MULTI-OBJECTIVE HIERARCHICAL REINFORCEMENT LEARNING FOR COMPLEX MULTI-DOMAIN, MULTI-TIMESCALE SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world decision-making problems often require integrating heterogeneous data sources, adapting to changing operating conditions, and optimizing multiple, often conflicting objectives across interacting domains and spatial and temporal scales. For these settings, we introduce *Systemic Contextual Multi-Objective Hierarchical Reinforcement Learning (SC-MOHRL)*, a three-level hierarchical RL framework with Pareto-based multi-objective optimization. SC-MOHRL addresses problems in which objectives and decisions operate at different temporal horizons by decoupling them across hierarchical levels, while explicitly coordinating their interactions through online Pareto-based decision-making. Beyond hierarchical objective decomposition, the method introduces a unified formulation with bidirectional contextual information flow, where lower-level contexts are propagated upward to inform high-level Pareto-based decision-making across temporal scales, while high-level decisions provide top-down guidance to lower levels. The proposed method is evaluated in a complex simulated urban system with three interacting objectives: traffic flow, air quality and livability. Simulation results and ablation studies demonstrate the feasibility of the framework and the contribution of its hierarchical and contextual information sharing mechanisms.

1 INTRODUCTION

Reinforcement learning (RL) formulates decision-making as an experience driven process, where an agent interacts with its environment and iteratively improves its policy. Real-world applications often require integrating heterogeneous data sources (e.g. behavioral and environmental signals), operating under non-stationarity and partial observability, and optimizing multiple, often conflicting objectives across interacting domains and spatial and temporal scales. We refer to such decision problems as systemic, as decisions and objectives in one domain may affect conditions and outcomes in others and therefore cannot be optimized independently. Multi-objective RL (MORL) (Vamplew et al., 2011) provides a principled basis for handling such trade-offs. In complex systems involving large state, action spaces, and long decision horizons, credit assignment and efficient policy learning become increasingly difficult, resulting in poor RL performance. Hierarchical RL (HRL) (Pateria et al., 2021) addresses this challenge by decomposing a long-horizon RL task into a hierarchy of subtasks, through a process called temporal abstraction (Kulkarni et al., 2016). Nevertheless, existing RL approaches, either purely multi-objective or hierarchical, or their combination multi-objective hierarchical RL (MOHRL), remain limited in capturing the coupled, dynamic, and discontinuous nature of real-world systems, particularly in the absence of mechanisms for incorporating contextual information and coordinating decisions across levels of abstraction.

Incorporating dynamic contextual information into MOHRL can enable more effective coordination across objectives, hierarchical levels, and timescales (Singh et al., 2025). Building on this idea, we present a new hierarchical RL formulation for multi-timescale multi-objective optimization. The key extension is that objectives and decisions are distributed across hierarchical levels operating at different temporal resolutions, while remaining coupled through the hierarchical decision process. Our Pareto-based, hierarchical decision-making framework *Systemic Contextual Multi-Objective Hierarchical Reinforcement Learning (SC-MOHRL)* coordinates these objective-timescale interactions through online Pareto-based decision-making, using bidirectional contextual information flow across

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

the hierarchy. Our framework employs a hybrid learning paradigm. The policies of the two lower hierarchical levels, which we will from now on refer to as the low and middle levels, are optimized using the on-policy Proximal Policy Optimization (PPO) algorithm, while strategic multi-objective decision-making is performed by an off-policy Pareto Conditioned Network (PCN) on the high level. We train and freeze the low- and middle-level policies first, and then optimize the high-level policy on stable learned dynamics. Such a training procedure addresses the instability in HRL, and thereby enables real-world complex simulations. An important enabler for coordinating the learning across levels is the bidirectional information flow across the hierarchy, from the lower to the high levels, and vice-versa. The low and middle levels summarize local observations into compact high-level representations becoming the context for decision-making at the upper levels. Lower level-specific information is extracted as objectives and propagated upward, so that the high-level policy can incorporate information about performance across the hierarchy when carrying out multi-objective optimization. The high level selects Pareto trade-offs and transforms the information into contextual guidance, further concatenated with local observations at the lower levels.

We demonstrate SC-MOHRL in a complex urban decision-making scenario involving traffic flow, air quality and livability, with three performance objectives, and three hierarchical levels operating at distinct temporal scales. This setting provides a challenging test case for the proposed approach, demonstrating its ability to jointly optimize conflicting objectives while coordinating decisions and information across multiple temporal levels. The key challenge addressed in this work is not hierarchical control or multi-objective optimization separately, but their interaction when objectives and decisions operate at different temporal scales. In such settings, objectives cannot be treated as a single vector optimized at one decision frequency, nor can hierarchical levels be optimized independently as decisions taken at one level alter the state and achievable trade-offs at the others. SC-MOHRL addresses this objective timescale mismatch by coupling hierarchical decision-making with online Pareto-based coordination. Lower-level contexts are propagated upward across the hierarchy to inform multi-objective reasoning, while high-level Pareto decisions provide top-down guidance to lower-level policies. This enables objectives, even conflicting, associated with different temporal processes to be coordinated without collapsing them into a fixed scalar reward. Our main contributions are: (1) a multi-timescale MOHRL formulation for objectives and decisions distributed across hierarchical levels operating at different temporal resolutions, (2) online Pareto-based cross-level coordination of objectives and decisions without fixed reward scalarization, (3) bidirectional cross-level context and information flow, coupling bottom-up information with top-down Pareto-based guidance, and (4) evaluation in a complex urban system with three temporal levels and three performance objectives: traffic flow, air quality and livability, demonstrating coordinated decision-making across strongly coupled processes and timescales.

2 Related Work

Real-world systems are characterized by structural complexity, uncertainty, and competing objectives across multiple timescales, for which nominal RL methods often struggle to achieve scalability, generalization, and robustness. MORL methods can be divided into single-policy (Schulman et al., 2017) and multi-policy (Liu et al., 2014) approaches, the former being computationally more efficient but requiring prior preference information and providing only a single trade-off solution, while the latter approximates a Pareto front, providing trade-off information among conflicting objectives (Barrett & Narayanan, 2008). Among multi-policy methods, scalarization-based approaches can extend deep Q-learning by projecting vector-valued returns onto preference vectors and iteratively searching for policies across the preference space (Mnih et al., 2015), or use a preference-adaptive multi-policy framework for learning under expected scalarized returns (Pyykölä et al., 2024). Pareto-based optimization methods instead learn a set of policies whose return vectors approximate the Pareto front (Van Moffaert & Nowé, 2014; Reymond et al., 2022; Röpke et al., 2024), thereby capturing different trade-offs among conflicting objectives, including solutions on non-convex regions of the Pareto fronts.

