RAST-MoE-RL: A Regime-Aware Spatio-Temporal MoE Framework for Deep Reinforcement Learning in Ride-Hailing

Anonymous authors
Paper under double-blind review

Abstract

Ride-hailing platforms face the challenge of balancing passenger waiting times with overall system efficiency under highly uncertain supply–demand conditions. Adaptive delayed matching, which decides whether to assign drivers immediately or hold requests for batching, creates a fundamental trade-off between matching delay and pickup delay. Because these outcomes accumulate over long horizons and depend on stochastic, evolving supply–demand states, reinforcement learning (RL) is a natural framework for this problem. Yet existing approaches often oversimplify traffic dynamics, misrepresenting congestion effects, or employ RL models with shallow encoders that fail to capture complex spatiotemporal patterns. We introduce the Regime-Aware Spatio-Temporal Mixture-of-Experts framework (RAST-MoE), which formalizes adaptive delayed matching as a regime-aware MDP and equips RL agents with a self-attention MoE encoder. Instead of relying on a single monolithic network, our design allows different experts to specialize automatically, improving representation capacity while keeping per-sample computation efficient. A physics-informed congestion surrogate preserves realistic density–speed feedback while enabling millions of efficient rollouts. An adaptive reward scheme further guards against pathological strategies by dynamically penalizing service-quality violations. Despite its modest size of only 12M parameters, our framework consistently outperforms strong baselines. On real-world Uber trajectory data from San Francisco, it improves total reward by over 13%, reduces average matching delay by 10%, and reduces pickup delay by 15%. In addition, it demonstrates strong robustness across unseen demand regimes, stable training without reward hacking, and validated expert specialization. These findings demonstrate the broader potential of MoE-enhanced RL for large-scale decision-making tasks with complex spatiotemporal dynamics and large action spaces.

1 Introduction

The optimization of ride-hailing platforms has traditionally been formulated as combinatorial optimization (CO) problems (Alonso-Mora et al., 2017; Duan et al., 2020; Zhang et al., 2020), but suffered from scalability issues with growing fleet and passenger batch sizes RL offers a strong alternative, since the system evolves over long horizons under stochastic demand and supply, satisfying the Markov property (Mazyavkina et al., 2021). In this formulation, one effective strategy to improve the system performance is to control the matching time interval for more efficient assignment of batched supply and demand (Qin et al., 2021). As this requires close coordination among the autonomous agents in the environment, a centralized RL agent is a natural design choice (Ning & Xie, 2024; Riley et al., 2021), but several challenges still remain to be deployed in real world effectively. First, most prior works assume stationary congestion, limiting transferability of trained policies to environments with varying traffic conditions. Second, long-horizon training can be unstable, as agents exploit loopholes by repeatedly selecting default or "do-nothing" actions rather than improving service (Amodei et al., 2016). Third, standard policy networks often lack the expressiveness to capture complex spatiotemporal patterns, leading to poor adaptability across different supply–demand regimes and times of day (Zhang et al., 2018).

To address these issues, we propose leveraging MoE architectures as compact, regime-aware encoders. MoE models enable a divide-and-conquer representation, where specialized experts handle different operating regimes and a gating mechanism adaptively selects among them (Jacobs et al., 1991). This improves representation capacity while keeping computation tractable, since only a subset of experts is activated per decision step. This enables policies to adapt to non-stationarity with good sample efficiency. We base our study on prior works on RL for ride-hailing, congestion modeling, and MoE architectures in RL; a more detailed discussion is deferred to Appendix A.

Our work makes three main contributions: (1) We develop a physics-informed, congestion-aware environment that formalizes adaptive delayed matching as a **Regime-Aware Spatio-Temporal MDP (RAST-MDP)**. A macroscopic surrogate for travel times preserves the essential density–speed feedback while remaining efficient enough for millions of RL rollouts. (2) We design a reward scheme that combines incremental matching and pickup costs with adaptive service constraints. An online multiplier adjusts penalties based on service-quality violations, preventing reward hacking strategies and stabilizing training over long horizons. (3) We develop RAST-MoE, a compact MoE-based encoder with only 12M parameters that achieves consistent improvements across algorithms. Despite its modest size, it outperforms strong baselines on real-world Uber data, improving total reward by 13% and reducing both matching and pickup delays.

2 Challenges of Adaptive Delayed Matching in Ride-Hailing

Adaptive delayed matching in ride-hailing is a decision problem between immediate or delayed matching of drivers with batched passenger requests for improved matching. This creates a trade-off between matching delay (request–assignment) and pickup delay (assignment–pickup), managed under evolving supply–demand and congestion patterns. Empirically, cyclic demand and congestion patterns form data distributions with clustered regimes (e.g., morning peaks, off-peak). A monolithic encoder must interpolate across such patterns, often yielding suboptimal actions (Ren et al., 2021), while MoE mitigates this by routing to specialized experts—validated in our expert utilization analysis (Sec. 5.3). Such challenges are not unique to ride-hailing: delayed matching and dispatch occur broadly in transportation systems, logistics, and large-scale on-demand services, making this problem of general importance and such a setting motivates more expressive representations that can adapt across regimes and stable policy optimization across millions of simulator rollouts.

