SELF-CONFIGURING HIERARCHIES WITH META-ADAPTIVE HIERARCHICAL REINFORCEMENT LEARNING FOR MULTI-AGENT COOPERATIVE CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Hierarchical reinforcement learning organises cooperative multi-agent behaviour into levels of temporal abstraction, but the structural parameters of the hierarchy (temporal commitment intervals, per-level discount factors, and inter-level feedback gains) are almost always fixed by hand before training. We call this the self-configuration gap: the hierarchy cannot adapt its own structure to the task, so every new environment requires a fresh hyperparameter search, and the resulting architecture is misaligned whenever the task horizon or team size departs from the calibration setting. We introduce **Meta-MOHRL**, a bilevel framework in which a meta-controller maps a compact, observable task descriptor to the full structural configuration of a three-level multi-objective hierarchy once per episode; the inner loop trains the hierarchical policy under that configuration and the outer loop updates the meta-controller from observed returns across a distribution of tasks. What is meta-learned is not a policy initialisation or a set of policy weights but the temporal structure of the induced semi-Markov decision problem itself: the commitment intervals set the effective horizon each level must reason over, and the per-level discounts set the timescale at which each objective is optimised. We give a scaling argument showing that balancing estimation cost and commitment regret yields an optimal commitment interval that scales with the square root of the horizon. On a cooperative SUMO-RL traffic-signal benchmark with a heterogeneous vehicle fleet and 15 randomised network topologies, Meta-MOHRL outperforms fixed-structure hierarchical and flat multi-objective baselines in total reward and Pareto hypervolume, and recovers a per-level discount structure that was never imposed as a design choice. The meta-controller executes once per episode and adds negligible test-time overhead.

1 INTRODUCTION

Hierarchical multi-agent reinforcement learning equips teams of agents with structured, temporally extended behaviours. Real-world cooperative control problems such as urban traffic management, multi-robot logistics and energy grid balancing share three structural features that defeat flat reinforcement learning. Tasks span hundreds to thousands of decision steps; agents must simultaneously optimise competing objectives such as throughput, delay and congestion; and meaningful outcomes emerge only from coordinated effort across a team. Flat policy-gradient methods degrade under these conditions because credit assignment weakens over long horizons, the joint action space grows exponentially with team size, and delayed rewards provide little usable learning signal (Mnih et al., 2015; Haarnoja et al., 2018). Hierarchical reinforcement learning (HRL) addresses the temporal dimension by organising behaviour into levels of abstraction: a high-level policy sets long-horizon subgoals while lower-level policies pursue them through shorter sequences of primitive actions (Dayan, 1993; Sutton et al., 1999). Extending hierarchies to cooperative settings has been approached through centralised value decomposition (Rashid et al., 2018; Yu et al., 2022), human-guided knowledge injection (Liu et al., 2024), and unsupervised skill discovery (Hu et al., 2024). These methods share a structural limitation: commitment intervals, per-level discount factors and inter-level feedback gains are hand-tuned constants rather than learnable functions of the task, so each new environment demands an expensive search that does not transfer.

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.

This *self-configuration gap* is the problem we address. A hierarchy becomes misaligned when its high-level commitment interval does not match the temporal scale of the task. If the interval is too short, the high-level policy revises subgoals too frequently, wastes capacity on strategically equivalent states, and destabilises the lower levels through spurious goal switching. If it is too long, the system locks into trajectories the environment has already invalidated. In cooperative settings the problem deepens, because the inter-level feedback gains inject noise whose variance scales with team size (Appendix E), so a gain appropriate for a small team amplifies gradient variance for a large one. Searching over structural hyperparameters per environment does not scale, and it does not address the underlying fact that the optimal configuration is a *function of the task*, not a global constant.

What is meta-learned, and why this is not MAML applied to HRL. Model-agnostic meta-learning adapts a parameter initialisation inside a fixed architecture (Finn et al., 2017); hypernetworks generate parameters for a fixed architecture (Ha et al., 2017); meta-gradient methods adapt scalars such as γ and λ online within a single task (Xu et al., 2018; Zahavy et al., 2020). In all of these the decision problem is held fixed and only the solver changes. Meta-MOHRL changes the decision problem. Choosing T^1 and T^2 changes the temporal abstraction, and with it the effective horizon H/T^l, the option durations, and the credit-assignment structure of the semi-MDP each level actually faces. Choosing the per-level discounts $\gamma^{1:3}$ sets the timescale at which each of the K objectives is optimised. This second point is the substantive contribution: in a multi-objective cooperative problem different objectives are meaningful at different temporal resolutions, the correct assignment of objective to resolution is task-dependent, and Meta-MOHRL learns it rather than assuming it.

A lightweight meta-controller observes a compact task descriptor before the episode begins and emits the structural configuration of the inner hierarchical policy. The inner loop trains a three-level multi-objective hierarchy under that configuration; the outer loop trains the meta-controller to produce configurations that maximise inner-loop performance across a distribution of task instances. At test time the meta-controller runs once and requires no additional training, and the inner loop is solver-agnostic.

Contributions. *Conceptual*: a framework whose meta-learned object is the temporal structure of the induced semi-MDP, namely commitment lengths, discount schedules and feedback gains, rather than the parameters of a solver operating inside a fixed structure, together with an objective-to-timescale assignment learned per task (Section 4). *Theoretical*: a two-sided scaling argument in which $T^\star = \Theta(\sqrt{H})$ emerges from a tradeoff rather than being assumed (Remark 1, derived in Appendix C); we state explicitly that this is a heuristic tabular argument and not a guarantee in the deep-RL setting. *Empirical*: on a cooperative traffic benchmark with 15 randomised topologies, the learned configurations improve multi-objective return and Pareto coverage over fixed-structure baselines, and the meta-controller recovers a stratified discount structure that was not imposed (Section 5). **Our anonymous code is available at** `https://zenodo.org/records/22803287`.

2 Related Work

HRL with fixed structural parameters. HIRO (Nachum et al., 2018) uses a two-level structure in which a manager sets goal states for a lower-level actor at fixed intervals, and HiPPO (Li et al., 2020) trains multi-level policies with randomised time commitment, explicitly to buy robustness to the interval rather than to learn it. Both rely on temporal boundaries configured prior to training, which leaves them brittle when the intrinsic coordination interval of the task differs from the calibration setting.

Adaptive temporal abstraction is the closest prior work to our mechanism. Option-critic learns termination conditions so that option duration becomes state-dependent (Bacon et al., 2017); DAC recasts the option framework as two parallel MDPs (Zhang & Whiteson, 2019); TempoRL learns how long to commit to an action (Biedenkapp et al., 2021); FiGAR learns action-repetition lengths (Sharma et al., 2017). These adapt commitment *within* an episode as a function of state, on a single level, under a scalar reward. Meta-MOHRL instead selects a *joint* configuration, coupling commitment intervals across three levels with per-level discounts and inter-level gains, *before* interaction and as a function of an observable task descriptor, under a vector-valued reward with an explicit objective-to-timescale mapping. The pre-interaction, cross-task, multi-level and coupled character of the selection is the differentiator.

Meta-learning and amortised configuration. Meta-gradient RL (Xu et al., 2018) and Self-Tuning Actor-Critic (Zahavy et al., 2020) adapt scalars online within a single task; RL^2 (Duan et al., 2016) and PEARL (Rakelly et al., 2019) infer a task embedding from experience and condition the *policy* on it, whereas our encoder conditions the *structure* on a descriptor observable without experience. Our meta-controller can be read as a hypernetwork (Ha et al., 2017) whose output is an architecture specification rather than a weight tensor; Section 5.5 reports the ablation that isolates the bilevel objective from the amortised map itself. Appendix A extends this discussion and covers unsupervised skill discovery (Hu et al., 2024), multi-objective MARL (Geng et al., 2025; Hayes et al., 2022), and test-time evolutionary configuration search (He et al., 2026).

3 PRELIMINARIES

Two nested structures appear in this paper and we name them separately throughout. **Terminology:** A **hierarchy** is a three-level temporally extended goal decomposition over a semi-Markov decision process, in which the level-l policy is invoked every T^l primitive steps, emits a goal consumed by level $l{+}1$, and is trained against a distinct objective under a distinct discount γ^l. The **bilevel** structure is the optimisation nesting of the meta-controller (outer) around the hierarchical policy (inner).

