Scalable Multi-Objective Reinforcement Learning with Fairness Guarantees using Lorenz Dominance

DIMITRIS MICHAILIDIS*, University of Amsterdam, Netherlands
WILLEM RÖPKE, Vrije Universiteit Brussel, Belgium
DIEDERIK M. ROIJERS, Vrije Universiteit Brussel & City of Amsterdam, Netherlands
SENNAY GHEBREAB, University of Amsterdam, Netherlands
FERNANDO P. SANTOS, University of Amsterdam, Netherlands

Multi-Objective Reinforcement Learning (MORL) aims to learn a set of policies that optimize trade-offs between multiple, often conflicting objectives. MORL is computationally more complex than single-objective RL, particularly as the number of objectives increases. Additionally, when objectives involve the preferences of agents or groups, incorporating fairness becomes both important and socially desirable. This paper introduces a principled algorithm that incorporates fairness into MORL while improving scalability to many-objective problems. We propose using Lorenz dominance to identify policies with equitable reward distributions and introduce λ-Lorenz dominance to enable flexible fairness preferences. We release a new, large-scale real-world transport planning environment and demonstrate that our method encourages the discovery of fair policies, showing improved scalability in two large cities (Xi'an and Amsterdam). Our methods outperform common multi-objective approaches, particularly in high-dimensional objective spaces.

JAIR Associate Editor: Ivor Tsang

JAIR Reference Format:
Dimitris Michailidis, Willem Röpke, Diederik M. Roijers, Sennay Ghebreab, and Fernando P. Santos. 2026. Scalable Multi-Objective Reinforcement Learning with Fairness Guarantees using Lorenz Dominance. *Journal of Artificial Intelligence Research* 85, Article 31 (March 2026), 32 pages. DOI: 10.1613/jair.1.19862

1 Introduction

Reinforcement Learning (RL) is a powerful framework for sequential decision-making, where agents learn to maximize long-term rewards by interacting with an environment (Wang et al. 2020). In most RL applications, rewards are constructed by aggregating multiple criteria (or objectives) into a single scalar value, typically via a weighted sum (Hayes et al. 2022). However, this approach assumes prior knowledge of the precise preferences among objectives—a condition that rarely holds in real-world settings. Moreover, many real-world problems inherently involve multiple, often conflicting, objectives. Defining a scalar reward function before training can therefore introduce bias into the learning process, potentially excluding policies that differ primarily in their objective weightings (Vamplew, Smith, et al. 2022).

*Corresponding Author.

Authors' Contact Information: Dimitris Michailidis, ORCID: 0000-0002-0106-1126, d.michailidis@uva.nl, University of Amsterdam, Amsterdam, Noord Holland, Netherlands; Willem Röpke, ORCID: 0000-0001-5045-6127, willem.ropke@vub.be, Vrije Universiteit Brussel, Brussels, Belgium; Diederik M. Roijers, ORCID: 0000-0002-2825-2491, diederik.roijers@vub.be, Vrije Universiteit Brussel & City of Amsterdam, Amsterdam, Noord Holland, Netherlands; Sennay Ghebreab, ORCID: 0009-0007-5788-4635, s.ghebreab@uva.nl, University of Amsterdam, Amsterdam, Noord Holland, Netherlands; Fernando P. Santos, ORCID: 0000-0002-2310-6444, f.p.santos@uva.nl, University of Amsterdam, Amsterdam, Noord Holland, Netherlands.

DOI: 10.1613/jair.1.19862

Multi-Objective Reinforcement Learning (MORL) addresses this challenge by defining a separate reward function for each objective (Hayes et al. 2022). This yields a set of candidate optimal policies, rather than a single solution, that decision-makers select according to their preferences. MORL has been successfully applied in various domains, such as decision-making under unknown preferences (Alegre, A. Bazzan, et al. 2022; Roijers et al. 2015), human-value alignment (Peschl et al. 2021; Rodriguez-Soto, Serramia, et al. 2022), robot locomotion (Cao and Zhan 2021), and multi-agent systems (Rădulescu et al. 2019; Röpke 2023; Vamplew, Smith, et al. 2022).

Single-policy MORL learns one policy based on predefined knowledge about decision-makers' preferences. However, such preferences are not always known at training time. Multi-policy methods in MORL handle unknown preferences by assuming a monotonically increasing utility function and optimizing all objectives simultaneously, approximating the Pareto front of optimal policies (Hayes et al. 2022; Mannion et al. 2021; Reymond, Bargiacchi, et al. 2022). Multi-policy methods face scalability challenges as the solution set can scale exponentially with the number of objectives. This issue becomes particularly severe in *many-objective* optimization, where the number of objectives is large (Nguyen et al. 2020; Perny et al. 2013). Consequently, multi-policy methods often struggle to scale efficiently in these scenarios, highlighting the need for further research in *many-objective RL* (Hayes et al. 2022).

Learning the entire Pareto front is often unnecessary, since some policies may be inherently undesirable (Osika, Salazar, et al. 2023). For example, in fairness-critical applications, some Pareto-non-dominated policies may result in unequal reward distributions across objectives. This is especially problematic when objectives represent the utilities of different societal groups (Cimpeana et al. 2023; Jabbari et al. 2017). Although egalitarian approaches such as maxmin or equal weighting can address this issue, they assume a predefined, exact preference over objectives, often limiting flexibility and sometimes yielding inefficient results (Siddique et al. 2020). This reveals a research gap in MORL: no current multi-policy method (a) guarantees fairness to the decision-maker, (b) allows control over fairness constraints, and (c) scales effectively to many-objective problems.

In this paper, we propose using Lorenz dominance to identify a subset of the Pareto front that ensures equitable reward distribution, without requiring predefined preferences. We extend this approach with λ-Lorenz dominance, enabling decision-makers to adjust the strictness of fairness constraints through a parameter λ. We formally show that λ-Lorenz dominance interpolates between Lorenz and Pareto dominance, providing decision-makers with fine-grained control over the degree of fairness. We also introduce Lorenz Conditioned Networks (LCN), a novel algorithm for optimizing λ-Lorenz dominance.

To support scalability in many-objective settings, we develop a new large-scale, multi-objective environment for planning transport networks in real-world cities with a flexible number of objectives. We conduct experiments in the cities of Xi'an (China) and Amsterdam (Netherlands) and show that LCN generates fair policy sets in large objective spaces. Since Lorenz-optimal sets are typically smaller than Pareto-optimal sets (Perny et al. 2013), LCN scales effectively, particularly in many-objective problems. We release the code and data used to generate our results alongside this paper [1].

2 Related Work

Our work intersects multi-policy methods for MORL (Hayes et al. 2022) and algorithmic fairness in sequential decision-making problems (Gajane et al. 2022).

2.1 Multi-Policy MORL

Early multi-policy methods in RL, such as Pareto Q-learning, were limited to small-scale environments (Moffaert and Nowé 2014; Parisi et al. 2016; Ruiz-Montiel et al. 2017). To improve scalability, many approaches assume linear decision-maker preferences, resulting in a simpler solution set called the convex coverage set (Abels

[1] GitHub repository: https://github.com/sias-uva/mo-transport-network-design

et al. 2019; Felten, Talbi, et al. 2024; Roijers et al. 2015). For example, GPI-LS—a state-of-the-art method and our baseline—decomposes the multi-objective problem into single-objective subproblems. Each subproblem uses a reward function that is a convex combination of the original vectorial reward. It then trains a neural network to approximate optimal policies for different weights (Alegre, A. L. C. Bazzan, et al. 2023). Beyond linear preferences, the Iterated Pareto Referent Optimization (IPRO) method uses a similar decomposition-based approach and provides strong theoretical guarantees. However, its performance degrades as the number of objectives increases (Röpke et al. 2024). In contrast, Pareto Conditioned Networks (PCNs) do not decompose the problem, but instead train a return-conditioned policy, (Delgrange et al. 2023; Reymond, Bargiacchi, et al. 2022; Reymond, Hayes, et al. 2022). PCNs have been applied in various domains, including water management (Osika, Radelescu, et al. 2025), pandemic intervention policies (M. Chen and Zilka 2025), autonomous cyber defence (O'Driscoll et al. 2025) and battery control (J. Hu et al. 2025). Similar to PCN, PD-MORL approximates the Pareto front by uniformly sampling preferences across the preference space (Basaklar et al. 2023), while C-MORL (Liu et al. 2024) bridges constrained policy optimization and MORL to efficiently discover the Pareto front through parallel policy training. We propose a method inspired by PCNs that focuses on fairness, avoiding the need to search the entire preference space. This enables scalability to higher dimensions, learning a set of fair policies and offering flexibility in setting the degree of fairness preference.

2.2 Fairness in MORL

Research on fairness in RL can be categorized along two main themes (Gajane et al. 2022): fairness in domains where individuals belong to protected groups (societal bias) and fairness in resource allocation problems (non-societal bias). Our work aligns closely with the first theme, focusing on the fair distribution of benefits among different societal groups. Group fairness has been studied in RL before, specifically in multi-agent scenarios (Ju et al. 2023; Satija et al. 2023), where agents learn individual policies. While we focus on single-agent RL, we assume that the agent's policies will affect groups of individuals, who may have conflicting preferences

Achieving fairness in RL often requires balancing multiple objectives. Many studies in this area incorporate diverse objectives into a single fairness-based reward function. This is typically achieved through linear reward scalarization (Blandin and Kash 2024; X. Chen et al. 2023; Rodriguez-Soto, Lopez-Sanchez, et al. 2021), nonlinear reward combinations, and welfare functions (e.g. the Generalized Gini Index) (Fan et al. 2023; X. Hu et al. 2023; Siddique et al. 2020), and other reward-shaping mechanisms (Kumar and Yeoh 2023; Mandal and Gan 2023; Yu et al. 2022; Zimmer et al. 2021). Alternatively, some methods adapt the reward function during training to satisfy fairness constraints (J. Chen et al. 2021). These approaches require encoding fairness principles into the reward functions a priori, requiring preference information before training. Our method avoids these assumptions.

Our work is closely related to (Cimpeana et al. 2023), which proposes a formal MORL fairness framework that encodes six fairness notions as objectives. The authors use PCNs to identify Pareto-optimal trade-offs among these fairness notions. While our method can be used within this framework, it differs by not predefining any specific fairness notion. Instead, it learns a set of non-dominated policies across all objectives, allowing the decision-maker to define their fairness criteria after training and select a policy accordingly.

Our method relies on Lorenz dominance, a criterion that favors policies with balanced reward distributions. Lorenz dominance has previously been used in multi-objective optimization methods (Bederina et al. 2024; Chabane et al. 2019; Fasihi et al. 2023), but its application in MORL has been limited. (Perny et al. 2013) first introduced the Lorenz criterion in Multi-Objective Markov Decision Processes (MOMDPs), providing much of the theoretical foundation we rely on this work. However, their experiments were limited to small-scale, randomly generated MOMDPs. Since then, some works inspired by the Lorenz curve have emerged; for example, (Siddique et al. 2020) uses the Generalized Gini Function to create a weighted sum in the Lorenz space for single-objective RL. Building on the framework of (Perny et al. 2013), we train a neural network to learn the full Lorenz front,

enabling flexible degrees of fairness for the decision-maker and demonstrating scalability to significantly larger and more realistic environments.

3 Preliminaries

In this section, we formally introduce Multi-Objective Reinforcement Learning (MORL) and Lorenz dominance.

3.1 Multi-Objective Reinforcement Learning

We consider reinforcement learning agents that interact with a Multi-Objective Markov Decision Process (MOMDP). An MOMDP is represented as a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathbf{P}, \mathcal{R}, \gamma \rangle$ consisting of a set of states $\mathcal{S}$, set of actions $\mathcal{A}$, transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, vector-based reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^d$, with $d \geq 2$ the number of objectives, and a discount factor $\gamma \in [0, 1)$. In an MOMDP, we consider deterministic policies $\pi : \mathcal{S} \to \mathcal{A}$, which map states to actions.

With vector-based rewards, there is generally no single optimal policy in the usual sense of scalar rewards (e.g. policy maximizing reward). Instead, we learn a set of optimal policies using a dominance criterion. In MORL, Pareto dominance is commonly applied, resulting in a solution set called the Pareto front.

Definition 1. (Pareto dominance) Consider two vectors $\boldsymbol{v}, \boldsymbol{v}' \in \mathbb{R}^d$. We say that $\boldsymbol{v}$ Pareto dominates $\boldsymbol{v}'$, denoted $\boldsymbol{v} \succ_P \boldsymbol{v}'$, when $\forall j \in \{1, \ldots, d\} : v_j \geq v'_j$ and $\boldsymbol{v} \neq \boldsymbol{v}'$.

In essence, $\boldsymbol{v}$ Pareto dominates $\boldsymbol{v}'$ when it is at least as good in all objectives and strictly better in at least one. For a set of vectors D, the Pareto front $\mathcal{F}(D)$ contains all non-Pareto-dominated vectors.

Definition 2. (Pareto front) Given a set of vectors $D \subseteq \mathbb{R}^d$, the Pareto front $\mathcal{F}(D)$ is the subset of vectors in D that are not Pareto-dominated by any other vector in D. Formally,

$$\mathcal{F}(D) = \{\boldsymbol{v} \in D \mid \nexists\, \boldsymbol{v}' \in D \text{ such that } \boldsymbol{v}' \succ_P \boldsymbol{v}\}.$$

3.2 Fairness in Many-Objective Reinforcement Learning

One common fairness approach in Multi-Objective Reinforcement Learning (MORL) is to treat all objectives as equally important, optimizing a single, equally weighted objective. However, this assumes absolute equality in reward distribution, which may be infeasible in certain problems, and yields a single policy, without offering options to the decision-maker. Another approach, inspired by Rawlsian justice theory and the *maxmin* principle, focuses on maximizing the minimum reward between objectives. However, this often results in solutions that are not efficient for all users (Siddique et al. 2020).

To train a multi-policy algorithm with fair trade-offs, it is necessary to identify all optimal trade-offs that achieve a fair distribution of rewards. To achieve this, we use Lorenz dominance, a refinement of Pareto dominance that considers the distribution of values within a vector (Perny et al. 2013). This concept, traditionally used in economics to assess income inequality (Shorrocks 1983), is adapted here for fairness in MORL.

Definition 3. (Lorenz dominance) Let $L(\boldsymbol{v})$ be the Lorenz vector of a vector $\boldsymbol{v} \in \mathbb{R}^d$, defined as follows:

$$L(\boldsymbol{v}) = \left(v_{(1)}, v_{(1)} + v_{(2)}, \cdots, \sum_{i=1}^{d} v_{(i)}\right), \tag{1}$$

where $v_{(1)} \leq v_{(2)} \leq \cdots \leq v_{(d)}$ are the values of the vector $\boldsymbol{v}$, sorted in increasing order. A vector $\boldsymbol{v}$ Lorenz dominates a vector $\boldsymbol{v}'$, when its Lorenz vector $L(\boldsymbol{v})$ Pareto dominates the Lorenz vector $L(\boldsymbol{v}')$ (Perny et al. 2013). We use $\boldsymbol{v} \succ_L \boldsymbol{v}'$ to denote that $\boldsymbol{v}$ Lorenz dominates $\boldsymbol{v}'$.

For a set of vectors D, the Lorenz front $\mathcal{L}(D)$ contains all vectors that are non-Lorenz-dominated.

Definition 4. (Lorenz front) Given a set of vectors $D \subseteq \mathbb{R}^d$, the Lorenz front $\mathcal{L}(D)$ is the subset of vectors in D that are not Lorenz-dominated by any other vector in D. Formally,

$$\mathcal{L}(D) = \{\boldsymbol{v} \in D \mid \nexists\, \boldsymbol{v}' \in D \text{ such that } L(\boldsymbol{v}') \succ_P L(\boldsymbol{v})\},$$

where $L(\boldsymbol{v})$ denotes the Lorenz vector of $\boldsymbol{v}$, and $\succ_P$ is the Pareto dominance relation.

Our approach builds on the Pigou-Dalton transfer principle from economics (Adler 2013; Perny et al. 2013). This principle states that a redistribution of value from a better-off to a worse-off component improves fairness, as long as it does not reverse their ranking. Formally, given a reward vector $\boldsymbol{v} \in \mathbb{R}^d$ with two components $v_i > v_j$ for some indices i and j, transferring a small amount ϵ ($0 < \epsilon \leq v_i - v_j$) from v_i to v_j yields a new vector $\boldsymbol{v}' = \boldsymbol{v} - \epsilon I_i + \epsilon I_j$, where $I_i \in \mathbb{R}^d$ is an *indicator vector* with a 1 in the ith position and 0 elsewhere (and similarly for I_j). This transformation preserves the total reward and ranking, but results in a more equitable distribution, and is therefore called a Pigou-Dalton transfer (Perny et al. 2013).