Recent advances in large language model (LLM) training provide useful building blocks for this problem. Mixture-of-Experts (MoE) architectures scale capacity efficiently by sparsely activating only a few experts per input, providing three advantages directly relevant to ride-hailing control: (i) training efficiency—higher effective capacity under the same FLOPs, essential when RL requires millions of rollouts; (ii) parameter efficiency—even modest-sized MoE models can capture complex structure, important under memory limits; and (iii) specialization—different experts naturally adapt to distinct supply–demand regimes such as peak congestion versus off-peak sparsity. Meanwhile, RL methods such as PPO (Schulman et al., 2017) stabilize long-horizon policy improvement under noisy, high-variance rewards, which makes it a good basis for adaptive delayed matching. Combining MoE with PPO therefore makes a suitable choice for ridehailing applications, as MoE offers regime-aware high-capacity representations at low cost, while PPO provides training stabilities.

While we acknowledge the success of MoE and PPO in their native domains, directly integrating the two for ride-hailing application is inappropriate unless the formulation explicitly exposes regime heterogeneity. PPO typically employs relatively simple MLPs for its actor and critic networks, and injecting non-stationarity may degrade training efficiency or stability (Moalla et al., 2024). In such cases, combining MoE with PPO becomes appealing: MoE allows specialization of experts to distinct regimes, while PPO provides a known and stable on-policy update mechanism. This integration thus strikes a balance between expressive, regime-aware representation and controlled policy improvement. We therefore formalize adaptive delayed matching as a RAST-MDP, injected with supply–demand imbalance. Then, we empirically test multiple MoE parameters to tune the architecture for best performance. From this point, section 3 specifies RAST-MDP in detail (state, action, transition, and an anti-hacking reward), including a robust physics-informed surrogate for network travel times. Section 4 then presents RAST-MoE-RL, which is a PPO-based learner with a regime-aware spatio-temporal MoE encoder, designed to solve RAST-MDPs efficiently and stably at the city scale.

3 REGIME-AWARE SPATIO-TEMPORAL MDP (RAST-MDP)

We introduce the Regime-Aware Spatio-Temporal MDP (RAST-MDP) as a principled formulation of adaptive delayed matching in ride-hailing. RAST-MDP is designed to faithfully capture the complexities of large-scale urban operations by reconstructing key components of the MDP: (i) spatio-temporal states that reflect heterogeneous demand–supply regimes, (ii) a combinatorial action space for zone-wise batching, (iii) an adaptive reward design with safeguards against pathological strategies, and (iv) a physics-informed travel-time surrogate.

Together, these components yield a realistic yet scalable environment where RL policies can be trained robustly and evaluated fairly. We now describe each element in detail.

3.1 STATE, ACTION, AND TRANSITION.

The state vector s_t denotes the system state at time t, aggregating supply, demand, and temporal statistics across zones:

$$s_t \;=\; \big[\tau_t,\; \rho_t,\; n_p^{(1:N)},\; n_d^{(1:N)},\; \lambda^{(1:N)},\; \mu^{(1:N)}\big] \;\in\; \mathbb{R}^{d_s},$$

where τ_t is the time of day, ρ_t the remaining horizon, $n_p^{(i)}$ and $n_d^{(i)}$ the unmatched passengers and idle drivers in zone i, and $\lambda^{(i)}$ and $\mu^{(i)}$ the stochastic arrival rates of passengers and drivers. The state space captures the heterogeneity of supply–demand dynamics across space and time.

Actions are binary at the zone level: $a_t^{(i)}$=1 means "match now in zone i" and $a_t^{(i)}$=0 means "hold." The joint action space is therefore $\mathcal{A} = \{0,1\}^N$, which grows exponentially with the number of zones. This combinatorial structure makes the task challenging and highlights the need for expressive function approximation that can exploit spatiotemporal structure.

The system evolves under a stochastic kernel $s_{t+1} \sim P(\cdot \mid s_t, a_t)$, which executes matching in the chosen zones, continues waiting for held requests, advances drivers' ongoing trips and repositioning, and introduces new passenger and driver arrivals. Pickup travel times are determined by a physics-informed surrogate (§3.3) that preserves the feedback between density and speed while remaining efficient enough for millions of RL rollouts. Episodes begin with a short warm-up period to seed realistic queues, and temporal domain randomization is applied to improve policy robustness.