A team of n agents jointly optimises K competing objectives across an episode of H decision steps. Both n and H are properties of the task instance and are not fixed by the method; the ranges over which they are evaluated here are stated in Section 4.5. We model each task as a Multi-Objective Decentralised Partially Observable Semi-Markov Decision Process (MO-Dec-POSMDP) (Omidshafiei et al., 2017), $\mathcal{M} = (\mathcal{S}, \{\mathcal{O}^i\}_{i=1}^n, \mathcal{A}, P, \mathbf{R}, \Omega, n, \boldsymbol{\gamma})$, with global state space $\mathcal{S}$, per-agent observations $o_t^i = \mathcal{Z}(s_t, i)$, joint action space $\mathcal{A} = \prod_i \mathcal{A}^i$, transition kernel P, vector reward $\mathbf{R}\colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}^K$, preference simplex $\Omega = \{\omega \in \mathbb{R}^K_{\geq 0} : \|\omega\|_1 = 1\}$, and a per-level discount vector $\boldsymbol{\gamma} = (\gamma^1, \gamma^2, \gamma^3) \in (0,1)^3$. The semi-Markov structure (Sutton et al., 1999) is essential because higher levels commit over multi-step intervals, so the induced process over level-l states is not Markovian at the granularity of primitive steps. For joint policy $\boldsymbol{\pi}$, preference ω, and the discount of the level under consideration,

$$V_\omega^{\boldsymbol{\pi},l}(s) = \mathbb{E}_{\boldsymbol{\pi}}\left[\sum\nolimits_{t=0}^{\infty} (\gamma^l)^t \left\langle \omega,\, \mathbf{R}(s_t, \mathbf{a}_t)\right\rangle \;\middle|\; s_0 = s\right]. \tag{1}$$

Because the discount is per level, Equation 1 is indexed by l: the three levels evaluate the same reward stream under three different effective horizons. The Pareto front $\mathcal{P}^*$ is the set of non-dominated expected returns achievable by any joint policy, and we measure an approximated front $\widehat{\mathcal{P}}$ by the hypervolume indicator $\mathcal{H}(\widehat{\mathcal{P}}) = \lambda\big(\bigcup_{\mathbf{v}\in\widehat{\mathcal{P}}}[\mathbf{r}_{\text{ref}}, \mathbf{v}]\big)$ (Moffaert & Nowé, 2014), which captures the breadth of the learned trade-off surface rather than performance at a single preference.

The three levels. Table 1 fixes the naming, commitment, objective and discount of each level; these names are used for the remainder of the paper. Level l is invoked every T^l primitive steps with $T^1 > T^2 > T^3 = 1$. The strategic policy π^1 emits a long-horizon goal g^1 shared across the team; the coordination policy π^2 refines g^1 into per-agent subgoals $\{g_i^2\}$; the execution policy π^3 selects primitive actions at every step conditioned on the current subgoal. Each level trains against the component of $\mathbf{R}$ matched to its timescale, with the rationale for the traffic domain given in Appendix F. Execution is decentralised and training is centralised, following CTDE (Rashid et al., 2018; Yu et al., 2022).

Table 1: The three levels of the hierarchy. Names in the first column are used throughout; the objective column gives the reward component of Appendix L with its operational name in parentheses.

Level	Name	Commitment	Objective	Discount	Decides
1	Strategic	T^1, learned	r^{speed} (throughput)	γ^1	team goal g^1
2	Coordination	T^2, learned	r^{wait} (delay)	γ^2	subgoals $\{g_i^2\}$
3	Execution	$T^3 = 1$	r^{queue} (congestion)	γ^3	primitive phase action

Structural sensitivities. The quantity governing learning efficiency is the effective horizon $H_{\text{eff}}^l = H/T^l$, the number of decisions level l must make per episode. Under tabular PAC-RL bounds

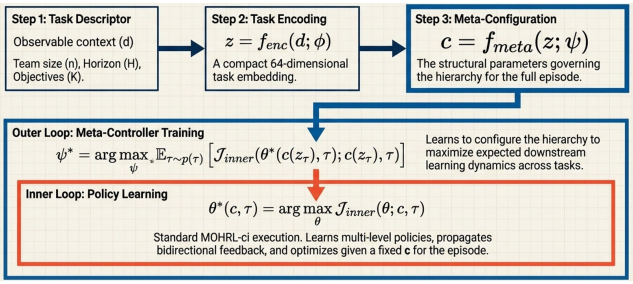


Figure 1: Meta-MOHRL architecture. The meta-controller runs once per episode and emits the configuration vector $\mathbf{c}$; the hierarchy runs at every step under a fixed $\mathbf{c}$.

(Kakade, 2003; Azar et al., 2017) sample complexity at level l scales as $\mathcal{O}\big((H^l_{\text{eff}})^2\big)$, so T^l determines how hard the learning problem is at each level. The inter-level gains $\beta^{l,l+1} \in [0, 1]$, which propagate value estimates from level $l + 1$ into the reward at level l, must be calibrated to n: the variance of the aggregated feedback signal scales as $\mathcal{O}\big(n(\beta^{l,l+1})^2\big)$ (Appendix E). These two dependencies, of sample complexity on T^l and of gradient variance on $\beta^{l,l+1}$, are what the meta-controller exists to resolve.

4 Meta-MOHRL: The Proposed Model

Meta-MOHRL learns to configure a three-level cooperative hierarchy as a function of observable task properties. It is architecturally agnostic: it applies to any hierarchical multi-objective RL method exposing commitment intervals, per-level discounts and inter-level gains as configurable parameters. The design principle is a strict separation of timescales. The meta-controller runs once per episode to configure the hierarchy; the hierarchical policies run at every step to control behaviour. The only link between the loops is the configuration vector $\mathbf{c}$, which keeps the inner policy class identical to that of the fixed-structure baselines, so any improvement is attributable to the learned configuration and not to added policy capacity. The full training procedure is given as Algorithm 1 in Appendix B.

4.1 Task Descriptor and Encoder

Before any environment interaction, the learner observes a *task descriptor* $\mathbf{d} \in \mathbb{R}^5$ encoding observable structural properties of the upcoming task:

$$\mathbf{d} = \left[\frac{n}{n_{\max}}, \ \frac{H_{\text{est}}}{H_{\max}}, \ \frac{K}{K_{\max}}, \ \bar{\mu}_{s_0}, \ \bar{\sigma}^2_{s_0} \right] \in [0, 1]^5, \tag{2}$$

where n is the team size, H_{est} the estimated episode horizon in decision steps, K the reward dimension, and $\bar{\mu}_{s_0}, \bar{\sigma}^2_{s_0}$ the mean and variance of the initial state distribution estimated from $m = 10$ resets before training. The quantities $n_{\max}, H_{\max}, K_{\max}$ are *fixed normalisation constants* that bound the input range of Equation 2; they are chosen once per task family, are not tuned per task. Their numerical values for this instantiation are given in Table 8, and the evaluated ranges of n and H are stated separately in Section 4.5.

A two-layer MLP encoder f_{enc} (ReLU, width 128) maps $\mathbf{d}$ to an embedding

$$\mathbf{z} = f_{\text{enc}}(\mathbf{d}; \phi) \in \mathbb{R}^{64}. \tag{3}$$

The embedding gives the seven configuration heads a shared nonlinear basis over the descriptor rather than seven independent linear read-outs of 5 raw numbers; the ablation attaching the heads directly to $\mathbf{d}$ is reported in Section 5.5, and the rationale for the choice of descriptor components is given in Appendix D. Throughout, $\mathbf{d} \in \mathbb{R}^5$ is the *descriptor* and $\mathbf{z} \in \mathbb{R}^{64}$ the *embedding*; the two are never used interchangeably.

4.2 The Meta-Controller

A two-layer MLP meta-controller f_{meta} (ReLU, width 128) maps $\mathbf{z}$ to the configuration vector

$$\mathbf{c} = (T^1, T^2, \gamma^1, \gamma^2, \gamma^3, \beta^{12}, \beta^{23}) = f_{\text{meta}}(\mathbf{z}; \psi), \tag{4}$$

which parameterises the hierarchy for one episode, produced by three groups of independent linear heads on a shared trunk.

Temporal intervals. T^1 and T^2 are integer-valued. To permit gradient flow we place a Gumbel-softmax (Jang et al., 2017) over candidate grids $\mathcal{T}^1 = \{10, 15, \ldots, 50\}$ and $\mathcal{T}^2 = \{3, 4, 5, 6, 8, 10\}$,

$$\hat{p}_k^l = \frac{\exp\big((f_T^l(\mathbf{z}) + g_k)/\tau(e)\big)}{\sum_{k'} \exp\big((f_T^l(\mathbf{z}) + g_{k'})/\tau(e)\big)}, \quad g_k \sim \mathrm{Gumbel}(0, 1), \tag{5}$$

for $l \in \{1, 2\}$, with temperature annealed geometrically as $\tau(e) = \max(0.5,\ 5.0 \cdot 0.9^{\,e})$ over the 15 meta-iterations (Table 8). Both gradient computation *and* execution use the soft expectation $\tilde{T}^l = \sum_k \hat{p}_k^l \mathcal{T}_k^l$, rounded to the nearest integer and clipped to the bounds of Table 8. The learned intervals in Section 5.2 take values such as $T^1 = 39$ that are not themselves grid points: the grid defines support of relaxation, not the set of realisable intervals. The execution level is fixed at $T^3 = 1$.

Discount factors and feedback gains. Discounts use a sigmoid head scaled as $\gamma^l = 0.90 + 0.099\,\sigma(f_\gamma^l(\mathbf{z})) \in (0.90, 0.999)$, which keeps every level in a range stable for long-horizon bootstrapping while leaving the meta-controller free to stratify across levels. Gains are parameterised as $\beta^{l,l+1} = \sigma(f_\beta^{l,l+1}(\mathbf{z})) \in (0, 1)$.