HRL decomposes complex decision-making into hierarchical sub-tasks (Pateria et al., 2021) allowing policies operating at different temporal and spatial scales to be learned separately. When integrated with multi-objective optimization, this hierarchy enables coordinated decision-making across conflicting objectives while preserving explicit trade-offs between them, facilitating consistent decision-making across multiple levels of abstraction and temporal resolutions (Sutton et al.,

1999). Combining MORL with hierarchical decision-making is challenging because objective trade-offs and temporal abstraction must be aligned across levels operating at different time scales, yet relatively few methodologies explicitly address this challenge. Recent advances in distributional MORL have further refined optimality concepts by considering preferences over entire return distributions rather than expectations alone (Cai et al., 2023), however these methods remain single level and therefore do not address how objective trade-offs should be coordinated across hierarchical decision levels and time scales. Likewise, methodologies joining HRL with MO optimization often rely on hard-coded task hierarchies that implicitly fix preferences, rather than modeling preferences explicitly as part of the learning problem (Zhao et al., 2020).

Recent MOHRL research introduces hierarchical structures to facilitate decision-making under multiple objectives in different real-world applications. Chang et al. (2022) and Ding et al. (2025) use preference-guided hierarchical goal selection to choose low-level policies with fixed rewards for achieving different objectives. In Kallaka et al. (2025); Su & Dong (2025); Choo & Oh (2026); Zhang et al. (2025a); Coskun et al. (2024), rewards corresponding to different objectives are scalarized for multi-objective optimization. However, these methods either do not explicitly optimize the Pareto front or evaluate the resulting Pareto solutions offline after training. Within the hierarchical approaches, some studies explicitly consider decision-making across multiple timescales. In Jin et al. (2025), high- and low-level policies operate over different action spaces and timescales but are jointly trained toward the same scalarized objective. In Gou et al. (2025), the high-level policy dynamically adjusts objective priorities to guide decentralized cooperative agents at the lower level, resulting in a single adaptive hierarchical policy. In Su et al., multiple temporal horizons are incorporated into actor-critic RL through components with different discount factors, without explicit hierarchical levels. The three-level EdgeTimer framework Hao et al. (2025), adapts decision timescales across hierarchical levels, but the RL agents optimize scalar rewards rather than Pareto-based objectives. Thus, while these approaches explicitly address temporal abstraction, they do not coordinate Pareto-based multi-objective trade-offs across hierarchy levels operating at different timescales.

A policy at one level cannot make meaningful decisions from its local information alone when the optimal action depends on conditions created at other levels. Context is a mechanism that makes the cross-level information usable. Luo et al. (2024) proposes a bidirectional-reachable Hierarchical Policy Optimization (BrHPO) algorithm incorporating a mutual response mechanism for context information exchange to improve subgoal-based HRL performance. However, it remains an open problem how contextual information can be incorporated into the learning process of a multi-objective hierarchy while retaining explicit Pareto trade-off reasoning. Singh et al. (2025) introduces context-intelligence, an explicit bi-directional cross-level communication mechanism to MOHRL that allows low-level policies to influence high-level decision-making. However, the work primarily addresses contextual information through relatively simple settings and not full algorithmic implementation and training in a full hierarchical learning framework while addressing multiple conflicting objectives. This leaves open how bidirectional context can be integrated into and learned within a MOHRL architecture. Our framework addresses this gap by incorporating cross-level contextual information directly into the hierarchical decision-making and learning process.

Most existing HRL methods communicate primarily top-down goals or sub-goals between hierarchical levels, while bottom-up information is typically derived from environment (Nachum et al., 2018). Some recent approaches additionally condition lower-level policies on multi-objective preferences (Su & Dong, 2025). In contrast, our SC-MOHRL framework enables explicit bidirectional information flow throughout the hierarchy. High-level actions constraint lower-level decision-making, while information about lower-level objectives and their outcomes can be propagated upward as context for higher-level decisions. This information may reflect conflicting objectives and directly condition high-level actions selection, while the high level uses an independently formulated reward vector for multi-objective optimization. Beyond directional information flow, HRL methods also differ in how learning and adaptation are coordinated across hierarchical levels. Some approaches train lower-level policies first and subsequently optimize higher-level policies while keeping the lower-level policies fixed (Haarnoja et al., 2018). Others alternate optimization between hierarchical levels (Drappo et al., 2024), or allow joint fine-tuning after initially training the levels separately (Zhang et al., 2025b). These approaches enable different forms of cross-level adaptation during training, but do not maintain online bidirectional cross-level coordination during deployment. In contrast,

SC-MOHRL maintains such coordination through bottom-up contextual and objective information and top-down Pareto-based guidance.