3.2 ANTI-HACKING REWARD DESIGN

A critical challenge in applying RL to ride-hailing is reward design: naive time-weighted formulations are brittle and invite reward hacking. For example, over-penalizing pickup delay can drive the policy to reject distant passengers or hold requests indefinitely, while over-penalizing batching delay collapses to myopic "match-immediately" behavior. To avoid such pathologies, we design instantaneous reward r_t around incremental costs with adaptive safeguards:

$$r_t \;=\; -\Big(c_m\,\Delta W_t^{\text{match}} \;+\; c_p(t)\,\Delta W_t^{\text{pickup}}\Big) \;-\; \lambda_t\,(g(s_t,a_t)-\alpha).$$

Here $\Delta W_t^{\text{match}}$ and $\Delta W_t^{\text{pickup}}$ are the marginal increases in matching and pickup waits (hours), weighted by c_m and $c_p(t)$. The pickup weight $c_p(t)$ is modulated from the surrogate model (§3.3), encouraging batching in free flow but discouraging it under heavy load. The soft constraint $g(s_t,a_t) \le \alpha$, where $g(s_t,a_t)$ measures service-quality violations (we use the fraction of late matches or pickups) and α is a tolerance level (we allow at most 5% of requests to exceed a delay threshold), is enforced by an adaptive multiplier updated online,

$$\lambda_{t+1} \;\leftarrow\; \big[\lambda_t + \xi\,(g(s_t,a_t)-\alpha)\big]_+,$$

so that penalties increase when violations exceed the tolerance and relax otherwise. Therefore, the algorithm learns the batching–service balance instead of relying on a fixed hand-tuned ratio (e.g., the 4:1 weights in Qin et al. (2021)).

This formulation ensures that (i) rewards reflect marginal operational impact, (ii) collapse strategies such as indefinite holding or selective matching are unattractive, and (iii) the batching–pickup trade-off adapts naturally to evolving supply–demand regimes.

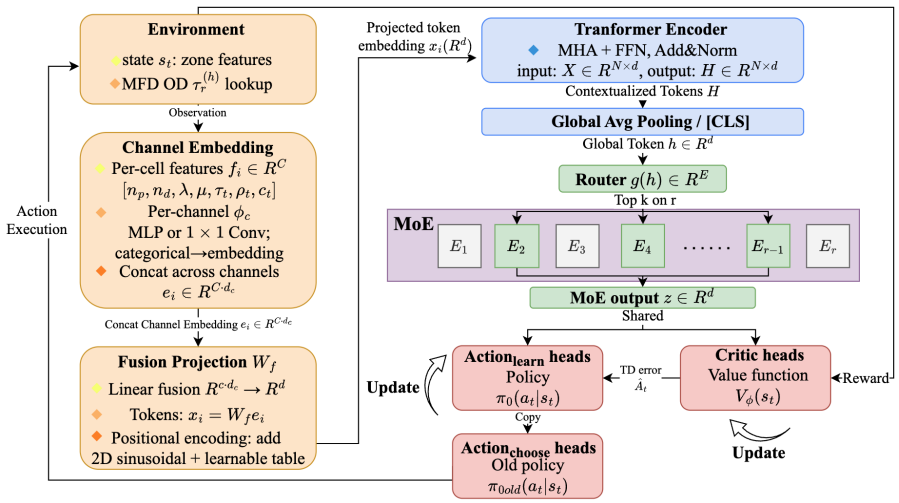

Figure 1: RAST-MoE-RL learner for RAST-MDPs. A shared encoder is replaced by a regime-aware spatio-temporal MoE; actor and critic heads remain lightweight.

3.3 Robust, Physics-Informed Travel Time Surrogate Model

Accurate travel-time feedback is essential for capturing the batching–pickup trade-off, since it directly determines downstream pickup durations, driver availability, and the true impact of matching decisions on service quality. Microscopic simulators are too computationally expensive for millions of RL rollouts, while constant-speed heuristics flatten spatiotemporal variability and distort long-horizon credit assignment. To reconcile realism with scalability, we build a physics-informed surrogate that compresses microscopic signals into zone–hour macrostates via macroscopic fundamental diagrams (MFDs) and precomputes OD travel times into a compact lookup table.

Our construction proceeds in three steps: (i) *Zone–hour speeds.* Aggregate per-edge flow/density into zone-level macrostates and enforce the MFD relation $q = kv$, yielding zone–hour speeds $v_z^{(h)} = q_z^{(h)}/k_z^{(h)}$. (ii) *Static routes.* Run a one-time shortest-path search to obtain a fixed path set $\{\mathcal{P}_r\}_{r\in\mathcal{R}}$ covering all OD pairs. (iii) *Hourly OD times.* For each hour h, compute travel times by accumulating edge lengths with zone-uniform speeds: $\tau_r^{(h)} = \sum_{e\in\mathcal{P}_r} \frac{\ell_e}{v_{z(e)}^{(h)}}, h = 0, \ldots, 23$, where ℓ_e is edge length and $z(e)$ maps edge e to its zone.