Lorenz-based fairness evaluates policies based on how equitably rewards are distributed across objectives. A solution is considered fairer if it can be obtained via a sequence of Pigou-Dalton transfers from another, implying that it Lorenz-dominates the other. For example, consider $\boldsymbol{v} = (8, 0)$. A transfer of $\epsilon = 3$ yields $\boldsymbol{v}' = (5, 3)$. Even though the total reward remains 8, and the rank between entries is preserved, $\boldsymbol{v}'$ is considered fairer under Lorenz dominance.

In MORL, we define vectors $\boldsymbol{v}^{\pi}, \boldsymbol{v}^{\pi'}$ as the expected return of the policies π, π', across all objectives of the environment, respectively. We define fair policies as those that are non-Lorenz-dominated. The set of non-dominated value vectors is called a *Lorenz coverage set*, which is usually (but not necessarily) significantly smaller than a Pareto coverage set (Perny et al. 2013). Our fairness approach satisfies the criteria outlined in (Siddique et al. 2020). It is Lorenz-*efficient* as the learned policies are non-Lorenz-dominated; it is *impartial*, since it treats all objectives as equally important, and it is *equitable*, as Lorenz dominance satisfies the Pigou-Dalton principle (Perny et al. 2013). In Figure 1, we show the difference between Pareto and Lorenz dominance. The latter extends the area of undesired solutions, allowing for fewer non-dominated solutions and providing fairness guarantees in the two objectives.

4 Flexible Fairness with λ-Lorenz Dominance

To give decision-makers fine-grained control over the fairness needs of their specific problem, we introduce a novel criterion, called λ-Lorenz dominance. λ-Lorenz dominance operates directly on the return vectors, without objective weights (such as the Generalized Gini Index (Siddique et al. 2020)). By selecting a single parameter $\lambda \in [0, 1]$, λ-Lorenz dominance allows decision-makers to balance Pareto and Lorenz optimality. We formally define λ-Lorenz dominance in Definition 5.

Definition 5 (λ-Lorenz dominance)**.** Let $\sigma(\boldsymbol{v})$ be the vector $\boldsymbol{v}$ sorted in increasing order. Given $\lambda \in [0, 1]$, a vector $\boldsymbol{v}$ λ-Lorenz dominates another vector $\boldsymbol{v}'$, denoted $\boldsymbol{v} \succ_\lambda \boldsymbol{v}'$ if,

$$\lambda\sigma(\boldsymbol{v}) + (1-\lambda)L(\boldsymbol{v}) \succ_P \lambda\sigma(\boldsymbol{v}') + (1-\lambda)L(\boldsymbol{v}'). \tag{2}$$

Intuitively, for $\lambda = 1$, it is assumed that the decision-maker cares equally about all objectives and thus may reorder them. This relaxation allows some non-Pareto-dominated vectors to become dominated. Consider, for example, $(4, 2)$ and $(1, 3)$. While no vector is Pareto-dominated, reordering the objectives in increasing order yields $(2, 4)$ and $(1, 3)$, in which case the second vector is now dominated. This approach may already reduce the size of the Pareto front, but does not yet achieve the same fairness constraints imposed by the Lorenz front. At the other extreme, setting $\lambda = 0$ ensures that the solution set is equal to the Lorenz front.

The λ-Lorenz front of a set D, denoted $\mathcal{L}(D; \lambda)$, contains all vectors that are pairwise non-λ-Lorenz-dominated. In Theorem 1, we formally show that the λ-Lorenz fronts form increasing nested sets as λ varies from 0 to 1:

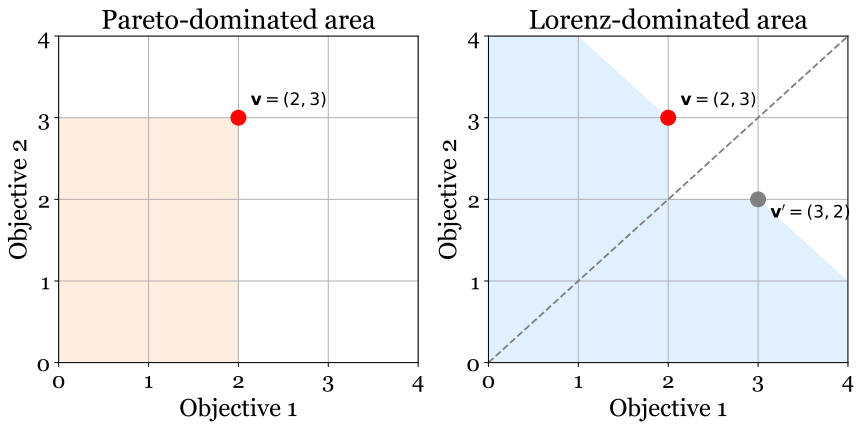

Fig. 1. The Pareto and Lorenz-dominated areas of vector $\boldsymbol{v}$. The Lorenz-dominated area includes the Pareto-dominated area, and is symmetric around the equality line, except for the symmetric vector $\boldsymbol{v}' = (3, 2)$. This creates an expanded dominance, resulting in fewer acceptable trade-offs.

lower λ values mean increasingly selective coverage sets that interpolate between the Pareto and Lorenz fronts. The full proof is provided in the supplementary material.

Theorem 1. *$\forall \lambda_1, \lambda_2 : 0 \leq \lambda_1 \leq \lambda_2 \leq 1$ and $\forall D \subset \mathbb{R}^d$ the following relations hold.*

$$\mathcal{L}(D) \subseteq \mathcal{L}(D; \lambda_1) \subseteq \mathcal{L}(D; \lambda_2) \subseteq \mathcal{F}(D). \tag{3}$$

Proof sketch. To prove Theorem 1, we provide three auxiliary results. First, we demonstrate that for all parameters $\lambda \in [0, 1]$ and vectors $\boldsymbol{v}, \boldsymbol{v}' \in \mathbf{R}^d$ we have that:

$$\boldsymbol{v} \succ_\lambda \boldsymbol{v}' \implies \boldsymbol{v} \succ_L \boldsymbol{v}'. \tag{4}$$

Together with some algebra, this result is subsequently used to show that for all parameters λ_1 and λ_2 such that $0 \leq \lambda_1 \leq \lambda_2 \leq 1$,

$$\boldsymbol{v} \succ_{\lambda_2} \boldsymbol{v}' \implies \boldsymbol{v} \succ_{\lambda_1} \boldsymbol{v}'. \tag{5}$$

Finally, we extend (Perny et al. 2013, Theorem 1) to show that Pareto dominance implies λ-Lorenz dominance as well. These components are combined to obtain the desired result. □

It is a straightforward corollary that for $\lambda = 1$, $\mathcal{L}(D; \lambda) \subseteq \mathcal{F}(D)$ while for $\lambda = 0$, $\mathcal{L}(D; \lambda) = \mathcal{L}(D)$. In Figure 2 (C) we illustrate conceptually how the λ controls the size of the coverage set to consider.

5 Lorenz Conditioned Networks

Lorenz Condition Networks (LCNs) are an adaptation of Pareto Conditioned Networks (PCNs) that aim to learn policies on the λ-Lorenz front. We use both the abbreviations PCN and PCNs (and likewise LCN and LCNs) interchangeably, depending on whether we refer to the framework as a whole (singular) or to the family of networks (plural). We begin this section by giving an overview of PCNs and identifying their drawbacks.

5.1 Background: Pareto Conditioned Networks (PCN)

A PCN is a supervised learning network designed for Multi-Objective Reinforcement Learning (MORL) (Reymond, Bargiacchi, et al. 2022). It enables a single neural network to learn a diverse set of policies that approximates the Pareto front of optimal trade-offs between objectives. A PCN is a conditioned network, meaning that it is trained to take as input both the environment state and a desired return vector, and output a probability distribution over actions.

Formally, the PCN policy is parameterized as: $\pi_\theta(a_t \mid s_t, \hat{h}_t, \hat{\boldsymbol{R}}_t)$, where s_t is the current state, $\hat{h}_t$ is the desired horizon, and $\hat{\boldsymbol{R}}_t$ is a desired return vector over the d objectives. The network is trained using supervised learning on transitions collected during exploration. Each transition includes the state, the action taken, and the return vector obtained from the trajectory, allowing the network to learn by imitating non-dominated behavior conditioned on return goals.

During training, PCN incrementally improves the quality of collected experiences by perturbing non-dominated return vectors in the experience replay (ER) buffer. A new desired return $\hat{\boldsymbol{R}}_t$ is sampled by adding noise to an existing non-dominated point (proportional to the standard deviation of current returns), which serves as a target for generating new exploratory trajectories. This process helps expand the diversity of collected policies and encourages coverage of the Pareto front.

To filter the ER buffer, PCN introduces a filtering heuristic that favors experiences closer to the Pareto front while preserving diversity. This is achieved by computing the Euclidean distance of each experience to the Pareto front (to prioritize proximity), and a crowding distance (Deb, Agrawal, et al. 2000) (to encourage spread in objective space), and then applying penalties to overrepresented areas.

Despite its effectiveness in approximating the Pareto front, PCN suffers from two key drawbacks:

(1) Lack of fairness control — PCN prioritizes Pareto optimality without considering fairness or equitable reward distribution across objectives, potentially favoring extreme, imbalanced solutions and leading to intractable learning in large state and objective spaces.
(2) Experience Replay volatility — As new experiences are collected, many older points are replaced or re-evaluated, causing instability during training.

LCNs belong to the same family of reward conditioned networks, also referred to as upside-down reinforcement learning, where a policy is trained as a single neural network through supervised learning (Kumar, Peng, et al. 2019; Reymond, Bargiacchi, et al. 2022). An LCN network learns multiple policies, each representing a Lorenz-optimal trade-off.

5.2 LCN Network

Just like PCN, LCN uses a single neural network to learn a policy $\pi_\theta(a_t|s_t, \hat{h}_t,\ \hat{\boldsymbol{R}}_t)$, which maps the current state s_t, the desired horizon $\hat{h}_t$ and the desired return $\hat{\boldsymbol{R}}_t$ to the next action a_t. Note that $\hat{\boldsymbol{R}}_t$ is a vector with dimension d equal to the number of objectives. The network receives an input tuple $\langle s_t, \hat{h}_t, \hat{\boldsymbol{R}}_t \rangle$ and returns a probability distribution over the set of potential next actions. It is trained with supervised learning on samples collected by the agent during exploration. The network updates its parameters using a cross-entropy loss function:

$$\mathcal{J}(y, \pi) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{a \in A} y_a^{(i)} \log \pi \left(a_t^{(i)} | s_t^{(i)}, h_t^{(i)}, \boldsymbol{R}_t^{(i)} \right), \tag{6}$$

where N is the batch size, $y_a^{(i)}$ is the i-th sample action taken by the agent (ground truth), $y_a^{(i)} = 1$ if $a_t = a$ and 0 otherwise and $\pi(a_t^{(i)}|s_t^{(i)}, h_t^{(i)}, \boldsymbol{R}_t^{(i)})$ represents the predicted probability of action a for the i-th sample, conditioned on its specific state $s_t^{(i)}$, horizon $h_t^{(i)}$, and return $\boldsymbol{R}_t^{(i)}$.

The training process involves sampling collected, non-dominated experiences and then training the policy with supervised learning to imitate these experiences. Given a sufficient number of good experiences, the agent will learn good policies.

5.3 Collecting Experiences

LCN learns the policy network π_θ by collecting experiences and storing them in an ER buffer. These experiences are then used to train the policy via supervised learning (Kumar, Peng, et al. 2019; Reymond, Bargiacchi, et al. 2022). Because action selection involves conditioning the network on a specified return, the primary mechanism for collecting higher-quality experiences is to iteratively improve the desired return used as the conditioning input.

To achieve this improvement, similarly to PCN, a non-dominated return is randomly sampled from the current non-dominated experiences in the ER buffer. This sampled return is then increased by a value drawn from a uniform distribution $U(0, \sigma_o)$, where σ_o represents the standard deviation of all non-dominated points in the ER buffer (Reymond, Bargiacchi, et al. 2022). The updated return is subsequently used as the input $\hat{\boldsymbol{R}}_t$ for the policy network. Through this iterative process—refining the condition, collecting improved experiences, and training the policy network on non-dominated experiences—the network progressively learns to approximate all non-dominated trade-offs, forming a Lorenz coverage set. To ensure that the ER buffer contains experiences that will contribute most to performance improvement, however, the buffer must be filtered to retain only the most useful experiences.

5.4 Filtering Experiences

PCN improves the experience replay buffer by filtering out experiences that are far away from the currently approximated Pareto front. This is done by calculating the distance of each collected experience to the closest non-Pareto-dominated point in the buffer. In addition, a *crowding distance* is calculated for each point, measuring its distance to its closest neighbors (Deb, Agrawal, et al. 2000). Points with many neighbors have a high crowding distance and are penalized, ensuring that ER experiences are distributed throughout the objective space (Reymond, Bargiacchi, et al. 2022). With this approach, the set of Pareto-optimal solutions can grow exponentially with the number of states and objectives (Perny et al. 2013), and maintaining a good ER buffer becomes a great challenge. This can be intuitively understood by a simple example provided in (Perny et al. 2013): consider a deterministic MOMDP with $N+1$ states, where each non-terminal state allows two actions with different two-objective reward vectors. The terminal state is absorbing, and assume a discount factor $\gamma = 1$. There are 2^{N+1} possible stationary deterministic policies, and exactly 2^N of them result in distinct value vectors at the initial state. These vectors take the form $(x, 2^N - 1 - x)$ for $x = 0, 1, \ldots, 2^N - 1$, and all lie on the Pareto front since improving one objective necessarily worsens the other. Therefore, the number of Pareto-optimal value vectors grows exponentially with N, the number of non-terminal decision points.

This explosion is a consequence of the definition of Pareto dominance, which treats all objectives as equally important and assumes no prior knowledge of user preferences. Any improvement in any objective is valued, resulting in a broad set of solutions. However, in many practical applications, such a solution set may be unnecessary or even undesirable. For instance, when objectives correspond to benefits for distinct individuals or groups (and we know that fairness is preferred), some Pareto-optimal solutions may be undesirable, as they result in highly unequal outcomes.

It is important to note that this issue is not exclusive to Pareto dominance; other dominance relations, such as Lorenz dominance, can theoretically also produce exponentially large solution sets. Empirically, for a fixed number of objectives, approximating a Lorenz-optimal set within an error bound from the Lorenz front can be computed in polynomial time on the size of the state and objective spaces. We provide empirical results

supporting this claim in Section 7. This aligns with the intuition that stricter dominance criteria, such as those that prioritize fairness, tend to reduce the number of acceptable solutions by implicitly encoding preference information without relying on explicit weights. This distinction is crucial for experience filtering: when prior knowledge about user preferences (e.g., fairness) is available, employing stricter dominance criteria can produce more computationally manageable ER buffers.

In Lorenz Conditioned Networks (LCN), the evaluation of each experience e_i in the Experience Replay buffer $\mathcal{B}$ is determined by its proximity to the nearest *non-Lorenz-dominated* point $l_j \in \mathcal{L}(\mathcal{B}) \subseteq \mathcal{B}$. Here, the proximity between experiences is measured in the objective space, where each sampled experience e_i is represented by its actual reward vector (not the condition used to generate the experience). Thus, the nearest non-Lorenz-dominated point is the one with the smallest distance in this space. In Figure 2 (A) we show an example of this distance calculation. To ensure meaningful distance comparisons across potentially differently scaled objectives, all objective vectors in the buffer are normalized to the $[0, 1]$ range before computing distances. We denote the distance between an experience e_i and a reference point t_i as $d_{e_i,t_i} = ||e_i - t_i||_2$, where $t_i = \min ||e_i - l_j||_2$ is the nearest non-Lorenz-dominated point, which we refer to as *reference point*. We formalize the final distance for the evaluation $d_{\text{Lorenz},i}$ as follows:

$$d_{\text{Lorenz},i} = \begin{cases} d_{e_i,t_i} & \text{if } d_{cd,i} > \tau_{cd} \\ \rho_{pen}(d_{e_i,t_i} + c) & \text{if } d_{cd,i} \leq \tau_{cd} \end{cases} \tag{7}$$

Where d_{cd} is the crowding distance of i and τ_{cd} is the crowding distance threshold. A constant penalty c is added to the points below the threshold, whose distance is additionally penalized by a penalty multiplier ρ_{pen}. For the experiments in this paper, we set $\rho_{pen} = 2$ and provide additional sensitivity analysis on Appendix D. The points in the ER buffer are sorted based on d_{Lorenz}, and those with the highest get replaced first when a better experience is collected. The threshold τ_{cd} separates serves to break high-density regions in the objective space and ensures diversity. Experiences with a crowding distance below this threshold lead to too similar policies and are therefore penalized. While this heuristic introduces a discontinuity at the threshold τ_{cd}, in practice this acts as a strong bias toward maintaining diversity in the buffer, which benefits stable training.

