B. C. da Silva. MO-Gym: A library of multi-objective reinforcement learning environments. In *Proceedings of the 34th Benelux Conference on Artificial Intelligence BNAIC/Benelearn 2022*, 2022.
- [2] L. N. Alegre, A. L. C. Bazzan, D. M. Roijers, A. Nowé, and B. C. da Silva. Sample-efficient multi-objective learning via generalized policy improvement prioritization. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS '23*, page 2003–2012, Richland, SC, 2023. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450394321.
- [3] L. N. Alegre, A. L. C. Bazzan, D. M. Roijers, A. Nowé, and B. C. da Silva. Sample-efficient multi-objective learning via generalized policy improvement prioritization. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS '23*, page 2003–2012, Richland, SC, 2023. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450394321.
- [4] D. Amir and O. Amir. Highlights: Summarizing agent behavior to people. In *Adaptive Agents and Multi-Agent Systems*, 2018. URL <https://api.semanticscholar.org/CorpusID:21755369>.
- [5] S. Bandaru, A. H. Ng, and K. Deb. Data mining methods for knowledge discovery in multi-objective optimization: Part a - survey. *Expert Systems with Applications*, 70:139–159, 2017. ISSN 0957-4174. doi: <https://doi.org/10.1016/j.eswa.2016.10.015>. URL <https://www.sciencedirect.com/science/article/pii/S0957417416305449>.
- [6] A. Castelletti, F. Pianosi, and M. Restelli. Tree-based fitted q-iteration for multi-objective markov decision problems. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2012. doi: 10.1109/IJCNN.2012.6252759.
- [7] J. G. Falcón-Cardona, H. Ishibuchi, and C. A. C. Coello. Riesz s-energy-based reference sets for multi-objective optimization. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2020. doi: 10.1109/CEC48606.2020.9185833.
- [8] F. Felten, L. N. Alegre, A. Nowé, A. L. C. Bazzan, E. G. Talbi, G. Danoy, and B. C. d. Silva. A toolkit for reliable benchmarking and research in multi-objective reinforcement learning. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*, 2023.
- [9] M. Giuliani, A. Castelletti, F. Pianosi, E. Mason, and P. M. Reed. Curses, tradeoffs, and scalable management: Advancing evolutionary multiobjective direct policy search to improve water reservoir operations. *Journal of Water Resources Planning and Management*, 142(2): 04015050, 2016.
- [10] C. F. Hayes, R. Rădulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Raymond, T. Verstraeten, L. M. Zintgraf, R. Dazeley, F. Heintz, et al. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.
- [11] S. H. Huang, K. Bhatia, P. Abbeel, and A. D. Dragan. Establishing appropriate trust via critical states. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3929–3936. IEEE, 2018.
- [12] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985. doi: 10.1007/BF01908075. URL <https://doi.org/10.1007/BF01908075>.
- [13] A. Jalalimanesh, H. S. Haghghi, A. Ahmadi, H. Hejazian, and M. Soltani. Multi-objective optimization of radiotherapy: distributed q-learning and agent-based simulation. *Journal of Experimental & Theoretical artificial intelligence*, 29(5):1071–1086, 2017.
- [14] I. Lage, D. Lifschitz, F. Doshi-Velez, and O. Amir. Exploring computational user models for agent policy summarization. In *IJCAI: proceedings of the conference*, volume 28, page 1401. NIH Public Access, 2019.
- [15] E. Leurent. An environment for autonomous driving decision-making. <https://github.com/eleurent/highway-env>, 2018.
- [16] C. Li and K. Czarnecki. Urban driving with multi-objective deep reinforcement learning. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19*, page 359–367, Richland, SC, 2019. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450363099.
- [17] D. Lizotte, M. Bowling, and S. Murphy. Efficient reinforcement learning with multiple reward functions for randomized controlled trial analysis. pages 695–702, 08 2010.