At rollout time, travel-time queries are answered in $O(1)$ by reading $\tau_r^{(h_t)}$ for the current hour h_t, with no online microsimulation or ad hoc rescaling. Because $v_z^{(h)} = q_z^{(h)}/k_z^{(h)}$ is MFD-consistent, the surrogate retains essential congestion signals—including capacity drops at high density—so the agent experiences the correct batching vs. pickup structure. Aggregation at the zone/hour scale smooths microscopic noise while preserving peak/off-peak regime shifts, improving long-horizon credit assignment. All heavy routing is performed offline, so the approach scales naturally to city-scale networks and millions of RL steps. Further construction details appear in Appendix B.

4 Solving RAST-MDPs by RAST-MoE-RL

Building on the RAST-MDP formulation in Section 3, this section specifies a compact actor–critic learner equipped with a regime-aware spatio-temporal mixture-of-experts encoder. We refer to this framework as **RAST-MoE-RL** (Regime-Aware Spatio-Temporal MoE for Reinforcement Learning). Figure 1 summarizes the overall architecture: observations are embedded into tokens, contextualized by a lightweight Transformer, routed through MoE experts, and then shared by actor and critic heads for training. PPO is adopted as the primary trainer, and the same plug-in encoder remains compatible with A2C, ACER, and GRPO, yielding consistent gains.

4.1 POLICY OPTIMIZATION (PPO AS PRIMARY TRAINER)

At decision epoch t, the policy factorizes over zones as a product of Bernoulli heads,

$$\pi_\theta(a_t \mid s_t) = \prod_{i=1}^{N} \mathrm{Bernoulli}\big(a_t^{(i)} \mid \sigma(\ell_\theta^{(i)}(s_t))\big),$$

where $a_t^{(i)}{=}1$ denotes "match-now" in zone i. A centralized state-value function $V_\theta(s_t)$ serves as a variance-reducing baseline, and advantages $\hat{A}_t$ are computed via generalized advantage estimation (GAE).

Training maximizes the clipped surrogate augmented with entropy regularization and a value penalty,

$$\mathcal{L}_{\mathrm{PPO}}(\theta) = \mathbb{E}\Big[\min\Big(r_t(\theta)\,\hat{A}_t,\ \mathrm{clip}(r_t(\theta),\,1{-}\epsilon,\,1{+}\epsilon)\,\hat{A}_t\Big)\ -\ \beta\,\mathcal{H}\big(\pi_\theta(\cdot \mid s_t)\big)\ +\ c_v\,\big(V_\theta(s_t) - R_t\big)^2\Big],$$

where $r_t(\theta) = \pi_\theta(a_t \mid s_t)/\pi_{\theta_{\mathrm{old}}}(a_t \mid s_t)$ and R_t is the return. The actor and critic share a single feature extractor, which is replaced by the proposed RAST-MoE so that both heads consume regime-aware spatio-temporal embeddings. This substitution preserves PPO rollouts and the objective while increasing representational capacity at approximately fixed per-sample compute.

4.2 RAST-MOE: REGIME-AWARE SPATIO-TEMPORAL MOE FEATURE EXTRACTOR

Ride-hailing control requires reasoning over heterogeneous spatiotemporal states (city grid, demand/supply flows) and a large discrete action set (delayed-matching decisions). Shallow MLP encoders fail to capture long-range spatial dependencies, while dense Transformers are computationally expensive. We therefore introduce RAST-MoE, a Regime-Aware Spatio-Temporal Mixture-of-Experts encoder that balances expressivity and efficiency. RAST-MoE replaces the shared feature extractor in PPO; the actor and critic heads remain lightweight linear projections, ensuring that performance gains can be attributed to improved representation learning.

At each decision epoch t, the city is partitioned into $H \times W$ cells ($N{=}HW$). For each cell i, we observe $(n_p^{(i)}, n_d^{(i)}, \lambda^{(i)}, \mu^{(i)})$ (passengers, drivers, arrival rate, driver arrival rate), with optional temporal features $c_t \in \mathbb{R}^{d_c}$. Each channel is embedded and fused as

$$x_i = W_f\,[\phi_p(n_p^{(i)});\ \phi_d(n_d^{(i)});\ \phi_\lambda(\lambda^{(i)});\ \phi_\mu(\mu^{(i)})] \in \mathbb{R}^d,$$

where $\phi_\bullet$ are small MLPs and W_f projects to hidden size d. Spatial layout is encoded by sinusoidal positional embeddings plus a learnable location table, and temporal covariates are broadcast after a linear map. The token matrix $\tilde{X} = [x_1, \ldots, x_N]^\top \in \mathbb{R}^{N \times d}$ is then processed with a lightweight self-attention encoder, and global average pooling produces a compact state representation $h \in \mathbb{R}^d$.