5.5 Improving the Filtering Mechanism with Reference Points

The nearest-point filtering method employed by previous works has two drawbacks. Firstly, during exploration, stored experiences undergo significant changes as the agent discovers new, improved trajectories. This leads to a volatile ER buffer and moving targets, posing stability challenges during supervised learning. Secondly, we know in advance that certain experiences, even if non-dominated, are undesirable due to their unfair distribution of rewards.

Consider, for example, vectors: $\boldsymbol{v} = (8, 0), \boldsymbol{w} = (3, 4)$ and their corresponding Lorenz vectors $L(\boldsymbol{v}) = (0, 8), L(\boldsymbol{w}) = (3, 7)$. Both $\boldsymbol{v}$ and $\boldsymbol{w}$ are non-Lorenz-dominated, and would typically be used as targets for evaluating other experiences. However, $\boldsymbol{v}$ is not a desirable target due to its unfair distribution of rewards (this is essentially a limitation of Lorenz dominance, when one objective is very large). To address these issues, we adopt the concept of reference points from Multi-Objective Optimization (Cheng et al. 2016; Deb and Jain 2014; Felten, Talbi, et al. 2024), and propose reference points for filtering the experiences. We introduce two reference point mechanisms: a **redistribution** mechanism and a **mean** reference point mechanism. Both of these reference points are optimistic, meaning that the agent seeks to minimize the distance to them.

5.5.1 Redistributed Reference Point (LCN-Redist). This mechanism draws inspiration from the Pigou-Dalton principle we introduced in Section 3.2. Under this axiomatic principle, any experience in the ER buffer can be adjusted to provide a more desirable one. We identify the experience with the highest sum of rewards and evenly

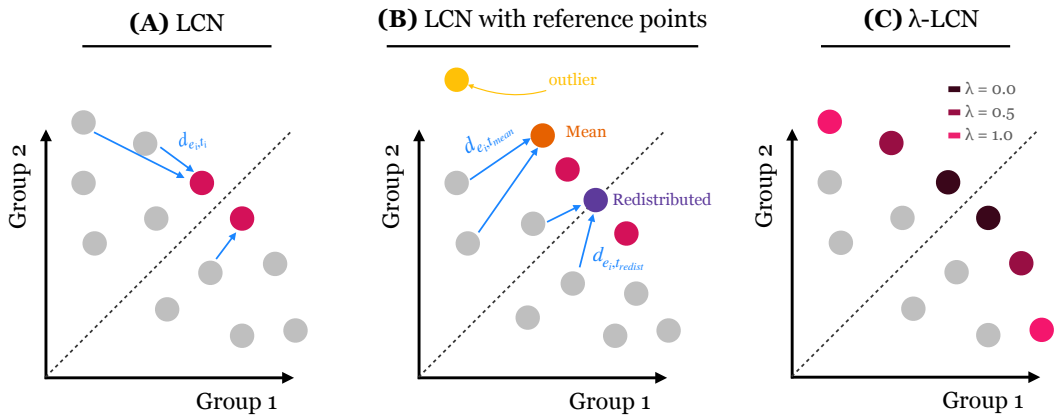

Fig. 2. Lorenz Conditioned Networks (LCNs) is a multi-policy method that offers fair trade-offs between different objectives. In this illustrative example, each group (Group 1 and Group 2) corresponds to a distinct objective related to group satisfaction (e.g. satisfied transportation demand). **(A)** Standard LCNs learn policies that balance objectives by exploring the trade-off space between them. **(B)** Reference points help accelerate training by filtering the Experience Replay buffer and guiding learning toward desirable solutions, while reducing the influence of outliers. **(C)** λ-LCN introduces flexibility in the fairness preferences, enabling the relaxation of fairness constraints to accommodate more diverse policies.

distribute the total reward across all dimensions of the vector. This is then assigned as the new *reference point* t for all experiences $e \in \mathcal{B}$:

$$t_{\text{redist}} = \frac{1}{n} \sum_{i=1}^{n} \left(\arg\max_{e \in D} \sum_{j=1}^{n} e_j \right)_i . \tag{8}$$

Note that t is now the same for all $e \in \mathcal{B}$. Subsequently, we measure the distances of all $e \in \mathcal{B}$ to this reference point and filter out those farthest from it, according to Equation 7 (replace t_i with t_{redist}). In Figure 2 (B), we illustrate this transfer mechanism.

5.5.2 Non-Dominated Mean Reference Point (LCN-Mean). Additionally, we propose an alternative, simpler reference point mechanism: a straightforward averaging of all non-Lorenz-dominated vectors in the experience replay (ER) buffer. This approach provides a non-intrusive method for incorporating collected experiences, while simultaneously smoothing out outlier non-dominated points. The reference point, denoted as t_{mean}, is defined as:

$$t_{\text{mean}} = \frac{1}{|\mathcal{L}(B)|} \sum_{l_j \in \mathcal{L}(B)} l_j, \tag{9}$$

where $\mathcal{L}(B) = \{l_1, l_2, \ldots, l_j\}$ represent the set of non-Lorenz-dominated experiences in the ER buffer. In Figure 2 (B) we show how this approach defines a reference point.

6 A Large Scale Many-Objective Environment

Existing discrete MORL benchmarks are often small-scale, with small state-action spaces or low-dimensional objective spaces (Lopez et al. 2018; Vamplew, Dazeley, et al. 2011). In addition, they do not cover allocation of resources such as public transport, where fairness in the distribution is crucial. We introduce a novel and

modular MORL environment, named the Multi-Objective Transport Network Design Problem (MO-TNDP). Built on MO-Gymnasium (Alegre, Felten, et al. 2022), the MO-TNDP environment simulates public transport design in cities of varying sizes and morphologies, addressing TNDP, an NP-hard optimization problem aiming to generate a transport line that maximizes the satisfied travel demand (Farahani et al. 2013).

In MO-TNDP, a city is represented as $H^{m \times n}$, a grid with equally sized cells. The mobility demand forecast between cells is captured by an Origin-Destination (OD) flow matrix $OD^{|H| \times |H|}$. Each cell $h \in H^{n \times m}$ is associated with a socioeconomic group $g \in \mathcal{R}$, which determines the dimensionality of the reward function. In this paper, we scale it from 2 to 10 groups (objectives).

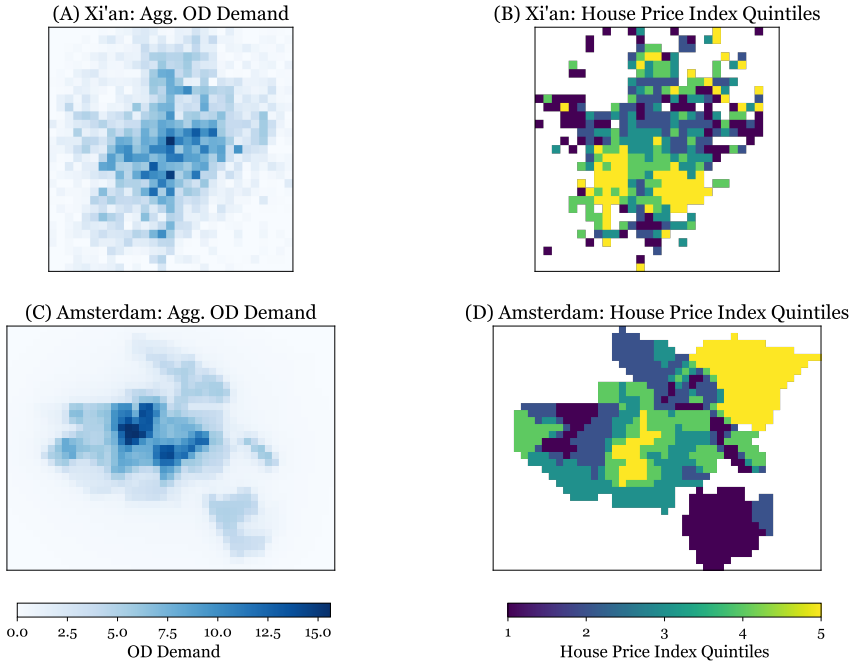

Fig. 3. Two real-world instances of the MO-TNDP environment in Xi'an (China) (Wei et al. 2020) and Amsterdam (Netherlands). (A) and (C) show the aggregate Origin-Destination Demand per cell (sum of incoming and outgoing flows) for Xi'an (A) and Amsterdam (C). (B) and (D) show the group membership of each cell, based on the house price index quintiles for Xi'an (B) and Amsterdam (D).

Episodes last a predefined number of steps. The agent traverses the city, connecting grid cells with eight available actions (movement in all directions). At each time step, the agent receives a vectorial reward of dimension $|\mathcal{R}|$, each corresponding to the percent satisfaction of the demand of each group. We formulate it as an MOMDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathbf{P}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ is the current location of the agent, $\mathcal{A}$ is the next direction of movement, and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}^d$ is the additional demand satisfied by taking the last action for each group.

Given the discrete and episodic nature of this particular problem, we set the discount factor γ to 1. The transition function $\mathbf{P}$ is deterministic, and each episode starts in the same state.

Additional directional constraints can be imposed on the agent's action space. The environment code enables developers to modify the city object, incorporating adjustments to grid size, OD matrix, cell group membership,

and directional constraints, making it adaptable to any city. It supports both creating transport networks from scratch and expanding existing ones. MO-TNDP is available online[2].

7 Experiments

To evaluate our methods and compare them against baselines, we conduct experiments in two environments. One of these is the *Deep Sea Treasure* environment — a widely used benchmark in Multi-Objective Reinforcement Learning (MORL) (Vamplew, Dazeley, et al. 2011). In this environment, an agent pilots a submarine through a grid-based ocean map, aiming to collect treasures located at various depths. The agent must navigate a trade-off between two conflicting objectives: maximizing treasure value and minimizing fuel consumption. Movement consumes fuel, while treasures located deeper in the sea yield higher rewards. This setup creates a trade-off between short, low-value paths and longer, more rewarding ones. Notably, because the time objective is modeled as a negative reward (i.e., a cost), only the baseline Linear Combination of Normals (LCN) method is applicable in our experiments.

In MO-TNDP, which we described in Section 6, we run experiments on two cities: Xi'an in China (841 cells, 20 episode steps) and Amsterdam in the Netherlands (1645 cells, 10 episode steps). Group membership for cells is determined by the average house price, which is divided into 2-10 equally sized buckets. Figure 3 illustrates two instances of the MO-TNDP environment for five objectives (a map of group membership for all objectives is provided in the supplementary material). LCNs are built using the MORL-Baselines library and the code is attached as supplementary material (Felten, Alegre, et al. 2023).

Through a Bayesian hyperparameter search of 100 runs, we tuned the batch size, learning rate, ER buffer size, number of layers, and hidden dimension across all reported models, environments, and objective dimensions (details in the supplementary material). We compare LCN with two state-of-the-art multi-policy baselines: PCN (Reymond, Bargiacchi, et al. 2022) and GPI-LS (Alegre, A. L. C. Bazzan, et al. 2023), on widely used MO evaluation metrics. To fairly compare them, we trained all algorithms for a maximum of 30,000 steps.

7.1 Evaluation Metrics

We use three common Multi-Objective (MO) evaluation metrics to compare our method with the baselines: one axiomatic metric (hypervolume) and two utility-based metrics (expected utility and Sen welfare). Axiomatic metrics evaluate the quality of a solution set by assuming that the true Pareto front represents the optimal solution. They do so without requiring any knowledge of the decision-maker's preferences over objectives, instead focusing purely on the Pareto-optimality and diversity of the solutions (Hayes et al. 2022). In contrast, utility-based metrics assume that the decision-maker has preferences over the objectives. These metrics either encode a specific utility function or assume a distribution over (or class of (Zintgraf et al. 2015)) possible utility functions to evaluate the solutions (Hayes et al. 2022).

Hypervolume: an axiomatic metric that measures the volume of a set of points relative to a specific reference point and is maximized for the Pareto front. In general, it assesses the quality of a set of non-dominated solutions, its diversity, and spread (Hayes et al. 2022). It's defined as:

$$\mathtt{HV}(CS, \boldsymbol{v}_{ref}) = \mathtt{Volume}\left(\bigcup_{\pi \in CS} \left[\boldsymbol{v}_{ref}, \boldsymbol{v}^{\pi}\right]\right), \tag{10}$$

where CS is the set of non-dominated policies, $\mathtt{Volume}(\cdot)$ computes the Lebesgue measure of the input space, and $[\boldsymbol{v}_{ref}, \boldsymbol{v}^{\pi}]$ is the box spanned by the reference and policy value.

[2]Githup repository: https://github.com/dimichai/mo-tndp

Expected Utility Metric (EUM): a utility-based metric that measures the expected utility of a given set of solutions, under some assumed distribution of utility functions. Since the true utility function of the decision-maker is unknown, we sample utility functions from a prior distribution and evaluate, for each sampled utility, the maximal score within the solutions set. The EUM score is then computed as the average utility across all sampled decision-makers. Specifically, in this paper, we assume that utilities can be expressed as linear combinations of the objective values, i.e., $u(\boldsymbol{v}^{\pi}) = w^{\top}\boldsymbol{v}^{\pi}$, where w is a non-negative weight vector representing a particular preference trade-off among objectives. For each weight vector w: $w_i \geq 0$ and $\sum_i w_i = 1$, a standard weighted-sum multi-objective utility function. We sample a set of 50 well-spaced weight vectors via the Riesz s-Energy method, which ensures diversity by spreading the samples as evenly as possible over the space of valid utility weights (see (Blank et al. 2021) for details). EUM measures the actual expected utility of the policies, and is more interpretable compared to axiomatic approaches like the hypervolume (Hayes et al. 2022). The score is defined as:

$$\texttt{EUM}(CS) = \mathbb{E}_{P_u}\left[\max_{\pi \in CS} u(\boldsymbol{v}^{\pi})\right], \tag{11}$$

where CS is the set of non-dominated policies, P_u is a distribution over utility functions (we use 100 equidistant weight vectors as was done in (Alegre, A. L. C. Bazzan, et al. 2023)) and $\max_{\pi \in CS} u(v^{\pi})$ is the value of the best policy in the CS, according to the utility function u, defined by the sampled weights.

Sen Welfare: this metric is based on a welfare function, inspired by Amartya Sen's social welfare theory. It integrates total efficiency and equality into a unified metric. Equality is quantified using the Gini coefficient, a widely used statistical measure of inequality originally developed to assess income or wealth distribution within a population. It is derived from the Lorenz curve, which plots the cumulative proportion of total reward (or resource) received by the bottom fraction of groups, ordered from the least to the most rewarded.

The Gini coefficient is defined as the area between the Lorenz curve and the line of perfect equality (a 45-degree line), normalized by the total area under the line of perfect equality. It ranges from 0 to 1, where 0 indicates perfect equality (all groups receive equal rewards), and 1 indicates maximal inequality (all rewards are concentrated in a single group) (Sen 1976).

Formally, the Sen Welfare score for a policy π is computed as:

$$\texttt{SW}(\pi) = \left(\sum_i \boldsymbol{v}_i^{\pi}\right)(1 - \texttt{GI}(\boldsymbol{v}^{\pi})), \tag{12}$$

where $\sum_i \boldsymbol{v}_i^{\pi}$ is the sum of the returns of all objectives in policy π, and $\texttt{GI}(\boldsymbol{v}^{\pi})$ is the Gini coefficient of the return vector $\boldsymbol{v}^{\pi}$. We use this metric for comparative purposes, reflecting a balanced scenario where both efficiency and equality are considered. Sen Welfare has been utilized in economic simulations employing RL before (Zheng et al. 2022). A higher Sen Welfare value signifies increased efficiency and equity.

7.2 Results

We use the *Deep Sea Treasure* environment as a sanity check to verify that our methods can effectively optimize in a simple, small-scale setting. In Table 1, we show that LCN performs on par with PCN on the *Deep Sea Treasure* environment. We next focus on the large-scale MO-TNDP environment. The results discussed here are based on five independently seeded runs. Figure 4 (a) presents a comparison between PCN, GPI-LS and LCN across all objectives in the MO-TNDP-Xi'an environment, and Figure 4 (b) compares the vanilla LCN to the reference point alternatives.