3 SC-MOHRL: A MULTI-OBJECTIVE HIERARCHICAL REINFORCEMENT LEARNING BASED FRAMEWORK

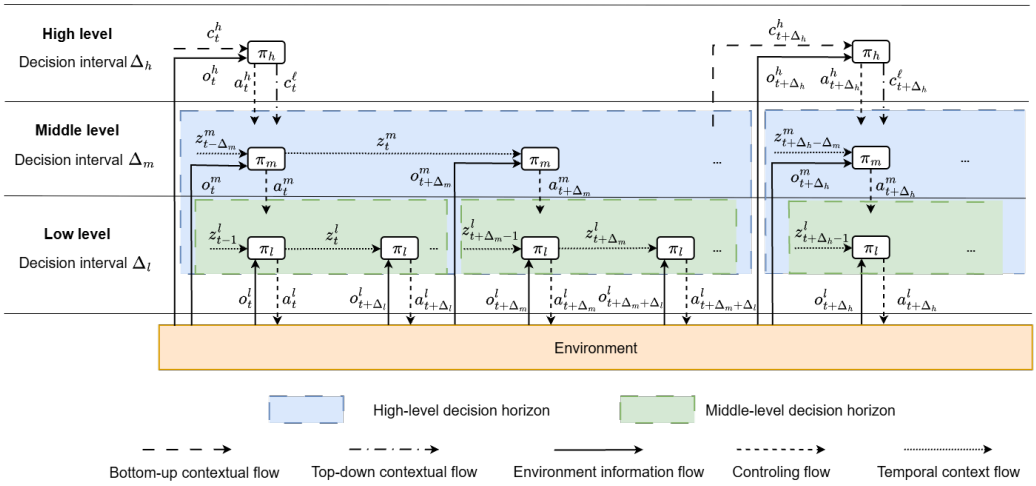


Figure 1: The SC-MOHRL framework. Each level $\hbar$ generates actions $a_t^{\hbar}, \hbar \in \{l, m, h\}$ at intervals $\Delta_{\hbar}$, conditioned on observations $o_t^{\hbar}$ and hierarchical context $c_t^{\hbar}$, lower levels additionally use temporal context $z_t^{\ell}, \ell \in \{l, m\}$. High-level Pareto guidance provides the lower-level hierarchical context, while lower-level information forms the high-level context for multi-objective reasoning. The lower-level temporal contexts concatenates the reward and action from the previous step.

Our SC-MOHRL framework for multi-timescale RL with support for conflicting objectives is presented in Figure 1. The framework decomposes the decision-making process into three hierarchical levels with different temporal resolutions: high, middle, and low levels. The two lower levels learn local control policies using PPO, while the high level performs Pareto-based multi-objective decision-making using PCN. Decisions and information are coupled across the hierarchies, high-level actions provide contextual guidance to lower levels, while lower level-specific objective information provide bottom-up context flow for high-level decision-making.

Training is performed in two stages. First, the low- and middle-level policies are trained jointly while high-level actions are sampled, allowing the lower levels to learn their coupled dynamics. The low- and middle-level policies are then frozen, and the high-level PCN is trained through interaction with them. The three levels act at their respective decision frequencies, maintaining bidirectional cross level coordination through top-down contextual guidance and bottom-up contextual information.

3.1 MULTI-TIMESCALE HIERARCHICAL FORMULATION

SC-MOHRL is formulated as a hierarchical Multi-objective Markov Decision Process (MOMDP), with the environment defined by the tuple $\mathcal{E} = (\mathcal{X}, \mathcal{A}, P, \mathbf{R}, \gamma)$

where $\mathcal{X}$ is the global state space with $x_t \in \mathcal{X}$ the global state of the environment at time step t, and $\mathcal{A}$ the action space. P represents the state transition probability under hierarchical actions $P : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to [0, 1]$. The single-objective reward function is extended to a vector-valued immediate reward function $\mathbf{R} : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to \mathbb{R}^d$, where $d \geq 2$ is the number of objectives and $\gamma \in [0, 1]$ is the discount factor weighting the importance of future rewards. The vector-valued value function ($\mathbf{V}^{\pi} \in \mathbb{R}^d$) of a multi-objective MDP policy is

$$\mathbf{V}^{\pi} = \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k \mathbf{r}_{k+1} | \pi, \mu], \tag{1}$$

where $\mathbf{r}_{k+1} = \mathbf{R}(x_k, a_k, x_{k+1})$ is the reward vector at step $k+1$, and μ is the initial-state distribution $\mu : \mathcal{X} \to [0, 1]$. In a single-objective setting, value functions provide means for solving the optimization problem, namely it is possible to find the optimal policy (π^*) by maximizing the expected cumulative discounted reward. However, in multi-objective setting this does not always hold (Hayes et al., 2022) and finding a Pareto front describing the trade-offs between conflicting objectives and providing a set of non-dominated policies is required. A policy is non-dominated if there exists no other policy whose value is at least as good in all objectives and strictly better in at least one objective

$$V^{\pi} \succ_p V^{\pi'} \Leftrightarrow (\forall i : V_i^{\pi} \geq V_i^{\pi'}) \wedge (\exists i : V_i^{\pi} > V_i^{\pi'}). \tag{2}$$

Pareto front consists of the value vectors of all non-dominated policies and Pareto coverage set is a set of policies whose value vectors lie on the Pareto Front

$$\Pi^* = \{\pi \in \Pi \mid \nexists \pi' \in \Pi : V^{\pi'} \succ_p V^{\pi}\}. \tag{3}$$

We define $\mathcal{H} = \hbar \in \{l, m, h\}$ as the set of different hierarchical decision levels, where l, m and h correspond to the low, middle and high levels, respectively. All hierarchical levels interact with the same environment, but operate on different representations of the global state according to their respective decision roles and at different temporal resolutions. As shown in Figure 1, Δ_h, Δ_m, Δ_l are high-, middle-, low-level decision intervals, respectively, satisfying $\Delta_h > \Delta_m > \Delta_l$. Let T denote the episode horizon, then the set of decision epochs for each hierarchical level are defined as

$$\mathcal{T}^{\hbar} = \{t \mid t \bmod \Delta_{\hbar} = 0\}, \qquad \hbar \in \mathcal{H}, \qquad t = 0, ..., T-1, \tag{4}$$

and t is the global discrete time step. For each hierarchical level $\hbar \in \mathcal{H}$ and level-$\hbar$ decision epoch $t_{\hbar} \in \mathcal{T}^{\hbar}$, the level-specific observations $o_{t_{\hbar}}^{\hbar}$ and actions $a_{t_{\hbar}}^{\hbar}$ with level-specific action space $\mathcal{A}^{\hbar}$ are

$$o_{t_{\hbar}}^{\hbar} = \phi_{\hbar}(x_{t_{\hbar}}), \qquad a_{t_{\hbar}}^{\hbar} \in \mathcal{A}^{\hbar}, \tag{5}$$

where $\phi^{\hbar} : \mathcal{X} \to \mathcal{O}^{\hbar}$ and $\mathcal{O}^{\hbar}$ are the observation mapping and space, respectively.