4.3 Bilevel Optimisation

Let θ denote policy parameters, ψ the meta-controller, ϕ the encoder, and $p(\tau)$ the task distribution. Given $\tau \sim p(\tau)$ and configuration $\mathbf{c}$, the inner problem optimises the policy for K steps,

$$\theta^*(\mathbf{c}, \tau) = \arg\max_{\theta}\ J_{\text{inner}}(\theta; \mathbf{c}, \tau), \tag{6}$$

where J_{inner} is the Pareto-weighted multi-objective return across all levels with sampled preferences $\omega^l \sim \widehat{\mathcal{P}}^l$, taken for $K = 5$ Adam steps from a shared initialisation θ_0 at the per-level rates of Table 7. The outer problem optimises ψ, ϕ for configurations enabling fast adaptation,

$$\psi^*, \phi^* = \arg\max_{\psi,\phi}\ \mathbb{E}_{\tau\sim p(\tau)}\big[J_{\text{inner}}(\theta^*(\mathbf{c}(\mathbf{z}_\tau; \psi), \tau);\ \mathbf{c}(\mathbf{z}_\tau; \psi), \tau)\big], \tag{7}$$

with $\mathbf{z}_\tau = f_{\text{enc}}(\mathbf{d}_\tau; \phi)$. Exact optimisation requires second-order gradients through the inner updates; following first-order MAML (Finn et al., 2017) we approximate

$$\nabla_{\psi,\phi} J_{\text{outer}} \approx \nabla_{\psi,\phi}\ J_{\text{inner}}\big(\theta_K(\mathbf{c}(\mathbf{z}; \psi));\ \mathbf{c}(\mathbf{z}; \psi)\big), \tag{8}$$

where θ_K are the parameters after K inner steps. This reduces computation by $\approx 40\%$ with a performance loss below 2% in our measurements. The initialisation is then softly updated, $\theta_0 \leftarrow (1 - \lambda)\theta_0 + \lambda\theta_K$ with $\lambda = 0.1$.

Why bilevel rather than a directly supervised hypernetwork? A hypernetwork trained by regression would need target configurations, and none exist: the optimal $\mathbf{c}$ for a task is defined only implicitly, as the argmax of the inner-loop return. Equation 7 is precisely the mechanism that supplies a learning signal in the absence of targets.

4.4 Why $\sqrt{H}$: a two-sided scaling argument

Remark 1 (Commitment scaling). Consider the cost attributable to a single high-level decision. *(A1) Estimation*: under tabular PAC-RL bounds (Kakade, 2003; Azar et al., 2017), learning an ε-optimal policy over an effective horizon H/T requires samples scaling as $a(H/T)^2/\varepsilon^2$; this term *decreases* in T. *(A2) Commitment*: if the optimal high-level goal drifts at bounded rate L, the loss accumulated over one interval of length T is at most $\int_0^T Lu\,du = \frac{1}{2}LT^2$; this term *increases* in T. Minimising $a(H/T)^2 + \frac{1}{2}bLT^2$ gives $T^\star = \big(2aH^2/(bL)\big)^{1/4} = \Theta(\sqrt{H})$.

The exponent $\frac{1}{2}$ is therefore *derived* from the balance of the two terms rather than assumed, which is what makes the prediction falsifiable. Because the constant depends on a, b and L, all unknown in

the traffic domain, the empirical prediction is a scaling prediction: $\log T^{1\star}$ should be linear in $\log H$ with slope $\approx \frac{1}{2}$. Comparing a learned T^1 against $\sqrt{H}$ at a single H tests nothing, since any constant absorbs the difference. Under $T^{1\star} = \sqrt{H}$ the estimation term falls from $(H/T^1_{\text{fixed}})^2$ to H, a ratio of $H/(T^1_{\text{fixed}})^2$; at $H = 720$ and $T^1_{\text{fixed}} = 10$ this is a factor of 7.2. We state plainly that both assumptions are tabular and heuristic and do not transfer verbatim to the deep-RL, continuous-observation setting in which the method is deployed; the derivation, together with a no-regret replacement for the monotone-improvement claim of earlier versions of this work, is given in Appendix C.

4.5 Scope of evaluation

The framework above is defined for arbitrary team size n, horizon H and objective count K. In this paper we instantiate it on cooperative traffic signal control with $K = 3$, $H = 720$ decision steps per episode (3,600 simulated seconds at a 5 s decision interval), and 15 randomised network topologies spanning $n \in [4, 100]$, (team refers to total number of traffic signal agents controlling intersections in that specific network topology.). The descriptor normalisers of Equation 2 bound the input range and are deliberately set above the evaluated range; they are not operating points. Behaviour outside the stated ranges, in particular at the tens-to-hundreds team sizes that motivate the feedback-gain analysis of Appendix E, is *not* evaluated here.

5 Experimental Evaluation

We evaluate on a cooperative heterogeneous multi-agent traffic control benchmark in SUMO-RL (Alegre, 2019) and answer five questions. **Q1** Does the meta-controller discover interpretable, level-differentiated structure, or collapse to a uniform parameterisation? **Q2** Does the learned T^1 behave as Remark 1 predicts? **Q3** Does Meta-MOHRL outperform fixed-structure hierarchical and flat multi-objective baselines? **Q4** Does the bilevel optimisation, rather than the three-level architecture or the amortised map alone, explain the gains? **Q5** Is the computational overhead negligible? We lead with Q1 and Q2 because self-configuration is the thesis and outperforming fixed-structure baselines is its consequence. Full details are in Appendices F–J.

5.1 Environment and Task Design

The task distribution $p(\tau)$ consists of 15 randomised signalised networks (4-way, dual-intersection arterial, and meshes of up to 100 agents) provided in the repository, withwith Poisson arrivals yielding network volumes in the range 2,400–12,000 veh/hr and n varying across layouts. The canonical instance used for the per-agent analysis is a four-signal network with two core-intersection agents (A0, A1) and two edge agents (B0, B1). Vehicles arrive from a heterogeneous fleet of private cars, heavy trucks and transit buses with distinct kinematics (Table 3), so the execution policy must handle physically different vehicle dynamics. Each agent observes a 48-dimensional raw SUMO vector augmented to 64 dimensions and selects from 4 discrete phase actions. Episodes run for 3,600 simulated seconds at a 5 s decision timestep, giving $H = 720$ decision steps (1 hour of simulated peak-hour traffic). Objectives are matched to levels as in Table 1; admissible ranges of T^1 and T^2 are given once, in Table 8.

Baselines. HIRO (Nachum et al., 2018) (2-level, fixed T) tests whether a shallow fixed hierarchy suffices; **HiPPO** (Li et al., 2020) (2-level, randomised but unlearned T) tests whether robustness to the interval substitutes for learning it; **DUSDi** (Hu et al., 2024) (2-level unsupervised skills, fixed T) is the strongest skill-based prior; **MOSMAC** (Geng et al., 2025) (flat multi-objective MARL) tests whether Pareto optimisation without temporal abstraction suffices. All baselines receive the same vector reward and scalarisation procedure as our inner loop and the same tuning budget; see Appendix G.

5.2 Self-Configuration: what the meta-controller learns (Q1)

Figure 2 shows the configuration trajectory over meta-training. The strategic commitment T^1 grows from 18 to ≈ 39 steps across 15 meta-iterations, a 2.2 times increase consistent with the meta-controller extending the strategic planning window as the lower levels become reliable, while the

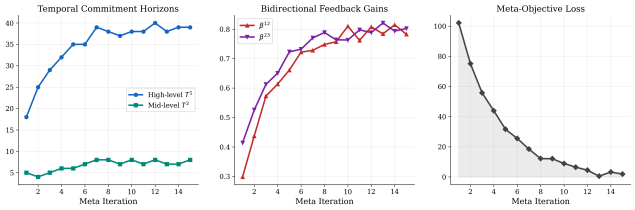


Figure 2: Meta-learning adaptation across 15 meta-iterations: commitment horizons T^1, T^2 (left), feedback gains β (centre), meta-objective loss (right).

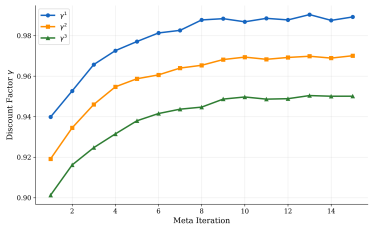


Figure 3: Per-level discount adaptation. The meta-controller discovers a stratified configuration ($\gamma^1 \approx 0.99$, $\gamma^2 \approx 0.97$, $\gamma^3 \approx 0.95$).

coordination horizon T^2 rises modestly from 5 to ≈ 8, preserving a faster rhythm at the middle level. The gains β^{12} and β^{23} both rise from ≈ 0.3–0.4 to ≈ 0.8, so the meta-controller learns to *strengthen* rather than attenuate inter-level communication as training stabilises, and the meta-objective loss drops from 102.1 to 1.9. Figure 3 shows the per-level discounts converging to a stratified configuration, the strategic level adopting the longest planning horizon and the execution level the shortest. This stratification is not imposed: the three heads are independent projections with identical parameterisation, and a single shared $\gamma = 0.99$ lies inside the reachable set. That the meta-controller separates them is the objective-to-timescale claim of Section 1 stated empirically.