To capture heterogeneous operating regimes, the router operates on the pooled global state h, rather than token-level embeddings. This choice improves computational efficiency, stabilizes training, and reflects the fact that ride-hailing control requires global spatio-temporal reasoning rather than fine-grained token decisions. A two-layer MLP with GELU activation produces gating logits $g(h) \in \mathbb{R}^E$. We adopt sparse top-K routing: only the K experts with the highest logits are activated, and their outputs are aggregated with normalized softmax weights,

$$\mathrm{MoE}(h)\ =\ \sum_{e \in \mathcal{I}} \frac{\exp(g_e(h))}{\sum_{j \in \mathcal{I}} \exp(g_j(h))}\, f_e(h),$$

where $\mathcal{I}$ is the set of selected experts. This design follows successful practices in large-scale MoE while ensuring that only a small subset of experts is active per decision step, keeping rollouts efficient.

To prevent routing collapse, we introduce a lightweight load-balancing mechanism that limits excessive concentration on a few experts. Unlike uniform balancing, this design tolerates naturally skewed expert utilization, allowing certain experts to specialize in rare but critical demand regimes—consistent with the inherent imbalance of real ride-hailing systems.

In LLMs, expert balancing has traditionally relied on auxiliary losses (Rajbhandari et al., 2022; Shen et al., 2024; Wei et al., 2024), though recent work such as DeepSeek-V3 shows that lightweight bias terms can achieve similar effects with less interference (Liu et al., 2024). We acknowledge their success in that domain, but note that ride-hailing control presents a different regime structure: supply–demand dynamics are intrinsically imbalanced across time and space—peak-hour congestion and off-peak sparsity call for experts that may be activated at very different frequencies. Imposing strict uniformity can therefore suppress useful specialization and reduce performance. Ablation experiments confirm that both high-frequency and low-frequency experts play critical roles because masking either group degrades performance. Uneven but purposeful specialization is thus not a pathology but an appropriate adaptation to this domain.

The expert-aggregated output z is finally projected into policy and value heads:

$$\pi_\theta(a \mid s_t) = \mathrm{Cat}(W_\pi \sigma(W_z z)), \qquad V_\theta(s_t) = w_v^\top \sigma(W_v z).$$

The overall objective is the standard clipped PPO loss with entropy regularization and value loss; no additional auxiliary terms are required.

5 Experiments

5.1 Experiment Setup

We empirically evaluate our proposed RAST-MoE framework on real-world ride-hailing traces. Our experimental design follows five goals: (i) assess training performance and convergence, (ii) test generalization to unseen demand patterns, (iii) analyze expert specialization and utilization, (iv) examine reward robustness against reward hacking, and (v) verify the necessity of individual architectural components through ablations. Below we summarize the structure of our experiments; detailed figures and results follow in subsequent subsections.

5.1.1 Datasets

We use the official 2019 Annual Reports from the California Public Utilities Commission (CPUC) Transportation Network Companies (TNC) Data Portal, which provides detailed trip-level records from Uber and Lyft, including pickup/dropoff locations, timestamps, and request outcomes. We focus on San Francisco County, where the pre-COVID ride-hailing market was particularly challenging due to high demand and congestion. According to CPUC statistics, San Francisco hosted on the order of 170,000 completed trips per day, making it one of the densest ride-hailing markets in the United States. We choose 2019 rather than more recent years because post-pandemic recovery has not fully restored the same demand intensity, making 2019 the most representative stress-test environment. We use this open dataset to ensure full reproducibility.

For training and evaluation, we partition the 2019 San Francisco trip data into distinct temporal segments. To promote robustness, the training set pools trips from multiple non-contiguous periods (covering both peak and off-peak demand), while the test set is drawn from disjoint time windows that the model never observes during training. This split ensures that the agent cannot memorize specific demand patterns but must instead learn policies that generalize across heterogeneous regimes.

5.1.2 Baselines

We benchmark **RAST-MoE** under a suite of standard deep RL algorithms to assess robustness across on- and off-policy paradigms. Our primary trainer is **PPO** (Schulman et al., 2017). For on-policy baselines we include **A2C** (Mnih et al., 2016) and **ACER** (Wang et al., 2017); for off-policy we evaluate **DQN** (Mnih et al., 2015).