7.3 LCN Outperforms PCN on Many-Objective Settings

As shown in Figure 4 (a), GPI-LS exhibits significantly lower performance across all objectives compared to PCN and LCN. (We limited GPI-LS to up to six objectives due to rapidly increasing runtime.) This performance gap is

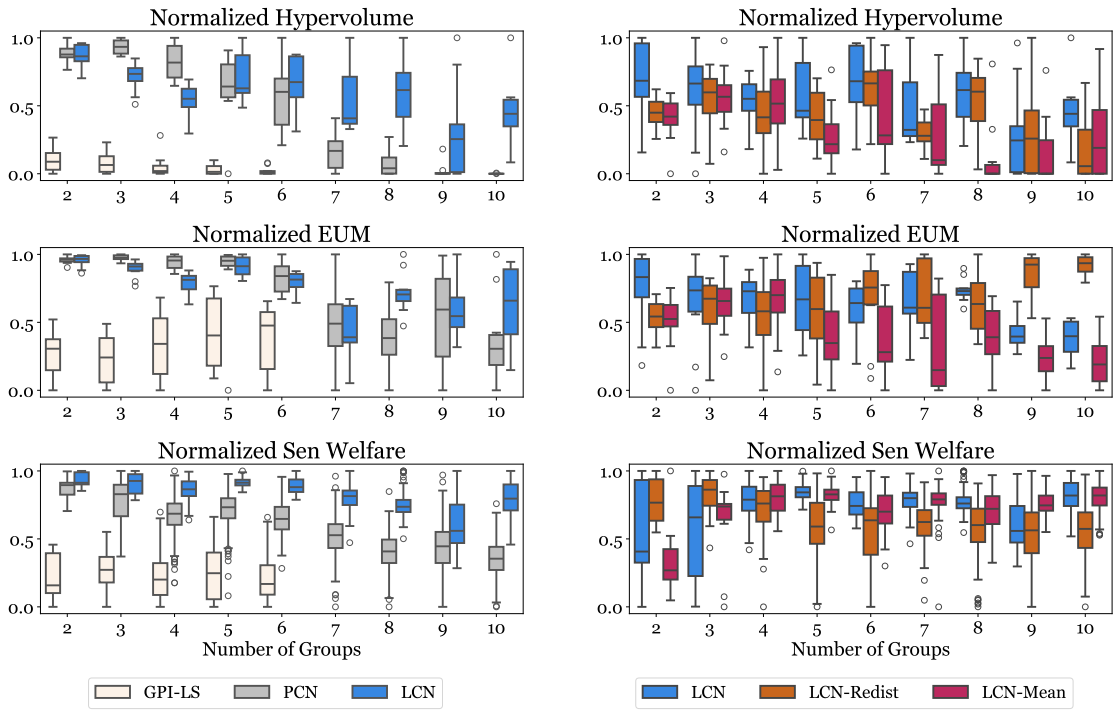

Fig. 4. (a) LCN outperforms PCN and GPI-LS across all objectives in the Sen Welfare measure (Xi'an). Additionally, LCN outperforms PCN in hypervolume when the number of objectives > 4 and in EUM for objectives > 6, showcasing its scalability over the objective space. (b) A comparison of the trained policies of the proposed LCN, LCN-Redist and LCN-Mean models.

Table 1. Results on the *Deep Sea Treasure* environment.

	GPI-LS	**PCN**	**LCN**
HV	22622.8 ± 55.4	22845.4 ± 9.6	22838.0 ± 0.0
EUM	53.86 ± 0.02	53.82 ± 0.02	53.76 ± 0.03

primarily due to the combination of the high-dimensional state-action-reward space and the limited number of time steps. The original experiments on larger domains were conducted for over 200k steps; to ensure a fair comparison, we restricted training to 30k steps.

We focus the remainder of the discussion on comparing PCN and LCN. PCN demonstrates strong performance on hypervolume and EUM metrics in environments with 2 to 6 objectives. This is expected, since PCN is designed to learn diverse, non-Pareto-dominated solutions that directly maximize these metrics. However, as the number of objectives increases, LCN begins to outperform PCN, even on hypervolume and EUM. This suggests that LCN is capable of learning a diverse set of policies that cover a broad region of the solution space, despite primarily targeting fair solutions.

Although this result may appear counterintuitive at first, it can be explained by the fact that the set of non-Pareto-dominated solutions grows rapidly with the number of objectives, making supervised learning

significantly more difficult for PCN. Notably, when the number of objectives exceeds seven, PCN's hypervolume performance collapses. In contrast, LCN continues to scale effectively, benefiting from the relatively smaller size of the non-Lorenz-dominated set.

LCN consistently outperforms PCN in Sen Welfare across all objectives. The Sen Welfare metric, which promotes solutions balancing efficiency and equality, shows that LCN excels in generating effective policies even when the solution space is constrained. In particular, LCN maintains its superior performance relative to PCN even as the number of objectives increases.

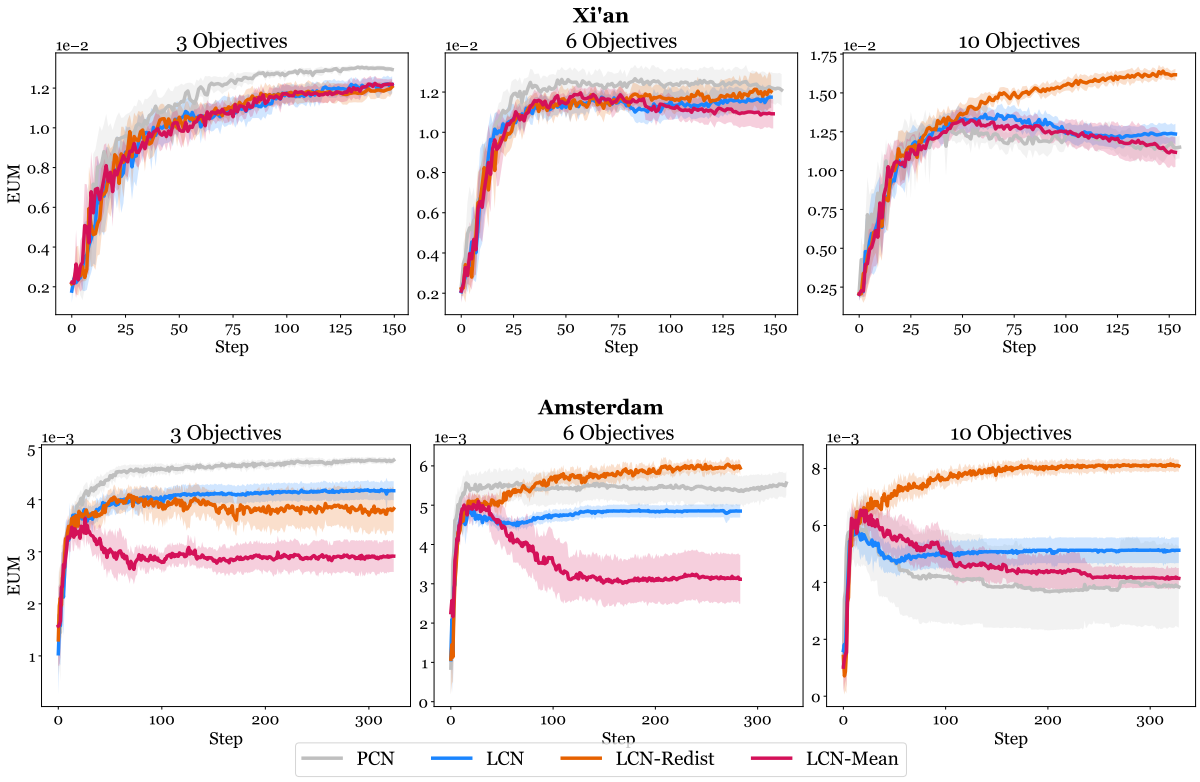

Fig. 5. Learning Curves for EUM on 3 and 10 objectives (curves for all objectives are in the supplementary material).

7.4 Reference Points Improve Training

In Figure 4 (b), we compare the reference point mechanisms (LCN-Redist and LCN-Mean) with vanilla LCN. While the axiomatic hypervolume metric shows minimal impact of reference points on the model's performance, the utility-based metrics, particularly EUM and Sen Welfare, reveal a different story. Vanilla LCN performs well when the objective space is limited. However, as the number of objectives increases, introducing reference points improves stability and outperforms using raw distances from non-dominated points.

Specifically, LCN-Redist demonstrates superior performance in EUM when $d > 5$. Although this seems counterintuitive, the redistribution mechanism creates a reference point with equal objectives. This ensures that all objectives are represented, even if they were absent in the original vector. This approach promotes policies

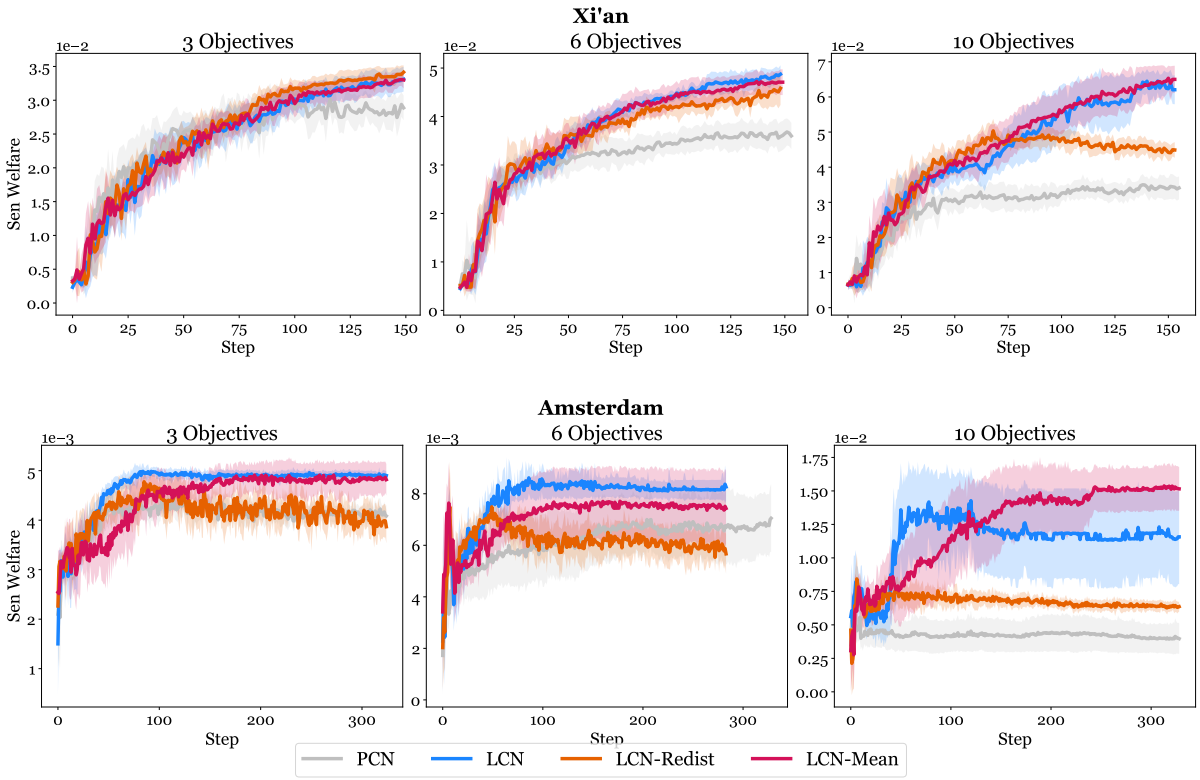

Fig. 6. Learning Curves for Sen Welfare on 3 and 10 objectives (curves for all objectives are in the supplementary material).

that achieve balanced trade-offs across the solution space. The effectiveness of LCN-Redist is further illustrated by the learning curves in Figure 5.

Conversely, LCN-Mean performs the worst in EUM as the number of objectives increases. This outcome can be attributed to the potential skewing of the mean vector by large disparities among dimensions. If a group is consistently underrepresented in the collected experiences, its mean dimension will be close to zero, negatively affecting EUM. However, LCN-Mean performs great in Sen Welfare across all objectives, offering better stability than LCN-Redist. This result is expected, as LCN is designed to maximize Sen Welfare. LCN-Mean effectively balances outliers and creates reference points that balance efficiency and equality with minimal intervention. The learning curves for Sen Welfare are presented in Figure 6.

In Figure 7, we examine how the cardinality of the final, non-dominated policy sets evolves as the number of objectives increases (in the Xi'an Environment). PCN exhibits steep growth: it offers fewer than ten policies for up to three objectives, but this number rises above sixty for ten objectives. This shows, empirically, the complexity discussed in Section 5.4, i.e., that as the number of objectives grows, the PCN policy set rapidly becomes large. In contrast, LCN and LCN-Mean maintain relatively stable cardinalities across all objective counts, with the trained policy set size remaining below ten throughout. This reflects the regularizing effect of reference points in constraining the solution space. LCN-Redist displays an interesting behavior: its cardinality remains low for up to six objectives; above six objectives it LCN-Redist's cardinality increases, approaching the size of PCN's set of non-dominated policies. While the redistribution mechanism curbs the growth of the ER buffer in fewer

objective settings, as dimensionality increases, the filtering retains more diverse experiences, which also explains why EUM remains high for LCN-Redist.

Overall, our experiments suggest the following guidance for decision-makers: if they care about obtaining many diverse policies that cover a wide range of fair trade-offs, they should choose LCN-Redist. However, if they prioritize strict fairness and are comfortable with a smaller set of policies, LCN is preferable for problems with fewer objectives, while LCN-Mean offers stable policies for higher-dimensional objective problems.

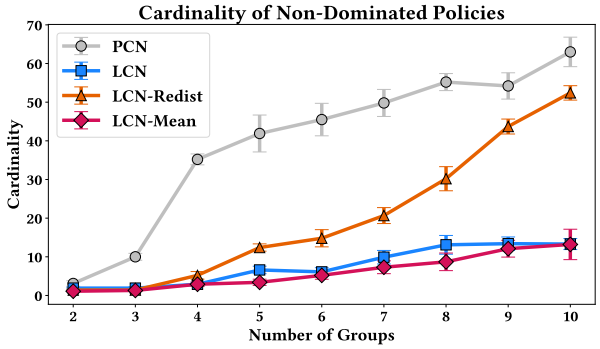

Fig. 7. Cardinality of the non-dominated policy sets after training, as the number of objectives increases (Xi'an environment). PCN grows rapidly, while LCN and LCN-Mean remain manageable throughout. LCN-Redist stays low up to six objectives before increasing similarly to PCN.

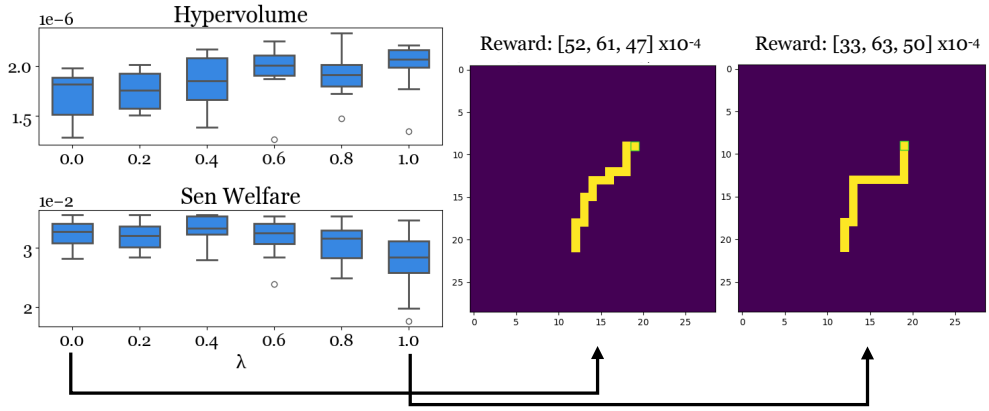

Fig. 8. λ-LCN offers flexibility between emphasizing fair distribution of rewards ($\lambda = 0$) and a relaxation that allows less fair alternatives ($\lambda = 1$).

7.5 λ-LCN Can be Used to Achieve Control over the Degree of Fairness Preference

In Figure 8, we illustrate the flexibility of λ-LCN in achieving diverse solutions across different fairness preference degrees. When $\lambda \approx 0$, there is little flexibility, as the goal is to find the most equally distributed policies. This results in high performance for Sen Welfare but, as expected, lower performance in terms of hypervolume due to

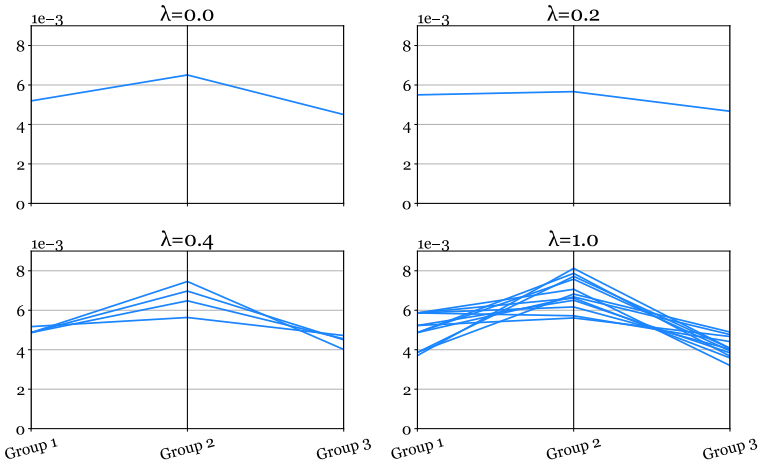

Fig. 9. Parallel coordinate plots of the approximated front for varying values of λ. As λ increases, the solutions become less constrained by fairness, resulting in a larger coverage set.

the concentrated solutions, which cover a smaller area. On the other hand, when $\lambda \approx 1$, the space of accepted solutions expands, leading to a larger hypervolume. However, by accepting less fair solutions, the overall Sen Welfare decreases, adding an extra layer of complexity to the decision-making process. On the right side of the figure, we also show two transport lines originating from the same cell, one for $\lambda = 0$ and the other for $\lambda = 1$. While the lines follow a similar direction, the placement of stations and the areas they traverse can differ substantially based on the degree of fairness preference.