3.2 Bidirectional Cross-level Information Flow

Following the definition for hierarchical formulation in Section 3.1, let ℓ denote the low or middle hierarchical level, where $\ell \in \{l, m\} \subset \mathcal{H}$. At each high-level decision time t_h, information from the middle and low levels is propagated upward as *hierarchical context* $c_{t_h}^{h}$ for high-level decision-making. The high-level reward is formulated, using hierarchical context information, as a vector composed of possible conflicting objectives. Correspondingly, high-level actions are propagated downward as *hierarchical context* $c_{t_\ell}^{\ell}$ and condition decision-making at the middle and low levels. The bidirectional hierarchical context mechanism is defined as

$$c_{t_h}^{h} = \{o_{\tau_\ell(t_h)}^{\ell}\}_{\ell \in \{l,m\}}, \qquad c_{t_\ell}^{\ell} = a_{\tau_h(t_\ell)}^{h}, \tag{6}$$

where, for $i < j$, $\tau_j(t_i)$ denotes the latest higher-level decision epoch before t_i, and $\tau_i(t_j)$ the latest lower-level decision epoch at or before t_j as

$$\tau_j(t_i) = \max\{t \in T^j \mid t < t_i\}, \qquad \tau_i(t_j) = \max\{t \in T^i \mid t \leq t_j\}. \tag{7}$$

3.3 Lower-level Policy Learning

Given the hierarchical contexts defined above, lower-level policies are optimized using recurrent PPO (Schulman et al., 2017), which is a widely used policy-gradient actor-critic RL framework. The *temporal context* $z_{t_\ell}^{\ell}$ combines the previous action and reward generated within the level ℓ

$$z_{t_\ell}^{\ell} = [a_{t_\ell - \Delta_\ell}^{\ell}, r_{t_\ell - \Delta_\ell}^{\ell}]. \tag{8}$$

The policy at the level ℓ is conditioned both on its level-specific observation $o^\ell_{t_\ell}$, temporal context $z^\ell_{t_\ell}$, and hierarchical context $c^\ell_{t_\ell}$. Consequently, the policy is represented as $\pi_{\theta_\ell}(a^\ell_{t_\ell}|o^\ell_{t_\ell}, z^\ell_{t_\ell}, c^\ell_{t_\ell})$. To capture the temporal dependencies in the level-specific decision process, the observation, temporal and hierarchical contexts are concatenated as $\xi^\ell_{t_\ell} = [o^\ell_{t_\ell}, z^\ell_{t_\ell}, c^\ell_{t_\ell}]$. The resulting input is processed by an LSTM module (Hochreiter & Schmidhuber, 1997) as

$$h^\ell_{t_\ell} = \text{LSTM}_\ell(\xi^\ell_{t_\ell}, h^\ell_{t_\ell - \Delta_\ell}). \tag{9}$$

To preserve both historical information and the current level-specific observation, the hidden state is concatenated with the current level-specific observation and projected through a fully connected layer to obtain a latent embedding e^ℓ_t that integrates both temporal and current information as

$$e^\ell_t = f_\ell([h^\ell_t, o^\ell_t]). \tag{10}$$

The embedding is shared by the actor and critic networks of PPO. The actor outputs the policy distribution $\pi_{\theta_\ell}(a^\ell_t|e^\ell_t)$, which is used for action selection. Meanwhile, the critic estimates the corresponding state value function for advantage computation.

The middle- and low-level policies interact with the same environment and are hierarchically coupled during trajectory collection. Each level maintains its own trajectory and independently optimizes its policy and value network using PPO. The actor and critic are optimized by PPO based on the recurrent representation e^ℓ_t.

3.4 High-level Pareto-based Optimization

High-level policy in the proposed SC-MOHRL is responsible for multi-objective optimization at the largest temporal scale. SC-MOHRL employs a Pareto Conditioned Network (PCN) (Reymond et al., 2022) to learn policies corresponding to different trade-offs among multiple objectives. The PCN-based high-level policy $\pi_{\theta_h}\left(a^h_{t_h} \mid o^h_{t_h}, \mathbf{G}^{\text{des}}_{t_h}, H_{t_h}\right)$ is conditioned on the desired multi-objective return vector $\mathbf{G}^{\text{des}}_{t_h}$ and the remaining horizon $H_{t_h} = \frac{T-t_h}{\Delta_h}$ at the decision epoch t_h. The multi-objective return vector from the trajectory at t_h is

$$\mathbf{G}^h_{t_h} = \sum_{j=0}^{\frac{T-t_h}{\Delta_h}-1} \gamma^j \mathbf{r}^h_{t_h + j\Delta_h}, \tag{11}$$

where $\mathbf{r}^h_{t_h} = F(c^h_{t_h})$ is the high-level reward vector which contains objectives extracted by function F from the lower hierarchical levels context $c^h_{t_h}$. During training, the achieved returns are evaluated according to Pareto dominance, and non-dominated trajectories provide candidate behaviors for constructing the PCN training datasets.

3.5 Two-stage Training Algorithm

There are two training stages for the proposed algorithm. At the first stage, the low- and middle-level policies are trained jointly, with the high-level actions sampled randomly. At the second stage, high-level policy is trained with the low- and middle-level policies frozen and loaded.

During the lower-level training stage, the parameters of the middle- and low-level policies, θ_m and θ_l, are iteratively updated according to their corresponding policy gradients. The parameters are also influenced by the policy from another level through the interaction within the shared environment

$$\begin{aligned} \theta_m^{(t_m+\Delta_m)} &= \theta_m^{(t_m)} + \alpha_m \widehat{\nabla}_{\theta_m} \mathcal{L}_m^{\text{CLIP}}(\theta_m^{(t_m)}; \theta_l^{(\tau_l(t_m))}), \\ \theta_l^{(t_l+\Delta_l)} &= \theta_l^{(t_l)} + \alpha_l \widehat{\nabla}_{\theta_l} \mathcal{L}_l^{\text{CLIP}}(\theta_l^{(t_l)}; \theta_m^{(\tau_m(t_l))}). \end{aligned} \tag{12}$$

α is the learning rate, and L^{CLIP} is the clipped surrogate objective in PPO (Schulman et al., 2017). After the lower-level training stage, the final trained parameters at low and middles levels are θ^*_m and

θ_l^*, where $(\theta_m^*, \theta_l^*) = (\theta_m^{(T)}, \theta_l^{(T)})$. In the high-level training stage, the high-level policy is trained with the frozen policies from the lower-level training stages with the cross-entropy loss $\mathcal{L}_h$ in PCN (Reymond et al., 2022).

$$\theta_h^* = \arg\min_{\theta_h} \mathcal{L}_h \left(\theta_h; \theta_m^*, \theta_l^*\right). \tag{13}$$

4 Performance Evaluation in Urban Simulation

The SC-MOHRL framework was evaluated in a simulated urban traffic environment, optimizing three sustainability-related objectives, traffic flow, air quality and livability. The simulation environment was created using Simulation of Urban MObility (SUMO) (Lopez et al., 2018), an open-source microscopic traffic simulator. Vehicle data were obtained from publicly available repositories, while the air quality and livability variables were parametrized according to values reported in (Järvi et al., 2024). SC-MOHRL was applied to model a hierarchy of interacting decision-makers, with each level corresponding to a different level of authority in the urban transportation system, that is urban planner, traffic authority, and road user. The high-level policy controlled the share of trips made by cars relative to walking, biking and bus users, middle-level policy set the speed limit, and the low-level homogeneous agents controlled their speed. The simulation was implemented using ten homogeneous agents. The framework considered three interacting objectives, traffic efficiency, air quality and livability. In the evaluation setup, only the first two were optimized directly, while livability was heavily determined by air quality and therefore did not introduce an independent optimization direction.