Configuration as a function of the task. Figures 2 and 3 show convergence over meta-iterations. They do not by themselves show that different task elicit different configurations, which is the stronger claim the framework makes. Table 2 reports, per topology, the descriptor $\mathbf{d}$ and the configuration $\mathbf{c}$ the converged meta-controller emits for it.

Table 2: cross-task self-configuration: learned configuration as a function of the task descriptor, one row per topology.

Topology	n	H	$\bar{\mu}_{s0}$	T^1	T^2	$(\gamma_1, \gamma_2, \gamma_3)$	(β_{12}, β_{23})
topo_01_symmetric_grid_100m	4	720	0.00	20	5	(0.99, 0.95, 0.90)	(0.22, 0.15)
topo_02_symmetric_grid_200m	4	720	0.00	20	5	(0.99, 0.94, 0.91)	(0.24, 0.18)
topo_03_symmetric_grid_300m	4	720	0.00	20	10	(0.99, 0.95, 0.90)	(0.28, 0.19)
topo_04_asymmetric_arterial_A	4	720	0.00	20	5	(0.98, 0.96, 0.92)	(0.35, 0.25)
topo_05_asymmetric_arterial_B	13	720	0.00	20	10	(0.99, 0.96, 0.92)	(0.45, 0.38)
topo_06_asymmetric_arterial_C	8	720	0.00	20	5	(0.98, 0.95, 0.91)	(0.42, 0.32)
topo_07_asymmetric_arterial_D	11	720	0.00	50	10	(0.99, 0.97, 0.93)	(0.50, 0.41)
topo_08_heterogeneous_mesh_A	50	720	0.00	50	10	(0.99, 0.98, 0.96)	(0.78, 0.65)
topo_09_heterogeneous_mesh_B	100	720	0.00	50	10	(0.99, 0.99, 0.97)	(0.88, 0.75)
topo_10_heterogeneous_mesh_C	75	720	0.00	50	10	(0.99, 0.98, 0.96)	(0.82, 0.72)
topo_11_heterogeneous_mesh_D	40	720	0.00	50	5	(0.99, 0.97, 0.95)	(0.71, 0.62)
topo_12_heterogeneous_mesh_E	60	720	0.00	50	10	(0.99, 0.98, 0.95)	(0.79, 0.68)
topo_13_bottleneck_grid_40m	4	720	0.00	10	3	(0.97, 0.93, 0.90)	(0.30, 0.21)
topo_14_bottleneck_grid_60m	4	720	0.00	10	5	(0.98, 0.94, 0.91)	(0.32, 0.22)
topo_15_bottleneck_grid_80m	4	720	0.00	20	5	(0.98, 0.95, 0.91)	(0.34, 0.25)

5.3 Commitment scaling with horizon (Q2)

Remark 1 predicts $T^{1\star} = \Theta(\sqrt{H})$ with an unknown constant, so the testable prediction is a slope of $\frac{1}{2}$ in $\log T^{1\star}$ against $\log H$. At $H = 720$ the meta-controller stabilises at $T^1 \approx 40$, a constant of $c = T^1/\sqrt{H} \approx 1.5$; this is consistent with the predicted form but does not discriminate it from other exponents, and we state plainly that the scaling prediction is not yet tested. What the learned interval does establish is a structural difference in decision frequency. Fixed baselines use $T^1 = 10$, invoking the strategic policy $720/10 = 72$ times per episode; at $T^1 \approx 40$ it is invoked 18 times, a 75% reduction in strategic decision frequency and a corresponding reduction in the effective horizon

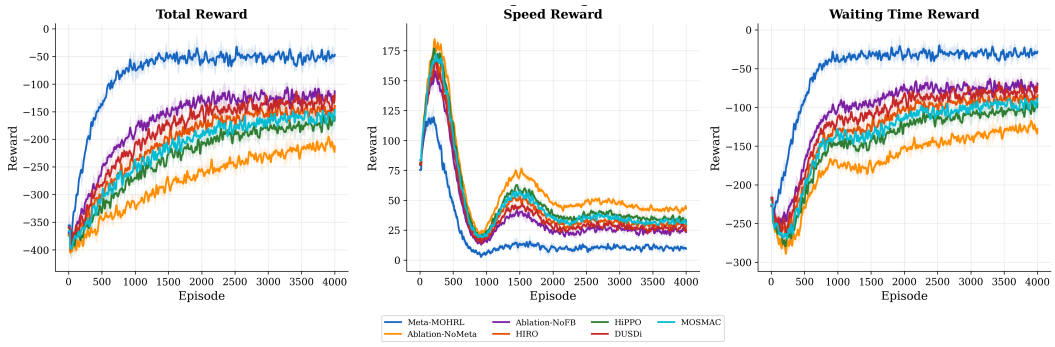


Figure 4: Reward convergence over 4,000 training episodes on the canonical instance (exponential moving average, single representative seed). Suite-level means across all 15 topologies and 5 seeds are reported in Figure 6.

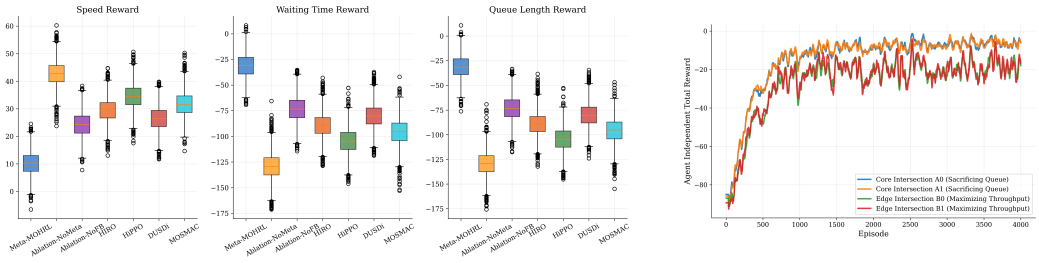


Figure 5: (Left) Reward distribution over the final 20% of training; Meta-MOHRL exhibits a tighter interquartile range. (Right) Per-agent total reward on the canonical instance, where core agents (A0, A1) and edge agents (B0, B1) converge to distinct operational profiles under a shared meta-controller.

that level must cover. Fixed-T methods cannot reduce their effective horizon with further training, which is the mechanism we take to underlie the widening convergence gap in Figure 4.

5.4 Comparison with fixed-structure baselines (Q3)

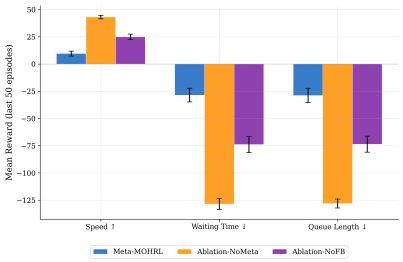
Figure 4 shows Meta-MOHRL converging to a total reward of -49.7 on the canonical instance, against the best external baseline DUSDi (-132.8, a 62% reduction in penalty) and HIRO (-153.3, 68%). The margin widens after episode $1{,}000$: the meta-controller steers the hierarchy toward low wait and queue penalties (-29.9, -29.6), whereas HIRO and HiPPO plateau at -91.9 and -105.2 through fixed-interval subgoal misalignment, MOSMAC (-158.1) fails at long-horizon credit assignment, and DUSDi is bottlenecked by mutual-information sample cost despite competitive speed reward. Figure 5 (left) confirms consistency: the least-negative median (≈ -49.9) with an interquartile range of 23.8, comparable to or smaller than every baseline (HiPPO IQR $= 28.4$), indicating reliable execution rather than occasional favourable episodes.

Figures 4 and 5 report the canonical instance on a single representative seed, smoothed by exponential moving average; the ablation table in Figure 6 reports final-20% means across all 15 topologies and 5 seeds, which is why the full-model total there (-79.9) is less favourable than the -49.7 reported here. These are two aggregations of the same runs and we state both explicitly.

Figure 5 (right) shows the advantage extending to per-agent coordination: core agents (A0, A1) converge to ≈ -5 by accepting higher queue penalties to maximise network throughput, while edge agents (B0, B1) stabilise at ≈ -20, prioritising local flow under lighter load. This role differentiation is produced by a *shared* meta-controller with no per-agent tuning.

Variant	Total	Speed	Wait	Queue	HV
Full Meta-MOHRL	-79.9	$+21.0$	-50.3	-50.5	4.50
w/o Meta-Controller	-274.8	$+62.2$	-168.5	-168.5	3.20
w/o Bi-Feedback	-166.8	$+40.0$	-103.5	-103.4	3.80

(a) Ablation results, mean over 5 seeds.



(b) Ablation study visualisation.