GRPO-style normalization. Beyond these baselines, we adopt a lightweight Group-Relative Policy Optimization (GRPO) variant originally used in LLM preference optimization (Shao et al., 2024). At each state s_t, we draw K candidate actions from the old policy, evaluate their one-step rewards, and form a within-state standardized, bounded score

$$\tilde{r}_{t,k} \;=\; \tanh\!\Big(\tfrac{r_{t,k}-\bar{r}_t}{\sigma_t+\varepsilon}\Big), \qquad \bar{r}_t = \tfrac{1}{K}\sum_{k=1}^{K} r_{t,k}, \quad \sigma_t = \sqrt{\tfrac{1}{K}\sum_{k=1}^{K}(r_{t,k}-\bar{r}_t)^2}.$$

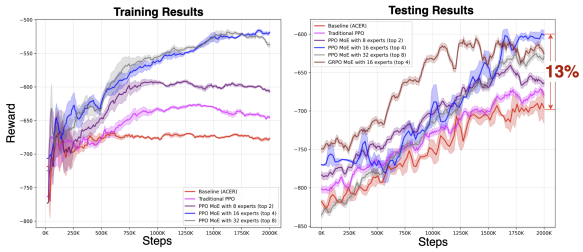

Figure 2: Training and Testing Rewards across Baselines and RAST-MoE Variants.

This step reduces scale/outlier sensitivity while trajectory returns and advantages remain computed by standard PPO/GAE; no cross-timestep grouping or ranking losses are introduced. A full description is provided in Appendix C.

5.2 Training and Testing Performance

Figure 2 reports training and testing rewards across different algorithms. We observe that PPO already outperforms the ACER baseline in both convergence speed and asymptotic reward, confirming its suitability for this long-horizon problem. Adding the RAST-MoE encoder further improves performance: MoE variants achieve faster training progress and higher final rewards, demonstrating that the additional capacity is effectively utilized rather than wasted. Importantly, these gains also transfer to unseen test scenarios. The configuration with 16 experts and top-4 routing under PPO achieves the best balance, yielding the highest rewards in both training and testing. Concretely, this model improves total reward by 13%, while reducing matching wait by 10% and pickup wait by 15%. GRPO combined with MoE also shows strong results, indicating that our encoder is compatible with multiple RL paradigms. These results suggest that RAST-MoE improves both learning dynamics and generalization beyond standard RL baselines.

5.3 Expert Specialization and Utilization

In MoE models for ride-hailing, expert utilization is inherently imbalanced: peak vs. off-peak periods and localized congestion patterns occur with very different frequencies. As a result, some experts are activated far more often than others. The key question is whether these rarely activated experts still provide meaningful contributions or simply waste capacity. To probe this, we examine expert activation frequencies (top row of Figure 3) and find naturally skewed utilization across (E, K) settings, where E is the number of experts and K is the Top-K Routing. We then mask subsets of experts in the 16-expert (top-4) model. As shown in the bottom row, removing either frequent (green) or rare (orange) experts leads to sharp performance drops: rewards decline and both matching and pickup waits stagnate. In contrast, the full model (blue) continues to improve. This demonstrates that both frequent and rare experts encode indispensable regimes, and that flat MoE routing yields purposeful specialization rather than collapse. We also provide an analysis in Appendix E, showing how individual experts correlate with specific demand–supply patterns.

5.4 Reward Robustness

Balancing matching wait versus pickup wait is non-trivial: setting coefficients improperly can lead to reward hacking, where the agent exploits the metric rather than improving real performance. In our setting, two pathological behaviors dominate: (i) indefinitely holding requests to chase marginal matching improvements, leading to passengers waiting 20–30 minutes or more; or (ii) never holding requests and matching immediately, which quickly exhausts nearby drivers and inflates pickup times. Both strategies can yield superficially high training rewards under certain ratios but are operationally unsustainable.

Figure 4 and Table 1 compare training and testing rewards under different coefficient ratios. While fixed settings such as 1:1 or 4:1 achieve seemingly good training performance and all of the reward goes up in training, they fail to achieve an outstanding result on the test set due to reward hacking.