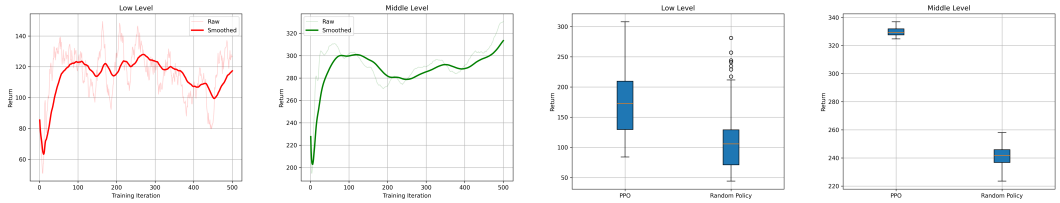
Direct comparison to existing MOHRL methods would require substantial adaptation to match the hierarchical structure, temporal coordination, objective formulation, and cross-level information flow of our setting. Such adaptations would introduce additional design choices and make it difficult to compare performance differences to the original methods themselves. We therefore evaluate SC-MOHRL through controlled component-wise comparisons and ablations that preserve the same hierarchical decision process and isolate the contributions of learned lower-level control and high-level Pareto-based coordination. The implementation details were as follows. Each episode lasted 20 hours. During one episode, low-, middle-, and high-level policies operated at decision intervals of 10 seconds, 10 minutes, and 50 minutes, respectively. The decision intervals were set to $\Delta_l = 1, \Delta_m = 60, \Delta_h = 300$, and the episode horizon $T = 7200$. One environment time step corresponded to 10 seconds. Therefore, each episode contained 7200, 120 and 24 decision epochs at the low, middle, and high levels, respectively.

4.1 Hierarchical PPO at Lower Levels

Both lower levels PPO models incorporate LSTM networks with learning rates 1.0×10^{-4}, discount factors 0.99, and 500 training episodes. The training and evaluation results are shown in Figure 2. In Figures 2a and 2b, the x-axis are training iterations, each including one episode, and y-axis are returns from each level. As shown in the figure, the agents learn useful actions early in training. The low-level return then stabilizes, with some fluctuations caused by updates to the middle-level policy that guides low-level decisions. Figures 2c and 2d indicate the evaluation results of the lower-level trained policies compared to random policies. Both levels achieve higher and more concentrated returns than random policies.

4.2 Multi-objective Optimization

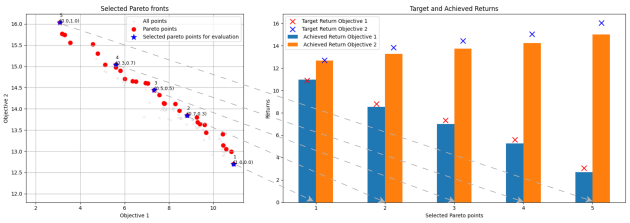
At the high level, the training set comprised of 100 transitions updated at every episode. At each training iteration, 20 Pareto points were selected to update the PCN. Figure 3a shows the Pareto evaluation results. Red points represent solutions under specific preferences, while the axes correspond to the two objectives. Figure 3b indicates the evaluation results on different selected Pareto points (blue stars in Figure 3a). These points cover different regions of the Pareto front, including both extreme and intermediate trade-offs between objectives. In Figure 3b, the cross markers indicate the desired returns, while the bars represent the actual returns achieved by the trained model. The



(a) (b) (c) (d)

Figure 2: Training and evaluation results on lower levels: (a) low-level, and (b) middle-level training returns, (c) low-level, and (d) middle-level evaluation results

evaluation results show that the trained policy is achieving the target return on different preferences. The hypervolume, overall mean error and error standard deviations are reported in Table 1.

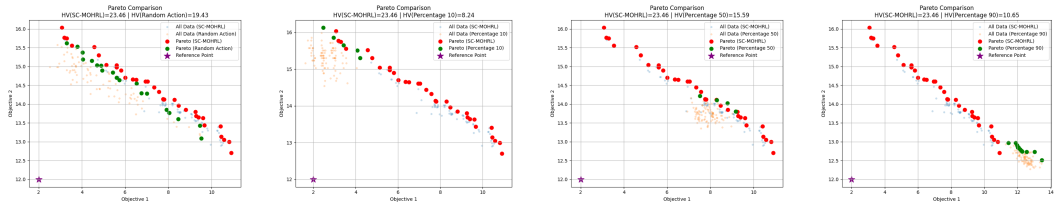


(a) (b)

Figure 3: High-level PCN Pareto front (a) and preference evaluation on selected Pareto points (b).

4.3 Ablation Study

To evaluate the contribution of the high-level policy, we replaced it with either random actions or fixed driving shares of 10% (low), 50% (middle), and 90% (high), while keeping the low- and middle-level policies unchanged. Figure 4 compares the resulting Pareto fronts. Using the same reference point (2,11), the proposed method achieves the highest hypervolume (Table 1), and unlike the fixed or random baselines, adapts the high-level decision-making to different preference settings. To evaluate the contribution of the learned lower-level policies, the trained high-level policy was kept fixed while the lower-level policies were replaced by either random actions or fixed-speed policies of 30, 50, and 70 km/h. Performance was evaluated at five selected points on the Pareto front (Figure 3) using the L2 distance between the target and achieved return vectors. Each point was evaluated over 50 Monte Carlo runs, and the mean errors and standard deviations over all points are reported in Table 1. As shown in Figure 5, the proposed method gives the lowest error at the first three Pareto points, while the 30 km/h policy gives lower errors and error standard deviations at the last two points, and the random policy also gives lower error but with a higher standard deviation. Overall, however, the learned lower-level policies provide the lowest mean error and error standard deviations across the evaluated preference points.