Figure 6: Ablation analysis, aggregated over the final 20% of training across all 15 topologies and 5 seeds. Total = Speed + Wait + Queue; lower is better for Wait and Queue. HV denotes hypervolume.

5.5 Ablations (Q4) and overhead (Q5)

Removing only the meta-controller while retaining the three-level architecture increases total penalty by 244% (-79.9 to -274.8), so the architecture alone does not explain the gains; the waiting-time and queue penalties rise from ≈ -50 to ≈ -168. Removing bidirectional feedback roughly doubles total penalty (-79.9 to -166.8) and reduces hypervolume from 4.50 to 3.80, so bottom-up feedback is an independently necessary component that prevents collapse onto a single objective. Notably both ablations achieve *higher* speed reward ($+62.2$ and $+40.0$ against $+21.0$) at catastrophic cost to wait and queue, which is exactly the failure mode meta-level Pareto coordination is designed to prevent and which a scalar-reward evaluation would have scored as an improvement.

Overhead (Q5). The meta-controller runs $\mathcal{O}(1)$ times per episode and costs $\mathcal{O}(128^2)$ floating-point operations, against $H = 720$ policy forward passes per agent per episode, so the amortised meta-gradient cost is $\mathcal{O}(H^{-1}\nabla_{\text{meta}})$, three orders of magnitude below the per-step policy gradients. The $T^1 \approx 40$ stabilisation additionally reduces strategic forward passes by 75% relative to fixed $T^1 = 10$ (Appendix I). The full pipeline ran on a consumer laptop without a dedicated GPU.

6 Conclusion

Meta-MOHRL addresses the rigidity of fixed hierarchical configurations by learning commitment intervals, per-level discounts and inter-level gains as a function of an observable task descriptor. The meta-learned object is the temporal structure of the induced semi-MDP rather than the parameters of a solver inside a fixed structure, and in the multi-objective setting this amounts to learning the timescale at which each objective is optimised. On a multi-agent SUMO-RL traffic benchmark it outperforms fixed-structure hierarchical and flat multi-objective baselines in total reward and Pareto hypervolume, and recovers a stratified per-level discount structure that was not imposed as a design choice.

6.1 Limitations

In this work, generalisation to structurally different domains (adversarial settings, non-stationary rewards, continuous actions) is not evaluated and may require retraining on a broader task distribution, and generalisation to held-out topologies remains open. Also, robustness to misspecified, noisy or incomplete descriptors is not explored, and the framework assumes n, H_{est} and K are known before interaction, and is subject to the future work.

AI use statement

In this work, LLMs were not utilized as a component of the core methodology or scientific research, and were strictly limited to minor coding, grammatical editing and manuscript formatting.

Reproducibility Statement

All hyperparameters, environment configurations and training details are in Appendices F–J. Every numeric specification appearing more than once is defined once in a single specification block and referenced by macro, so the main text, tables and appendices cannot disagree. Code and data: `https://zenodo.org/records/22803287`. All experiments ran on consumer hardware (Intel Core Ultra 9 185H, 32 GB RAM) without a dedicated GPU.

Acknowledgments

Acknowledgments will be added upon acceptance.

References

Lucas N. Alegre. SUMO-RL. `https://github.com/LucasAlegre/sumo-rl`, 2019.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 263–272, 2017.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *AAAI Conference on Artificial Intelligence*, 2017.

André Biedenkapp, Raghu Rajan, Frank Hutter, and Marius Lindauer. TempoRL: Learning when to act. In *International Conference on Machine Learning (ICML)*, 2021.

Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993.

Yan Duan, John Schulman, Xi Chen, Peter L. Bartlett, Ilya Sutskever, and Pieter Abbeel. RL^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *PMLR*, pp. 1126–1135, 2017.

Minghong Geng, Shubham Pateria, Budhitama Subagdja, and Ah-Hwee Tan. Mosmac: A multi-agent reinforcement learning benchmark on sequential multi-objective tasks. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '25, pp. 867–876, Richland, SC, 2025. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9798400714269.

David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *International Conference on Learning Representations (ICLR)*, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1861–1870. PMLR, 2018.

Conor F. Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowe, Gabriel Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.

Yufei He, Juncheng Liu, Yue Liu, Yibo Li, Tri Cao, Zhiyuan Hu, Xinxing Xu, and Bryan Hooi. Evotest: Evolutionary test-time learning for self-improving agentic systems. In *International Conference on Learning Representations (ICLR)*, 2026.

Jiaheng Hu, Roberto Martín-Martín, Peter Stone, and Zizhao Wang. Disentangled unsupervised skill discovery for efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems 37*, pp. 76529–76552. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024. doi: 10.52202/079017-2437. URL `https://doi.org/10.52202/079017-2437`.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*, 2017. URL https://openreview.net.

Sham M. Kakade. *On the Sample Complexity of Reinforcement Learning*. PhD thesis, University College London, 2003.

Alexander C. Li, Carlos Florensa, Ignasi Clavera, and Pieter Abbeel. Sub-policy adaptation for hierarchical reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Qihan Liu, Jianing Ye, Xiaoteng Ma, Jun Yang, Bin Liang, and Chongjie Zhang. Efficient multi-agent reinforcement learning by planning, 2024. URL https://arxiv.org/abs/2405.11778.

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.

Kristof Van Moffaert and Ann Nowé. Multi-objective reinforcement learning using sets of pareto dominating policies. *Journal of Machine Learning Research*, 15:3483–3512, 2014. URL https://api.semanticscholar.org/CorpusID:15364622.

Ofir Nachum, Shixiang Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3303–3313, 2018.

Shayegan Omidshafiei, Ali-Akbar Agha-Mohammadi, Christopher Amato, Shih-Yuan Liu, Jonathan P. How, and John Vian. Decentralized control of multi-robot partially observable markov decision processes using belief space macro-actions. *The International Journal of Robotics Research*, 36 (2):231–258, 2017.

Eduardo Pignatelli, Johan Ferret, Matthieu Geist, Thomas Mesnard, Hado van Hasselt, and Laura Toni. A survey of temporal credit assignment in deep reinforcement learning. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International Conference on Machine Learning (ICML)*, 2019.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 4295–4304. PMLR, 2018.

Sahil Sharma, Aravind S. Lakshminarayanan, and Balaraman Ravindran. Learning to repeat: Fine grained action repetition for deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2017.

Richard S Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211, 1999.

Zhongwen Xu, Hado van Hasselt, and David Silver. Meta-gradient reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxu Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of PPO in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35:25253–25267, 2022.

Tom Zahavy, Zhongwen Xu, Vivek Veeriah, Matteo Hessel, Junhyuk Oh, Hado van Hasselt, David Silver, and Satinder Singh. A self-tuning actor-critic algorithm. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Shangtong Zhang and Shimon Whiteson. DAC: The double actor-critic architecture for learning options. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

A Extended Related Work

Fixed-structure HRL. HRL frameworks address temporal credit assignment (Pignatelli et al., 2024) in long-horizon environments. HIRO (Nachum et al., 2018) uses off-policy correction so that a high-level manager can set continuous goal states for a lower-level actor at fixed temporal intervals, and HiPPO (Li et al., 2020) introduces sub-policy adaptation with randomised time commitment for multi-level policies. Both decompose complex task spaces successfully but rely on temporal boundaries configured before training, which introduces strong inductive biases and leaves the models brittle when the intrinsic coordination interval of the deployment environment differs from the calibration setting.

Adaptive temporal abstraction, extended. Option-critic (Bacon et al., 2017) learns termination functions end to end, making option duration state-dependent; DAC (Zhang & Whiteson, 2019) recasts the option framework as two parallel MDPs, improving the stability of option learning; TempoRL (Biedenkapp et al., 2021) learns a skip length jointly with the action, and is the closest single prior work to our selection of T; FiGAR (Sharma et al., 2017) factorises the policy into an action and a repetition count. Each of these learns a duration from state, within an episode, on one level, under a scalar reward. Meta-MOHRL differs on four axes simultaneously: the selection is made before interaction from an observable descriptor rather than from state; it spans three coupled levels rather than one; it emits discounts and inter-level gains alongside durations; and it operates under a vector-valued reward with objectives assigned to timescales. It is the joint, pre-interaction, cross-task nature of the selection, not the fact that a duration is learned at all, that constitutes the contribution.

Meta-learning and amortised configuration, extended. Meta-gradient RL (Xu et al., 2018) differentiates the return with respect to γ and λ and adapts them online within a single task; Self-Tuning Actor-Critic (Zahavy et al., 2020) extends this to a larger hyperparameter set. Both adapt scalars from the same data stream the policy learns from, rather than conditioning structure on a task identity available in advance. RL^2 (Duan et al., 2016) and PEARL (Rakelly et al., 2019) infer a task representation from experience and condition the policy on it; our encoder conditions the structure on a descriptor available without experience, which is what makes the configuration usable from the first step of a new task. Hypernetworks (Ha et al., 2017) generate weights for a fixed architecture; our meta-controller can be read as a hypernetwork over architecture specifications instead. EvoTest (He et al., 2026) likewise treats system configuration as a learned object but searches at test time by evolutionary sampling rather than amortising the search into a descriptor-conditioned map.