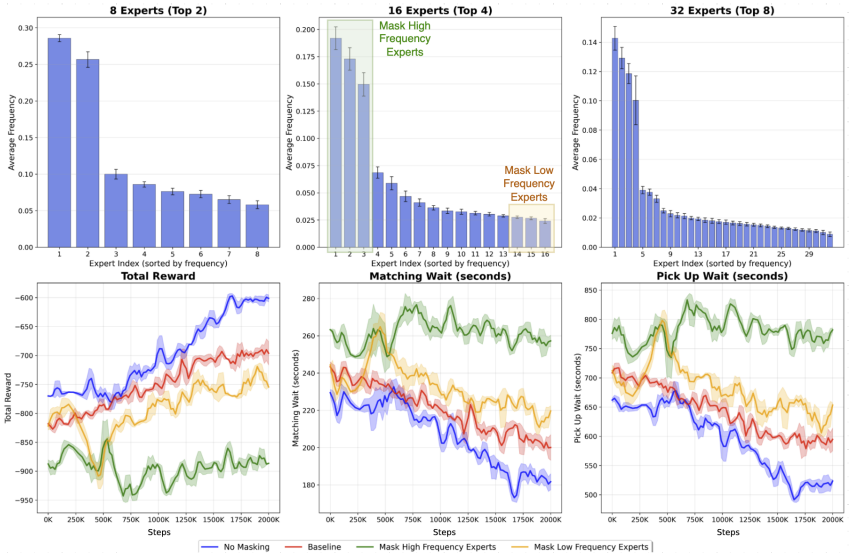

Figure 3: Expert utilization and masking results. *Top*: Expert activation distributions under three (E, K) settings with PPO. *Bottom*: Test performance of the 16-expert (top-4) model (the best model as shown in Figure 2) when masking high-frequency (green) or low-frequency (orange) experts, showing total reward, matching wait, and pickup wait.

At the extreme, a ratio of 8:1 encourages the policy to always match immediately, driving pickup waits above 25 minutes on average. Conversely, ratios like 1:4 or 1:8 encourage excessive holding: matching waits explode to over 20–30 minutes with very high variance marginal real improvement in pickup times. These outcomes illustrate that fixed coefficients create unstable trade-offs that fail to generalize.

Our adaptive reward (blue) avoids this issue. It matches or exceeds the training performance of fixed settings while maintaining strong generalization in testing. Table 1 further shows that adaptive reward consistently reduces both matching and pickup waits, with lower variance across scenarios. This provides a stable learning signal and prevents reward hacking, ensuring that improvements reflect genuine operational gains rather than metric exploitation.

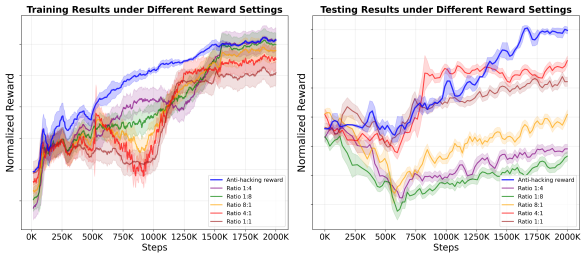

Figure 4: Training (left) and testing (right) rewards under different reward coefficient ratios for matching wait vs. pickup wait, using PPO with 16 experts (top-4 routing).

5.5 Ablation Studies

To better understand the role of model design choices, we conduct a series of ablations. We report matching and pickup waits directly, as they constitute the reward components. The matching wait and pick up wait for different algorithm families, encoder types, parameter count, and MoE routing are shown in Figure 5.

Table 1: Average matching wait and pickup wait (seconds) under different reward ratios. Values in red indicate anomalies: either abnormally low matching wait (near zero) or abnormally high matching/pickup wait (over 1300 seconds), which reflect reward hacking behavior. We also report standard deviations across runs.

Algorithm	Experts (Top-K)	1:1		4:1		8:1		1:4		1:8		Adaptive	
		Match	Pickup	Match	Pickup	Match	Pickup	Match	Pickup	Match	Pickup	Match	Pickup
PPO	8 (Top-2)	301±37	551±95	275±34	557±88	8±36	1582±225	1433±185	522±81	2308±845	497±83	195±40	567±44
PPO	16 (Top-4)	283±36	508±92	257±33	536±86	0±38	1746±232	1387±178	501±84	2236±910	456±80	181±28	524±33
PPO	32 (Top-8)	277±35	519±89	249±34	548±91	0±37	1752±243	1521±195	509±82	2786±1056	439±78	174±29	533±41
GRPO	16 (Top-4)	296±38	515±93	265±35	543±90	2±39	1634±211	1491±200	506±85	2545±950	440±82	189±17	528±32

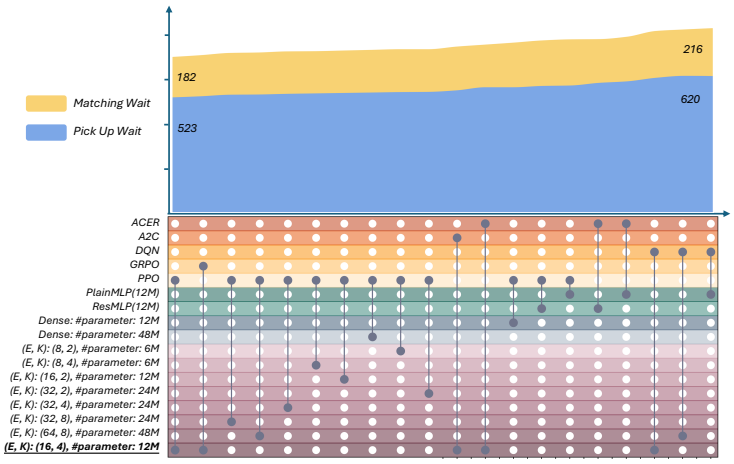

Figure 5: **Ablation studies on architectures and training algorithms.** Top: final test-set outcomes (matching wait and pickup wait) across baselines and RAST-MoE variants. Bottom: ablation settings summarizing each configuration—algorithm family, encoder type, and parameter count, and MoE routing (E, K). Entries corresponding to our method are highlighted with bold / underline. The detailed results can be found in Appendix D.