In Figure 9, we present parallel coordinate plots of the learned coverage sets for various values of λ. Here, too, we demonstrate that as λ increases, the flexibility for distributing rewards less fairly among different groups also increases. This leads to an expanded coverage set, providing the decision maker with more policy trade-offs.

8 Conclusion

We addressed key challenges in multi-objective, multi-policy reinforcement learning by proposing methods that perform well over large state-action and objective spaces. We developed a new multi-objective environment for simulating Public Transport Network Design, thereby enhancing the applicability of MORL to real-world scenarios. We proposed LCN, an adaptation of state-of-the-art methods that outperforms baselines in high-reward dimensions. Finally, we present an effective method for controlling the fairness constraint. These contributions move the research field toward more realistic and applicable solutions in real-world contexts, thereby advancing the state-of-the-art in algorithmic fairness in sequential decision-making and MORL.

Acknowledgments

This research was in part supported by the European Union's Horizon Europe research and innovation program under grant agreement No 101120406 (PEER). DM is supported by the Innovation Center for AI (ICAI, The Netherlands). WR is supported by the Research Foundation – Flanders (FWO), grant number 1197622N. F. P. Santos acknowledges funding by the European Union (ERC, RE-LINK, 101116987).

References

A. Abels, D. Roijers, T. Lenaerts, A. Nowé, and D. Steckelmacher. May 2019. "Dynamic Weights in Multi-Objective Deep Reinforcement Learning." en. In: *Proceedings of the 36th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, (May 2019), 11–20. Retrieved Dec. 15, 2023 from https://proceedings.mlr.press/v97/abels19a.html.

M. D. Adler. May 2013. *The Pigou-Dalton Principle and the Structure of Distributive Justice*. en. SSRN Scholarly Paper. Rochester, NY, (May 2013). doi:10.2139/ssrn.2263536.

L. N. Alegre, A. L. C. Bazzan, D. M. Roijers, A. Nowé, and B. C. da Silva. 2023. "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). International Foundation for Autonomous Agents and Multiagent Systems, London, United Kingdom, 2003–2012. ISBN: 9781450394321.

L. N. Alegre, F. Felten, E.-G. Talbi, G. Danoy, A. Nowé, A. L. C. Bazzan, and B. C. da Silva. 2022. "MO-Gym: A Library of Multi-Objective Reinforcement Learning Environments." In: *Proceedings of the 34th Benelux Conference on Artificial Intelligence BNAIC/Benelearn 2022*.

L. N. Alegre, A. Bazzan, and B. C. D. Silva. June 2022. "Optimistic Linear Support and Successor Features as a Basis for Optimal Policy Transfer." en. In: *Proceedings of the 39th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, (June 2022), 394–413. Retrieved Dec. 15, 2023 from https://proceedings.mlr.press/v162/alegre22a.html.

T. Basaklar, S. Gumussoy, and U. Y. Ogras. 2023. *PD-MORL: Preference-Driven Multi-Objective Reinforcement Learning Algorithm*. (2023). https://arxiv.org/abs/2208.07914 arXiv: 2208.07914 [cs.LG].

M. B. Bederina, D. Chaabane, and T. Lust. 2024. "Generating fair solutions of minimal cost." In: *ECAI 2024*. IOS Press, 4076–4083.

J. Blandin and I. A. Kash. 2024. *Group Fairness in Reinforcement Learning via Multi-Objective Rewards*. (2024). https://openreview.net/forum?id=cueEUSG7lE.

J. Blank, K. Deb, Y. Dhebar, S. Bandaru, and H. Seada. 2021. "Generating Well-Spaced Points on a Unit Simplex for Evolutionary Many-Objective Optimization." *IEEE Transactions on Evolutionary Computation*, 25, 1, 48–60. doi:10.1109/TEVC.2020.2992387.

Y. Cao and H. Zhan. 2021. "Efficient multi-objective reinforcement learning via multiple-gradient descent with iteratively discovered weight-vector sets." *Journal of Artificial Intelligence Research*, 70, 319–349.

B. Chabane, M. Basseur, and J.-K. Hao. 2019. "Lorenz dominance based algorithms to solve a practical multiobjective problem." *Computers & Operations Research*, 104, 1–14. doi:https://doi.org/10.1016/j.cor.2018.12.003.

J. Chen, Y. Wang, and T. Lan. May 2021. "Bringing Fairness to Actor-Critic Reinforcement Learning for Network Utility Optimization." In: *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*. ISSN: 2641-9874. (May 2021), 1–10. doi:10.1109/INFOCOM42981.2021.9488823.

M. Chen and M. Zilka. 2025. "Learning Pareto-Optimal Pandemic Intervention Policies with MORL." *arXiv preprint arXiv:2510.03340*.

X. Chen, T. Wang, B. W. Thomas, and M. W. Ulmer. July 2023. "Same-day delivery with fair customer service." *European Journal of Operational Research*, 308, 2, (July 2023), 738–751. doi:10.1016/j.ejor.2022.12.009.

R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff. 2016. "A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization." *IEEE Transactions on Evolutionary Computation*, 20, 5, 773–791. doi:10.1109/TEVC.2016.2519378.

A. Cimpeana, C. Jonkerb, P. Libina, and A. Nowéa. 2023. "A Multi-objective Framework For Fair Reinforcement Learning." In: *Multi-Objective Decision Making Workshop 2023*.

K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. 2000. "A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II." en. In: *Parallel Problem Solving from Nature PPSN VI* (Lecture Notes in Computer Science). Ed. by M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel. Springer, Berlin, Heidelberg, 849–858. ISBN: 978-3-540-45356-7. doi:10.1007/3-540-45356-3_83.

K. Deb and H. Jain. 2014. "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints." *IEEE Transactions on Evolutionary Computation*, 18, 4, 577–601. doi:10.1109/TEVC.2013.2281535.

F. Delgrange, M. Reymond, A. Nowe, and G. Pérez. May 2023. "WAE-PCN: Wasserstein-autoencoded Pareto Conditioned Networks." English. In: *Proc. of the Adaptive and Learning Agents Workshop (ALA 2023)*. Ed. by F. Cruz, C. Hayes, C. Wang, and C. Yates. Vol. https://alaworkshop2023.github.io/. 2023 Adaptive and Learning Agents Workshop at AAMAS, ALA 2023 ; Conference date: 29-05-2023 Through 30-05-2023. (May 2023), 1–7. https://alaworkshop2023.github.io.

Z. Fan, N. Peng, M. Tian, and B. Fain. May 2023. "Welfare and Fairness in Multi-objective Reinforcement Learning." In: *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems* (AAMAS '23). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, (May 2023), 1991–1999. ISBN: 978-1-4503-9432-1. Retrieved Jan. 30, 2024 from.

R. Z. Farahani, E. Miandoabchi, W. Y. Szeto, and H. Rashidi. 2013. "A review of urban transportation network design problems." *European journal of operational research*, 229, 2, 281–302.

M. Fasihi, R. Tavakkoli-Moghaddam, and F. Jolai. Apr. 2023. "A bi-objective re-entrant permutation flow shop scheduling problem: minimizing the makespan and maximum tardiness." *Operational Research*, 23, 2, (Apr. 2023), 29. doi:10.1007/s12351-023-00770-0.

F. Felten, L. N. Alegre, A. Nowé, A. L. C. Bazzan, E. G. Talbi, G. Danoy, and B. C. d. Silva. 2023. "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)*.

F. Felten, E.-G. Talbi, and G. Danoy. Feb. 2024. "Multi-Objective Reinforcement Learning Based on Decomposition: A Taxonomy and Framework." *J. Artif. Int. Res.*, 79, (Feb. 2024), 45 pages. doi:10.1613/jair.1.15702.

T. Feng and J. Zhang. 2014. "Multicriteria evaluation on accessibility-based transportation equity in road network design problem." en. *Journal of Advanced Transportation*, 48, 6, 526–541. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/atr.1202. doi:10.1002/atr.1202.

P. Gajane, A. Saxena, M. Tavakol, G. Fletcher, and M. Pechenizkiy. May 2022. *Survey on Fair Reinforcement Learning: Theory and Practice*. arXiv:2205.10032 [cs]. (May 2022). doi:10.48550/arXiv.2205.10032.

C. F. Hayes et al.. Apr. 2022. "A practical guide to multi-objective reinforcement learning and planning." en. *Autonomous Agents and Multi-Agent Systems*, 36, 1, (Apr. 2022), 26. doi:10.1007/s10458-022-09552-y.

J. Hu, K. Xie, Z. Fang, X. Li, J. Yan, and Z. Zhang. Aug. 2025. "Optimize Battery Control: A Multi-Objective Evolutionary Ensemble Reinforcement Learning Approach." In: *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*. Ed. by J. Kwok. AI4Tech: AI Enabling Technologies. International Joint Conferences on Artificial Intelligence Organization, (Aug. 2025), 9214–9222. doi:10.24963/ijcai.2025/1024.

X. Hu, Y. Zhang, H. Xia, W. Wei, Q. Dai, and J. Li. Jan. 2023. "Towards Fair Power Grid Control: A Hierarchical Multi-Objective Reinforcement Learning Approach." *IEEE Internet of Things Journal*, PP, (Jan. 2023), 1–1. doi:10.1109/JIOT.2023.3314522.

S. Jabbari, M. Joseph, M. Kearns, J. Morgenstern, and A. Roth. June 2017. "Fairness in Reinforcement Learning." In: *Proceedings of the 34th International Conference on Machine Learning* (Proceedings of Machine Learning Research). Ed. by D. Precup and Y. W. Teh. Vol. 70. PMLR, (June 2017), 1617–1626. https://proceedings.mlr.press/v70/jabbari17a.html.

P. Ju, A. Ghosh, and N. B. Shroff. 2023. *Achieving Fairness in Multi-Agent Markov Decision Processes Using Reinforcement Learning*. (2023). https://arxiv.org/abs/2306.00324 arXiv: 2306.00324 [cs.LG].

A. Kumar and W. Yeoh. 2023. "Fairness in Scarce Societal Resource Allocation: A Case Study in Homelessness Applications." In: *Proceedings of the workshop" Autonomous Agents for Social Good". online*.

A. Kumar, X. B. Peng, and S. Levine. 2019. *Reward-Conditioned Policies*. (2019). https://arxiv.org/abs/1912.13465 arXiv: 1912.13465 [cs.LG].

R. Liu, Y. Pan, L. Xu, L. Song, J. Bian, P. You, and Y. Chen. 2024. "C-morl: Multi-objective reinforcement learning through efficient discovery of pareto front." *arXiv preprint arXiv:2410.02236*.

P. A. Lopez et al.. Nov. 2018. "Microscopic Traffic Simulation using SUMO." In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. ISSN: 2153-0017. (Nov. 2018), 2575–2582. doi:10.1109/ITSC.2018.8569938.

D. Mandal and J. Gan. Feb. 2023. *Socially Fair Reinforcement Learning*. arXiv:2208.12584 [cs]. (Feb. 2023). Retrieved Jan. 23, 2024 from http://arxiv.org/abs/2208.12584.

P. Mannion, F. Heintz, T. G. Karimpanal, and P. Vamplew. 2021. "Multi-objective decision making for trustworthy AI." In: *Proceedings of the Multi-Objective Decision Making (MODeM) Workshop*.

K. V. Moffaert and A. Nowé. 2014. "Multi-Objective Reinforcement Learning using Sets of Pareto Dominating Policies." *Journal of Machine Learning Research*, 15, 107, 3663–3692. Retrieved Jan. 23, 2024 from http://jmlr.org/papers/v15/vanmoffaert14a.html.

T. T. Nguyen, N. D. Nguyen, P. Vamplew, S. Nahavandi, R. Dazeley, and C. P. Lim. Nov. 2020. "A multi-objective deep reinforcement learning framework." *Engineering Applications of Artificial Intelligence*, 96, (Nov. 2020), 103915. doi:10.1016/j.engappai.2020.103915.

R. O'Driscoll, C. Hagen, J. Bater, and J. Adams. 2025. "Multi-Objective Reinforcement Learning for Automated Resilient Cyber Defence." *Applied AI Letters*, 6, 3, e70007.

Z. Osika, R. Radelescu, J. Z. Salazar, F. Oliehoek, and P. K. Murukannaiah. 2025. "Multi-Objective Reinforcement Learning for Water Management." *arXiv preprint arXiv:2505.01094*.

Z. Osika, J. Z. Salazar, D. M. Roijers, F. A. Oliehoek, and P. K. Murukannaiah. 2023. "What Lies beyond the Pareto Front? A Survey on Decision-Support Methods for Multi-Objective Optimization." In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*. ijcai.org, 6741–6749. doi:10.24963/IJCAI.2023/755.

S. Parisi, M. Pirotta, and M. Restelli. 2016. "Multi-objective reinforcement learning through continuous Pareto manifold approximation." *Journal of Artificial Intelligence Research*, 57, 187–227.

P. Perny, P. Weng, J. Goldsmith, and J. Hanna. Sept. 2013. *Approximation of Lorenz-Optimal Solutions in Multiobjective Markov Decision Processes*. arXiv:1309.6856 [cs]. (Sept. 2013). doi:10.48550/arXiv.1309.6856.

M. Peschl, A. Zgonnikov, F. A. Oliehoek, and L. C. Siebert. Dec. 2021. *MORAL: Aligning AI with Human Norms through Multi-Objective Reinforced Active Learning*. arXiv:2201.00012 [cs]. (Dec. 2021). doi:10.48550/arXiv.2201.00012.

R. Rădulescu, P. Mannion, D. M. Roijers, and A. Nowé. 2019. "Equilibria in Multi-Objective Games: a Utility-based Perspective." In: *Proceedings of the adaptive and learning agents workshop (ALA-19) at AAMAS*.

M. Reymond, E. Bargiacchi, and A. Nowé. 2022. "Pareto Conditioned Networks." In: *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems* (AAMAS '22). International Foundation for Autonomous Agents and Multiagent Systems, Virtual Event, New Zealand, 1110–1118. ISBN: 9781450392136.

M. Reymond, C. F. Hayes, et al.. Apr. 2022. *Exploring the Pareto front of multi-objective COVID-19 mitigation policies using reinforcement learning*. arXiv:2204.05027 [cs, q-bio]. (Apr. 2022). doi:10.48550/arXiv.2204.05027.

M. Rodriguez-Soto, M. Lopez-Sanchez, and J. A. R. Aguilar. Aug. 2021. "Multi-Objective Reinforcement Learning for Designing Ethical Environments." en. In: vol. 1. ISSN: 1045-0823. (Aug. 2021), 545–551. doi:10.24963/ijcai.2021/76.

M. Rodriguez-Soto, M. Serramia, M. Lopez-Sanchez, and J. A. Rodriguez-Aguilar. Jan. 2022. "Instilling moral value alignment by means of multi-objective reinforcement learning." en. *Ethics and Information Technology*, 24, 1, (Jan. 2022), 9. doi:10.1007/s10676-022-09635-0.

D. M. Roijers, S. Whiteson, and F. A. Oliehoek. Mar. 2015. "Computing Convex Coverage Sets for Faster Multi-objective Coordination." en. *Journal of Artificial Intelligence Research*, 52, (Mar. 2015), 399–443. doi:10.1613/jair.4550.

W. Röpke. 2023. "Reinforcement Learning in Multi-Objective Multi-Agent Systems." In: *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems* (AAMAS '23). International Foundation for Autonomous Agents and Multiagent Systems, London, United Kingdom, 2999–3001. ISBN: 9781450394321.

W. Röpke, M. Reymond, P. Mannion, D. M. Roijers, A. Nowé, and R. Rădulescu. 2024. *Divide and Conquer: Provably Unveiling the Pareto Front with Multi-Objective Reinforcement Learning*. (2024). https://arxiv.org/abs/2402.07182 arXiv: 2402.07182 [cs.LG].

M. Ruiz-Montiel, L. Mandow, and J.-L. Pérez-de-la-Cruz. Nov. 2017. "A temporal difference method for multi-objective reinforcement learning." *Neurocomputing*. Multiobjective Reinforcement Learning: Theory and Applications 263, (Nov. 2017), 15–25. doi:10.1016/j.neucom.2016.10.100.