(a) Random policy (b) Driving percentage 10 (c) Driving percentage 50 (d) Driving percentage 90

Figure 4: Pareto front comparisons between the proposed method and 4a, random high-level actions, and 4b, 4c, and 4d, with driving percentages fixed at 10%, 50%, and 90%, respectively.

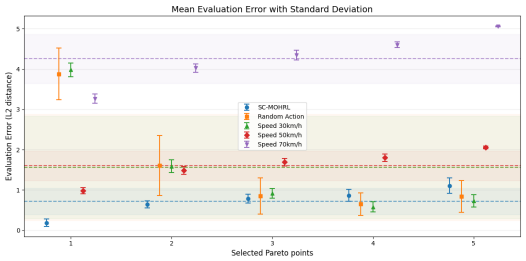


Figure 5: Mean error and standard deviation for the proposed method compared to one without trained lower-level policies on the five selected Pareto points.

Table 1: Performance comparison of the proposed methods and ablated variants

Lower-level policies	**High-level policy**	**Trade-off**	**Hypervolume ↑**	**Mean error ↓**	**Std ↓**
trained	trained	✓	**22.46**	**0.7165**	**0.3321**
trained	random	×	19.43	–	–
trained	driving 10%	×	8.24	–	–
trained	driving 50%	×	15.59	–	–
trained	driving 90%	×	10.65	–	–
random	trained	✓	–	1.5639	1.3140
speed 30 km/h	trained	✓	–	1.5591	1.2707
speed 50 km/h	trained	✓	–	1.6032	0.3689
speed 70 km/h	trained	✓	–	4.2574	0.6087

5 Conclusions

We present SC-MOHRL, a three-level hierarchical RL framework with Pareto-based multi-objective optimization that integrates bidirectional cross-hierarchy contextual information and decouples decision-making across different temporal resolutions. The method is evaluated under a complex urban system, and the evaluation results demonstrate the feasibility of the policies. The challenge in the current framework is that the performance of the offline RL relies heavily on the quality of pre-collected datasets. The proposed method was evaluated in a single complex system that provides a realistic and sufficiently demanding test environment. Extending the evaluation to additional systems would require substantial effort to construct and validate comparable simulation environments, making such an extension beyond the scope of the present study. In future work, we plan to formulate the problem as Partially Observable Markov Decision Process (POMDP), allowing the method to explicitly account for incomplete and uncertain observations of the system state.

AI use statement

In this work,we have not used generative AI tools for helping develop theoretical models or conceptual frameworks, design or provide feedback on research methodology or experiments, implement methods, assist with translation, support qualitative and thematic data analysis, interpret results, and generate synthetic data sets, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, clean and reformat dataset are not applicable to this work.

Additionally, we used generative AI tools for creating and editing software codes, aiding writing to improve readability (mostly checking grammar and spelling errors), and helping generate LaTeX equation expressions (mostly in Section 3 and Appendix A). We have reviewed all AI-assisted work. LLM-generated codes were verified and tested for correctness, the LLM-generated texts were carefully checked and modified. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility statement

To ensure reproducibility of the experimental result of SC-MOHRL, we provide the training codes in the supplementary material. The detailed design of the states, actions, rewards on each hierarchical level is presented in Appendix A, and the training details and parameters are described in Section 4.

References

Leon Barrett and Srini Narayanan. Learning all optimal policies with multiple criteria. In *Proceedings of the 25th international conference on Machine learning*, pp. 41–47, 2008.

Xin-Qiang Cai, Pushi Zhang, Li Zhao, Jiang Bian, Masashi Sugiyama, and Ashley Llorens. Distributional pareto-optimal multi-objective reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 15593–15613. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/32285dd184dbfc33cb2d1f0db53c23c5-Paper-Conference.pdf.

Jingru Chang, Dong Yu, Zheng Zhou, Wuwei He, and Lipeng Zhang. Hierarchical reinforcement learning for multi-objective real-time flexible scheduling in a smart shop floor. *Machines*, 10(12): 1195, 2022.

Minsol Choo and Sangyoon Oh. Gasched: Goal-adaptive hierarchical reinforcement learning for multi-objective hpc job scheduling. In *2026 IEEE 26th International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pp. 188–197. IEEE, 2026.

Serdar Coskun, Ozan Yazar, Fengqi Zhang, Lin Li, Cong Huang, and Hamid Reza Karimi. A multi-objective hierarchical deep reinforcement learning algorithm for connected and automated hevs energy management. *Control Engineering Practice*, 153:106104, 2024. doi: https://doi.org/10.1016/j.conengprac.2024.106104.

Linshan Ding, Zailin Guan, Dan Luo, and Lei Yue. Data-driven hierarchical multi-policy deep reinforcement learning framework for multi-objective multiplicity dynamic flexible job shop scheduling. *Journal of Manufacturing Systems*, 80:536–562, 2025.

Gianluca Drappo, Alberto Maria Metelli, and Marcello Restelli. A provably efficient option-based algorithm for both high-level and low-level learning. *Reinforcement Learning Journal*, 2:819–839, 2024.

Pak Lun Fung, Salla Sillanpää, Jarkko V Niemi, Anu Kousa, Hilkka Timonen, Martha Arbayani Zaidan, Erkka Saukko, Markku Kulmala, Tuukka Petäjä, and Tareq Hussein. Improving the current air quality index with new particulate indicators using a robust statistical approach. *Science of The Total Environment*, 844:157099, 2022.

Ákos Gosztonyi. *The Weight of Urban Air: Investigating the Disproportionate Burden of Urban Air Pollution in Helsinki Metropolitan Area, Finland*. PhD thesis, University of Helsinki, 2026. URL https://helda.helsinki.fi/items/ec59a2bb-d6ea-45c4-839d-6a777730aa69.

Yingdong Gou, Siwen Wei, Kai Xu, Jiancheng Liu, Ke Li, Bo Li, Zaikun Han, Xin Lai, Yuqing Lin, Dingrui Xue, et al. Hierarchical reinforcement learning with kill chain-informed multi-objective optimization to enhance resilience in autonomous unmanned swarm. *Neural Networks*, pp. 108255, 2025.