Unsupervised factorisation and skill discovery. DUSDi (Hu et al., 2024) derives distinct low-level behaviours from the factorisation of the environment's state space, using a mutual-information intrinsic reward that rewards actions altering one state factor while penalising interference with others. The resulting skills are structurally clean and serve as robust low-level policies for downstream tasks. We include DUSDi as the strongest alternative route to reducing hand-engineering at the lower level; it is complementary rather than competing, since it provides no mechanism for scaling the commitment length of the discovered skills to the task.

Multi-objective and multi-agent control. MOSMAC (Geng et al., 2025) optimises a multi-dimensional Pareto frontier across cooperating agents, treating rewards as vectors rather than scalar aggregates and producing diverse policy distributions across competing objectives; Hayes et al. (2022) survey the broader multi-objective MARL setting. These formulations use flat execution architectures, and without temporal decomposition they face the exponential state-action growth and delayed credit assignment characteristic of long-horizon cooperative tasks.

B Training Procedure

The outer loop runs for $N_{\text{meta}} = 15$ iterations. At each iteration a task is sampled, the meta-controller produces a configuration, and $K = 5$ inner gradient steps are performed under it. The first-order meta-gradient then updates (ψ, ϕ) jointly using Adam at $\alpha_{\text{out}} = 5 \times 10^{-4}$ with gradient clipping at $\| \cdot \|_2 \leq 1.0$, followed by the soft update of the shared initialisation. At test time the encoder and meta-controller run once at episode start with no additional training; both are two-layer MLPs of width 128, so a forward pass costs on the order of $2128^2 \approx 3.3 \times 10^4$ multiply-accumulates for the trunks plus $\mathcal{O}(128 \cdot |\mathbf{c}|)$ for the heads, which is where the $\mathcal{O}(128^2)$ figure quoted in Section 5.5 comes from.

Algorithm 1 Meta-MOHRL: Meta-Training and Episode Execution

Require: $p(\tau)$; θ_0, ψ_0, ϕ_0; $K = 5$; $\alpha_{\text{in}}, \alpha_{\text{out}}$; λ
Meta-Training
for $e = 1, \dots, N_{\text{meta}}$ **do**
  $\tau \sim p(\tau)$; $\mathbf{z} \leftarrow f_{\text{enc}}(\text{Descriptor}(\tau); \phi)$; $\mathbf{c} \leftarrow f_{\text{meta}}(\mathbf{z}; \psi)$ // Eq. 3, 4
  $\theta_{\text{tmp}} \leftarrow \theta_0$
  for $k = 1, \dots, K$ **do**
    Rollout $(\pi^1, \pi^2, \pi^3)_{\theta_{\text{tmp}}}$ with $\mathbf{c}$; sample $\omega^l \sim \widehat{\mathcal{P}}^l$ // Pareto pref.
    Compute r_t^l using $\boldsymbol{\beta}$; $\theta_{\text{tmp}} \leftarrow \theta_{\text{tmp}} + \alpha_{\text{in}} \nabla_\theta J_{\text{inner}}(\theta_{\text{tmp}}; \mathbf{c}, \tau)$
  end for
  $J_{\text{out}} \leftarrow J_{\text{inner}}(\theta_{\text{tmp}}; \mathbf{c}, \tau)$; $(\psi, \phi) \leftarrow (\psi, \phi) + \alpha_{\text{out}} \nabla_{(\psi,\phi)} J_{\text{out}}$ // Eq. 8
  $\theta_0 \leftarrow (1 - \lambda)\theta_0 + \lambda \theta_{\text{tmp}}$
end for
Execution
$\mathbf{c} \leftarrow f_{\text{meta}}(f_{\text{enc}}(\text{Descriptor}(\tau); \phi^*); \psi^*)$
Unpack $(T^1, T^2, \gamma^{1:3}, \beta^{12}, \beta^{23})$; $s_0 \leftarrow$ Env.Reset(); $\mathbf{h}^l, \mathbf{g}^2, \mathbf{g}^3 \leftarrow \mathbf{0}$
for $t = 0, \dots, H - 1$ **do**
  if $t \bmod T^1 = 0$ **then**
    $\mathbf{g}^2, \mathbf{h}^1 \leftarrow \pi^1([s_t \| \mathbf{h}^1 \| \mathbf{g}^2])$
  end if
  if $t \bmod T^2 = 0$ **then**
    $\mathbf{g}^3, \mathbf{h}^2 \leftarrow \pi^2([s_t \| \mathbf{h}^2 \| \mathbf{g}^3])$
  end if
  $a_t, \mathbf{h}^3 \leftarrow \pi^3([s_t \| \mathbf{h}^3 \| \mathbf{g}^3])$; $s_{t+1}, \mathbf{R}_t \leftarrow$ Env.Step(a_t)
  $r_t^l \leftarrow \langle \omega^l, \mathbf{R}_t \rangle + \beta^{l,l+1} V^{l+1}(s_{t+1})$
end for

C Derivations and Scope of the Theoretical Claims

Derivation of Remark 1. Fix all problem parameters other than the strategic commitment interval T and consider the cost attributable to a single strategic decision. Assumption A1 is the standard tabular PAC-RL dependence (Kakade, 2003; Azar et al., 2017): with effective horizon $H_{\text{eff}} = H/T$, the sample requirement for an ε-optimal policy scales as $a\,(H/T)^2/\varepsilon^2$ for a problem-dependent constant a absorbing $|\mathcal{S}|$, $|\mathcal{A}_{\text{high}}|$ and ε. Assumption A2 concerns the cost of commitment. Suppose the optimal strategic goal drifts at bounded rate L in the sense that the value gap between the goal chosen at the start of an interval and the optimal goal u steps later is at most Lu. Then the loss accumulated while holding a goal fixed for T steps is at most $\int_0^T Lu\,du = \frac{1}{2}LT^2$. Writing the total per-decision cost as $f(T) = a(H/T)^2 + \frac{1}{2}bLT^2$ and setting $f'(T) = -2aH^2/T^3 + bLT = 0$ gives $T^4 = 2aH^2/(bL)$, hence $T^\star = (2a/(bL))^{1/4}\sqrt{H} = \Theta(\sqrt{H})$, and $f''(T^\star) > 0$ confirms a minimum. The exponent $\frac{1}{2}$ is therefore a consequence of the two exponents in A1 and A2 and not an input to the argument; if A2 were linear in T rather than quadratic the minimiser would be $\Theta(H^{2/3})$, which is why the quadratic form is stated explicitly rather than assumed silently.

Scope. Neither assumption transfers verbatim to the setting in which we deploy the method. A1 is a tabular bound and the SUMO environment has continuous observations and function approximation; A2 posits a uniform Lipschitz drift rate, whereas traffic demand is piecewise stationary with abrupt transitions. Remark 1 should therefore be read as a scaling heuristic that generates a falsifiable prediction (slope $\frac{1}{2}$ on a log-log plot of $T^{1\star}$ against H) rather than as a guarantee. We report in Section 5.3 that this prediction is not yet tested.

Remark 2 (comparison to the best fixed configuration). Let $\mathcal{C}$ be the finite class of configurations reachable by the meta-controller and $J(\mathbf{c})$ the inner-loop return under a fixed $\mathbf{c}$. If the outer loop is run as a no-regret online learner over $\mathcal{C}$, then after M meta-iterations $\frac{1}{M}\sum_{e=1}^{M} J(\mathbf{c}_e) \geq \max_{\mathbf{c}\in\mathcal{C}} J(\mathbf{c}) - \mathcal{O}\big(\sqrt{\log|\mathcal{C}|/M}\big)$. We state this as a design target, not a property of our implementation: first-order MAML with Gumbel-softmax sampling is not a no-regret algorithm and is not guaranteed to reach the global optimum of Equation 7. Earlier versions of this work claimed that the adaptive scheme provably dominates the best fixed configuration; that claim does not hold as stated, because it assumed attainment of the global optimum. What survives is weaker and still useful: the configuration class *contains* the best fixed configuration, so nothing is lost by construction, and a no-regret outer loop

would close the gap at the stated rate. The empirical comparison in Section 5.4 is what supports the claim in practice.

D Task Descriptor Design

The 5 components of Equation 2 were chosen because each is observable before interaction and each is either a determinant of one of the two structural sensitivities of Section 3 or a cheap proxy for initial-state difficulty. The team size n drives the feedback-gain variance term of Appendix E; the estimated horizon H_{est} drives the commitment-interval term through Remark 1; the reward dimension K determines the dimension of the preference simplex; and the two initial-state moments distinguish congested from free-flowing starts at otherwise identical (n, H, K). Removing any one of them removes the information the corresponding head would need. We do not claim the set is sufficient to characterise arbitrary traffic tasks: robustness to incomplete or misspecified descriptors is untested and is recorded in Section 6.1. The embedding dimension of 64 balances expressiveness against overfitting given only 15 meta-iterations of outer-loop data.