H. Satija, A. Lazaric, M. Pirotta, and J. Pineau. 2023. "Group Fairness in Reinforcement Learning." *Trans. Mach. Learn. Res.*, 2023. https://openreview.net/forum?id=JkIH4MeOc3.

M. Schläpfer et al.. May 2021. "The universal visitation law of human mobility." en. *Nature*, 593, 7860, (May 2021), 522–527. Number: 7860 Publisher: Nature Publishing Group. doi:10.1038/s41586-021-03480-9.

A. Sen. 1976. "Poverty: An Ordinal Approach to Measurement." *Econometrica*, 44, 2, 219–231. Publisher: [Wiley, Econometric Society]. doi:10.2307/1912718.

A. F. Shorrocks. 1983. "Ranking Income Distributions." *Economica*, 50, 197, 3–17. Retrieved Jan. 19, 2024 from http://www.jstor.org/stable/2554117.

U. Siddique, P. Weng, and M. Zimmer. July 2020. "Learning Fair Policies in Multi-Objective (Deep) Reinforcement Learning with Average and Discounted Rewards." In: *Proceedings of the 37th International Conference on Machine Learning* (Proceedings of Machine Learning Research). Ed. by H. D. III and A. Singh. Vol. 119. PMLR, (July 2020), 8905–8915.

P. Vamplew, R. Dazeley, A. Berry, R. Issabekov, and E. Dekker. July 2011. "Empirical evaluation methods for multiobjective reinforcement learning algorithms." en. *Machine Learning*, 84, 1, (July 2011), 51–80. doi:10.1007/s10994-010-5232-5.

P. Vamplew, B. J. Smith, et al.. July 2022. "Scalar reward is not enough: a response to Silver, Singh, Precup and Sutton (2021)." en. *Autonomous Agents and Multi-Agent Systems*, 36, 2, (July 2022), 41. doi:10.1007/s10458-022-09575-5.

H.-n. Wang, N. Liu, Y.-y. Zhang, D.-w. Feng, F. Huang, D.-s. Li, and Y.-m. Zhang. Dec. 2020. "Deep Reinforcement Learning: A Survey." en. *Frontiers of Information Technology & Electronic Engineering*, 21, 12, (Dec. 2020), 1726–1744. doi:10.1631/FITEE.1900533.

Y. Wei, M. Mao, X. Zhao, J. Zou, and P. An. Aug. 2020. "City Metro Network Expansion with Reinforcement Learning." en. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, Virtual Event CA USA, (Aug. 2020), 2646–2656. ISBN: 978-1-4503-7998-4. doi:10.1145/3394486.3403315.

E. Y. Yu, Z. Qin, M. K. Lee, and S. Gao. Oct. 2022. *Policy Optimization with Advantage Regularization for Long-Term Fairness in Decision Systems*. arXiv:2210.12546 [cs]. (Oct. 2022). doi:10.48550/arXiv.2210.12546.

S. Zheng, A. Trott, S. Srinivasa, D. C. Parkes, and R. Socher. May 2022. "The AI Economist: Taxation policy design via two-level deep multiagent reinforcement learning." *Science Advances*, 8, 18, (May 2022), eabk2607. Publisher: American Association for the Advancement of Science. doi:10.1126/sciadv.abk2607.

M. Zimmer, C. Glanois, U. Siddique, and P. Weng. July 2021. "Learning Fair Policies in Decentralized Cooperative Multi-Agent Reinforcement Learning." en. In: *Proceedings of the 38th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, (July 2021), 12967–12978. Retrieved Jan. 23, 2024 from https://proceedings.mlr.press/v139/zimmer21a.html.

L. M. Zintgraf, T. V. Kanters, D. M. Roijers, F. Oliehoek, and P. Beau. 2015. "Quality assessment of MORL algorithms: A utility-based approach." In: *Benelearn 2015: proceedings of the 24th annual machine learning conference of Belgium and the Netherlands*.

A Theoretical Results

In this section, we provide the detailed theoretical results for λ-LCN that we sketched in the main paper. In particular, we show that λ-Lorenz dominance is a generalization of Lorenz dominance that can be used to flexibly set a desired fairness level. Moreover, by decreasing λ, the resulting solution set moves monotonically closer to the Lorenz front.

We first introduce a necessary auxiliary result in lemma 2 which shows that λ-Lorenz dominance implies Lorenz dominance.

Lemma 2. *$\forall \lambda \in [0,1]$ and $\forall \boldsymbol{v}, \boldsymbol{v}' \in \mathbb{R}^d$,*

$$\boldsymbol{v} \succ_\lambda \boldsymbol{v}' \implies \boldsymbol{v} \succ_L \boldsymbol{v}'. \tag{13}$$

Proof. By contradiction, assume there is some $\boldsymbol{v}, \boldsymbol{v}'$ and λ such that $\boldsymbol{v} \succ_\lambda \boldsymbol{v}'$ but $\boldsymbol{v}$ does not Lorenz dominate $\boldsymbol{v}'$. Then there is some smallest index k such that $L(\boldsymbol{v})_k < L(\boldsymbol{v}')_k$. Since $\sigma(\boldsymbol{v})_1 = L(\boldsymbol{v})_1$ and $\boldsymbol{v} \succ_\lambda \boldsymbol{v}'$, we know that $\sigma(\boldsymbol{v})_1 \geq \sigma(\boldsymbol{v}')_1$. Then for all indices $i \in \{1, \dots, k-1\}$ we have $\sum_{j=1}^{i} \sigma(\boldsymbol{v})_j \geq \sigma(\boldsymbol{v}')_j$. Given that $\sum_{j=1}^{k} \sigma(\boldsymbol{v})_j < \sum_{j=0}^{k} \sigma(\boldsymbol{v}')_j$, we have that $\sigma(\boldsymbol{v})_k < \sigma(\boldsymbol{v}')_k$. However, this implies that $\forall \lambda \in [0,1], \lambda\sigma(\boldsymbol{v})_k + (1-\lambda)L(\boldsymbol{v})_k < \sigma(\boldsymbol{v}')_k + (1-\lambda)L(\boldsymbol{v}')_k$ and therefore $\boldsymbol{v}$ does not λ-Lorenz dominate $\boldsymbol{v}'$ leading to a contradiction. □

In theorem 3 we use lemma 2 to demonstrate that when some vector λ-Lorenz dominates, it necessarily also dominates it for any smaller λ.

Theorem 3. *$\forall \lambda_1, \lambda_2 : 0 \leq \lambda_1 \leq \lambda_2 \leq 1$ and $\forall \boldsymbol{v}, \boldsymbol{v}' \in \mathbb{R}^d$,*

$$\boldsymbol{v} \succ_{\lambda_2} \boldsymbol{v}' \implies \boldsymbol{v} \succ_{\lambda_1} \boldsymbol{v}'. \tag{14}$$

Proof. Let $\boldsymbol{v} \succ_{\lambda_2} \boldsymbol{v}'$. From Definition 3, this can equivalently be written as $\lambda_2\sigma(\boldsymbol{v}) + (1-\lambda_2)L(\boldsymbol{v}) \succ_P \lambda_2\sigma(\boldsymbol{v}') + (1-\lambda_2)L(\boldsymbol{v}')$. In addition, from lemma 2 we know that since $\boldsymbol{v} \succ_{\lambda_2} \boldsymbol{v}' \implies \boldsymbol{v} \succ_L \boldsymbol{v}'$. Then, for any index $i \in \{1, \dots, d\}$ we have that,

$$\lambda_2\sigma(\boldsymbol{v})_i + (1-\lambda_2)L(\boldsymbol{v})_i \geq \lambda_2\sigma(\boldsymbol{v}')_i + (1-\lambda_2)L(\boldsymbol{v}')_i \tag{15}$$

$$\frac{\lambda_1}{\lambda_2}\lambda_2\sigma(\boldsymbol{v})_i + \frac{\lambda_1}{\lambda_2}(1-\lambda_2)L(\boldsymbol{v})_i \geq \frac{\lambda_1}{\lambda_2}\lambda_2\sigma(\boldsymbol{v}')_i + \frac{\lambda_1}{\lambda_2}(1-\lambda_2)L(\boldsymbol{v}')_i \tag{16}$$

$$\lambda_1\sigma(\boldsymbol{v})_i + (\frac{\lambda_1}{\lambda_2} - \lambda_1)L(\boldsymbol{v})_i \geq \lambda_1\sigma(\boldsymbol{v}')_i + (\frac{\lambda_1}{\lambda_2} - \lambda_1)L(\boldsymbol{v}')_i \tag{17}$$

$$\lambda_1\sigma(\boldsymbol{v})_i + (1-\lambda_1)L(\boldsymbol{v})_i \geq \lambda_1\sigma(\boldsymbol{v}')_i + (1-\lambda_1)L(\boldsymbol{v}')_i \tag{18}$$

where the last step holds since $\lambda_1 \leq \frac{\lambda_1}{\lambda_2} \leq 1$ and $L(\boldsymbol{v})_i \geq L(\boldsymbol{v}')_i$ by lemma 2. □

We now contribute an additional auxiliary result which guarantees that when some vector Pareto dominates another, it also λ-Lorenz dominates the vector. This result is an extension of the fact that Pareto dominance implies Lorenz dominance.

Lemma 4. *$\forall \lambda \in [0,1]$ and $\forall \boldsymbol{v}, \boldsymbol{v}' \in \mathbb{R}^d$,*

$$\boldsymbol{v} \succ_P \boldsymbol{v}' \implies \boldsymbol{v} \succ_\lambda \boldsymbol{v}'. \tag{19}$$

Proof. From Definition 3, λ-Lorenz dominance can be written as $\lambda\sigma(\boldsymbol{v}) + (1-\lambda)L(\boldsymbol{v}) \succ_P \lambda\sigma(\boldsymbol{v}') + (1-\lambda)L(\boldsymbol{v}')$. Let us first recall from Theorem 1 of (Perny et al. 2013) that $\boldsymbol{v} \succ_P \boldsymbol{v}' \implies \boldsymbol{v} \succ_L \boldsymbol{v}'$. It is then necessary to demonstrate an analogous result for $\boldsymbol{v} \succ_P \boldsymbol{v}' \implies \sigma(\boldsymbol{v}) \succ_P \sigma(\boldsymbol{v}')$. We prove this by induction.

Let $\boldsymbol{v} = (v_1, v_2)$ and $\boldsymbol{v}' = (v'_1, v'_2)$. By contradiction, assume that $\boldsymbol{v} \succ_P \boldsymbol{v}'$ but $\sigma(\boldsymbol{v})$ does not Pareto dominate $\sigma(\boldsymbol{v}')$. This implies that there is some index i such that $\sigma(\boldsymbol{v})_i < \sigma(\boldsymbol{v}')_i$. Let us consider the four cases for $\sigma(\boldsymbol{v})$ and $\sigma(\boldsymbol{v}')$.

If $\sigma(\boldsymbol{v}) = \boldsymbol{v}$ and $\sigma(\boldsymbol{v}') = \boldsymbol{v}'$ this cannot occur. Moreover, if both vectors are in reverse it also cannot be the case.

When $\sigma(\boldsymbol{v}) = (v_1, v_2)$ and $\sigma(\boldsymbol{v}') = (v'_2, v'_1)$, v'_2 cannot be greater than v_1 since by transitivy then also $v'_1 > v_1$ which is a contradiction. Furthermore, v'_1 cannot be greater than v_2 because then $v'_1 > v_1$ which is again a contradiction.

The final case, where $\sigma(\boldsymbol{v}) = (v_2, v_1)$ and $\sigma(\boldsymbol{v}') = (v'_1, v'_2)$, gives the same contradictions.

Assuming that the result holds for vectors of dimension d, we demonstrate that it must also hold for $d+1$. Let $\boldsymbol{v}, \boldsymbol{v}' \in \mathbb{R}^d$ and $\boldsymbol{v} \succ_P \boldsymbol{v}'$. Consider now $(\boldsymbol{v}, a)$ and $(\boldsymbol{v}', b)$ which are extensions of the vectors and $(\boldsymbol{v}, a) \succ_P (\boldsymbol{v}', b)$. Then $a > b$. By contradiction, assume again that $\sigma(\boldsymbol{v}, a)$ does not dominate $\sigma(\boldsymbol{v}', b)$. Let i be the smallest index where $\sigma(\boldsymbol{v}, a)_i < \sigma(\boldsymbol{v}', b)_i$. There are four cases where this may occur. Either this happens when a and b have both not been inserted yet, a has been inserted but not b, b has been inserted but not a and both have been inserted. Clearly, when neither or both were inserted, this leads to a contradiction.

If only b was inserted, by transitivity we have that $\sigma(\boldsymbol{v}, a)_i < \sigma(\boldsymbol{v}', b)_{i+1}$ and therefore $\sigma(\boldsymbol{v})_i < \sigma(\boldsymbol{v}')_i$ which is a contradiction. Lastly, if only a was inserted then $\sigma(\boldsymbol{v}, a)_i < \sigma(\boldsymbol{v}', b)_i$ and $\sigma(\boldsymbol{v}', b)_i < b$ implying that $a < b$ and leading to a contradiction. □

Finally, we provide a proof for Theorem 1 of the main paper. This result demonstrates that when λ is 1, the solution set starts closest to the Pareto front and decreasing λ to 0 monotonically reduces the solution set until it results in the Lorenz front. As such, by selecting a λ, a decision-maker can determine their preferred balance between Pareto optimality and Lorenz fairness. We first restate the theorem below and subsequently provide the proof.

Theorem 5. *(Referred to as Theorem 1 in the main text)* $\forall \lambda_1, \lambda_2 : 0 \leq \lambda_1 \leq \lambda_2 \leq 1$ *and* $\forall D \subset \mathbb{R}^d$ *the following relations hold.*

$$\mathcal{L}(D) \subseteq \mathcal{L}(D; \lambda_1) \subseteq \mathcal{L}(D; \lambda_2) \subseteq \mathcal{F}(D). \tag{20}$$

Proof. From lemma 4 we are guaranteed that $\boldsymbol{v} \succ_P \boldsymbol{v}'$ implies $\boldsymbol{v} \succ_\lambda \boldsymbol{v}'$ and therefore $\forall \lambda \in [0, 1], \mathcal{L}(D; \lambda) \subseteq \mathcal{F}(D)$. In addition, theorem 3 guarantees that $\mathcal{L}(D; \lambda_1) \subseteq \mathcal{L}(D; \lambda_2)$. Finally, given that $\mathcal{L}(D) = \mathcal{L}(D; 1)$ we have that $\forall \lambda \in [0, 1], \mathcal{L}(D) \subseteq \mathcal{L}(D; \lambda)$. □

B Preparing the Xi'an and Amsterdam Environments

The MO-TNDP environment released with this paper is adaptable for training an agent in any city, provided there are three elements: grid size (defined by the number of rows and columns), OD matrix, and group membership assigned to each grid cell. The grid size is specified as an argument in the constructor of the environment object, along with the file paths leading to the CSV files containing the OD matrix and group membership data. We have configured the environments for both Xi'an and Amsterdam, and these are included alongside the code for the environment.

Xi'an environment preparation. We generated the Xi'an environment utilizing the data provided in (Wei et al. 2020). [3]. The city is divided into a grid of dimensions $H^{29\times29}$, with cells of equal size ($1km^2$). The OD demand matrix was formulated using GPS data gathered from 25 million mobile phones, with their movements tracked over a one-month period Additionally, each cell is assigned an average house price index, which is categorized into quintiles. Figure 10 provides a comprehensive breakdown of the city into various sized groups.

Amsterdam environment preparation. We generate and release the data associated with the Amsterdam environment. The city is divided into a grid of dimensions $H^{35\times47}$, consisting of equally sizes cells of $0.5km^2$. The choice of this cell size takes into consideration Amsterdam's smaller size compared to Xi'an. Since GPS data is unavailable for Amsterdam, we estimate the OD demand using the recently published universal law of human mobility, which states that the total mobility flow between two areas, denoted as i and j, depends on their distance and visitation frequency (Schläpfer et al. 2021). The estimation is computed using the formula:

$$OD_{ij} = \mu_j \mathrm{K_i} / d_{ij}^2 \ln(f_{max}/f_{min}) \tag{21}$$

[3] source: https://github.com/weiyu123112/City-Metro-Network-Expansion-with-RL

Where K_i represents the total area of the origin location i, d_{ij}^2 is the (Manhattan) distance between i, j and μ_j is the magnitude of flows, calculated as follows:

$$\mu_j \approx \rho_{pop}(j) rad_j^2 f_{max} \tag{22}$$

Where rad_j^2 is the radius of area j. The flows are estimated for a full week, and in the model, this is accomplished by setting f_{min} and f_{max} to 1/7 and 7 respectively. Since the grid cells are of equal size in our case, the term K can be omitted from the calculation. An illustration of the Amsterdam environment is presented in A detailed breakdown of the city into different-sized groups is depicted in Figure 11.

Similar to the Xi'an environment, each cell in Amsterdam is associated with an average house price, sourced from the publicly available statistical bureau of the Netherlands dataset [4].