- [18] Y. Oh, A. Ullah, and W. Choi. Multi-objective reinforcement learning for power allocation in massive mimo networks: A solution to spectral and energy trade-offs. *IEEE Access*, 2023.
- [19] Z. Osika, J. ZatarainSalazar, D. M. Roijers, F. A. Oliehoek, and P. K. Murukannaiah. What lies beyond the Pareto front? A survey on decision-support methods for multi-objective optimization. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence, IJCAI '23*, pages 6741–6749, Macao, S.A.R., 2023.
- [20] Z. Osika, J. Zatarain-Salazar, F. A. Oliehoek, and P. K. Murukannaiah. Appendix: Navigating trade-offs: Policy summarization for multi-objective reinforcement learning, 2024. URL <https://zenodo.org/records/13354479>. Accessed: 2024-08-21.
- [21] S. Petchrompo, D. W. Coit, A. Brintrup, A. Wannakrairot, and A. K. Parlikad. A review of pareto pruning methods for multi-objective optimization. *Computers & Industrial Engineering*, 167: 108022, 2022. ISSN 0360-8352. doi: <https://doi.org/10.1016/j.cie.2022.108022>. URL <https://www.sciencedirect.com/science/article/pii/S0360835222000924>.
- [22] J. D. Quinn, P. M. Reed, M. Giuliani, and A. Castelletti. What is controlling our control rules? opening the black box of multireservoir operating policies using time-varying sensitivity analysis. *Water Resources Research*, 55(7):5962–5984, 2019. doi: <https://doi.org/10.1029/2018WR024177>. URL <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018WR024177>.
- [23] D. M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley. A survey of multi-objective sequential decision-making. *J. Artif. Int. Res.*, 48(1): 67–113, oct 2013. ISSN 1076-9757.
- [24] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.
- [25] E. Schubert and P. J. Rousseeuw. Faster k-medoids clustering: improving the pam, clara, and clarans algorithms. In *Similarity Search and Applications: 12th International Conference, SISAP 2019, Newark, NJ, USA, October 2–4, 2019, Proceedings 12*, pages 171–187. Springer, 2019.
- [26] S. Sreedharan, S. Srivastava, and S. Kambhampati. Tldr: Policy summarization for factored ssp problems using temporal abstractions. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, pages 272–280, 2020.
- [27] N. Topin and M. Veloso. Generation of policy-level explanations for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2514–2521, 2019.
- [28] L. Torrey and M. Taylor. Teaching on a budget: Agents advising agents in reinforcement learning. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13*, page 1053–1060, Richland, SC, 2013. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450319935.
- [29] T. Ulrich. Pareto-set analysis: Biobjective clustering in decision and objective spaces. *Journal of Multi-Criteria Decision Analysis*, 20(5-6): 217–234, 2013. doi: <https://doi.org/10.1002/mcda.1477>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/mcda.1477>.
- [30] K. Van Moffaert, M. M. Drugan, and A. Nowé. Sclarized multi-objective reinforcement learning: Novel design techniques. In *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 191–199, 2013. doi: 10.1109/ADPRL.2013.6615007.
- [31] W. Wang and M. Sebag. Hypervolume indicator and dominance reward based multi-objective monte-carlo tree search. *Machine Learning*, 92:403 – 429, 2013. URL <https://api.semanticscholar.org/CorpusID:1464172>.
- [32] J. Wu, Y. Ye, and J. Du. Multi-objective reinforcement learning for autonomous drone navigation in urban areas with wind zones. *Automation in Construction*, 158:105253, 2024.
- [33] Z. Xiong, B. Luo, B.-C. Wang, X. Xu, and T. Huang. Multi-objective battery charging strategy based on deep reinforcement learning. *IEEE Transactions on Transportation Electrification*, 2023.
- [34] J. Xu, Y. Tian, P. Ma, D. Rus, S. Sueda, and W. Matusik. Prediction-guided multi-objective reinforcement learning for continuous robot control. In *International conference on machine learning*, pages 10607–10616. PMLR, 2020.