E Extension to Adaptive Feedback Gains

For cooperative tasks with n agents, each contributing an independent stochastic policy π_i^3 at the execution level, the aggregated value function $V^2(s_{t+1})$ entering the strategic reward is a mean over n terms. Under standard regularity conditions on policy stochasticity, the variance of the resulting policy gradient for π^1 is

$$\mathrm{Var}\left[\nabla_\theta J^1\right] \;\propto\; n\,(\beta^{12})^2\,\sigma^2_{\text{low}}, \tag{9}$$

where σ^2_{low} is the per-agent policy variance. For fixed β^{12} this variance grows linearly with n, degrading the signal-to-noise ratio of the strategic gradient. The meta-controller can address this because β^{12} and β^{23} are conditioned on the embedded agent count, the first component of Equation 2. Setting $\beta^{12} \approx c_\beta/\sqrt{n}$ keeps $\mathrm{Var}[\nabla_\theta J^1]$ approximately constant in n, and the meta-training objective creates pressure toward this schedule, since configurations producing high J_{inner} on large-n tasks are precisely those that stabilise the strategic gradient.

Status of this section. This is an analysis of what the framework is expected to learn, not a report of what it was observed to learn. Because team size is not varied in the present evaluation (Section 6.1), the emergence of the $1/\sqrt{n}$ schedule is untested. We retain the analysis because it motivates the inclusion of n in the descriptor, and we flag it as a prediction so that it is not mistaken for a result.

F Environment and Experimental Details

SUMO-RL network configuration. Experiments use Eclipse SUMO v1.12.0 with the `sumo-rl` Python interface (Alegre, 2019), driven through LibSumo on Windows 11. Networks are generated programmatically with lane width 3.2 m, three lanes per approach (through, right-turn, protected left-turn), a speed limit of 50 km/h (13.89 m/s), approach lanes of 250 m of which the final 50 m form a deceleration section before the stop line, yellow phase 3 s, and minimum green phase 5 s. The SUMO internal clock advances at 1 s and agents act every 5 s, so each 3,600-second episode comprises $H = 720$ decision steps, equivalent to 1 hour of simulated peak-hour traffic.

Heterogeneous vehicle fleet. Table 3 is the authoritative fleet specification; no other fleet description appears in this paper. Vehicle classes are spawned with independent Poisson arrival processes calibrated to peak-hour arterial counts.

Table 3: Heterogeneous vehicle type specifications (authoritative).

Type	Accel. (m/s^2)	Max Speed (m/s)	Length (m)	Arrival share
Private car	2.6	13.89	4.5	70%
Heavy truck	1.3	11.11	12.0	15%
Transit bus	1.5	12.50	10.0	15%

The heterogeneity is operationally significant: trucks and buses require earlier green-phase allocation to clear the stop line before the yellow phase, and their lower acceleration means a queue of trucks contributes disproportionately to waiting-time penalties relative to the same number of private cars. This couples the execution-level congestion objective to the coordination-level delay objective, so optimal phase actuation must account for vehicle type and not merely queue length.

Agent observation and action spaces. Each signal agent receives a local 48-dimensional observation per decision step: lane densities (12 values, 3 lanes $\times$ 4 approaches), queue lengths (12), mean speeds (12), phase encoding (8, one-hot current phase plus normalised elapsed time in phase), and waiting time (4, cumulative per approach normalised by episode length). Context and memory fusion augments this to 64 dimensions. The action space is discrete with 4 choices: advance to the next phase in the ring-barrier sequence, maintain the current phase, request early phase termination, or activate the protected left-turn phase. Transitions violating the minimum-green constraint are masked.

Objective-to-level assignment rationale. The assignment in Table 1 follows from the temporal structure of each signal. r^{speed} is a network-level average that changes slowly and whose improvement requires phase coordination across intersections over many steps, so assigning it to the strategic level makes that policy plan at the timescale on which its own objective actually moves. r^{wait} accumulates per vehicle and is sensitive to sequences of green-phase decisions over T^2-length bursts, and green-wave coordination between adjacent agents requires exactly this medium horizon. r^{queue} responds immediately to phase actuation and supplies the dense per-step signal that drives execution. Assigning all three objectives to a single level, as flat multi-objective methods do, forces the policy to plan at one timescale for signals that are operationally meaningful at three, and it is this mismatch that the learned per-level discounting resolves.

G Baselines and Comparison Rationale

Table 4: Structural comparison of evaluated methods.

Method	Exec. Arch.	Reward Treatment	Temporal Commitment	Configuration Mechanism
HIRO	2-level hierarchy	Scalarised	Fixed horizon c	Off-policy correction
HiPPO	2-level hierarchy	Scalarised	Randomised, unlearned	Sub-policy adaptation
DUSDi	2-level (skills + selector)	Scalarised	Fixed skill horizon	Mutual-information reward
MOSMAC	Flat monolithic	Pareto (multi-objective)	Single-step	SAC distribution
Meta-MOHRL	**3-level hierarchy**	**Pareto (multi-objective)**	**Task-conditioned, learned per episode**	**Bilevel meta-optimisation**

Each baseline isolates a specific structural dimension of the contribution. Nachum et al. (2018) and Li et al. (2020) represent fixed-structure hierarchical RL: both decompose long-horizon tasks into subgoals but commit to temporal boundaries that are not a function of the task, so comparing against them tests the central hypothesis that this rigidity becomes a bottleneck when the intrinsic task timescale shifts between instances. DUSDi (Hu et al., 2024) asks whether automatically deriving reusable low-level skills suffices when no adaptive high-level temporal manager exists. MOSMAC (Geng et al., 2025) tests whether Pareto coverage alone meets the credit-assignment demands of long-horizon cooperative tasks. Together the four dissect the problem space along hierarchy depth, temporal adaptability, unsupervised skill structure and multi-objective coverage.

H Environmental and Simulation Specifications

Table 5 gives the simulation mechanics separately from the learning hyperparameters of Appendix J. Every entry is generated from the same specification block as the main text, so no value here can disagree with the value used elsewhere.

Table 5: Environmental and simulation specifications.

Category	Component	Specification / Configuration
Simulation Infrastructure	Engine version	Eclipse SUMO v1.12.0
	Interfacing API	LibSumo (TraCI-compatible API, Windows 11)
	Hardware profile	Intel Core Ultra 9 185H (16C/22T), 32 GB RAM, Intel Arc integrated GPU (ASUS Zenbook Duo)
	Simulation precision	1 s SUMO clock; 5 s per decision step; $H = 720$ decisions per 3,600 s episode (1 h)
Topological Configurations	Intersection architectures	15 randomised layouts (4-way, dual, grid)
	Traffic generation model	Poisson arrivals (0–3200 veh/hr)
	Lane specifications	250 m approaches; final 50 m deceleration section
	Team size range	$n \in [4, 100]$
	Active vehicles	1,000–5,000
Vehicle Fleet Profile	Vehicle types	Private car, heavy truck, transit bus (Table 3)
	Kinematics	As Table 3; min gap 2.5 m
State & Context	Raw observation	48 dims: lane density, queue, speed, phase, wait
	Augmented state ($\tilde{s}_t$)	64 dims (context + memory fused)
	Task descriptor $\mathbf{d}$	5 dims, Equation 2
	Task embedding $\mathbf{z}$	64 dims, Equation 3
Multi-Objective Ranges	o_1: Speed	$[0, 1]$, normalised by $v_{\max}$
	o_2: Wait time	≤ 0 by Equation 11; episode range $[-8000, 0]$ s
	o_3: Queue	≤ 0 by Equation 12; episode range $[-12000, 0]$

I Computational Complexity

Meta-MOHRL's total inference work is *lower* than the fixed-structure baselines despite the added meta-controller, because the learned $T^1 \approx 40$ reduces strategic forward passes relative to $T^1_{\text{fixed}} = 10$.

Table 6: Theoretical computational complexity. $|S|$ is the state dimension, K the number of objectives, c a fixed temporal boundary, T^1 the adaptive boundary, and $H = 720$ the episode length in decision steps.