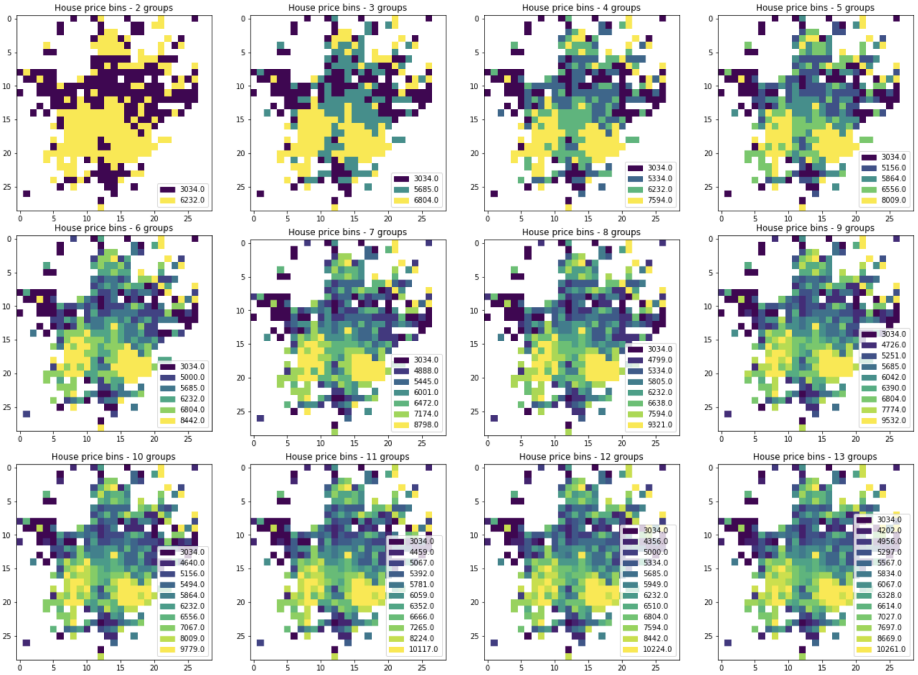

Fig. 10. MO-TNDP Xi'an Environment with different groups based on equally sized buckets of the average house price index.

C Experiment Reproducibility Details

Each model presented in the paper for training MO-TNDP was trained for 30000 steps. Hyperparameters were tuned via a Bayes search over 100 settings, with the following ranges:

PCN/LCN

- Batch size: [128, 256]
- Learning Rate: [0.1, 0.01]
- Number of Linear Layers: [1, 2]
- Hidden Dims: [64, 128]

[4] source: https://www.cbs.nl/nl-nl/maatwerk/2019/31/kerncijfers-wijken-en-buurten-2019

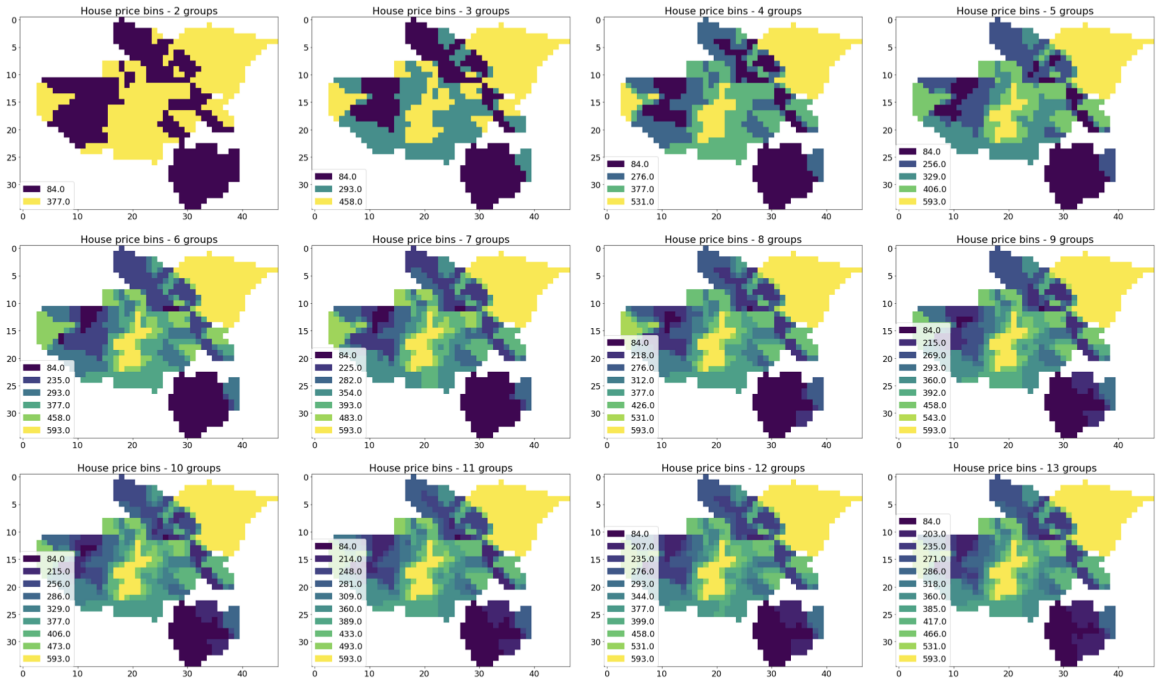

Fig. 11. MO-TNDP Amsterdam Environment with different groups based on equally sized buckets of the average house price index.

- Experience Replay Buffer Size: [50, 100]
- Model Updates: [5, 10]

GPI-LS

- Network Architecture: [64, 64, 64]
- Learning Rate: [0.00001, 0.0001, 0.001, 0.01]
- Batch Size: [16, 32, 64, 128, 256, 512]
- Buffer Size: [256, 512, 2048, 4096, 8192, 16384, 32768]
- Learning Starts: 50
- Target Net Update Frequency: [10, 20, 50, 100]
- Gradient Updates: [1, 2, 5]

fig. 12 shows the architecture used for the policy network. In the provided code, we provide the exact commands to reproduce all of our experiments, including the environments, hyperparameters, and seeds used to generate our results. Furthermore, the details of the hyperparameters we used for each experiment are available on a public Notion page [5]. Finally, we commit to sharing the output model weights upon request.

D Sensitivity Analysis of Crowding Distance Threshold and Penalty

We conduct a sensitivity analysis on the effects of the crowding-distance threshold τ_{cd} and the penalty multiplier ρ_{pen}. When the model filters experiences to maintain a consistent Experience Replay (ER) buffer, it applies a

[5]https://aware-night-ab1.notion.site/Project-B-MO-LCN-Experiment-Tracker-b4d21ab160eb458a9cff9ab9314606a7

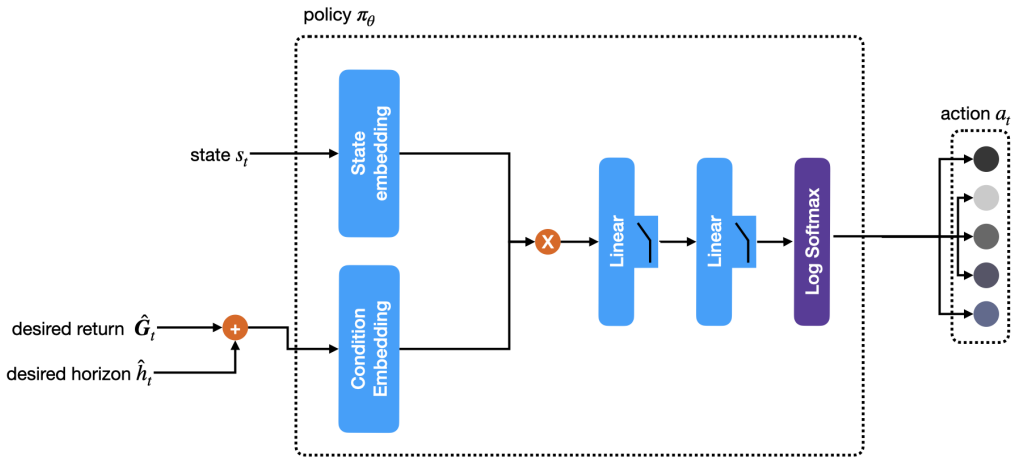

Fig. 12. Architecture of the policy network.

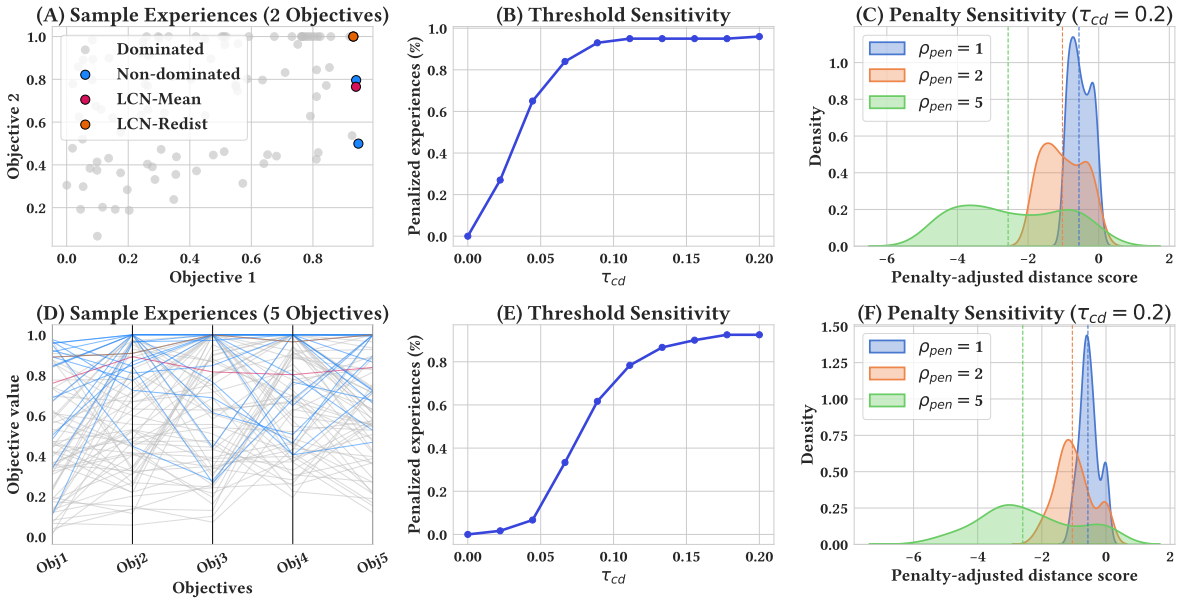

Fig. 13. Sensitivity analysis of the crowding-distance penalty: Xi'an Environment, 2 groups (A, B, C) and 5 groups (D, E, F). (A) Randomly sampled experiences, with the non-dominated set and the LCN-Mean and LCN-Redist reference points highlighted. (B) Fraction of sampled experiences that receive a crowding penalty as a function of the crowding-distance threshold (τ_{cd}).(C) Kernel density estimates (KDEs) of the total penalty applied to the experiences for different penalty multipliers, assuming $\tau_{cd} = 0.2$.

distance metric together with a crowdedness threshold. Experiences that lie too close to each other, i.e., those

whose crowding distance exceeds the threshold, are penalized to promote diversity, as the objective is to train a broad set of policies for the decision-maker. We examine, in a theoretical setting, how variations in τ_{cd} and ρ_{pen} influence both the proportion of penalized samples and the overall performance of the algorithm.

In Figure 13, we generated random sets of experiences in a 2-dimension (Panel A) and 5-dimension setting (Panel D). We show the non-dominated points, as well as the LCN-Mean and LCN-Redist reference points. In Panels C and E we show the sensitivity of the generated experiences to the crowding distance threshold.

The influence of the crowding distance threshold (τ_{cd}) varies significantly with the dimensionality of the objective space. In low dimensions (e.g., 2D), experiences cluster closer in the crowding-distance metric. Consequently, the sensitivity to τ_{cd} is high initially, with the proportion of penalized points peaking sharply at a very low threshold (e.g., $\tau_{cd} \approx 0.1$), where nearly all points are penalized. This calls for setting a low τ_{cd} in low-dimensional settings for the crowding distance penalty to remain effective. Conversely, in higher dimensions (e.g., 5D), the crowding distance penalty peaks at a later stage (e.g., $\tau_{cd} \approx 0.2$). The points are naturally more spread out, giving the designer finer control over the number of penalized experiences.

The penalty multiplier (ρ_{pen}) impacts the distribution of the penalty-adjusted distance scores. With a low penalty multiplier ($\rho_{pen} = 1$), the distribution of the adjusted scores is concentrated close to zero, with low variance and thin tails, leading to a minimal scaling effect beyond the crowding distance. However, increasing ρ_{pen} significantly skews the distribution, introducing heavier tails and increasing the variance of the penalty. This high multiplier aggressively penalizes points below the threshold, meaning points just below the threshold are greatly favored over those further away. A higher ρ_{pen} is desirable in low-diversity environments (where policies are very close) to discriminate between them. A lower ρ_{pen} suffices in high-diversity environments where distance itself is a sufficient discriminator. In this paper, we scale our environment to high dimensions, however, we operate in a relatively low-variance environment with hard constraints on the generated policies, hence we opted for a middle-point penalty of $\rho_{pen} = 2$.

E Additional Results

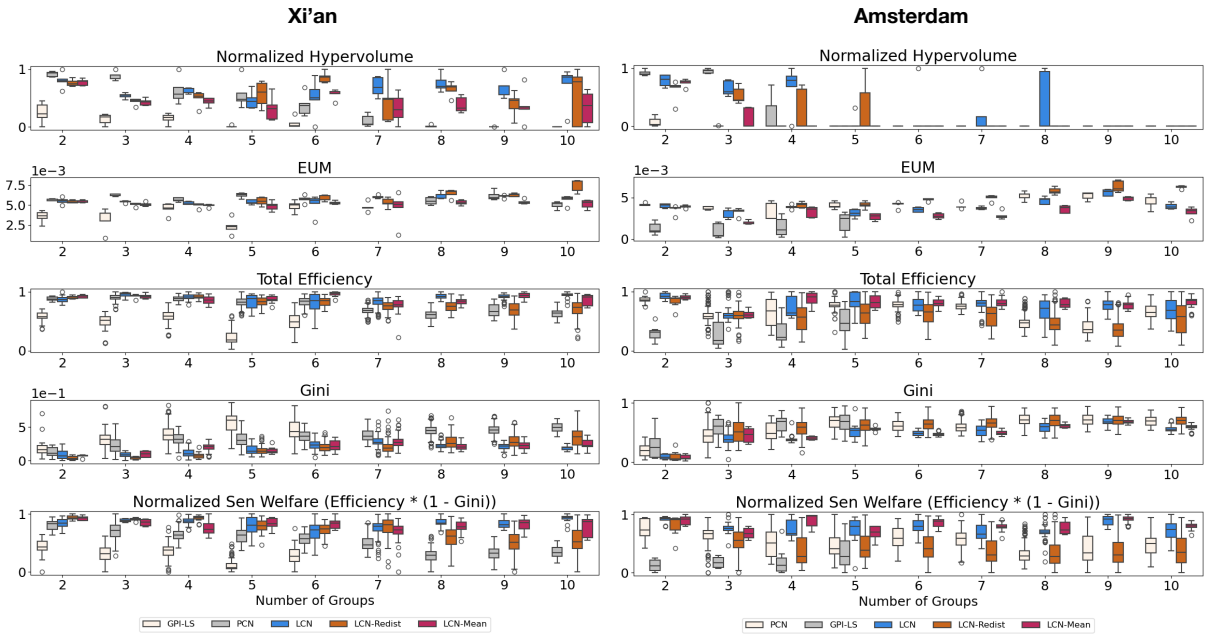

Fig. 14. Full results for the MO-TNDP Amsterdam and Xi'an Environments.

Total efficiency: measures how effectively a generated line captures the total travel demand of a city. It is calculated as the simple sum of all elements in the value vector – the sum of all satisfied demands for each group.

Gini coefficient: quantifies the reward distribution among various groups in the city. A value of 0 indicates perfect equality (equal percentage of satisfied OD flows per group), while 1 represents perfect inequality (only the demand of one group is satisfied). Although traditionally used to assess income inequality (Zheng et al. 2022), it has also been employed in the context of transportation network design (Feng and Zhang 2014). In fig. 14 and table 2, we show detailed results on both Xi'an and Amsterdam for all objectives. fig. 15 and fig. 16 show the learning curves of PCN and LCN for all objectives.

Table 2. Results of all models, for 1–10 objectives. Underline indicates the best results.