Tuomas Haarnoja, Kristian Hartikainen, Pieter Abbeel, and Sergey Levine. Latent space policies for hierarchical reinforcement learning. In *International Conference on Machine Learning*, pp. 1851–1860. PMLR, 2018.

Yijun Hao, Shusen Yang, Fang Li, Yifan Zhang, Shibo Wang, and Xuebin Ren. Learning adaptive multi-timescale scheduling for mobile edge computing. *IEEE Transactions on Mobile Computing*, 24(8):7297–7311, 2025.

C.F. Hayes, R. Rădulescu, E. Bargiacchi, and et al. A practical guide to multi-objective reinforcement learning and planning. *Auton Agent Multi-Agent Syst*, 36(26), 2022. doi: 10.1007/s10458-022-09552-y.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Leena Järvi, Sasu Karttunen, Nahid Atashi, and Xiaoyu Li. Advancements in understanding the local-scale urban air quality and impact of urban planning using large eddy simulation. In *AGU Fall Meeting Abstracts*, volume 2024, pp. A34H–06, 2024.

Guizhe Jin, Zhuoren Li, Bo Leng, Ran Yu, Lu Xiong, and Chen Sun. Multi-timescale hierarchical reinforcement learning for unified behavior and control of autonomous driving. *IEEE Robotics and Automation Letters*, 2025.

Ruechuda Kallaka, Junjie Zhao, Rodolphe Fremond, and Antonios Tsourdos. Hierarchical reinforcement learning for multi-objective uav routing considering operational complexities. In *2025 Integrated Communications, Navigation and Surveillance Conference (ICNS)*, pp. 1–13. IEEE, 2025.

Tejas D. Kulkarni, Karthik R. Narasimhan, Ardavan Saeedi, and Joshua B. Tenenbaum. Hierarchical deep reinforcement learning: integrating temporal abstraction and intrinsic motivation. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 3682–3690, 2016.

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

Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wiessner. Microscopic traffic simulation using sumo. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2575–2582, 2018. doi: 10.1109/ITSC.2018.8569938.

Yu Luo, Fuchun Sun, Tianying Ji, and Xianyuan Zhan. Bidirectional-reachable hierarchical reinforcement learning with mutually responsive policies. *Reinforcement Learning Journal*, 2:733–762, 2024.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

Shubham Pateria, Budhitama Subagdja, Ah-hwee Tan, and Chai Quek. Hierarchical reinforcement learning: A comprehensive survey. *ACM Comput. Surv.*, 54(5), 2021. doi: 10.1145/3453160.

Sara Pyykölä, Klavdiya Olegovna Bochenina, and Laura Ruotsalainen. Conciliator steering: Imposing user preference in multi-objective reinforcement learning. *Transactions on Machine Learning Research*, 2024.

Mathieu Reymond, Eugenio Bargiacchi, and Ann Nowe. Pareto conditioned networks. In *The 21st International Conference on Autonomous Agents and Multiagent Systems*, pp. 1110–1118. IFAAMAS, May 2022.

Willem Röpke, Mathieu Reymond, Patrick Mannion, Diederik M Roijers, Ann Nowé, and Roxana Rădulescu. Divide and conquer: Provably unveiling the pareto front with multi-objective reinforcement learning. *arXiv preprint arXiv:2402.07182*, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Niharika Singh, Volodymyr Beimuk, Matti Leinonen, and Laura Ruotsalainen. Contextual intelligence based multi-objective hierarchical reinforcement learning. 2025.

Jiantao Su and Saohua Dong. Multi-objective optimization for dynamic logistics scheduling based on hierarchical deep reinforcement learning. *Scientific Reports*, 15(1):33544, 2025.

Zhan Su, Peixi Peng, Xinyu Hu, Cong Li, Yisen Zhao, Zhuojian Li, Yonghong Tian, and Fanqi Shen. Multi-timescale reinforcement learning by value reconstruction. In *Forty-third International Conference on Machine Learning*.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: a framework for temporal abstraction in reinforcement learning. *Artif. Intell.*, 112(1–2):181–211, 1999. doi: 10.1016/S0004-3702(99)00052-1.

Peter Vamplew, Richard Dazeley, Adam Berry, Rustam Issabekov, and Evan Dekker. Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Machine Learning*, 84 (1):51–80, 2011. doi: 10.1007/s10994-010-5232-5.

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

Haijian Zhang, Yao Ge, Xiuyuan Zhao, and Jiyuan Wang. Hierarchical deep reinforcement learning for multi-objective integrated circuit physical layout optimization with congestion-aware reward shaping. *IEEE Access*, 2025a.

Ruize Zhang, Sirui Xiang, Zelai Xu, Feng Gao, Shilong Ji, Wenhao Tang, Wenbo Ding, Chao Yu, and Yu Wang. Mastering multi-drone volleyball through hierarchical co-self-play reinforcement learning. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 5278–5300. PMLR, 27–30 Sep 2025b. URL `https://proceedings.mlr.press/v305/zhang25n.html`.

Dongyang Zhao, Liang Zhang, Bo Zhang, Lizhou Zheng, Yongjun Bao, and Weipeng Yan. Mahrl: Multi-goals abstraction based deep hierarchical reinforcement learning for recommendations. In *SIGIR*, pp. 871–880, 2020. doi: 10.1145/3397271.3401170.

A Appendix: Environment Settings in the Simulation

State Space

The state of the low level is formulated as:

$$o_{t_l}^l = [v_{t_l}, \Delta v_{t_l}, D_{t_l}], \tag{14}$$

where v_{t_l} is the average speed of a vehicle, $\Delta v_{t_l} = v_{t_l} - v_{t_l - \Delta_l}$, and d_{t_l} is the vehicle's distance to the destination.

The simulation area is divided into cells in which air quality is measured and represented by a vector $\mathbf{AQ}_t$. The considered pollutants are particulate matter with a diameter below 2.5μm (PM2.5), black carbon (BC), and the total particle number concentration (Ntot). The pollutant concentration is defined by the function

$$\mathbf{AQ}_{t_m} = f_{\mathrm{AQ}}\left(N_{\mathrm{veh},t_m}, \bar{v}_{t_m}\right) \in \mathbb{R}_{\geq 0}^{N_c \times 3}, \qquad f_{\mathrm{AQ}} : \mathbb{N}_0 \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}^{N_c \times 3}, \tag{15}$$

where N_{veh,t_m} is the number of vehicles, $\bar{v}_{t_m}$ is the average vehicle velocity, and N_c is the number of cells. f_{AQ} is a mathematical model based on Parallelized Large-Eddy Simulation (LES) Model (PALM) data (Järvi et al., 2024).