Algorithm	Inference (per step)	Optimisation	Amortised Overhead
HIRO	$\mathcal{O}(\lvert S\rvert + \frac{\lvert S\rvert}{c})$	$\mathcal{O}(H \cdot \nabla_{actor})$	None
HiPPO	$\mathcal{O}(\lvert S\rvert + \frac{\lvert S\rvert}{c})$	$\mathcal{O}(H \cdot \nabla_{PPO})$	None
DUSDi	$\mathcal{O}(\lvert S\rvert \log \lvert S\rvert)$	$\mathcal{O}(H \cdot \nabla_{actor} + MI_{bound})$	Heavy intrinsic scaling
MOSMAC	$\mathcal{O}(K \cdot \lvert S\rvert)$	$\mathcal{O}(K \cdot H \cdot \nabla_{actor})$	None
Meta-MOHRL	$\mathcal{O}\left(K \cdot \lvert S\rvert + \frac{\lvert S\rvert}{T^1}\right)$	$\mathcal{O}(K \cdot H \cdot \nabla_{inner})$	$\mathcal{O}\left(\frac{1}{H} \cdot \nabla_{meta}\right)$ (negligible)

J Hyperparameters

All experiments were conducted on a consumer laptop (ASUS Zenbook Duo UX8406MA) with an Intel Core Ultra 9 185H CPU (16 cores, 22 threads, 3.8 GHz base) and 32 GB unified memory, without a dedicated GPU. The full training pipeline (7 algorithms $\times$ 4,000 episodes $\times$ 5 seeds) completed within 72 hours, indicating that the bilevel optimisation imposes no specialised compute requirement beyond those of the baselines it wraps. Eclipse SUMO v1.12.0 was interfaced via LibSumo on Windows 11. Experiment pipelines, topology generation and Pareto evaluation are orchestrated by Python; metrics and hypervolumes are provided in the code. Training trajectories are smoothed with a moving-average window for visualisation only; all reported numbers are computed on unsmoothed data.

Table 7: Inner loop (hierarchical policies) hyperparameters.

Parameter	Value	Notes
Optimiser	Adam	
Activation functions	ReLU	Context encoders
Execution-level learning rate (α_3)	1×10^{-3}	Primitive actions
Coordination-level learning rate (α_2)	5×10^{-4}	Subgoals
Strategic-level learning rate (α_1)	5e-4	Team goal
Batch size	64	Sampled from experience buffers
Critic Polyak τ	0.005	Soft target network updates
MLP hidden layers	$[256, 256]$	Action-value approximation
Replay buffer size ($\mathcal{D}$)	1×10^5	Separated per level
Decision timestep (Δt)	5 s	Simulation clock step
Decision steps per episode (H)	720	$= 3{,}600$ s / 5 s (1 h)
Environments	15	Randomised multi-agent topologies
Raw observation dim	48	SUMO lane/phase/wait features
Augmented observation dim ($\tilde{s}_t$)	64	Context + memory fused
Action dimension	4	Discrete signal phases (per agent)
Inner steps per meta-iteration (K)	5	

Table 8: Outer loop (meta-controller) hyperparameters, configuration bounds, and descriptor normalisation constants. The T bounds below are the only admissible ranges stated in this paper; the Gumbel candidate grids of Equation 5 lie inside them. The normalisers in the last block bound the input range of Equation 2 and are not evaluated operating points; the evaluated ranges of n and H are in Section 4.5.

Parameter	Value	Notes
Optimiser	Adam	
Meta learning rate α_{out}	5×10^{-4}	Shared across heads
Gradient clipping	$\|\cdot\|_2 \leq 1.0$	
Encoder / meta-controller hidden	$[128, 128]$	Two-layer MLPs, ReLU
Task descriptor dim	5	Equation 2
Task embedding dim	64	Equation 3
Gumbel temperature (init / min)	5.0 / 0.5	
Gumbel temperature decay	0.9 per meta-iteration	$\tau(e) = \max(0.5,\ 5.0 \cdot 0.9^e)$
Meta-iterations M	15	
Soft-update coefficient λ	0.1	$\theta_0 \leftarrow (1-\lambda)\theta_0 + \lambda\theta_K$
T^1 bounds	$[10, 50]$ steps	Strategic commitment
T^2 bounds	$[3, 10]$ steps	Coordination commitment
T^3	1 (fixed)	Execution
$\gamma^1, \gamma^2, \gamma^3$ bounds	$(0.90,\ 0.999)$	$\gamma^l = 0.90 + 0.099\,\sigma(\cdot)$
β^{12}, β^{23} range	$(0.0, 1.0)$	Adaptive feedback gains
Normaliser $n_{\max}$	40	Descriptor scaling only; not an evaluated n
Normaliser $H_{\max}$	4,000	Descriptor scaling only; not an evaluated H
Normaliser $K_{\max}$	5	Descriptor scaling only

K Multi-Seed Reproducibility

Meta-MOHRL's total, speed and waiting-time reward across 5 independent seeds (42–46) is shown in Figure 7. All seeds follow a consistent convergence trajectory, with inter-seed standard deviation below ± 1.5 throughout training, indicating that the reported gains are robust to random initialisation rather than artefacts of a favourable seed.

Table 9: Baseline-specific configuration for the DUSDi skill-discovery comparison. These settings parameterise the baseline, not Meta-MOHRL.

Parameter	Value	Description
Mixture density network	Gaussian	Maps augmented state to z-space
Individual latent dim	16	Per-vehicle local queueing
Team latent dim	32	Global intersection pressure
Mixing network width	$[32, 32]$	Value decomposition (QMIX-style)
Disentanglement weight λ_d	0.05	Sparsity regulariser on value overlap
Context decoder arch.	$[64, 64]$	Hidden trajectory state reconstruction
MARL architecture	CTDE	Centralised train, decentralised execution

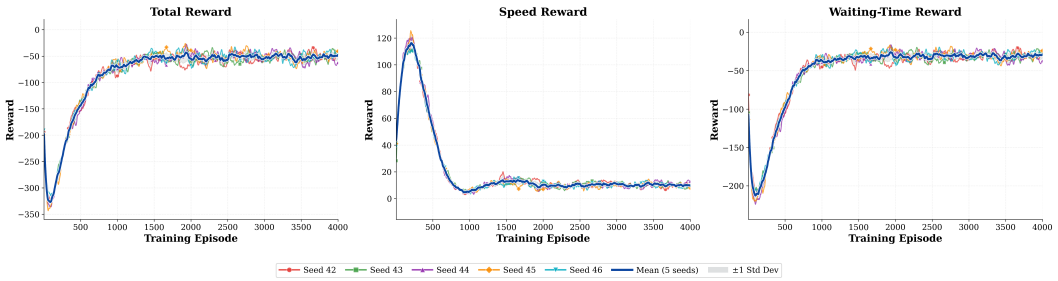


Figure 7: Meta-MOHRL training convergence across 5 seeds (42–46). Shaded region: ± 1 std.

L Reward and Pareto Relationship

At each decision step t the environment returns $\mathbf{R}_t = [r_t^{\text{speed}}, r_t^{\text{wait}}, r_t^{\text{queue}}]$:

$$r_t^{\text{speed}} = \frac{1}{|\mathcal{V}_t|} \sum_{v \in \mathcal{V}_t} \frac{v_{\text{speed}}}{v_{\max}} \in [0, 1], \tag{10}$$

$$r_t^{\text{wait}} = -\frac{1}{|\mathcal{V}_t|} \sum_{v \in \mathcal{V}_t} w_v \leq 0, \tag{11}$$

$$r_t^{\text{queue}} = -\frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} q_l \leq 0, \tag{12}$$

where $\mathcal{V}_t$ is the set of vehicles in the network at step t, v_{speed} and $v_{\max}$ the current and maximum speed of vehicle v, w_v the accumulated waiting time of v in seconds, $\mathcal{L}$ the set of lanes, and q_l the normalised queue length of lane l. Both r^{wait} and r^{queue} are non-positive by construction, which is why the episode ranges in Table 5 have zero as their upper bound.

Relation to the Pareto front. The three objectives conflict, since maximising arterial speed increases waiting time on crossing approaches, so no single optimal solution exists. Maintaining the reward as a vector lets the inner-loop networks map a multi-dimensional value space and learn a surface of non-dominated solutions, on which no solution improves one objective without worsening another.

Pareto front coverage. In Figure 8, Meta-MOHRL policies spread across the observable trade-off region while competing methods collapse into tight clusters biased toward their dominant objective: HIRO clusters near the speed axis, and HiPPO and DUSDi form overlapping mid-range groups that fail to span the extremes. Meta-MOHRL achieves the highest hypervolume ($\mathcal{H} = 4.50$), exceeding the no-feedback ablation (3.80), DUSDi (3.70), HIRO (3.60), MOSMAC (3.50), HiPPO (3.40) and the no-meta-controller ablation (3.20). The 40.6% gap between the full model and the no-meta-controller variant indicates that the adaptive configuration, not the three-level architecture alone, is the primary driver of Pareto coverage.

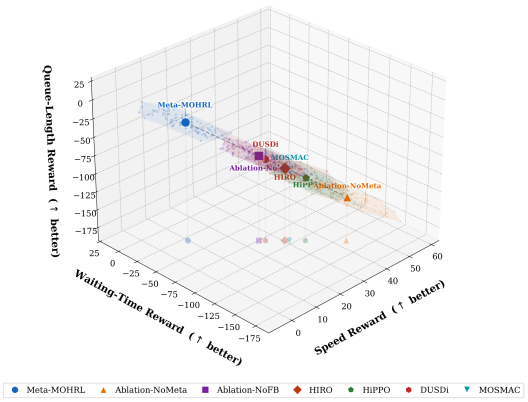


Figure 8: 3D Pareto front in the Speed $\times$ Wait $\times$ Queue objective space. Meta-MOHRL policies (blue) spread across the full trade-off region, while baselines collapse into narrow clusters.