Normalized hypervolume									
	Number of Objectives								
Xi'an	2	3	4	5	6	7	8	9	10
GPI-LS	0.11 ± 0.03	0.08 ± 0.02	0.05 ± 0.03	0.03 ± 0.01	0.02 ± 0.01	−−	−−	−−	−−
PCN	0.89 ± 0.02	0.93 ± 0.02	0.83 ± 0.04	0.63 ± 0.08	0.56 ± 0.08	0.17 ± 0.04	0.07 ± 0.03	0.02 ± 0.02	0.00 ± 0.00
LCN	0.86 ± 0.03	0.71 ± 0.03	0.54 ± 0.04	0.71 ± 0.06	0.67 ± 0.06	0.55 ± 0.08	0.60 ± 0.08	0.29 ± 0.11	0.46 ± 0.07
LCN-Redist.	0.79 ± 0.01	0.69 ± 0.03	0.48 ± 0.06	0.60 ± 0.04	0.64 ± 0.05	0.39 ± 0.03	0.53 ± 0.08	0.31 ± 0.11	0.19 ± 0.08
LCN-Mean	0.78 ± 0.02	0.70 ± 0.02	0.52 ± 0.07	0.51 ± 0.05	0.49 ± 0.08	0.37 ± 0.08	0.12 ± 0.08	0.15 ± 0.08	0.28 ± 0.10
Amsterdam	2	3	4	5	6	7	8	9	10
GPI-LS	0.12 ± 0.05	0.22 ± 0.06	0.06 ± 0.06	0.00 ± 0.00	−−	−−	−−	−−	−−
PCN	0.97 ± 0.01	0.80 ± 0.04	0.17 ± 0.11	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
LCN	0.80 ± 0.03	0.67 ± 0.04	0.41 ± 0.14	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
LCN-Redist.	0.79 ± 0.03	0.53 ± 0.03	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
LCN-Mean	0.81 ± 0.01	0.27 ± 0.05	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
Normalized EUM									
Xi'an	2	3	4	5	6	7	8	9	10
GPI-LS	0.27 ± 0.06	0.24 ± 0.06	0.33 ± 0.08	0.42 ± 0.08	0.38 ± 0.08	−−	−−	−−	−−
PCN	0.96 ± 0.01	0.97 ± 0.01	0.94 ± 0.02	0.86 ± 0.10	0.83 ± 0.04	0.57 ± 0.09	0.55 ± 0.04	0.39 ± 0.07	0.27 ± 0.05
LCN	0.95 ± 0.01	0.89 ± 0.02	0.79 ± 0.02	0.91 ± 0.02	0.80 ± 0.02	0.50 ± 0.06	0.74 ± 0.03	0.42 ± 0.04	0.38 ± 0.04
LCN-Redist.	0.92 ± 0.01	0.88 ± 0.02	0.73 ± 0.04	0.89 ± 0.02	0.83 ± 0.04	0.52 ± 0.07	0.63 ± 0.07	0.86 ± 0.05	0.92 ± 0.02
LCN-Mean	0.91 ± 0.01	0.89 ± 0.01	0.78 ± 0.04	0.84 ± 0.02	0.71 ± 0.03	0.26 ± 0.09	0.39 ± 0.07	0.25 ± 0.05	0.22 ± 0.06
Amsterdam	2	3	4	5	6	7	8	9	10
GPI-LS	0.30 ± 0.07	0.85 ± 0.05	0.92 ± 0.02	0.67 ± 0.07	−−	−−	−−	−−	−−
PCN	0.99 ± 0.00	0.95 ± 0.01	0.75 ± 0.04	0.80 ± 0.04	0.76 ± 0.04	0.62 ± 0.09	0.59 ± 0.07	0.36 ± 0.10	0.32 ± 0.09
LCN	0.93 ± 0.02	0.70 ± 0.03	0.21 ± 0.05	0.55 ± 0.02	0.58 ± 0.02	0.47 ± 0.04	0.79 ± 0.02	0.44 ± 0.03	0.50 ± 0.03
LCN-Redist.	0.92 ± 0.02	0.55 ± 0.08	0.46 ± 0.07	0.79 ± 0.02	0.86 ± 0.02	0.89 ± 0.03	0.90 ± 0.02	0.88 ± 0.02	0.92 ± 0.01
LCN-Mean	0.94 ± 0.01	0.15 ± 0.06	0.21 ± 0.05	0.17 ± 0.08	0.12 ± 0.07	0.11 ± 0.02	0.12 ± 0.03	0.22 ± 0.01	0.36 ± 0.02
Normalized Sen Welfare									
Xi'an	2	3	4	5	6	7	8	9	10
GPI-LS	0.21 ± 0.04	0.26 ± 0.02	0.23 ± 0.01	0.25 ± 0.01	0.20 ± 0.01	−−	−−	−−	−−
PCN	0.87 ± 0.01	0.78 ± 0.01	0.67 ± 0.01	0.71 ± 0.01	0.64 ± 0.01	0.51 ± 0.01	0.46 ± 0.01	0.45 ± 0.01	0.43 ± 0.00
LCN	0.93 ± 0.01	0.91 ± 0.02	0.86 ± 0.02	0.91 ± 0.00	0.87 ± 0.01	0.79 ± 0.01	0.77 ± 0.01	0.62 ± 0.02	0.81 ± 0.01
LCN-Redist.	0.97 ± 0.01	0.96 ± 0.01	0.82 ± 0.02	0.77 ± 0.01	0.77 ± 0.01	0.62 ± 0.01	0.59 ± 0.01	0.54 ± 0.01	0.56 ± 0.01
LCN-Mean	0.90 ± 0.01	0.92 ± 0.02	0.87 ± 0.01	0.90 ± 0.01	0.84 ± 0.01	0.78 ± 0.01	0.71 ± 0.01	0.76 ± 0.01	0.80 ± 0.01
Amsterdam	2	3	4	5	6	7	8	9	10
GPI-LS	0.16 ± 0.03	0.54 ± 0.03	0.47 ± 0.03	0.32 ± 0.01	−−	−−	−−	−−	−−
PCN	0.82 ± 0.02	0.69 ± 0.01	0.42 ± 0.01	0.43 ± 0.01	0.49 ± 0.01	0.42 ± 0.01	0.34 ± 0.01	0.37 ± 0.01	0.22 ± 0.01
LCN	0.93 ± 0.04	0.88 ± 0.01	0.45 ± 0.07	0.61 ± 0.01	0.66 ± 0.02	0.73 ± 0.02	0.45 ± 0.01	0.89 ± 0.01	0.62 ± 0.04
LCN-Redist.	0.86 ± 0.07	0.71 ± 0.02	0.36 ± 0.03	0.40 ± 0.02	0.43 ± 0.02	0.41 ± 0.01	0.34 ± 0.01	0.42 ± 0.01	0.38 ± 0.01
LCN-Mean	0.99 ± 0.01	0.85 ± 0.02	0.90 ± 0.02	0.54 ± 0.03	0.47 ± 0.02	0.83 ± 0.01	0.39 ± 0.01	0.90 ± 0.01	0.82 ± 0.02
Gini Index (the lower the better)									
Xi'an	2	3	4	5	6	7	8	9	10
GPI-LS	0.30 ± 0.05	0.30 ± 0.03	0.46 ± 0.02	0.50 ± 0.02	0.49 ± 0.01	−−	−−	−−	−−
PCN	0.10 ± 0.01	0.18 ± 0.01	0.30 ± 0.01	0.30 ± 0.00	0.30 ± 0.00	0.34 ± 0.00	0.44 ± 0.00	0.43 ± 0.00	0.46 ± 0.00
LCN	0.06 ± 0.01	0.09 ± 0.01	0.19 ± 0.01	0.25 ± 0.01	0.23 ± 0.01	0.25 ± 0.00	0.33 ± 0.01	0.30 ± 0.00	0.30 ± 0.00
LCN-Redist.	0.01 ± 0.00	0.05 ± 0.00	0.18 ± 0.01	0.23 ± 0.01	0.21 ± 0.01	0.27 ± 0.01	0.33 ± 0.01	0.40 ± 0.01	0.45 ± 0.00
LCN-Mean	0.07 ± 0.01	0.10 ± 0.01	0.24 ± 0.01	0.25 ± 0.01	0.23 ± 0.01	0.26 ± 0.01	0.39 ± 0.01	0.30 ± 0.00	0.32 ± 0.00
Amsterdam	2	3	4	5	6	7	8	9	10
GPI-LS	0.54 ± 0.05	0.64 ± 0.02	0.62 ± 0.01	0.70 ± 0.01	−−	−−	−−	−−	−−
PCN	0.13 ± 0.02	0.48 ± 0.01	0.58 ± 0.01	0.67 ± 0.01	0.63 ± 0.01	0.66 ± 0.01	0.71 ± 0.01	0.72 ± 0.01	0.74 ± 0.01
LCN	0.03 ± 0.02	0.37 ± 0.01	0.43 ± 0.01	0.52 ± 0.01	0.47 ± 0.01	0.50 ± 0.00	0.60 ± 0.00	0.67 ± 0.01	0.57 ± 0.01
LCN-Redist.	0.06 ± 0.04	0.45 ± 0.02	0.61 ± 0.02	0.64 ± 0.01	0.62 ± 0.01	0.64 ± 0.01	0.69 ± 0.01	0.69 ± 0.00	0.71 ± 0.00
LCN-Mean	0.01 ± 0.00	0.40 ± 0.02	0.40 ± 0.01	0.56 ± 0.02	0.54 ± 0.01	0.50 ± 0.01	0.60 ± 0.01	0.65 ± 0.00	0.58 ± 0.00

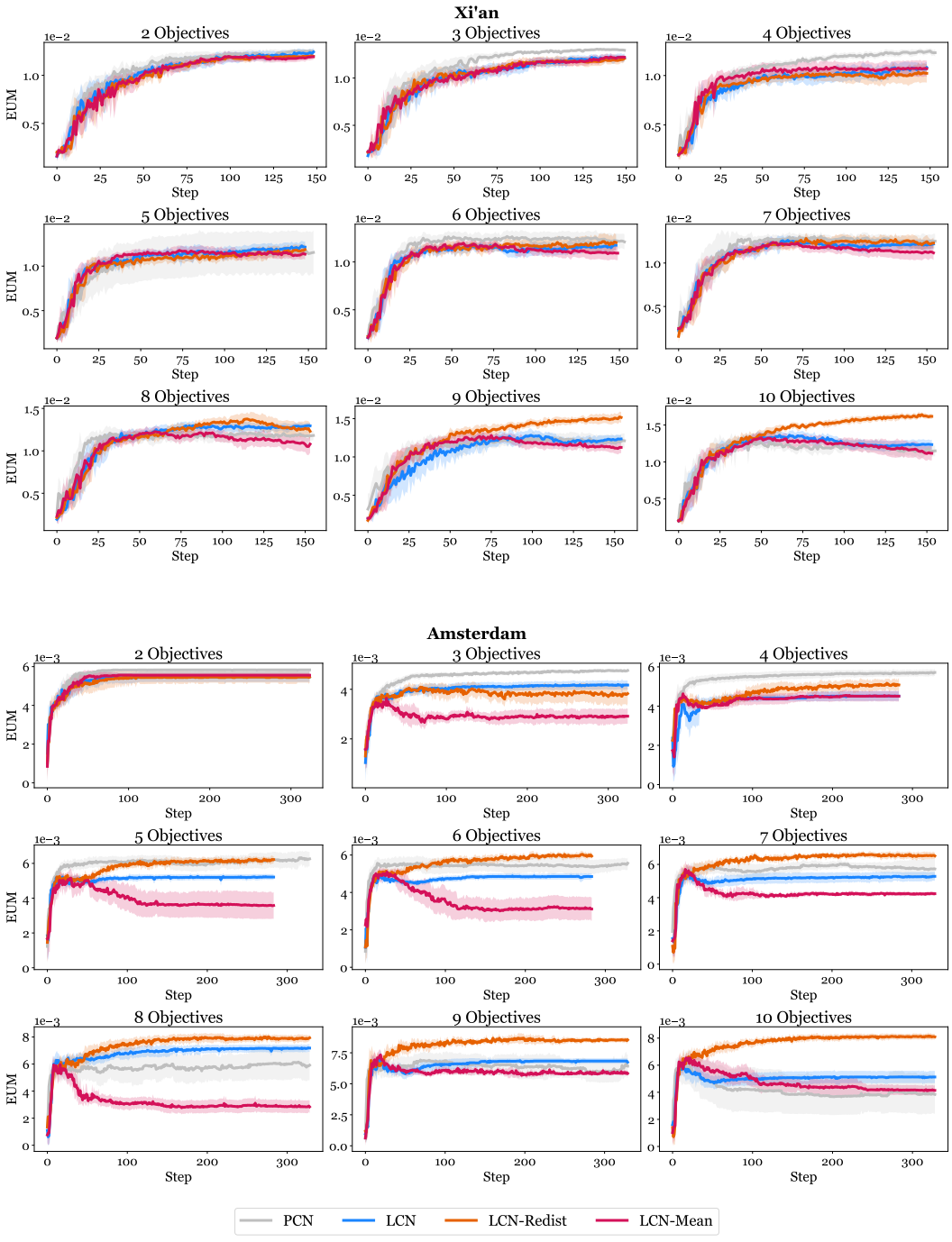

Fig. 15. Learning curves for EUM.

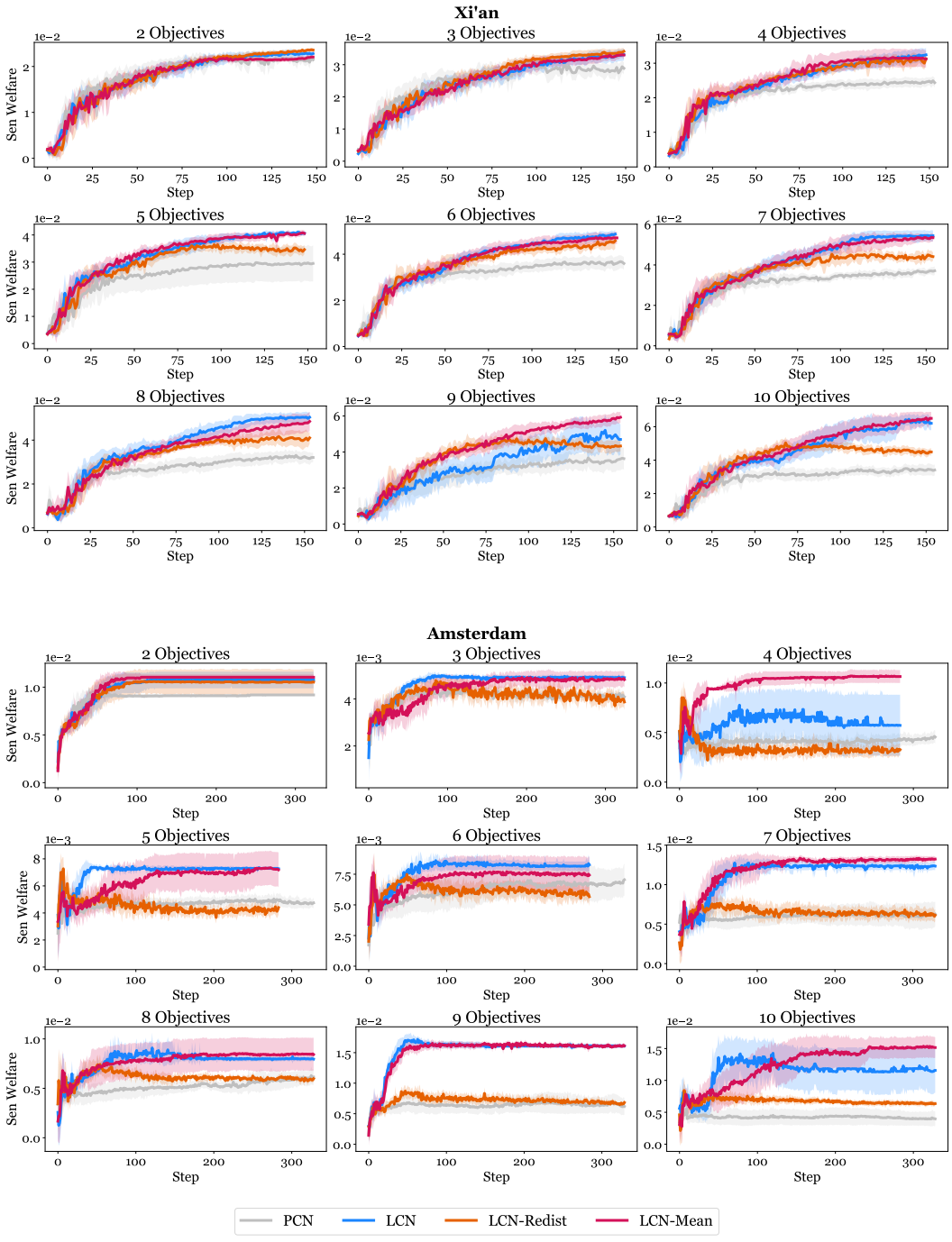

Fig. 16. Learning curves for Sen Welfare.

Received 17 July 2025; accepted 12 February 2026