States of the middle and high levels are

$$o_t^j = [\bar{\mathbf{AQ}}_{t_j}, \bar{v}_{t_j}, p_{veh,t_j}], \quad j \in \mathcal{H}_l^+ = \{m, h\}, \tag{16}$$

where $\bar{\mathbf{AQ}}_{t_j}$ is the mean air quality in the area,

$$\bar{\mathbf{AQ}}_{t_j} = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathbf{AQ}_{i,t_j} \in \mathbb{R}^3_{\geq 0}, \tag{17}$$

$\bar{v}_{t_j}$ is the average vehicle speed, and p_{veh,t_j} is the proportion of agents using driving as their mobility mode.

ACTION SPACE

Low-level objective is to improve traffic efficiency by enabling each agent to reach its destination as quickly as possible. At low-level decision steps, the agent selects an action from the action space $\mathcal{A}^l$

$$a^l_{t_l} \in \mathcal{A}^l = \{a^+, a^-, a^0\}, \tag{18}$$

where a^+, a^- and a^0 represent acceleration, deceleration and maintaining the current motion state, respectively.

Actions at middle level are defined as

$$a^m_{t_m} = v^{\lim}, \qquad v^{\lim} \in \mathcal{A}^m = \{30, 50, 70\}\ \mathrm{km/h}, \tag{19}$$

where $v^{\lim}$ is the speed limit.

At the high level, the policy adjusts the percentage of agents using driving as their mobility mode. At each high-level decision step, the agent selects an action from

$$a^h_{t_h} \in \mathcal{A}^h = \{+\delta, -\delta, 0\}, \tag{20}$$

where δ is a fixed adjustment to the percentage of agents using driving as their mobility mode. The actions $+\delta$, $-\delta$, and 0 correspond to increasing, decreasing, and maintaining this percentage, respectively.

REWARD FUNCTION FORMULATION

At the low level, the reward corresponds to the agents progress on reaching its destination and thus the traffic efficiency in the environment. The reward function is defined as

$$r^l_{t_l} = \frac{D_{t_l} - D_{t_l+\Delta_t}}{D_0}, \tag{21}$$

where $D_{t_l} - D_{t_l+\Delta_t}$ is the distance remaining after a time step and D_0 is the total distance to destination at the start of the travel.

At the middle level, the reward is a combination of an air quality reward and traffic penalty. The air quality reward function compares the observed air quality with predefined that indicate the maximum acceptable pollution concentration in the area. The overall air quality is defined by Equation 15. The reward is determined by comparing the resulting air quality values with the threshold vector $\mathbf{q} \in \mathbb{R}^3_{\geq 0}$ containing air quality thresholds for PM2.5, BC and Ntot, provided by Fung et al. (2022).

$$r_{AQ,t_m} = \begin{cases} y_{t_m}, & y_{t_m} \geq 0, \\ -y^2_{t_m}, & y_{t_m} < 0, \end{cases} \tag{22}$$

where

$$y_{t_m} = \min_{j \in \{1,2,3\}} \left(q_j - \bar{AQ}^{(j)}_{t_m} \right). \tag{23}$$

The traffic penalty $p_{t_m}^{\text{traf}}$ is calculated as

$$p_{t_m}^{\text{traf}} = \begin{cases} p_{\min} + \dfrac{p_{\max} - p_{\min}}{1 + \exp\left(k_s(\bar{v}_{t_m} - v_0)\right)}, & \bar{v}_{t_m} < v_{\text{th}}, \\ 0, & \bar{v}_{t_m} \geq v_{\text{th}}. \end{cases} \tag{24}$$

The parameters $p_{\min}$ and $p_{\max}$ represent the minimum and maximum penalty factors, respectively, v_0 defines the midpoint of the logistic function, k_s controls the steepness of the function and v_{th} denotes the speed threshold above which no traffic penalty is applied.

The middle-level reward is defined as the combination of the air quality reward and traffic penalty

$$r_{t_m}^m = r_{AQ,t_m} - p_{t_m}^{\text{traf}}. \tag{25}$$

The simulated urban traffic environment optimizes three objectives: traffic efficiency, air quality and livability. PCN is used at the high level to optimize the first two objectives: traffic efficiency and air quality. A reward vector is formulated for the multi-objective approach by combining information from the low- and middle-levels. Based on Equation 26,

$$\mathbf{r}_{t_h}^h = F(c_{t_h}^h) = [\bar{r}_{t_h}^l, r_{AQ,t_h}]. \tag{26}$$

where $\bar{r}_{t_h}^l$ is the mean reward from low-level homogeneous agents.

The third objective is livability, which combines air quality with demographic characteristics. Since the demographic factors are fixed during optimization, changes in livability are driven by changes in air quality. Thus, although livability is not represented as a separate objective in the PCN, it is implicitly optimized through the air quality objective. The livability function r_{liv} is defined by a relocation rate as:

$$r_{\text{liv}} = 1 - \phi_{\text{relocation_rate}}. \tag{27}$$

The relocation rate is calculated using a logit model (adapted from Gosztonyi (2026))

$$\phi_{\text{relocation_rate}} = \frac{1}{1 + \mathrm{e}^{-x}}, \tag{28}$$

where

$$\begin{aligned} x = \beta_1 &+ \beta_2\, x_{\text{gender}} + \beta_3\, x_{\text{age}} + \beta_4\, x_{\text{age}}^2 \\ &+ \beta_5\, x_{\text{language}} + \beta_6\, \log(x_{\text{income}}) + \beta_7\, x_{\text{renter}} \\ &+ \beta_8\, x_{\text{housing_size}} + \beta_9\, x_{\text{mean_PM2.5}} + \beta_{10}\, x_{\text{pop_density}}, \end{aligned} \tag{29}$$

and β_1, ..., β_{10} are demographic and area related values gender, age, language, income, renter, housing size and population density. Mean PM2.5 concentration is calculated using Equation 17.