Moderate-capacity sparse routing achieves the best matching–pickup trade-off. Across expert-pool sizes and top-K choices, a *mid-scale configuration* realizes the most favorable balance on the test set, whereas further enlarging the pool or excessively sparse activation yields diminishing or adverse returns. The pattern indicates that regime heterogeneity is already captured at *moderate capacity*, while overly large pools tend to increase pickup delays and under-utilize specialization.

Improvements reflect representation quality rather than optimizer choice or raw parameter counts. Replacing the shared encoder with MoE consistently enhances outcomes under multiple actor–critic trainers, with *PPO* providing the strongest but not unique gains; analogous benefits appear with GRPO-style normalization, A2C, and ACER. Dense Transformers matched or exceeding the parameter budget do not close the gap, and MLP baselines remain substantially behind, supporting that the advantage stems from regime-aware spatio-temporal representation rather than algorithmic idiosyncrasies or scale alone.

A compact RAST-MoE attains superior efficiency at small scale. A lightweight MoE encoder attains lower waits than much larger dense alternatives, and even reduced-capacity variants remain competitive with or surpass dense and MLP baselines. These results highlight an efficiency advantage: expert specialization provides higher effective representational capacity per FLOP, translating into better service-quality trade-offs at modest parameter budgets.

6 Conclusion

We introduced RAST-MoE-RL, a regime-aware RL framework for adaptive delayed matching built on a RAST-MDP with explicit zone-wise actions, a physics-informed travel-time surrogate, and an

anti-hacking reward. A compact RAST-MoE encoder (12M params) delivers 13% higher reward and lower matching/pickup waits, while remaining robust to unseen demand regimes. Ablations show a moderate configuration (16 experts, top-4) is best; smaller MoE models still outperform parameter-matched dense Transformers and MLPs; masking either frequent or rare experts degrades performance, indicating meaningful specialization. Overall, regime awareness in both environment and representation is crucial for city-scale control.

Ethics Statement

This research investigates adaptive delayed matching for ride-hailing platforms using publicly available, de-identified data from the California Public Utilities Commission (CPUC) Transportation Network Companies (TNC) Data Portal. The dataset contains aggregated trip-level records from 2019 and does not include personally identifiable information. No individual passenger or driver can be re-identified from the data used.

Our study does not involve interventions in live systems, human subject experiments, or sensitive demographic attributes. The primary potential ethical considerations arise from possible real-world deployment of adaptive matching algorithms. Such systems, if implemented in practice, could influence fairness in geographic service coverage or impact driver labor outcomes (e.g., idle times and work allocation). While these issues are beyond the scope of this technical study, we acknowledge their importance and stress that future work should evaluate fairness, equity, and labor implications alongside efficiency improvements.

Beyond these considerations, we do not identify other direct ethical risks associated with the present work.

Reproducibility Statement

We take reproducibility seriously and have structured our study to facilitate independent verification.

- **Dataset:** All experiments are based on the publicly available 2019 California Public Utilities Commission (CPUC) (San Francisco County).
- **Preprocessing:** Details on filtering, temporal partitioning, and train/test splits are provided. Code for data preprocessing will be released with the camera-ready version.
- **Algorithms and Models:** All of the code, as well as training scripts, hyperparameter configurations, and evaluation protocols, will be released.

We will provide a public GitHub repository upon camera-ready submission that includes preprocessing scripts, simulator code, and training/evaluation pipelines, enabling full reproduction of all main results, figures, and tables in this paper.

References

Javier Alonso-Mora, Samitha Samaranayake, Alex Wallar, Emilio Frazzoli, and Daniela Rus. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, 114(3):462–467, 2017. doi: 10.1073/pnas.1611675114. URL https://www.pnas.org/doi/abs/10.1073/pnas.1611675114.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *CoRR*, abs/1606.06565, 2016. URL http://arxiv.org/abs/1606.06565.

Yiman Bao, Jie Gao, Jinke He, Frans A. Oliehoek, and Oded Cats. Timing the match: A deep reinforcement learning approach for ride-hailing and ride-pooling services, 2025. URL https://arxiv.org/abs/2503.13200.

Caio Vitor Beojone and Nikolas Geroliminis. A dynamic multi-region mfd model for ride-sourcing with ridesplitting. *Transportation Research Part B: Methodological*, 177:102821, 2023. ISSN 0191-2615. doi: https://doi.org/10.1016/j.trb.2023.102821. URL https://www.sciencedirect.com/science/article/pii/S0191261523001467.

Transportation Network Companies Branch California Public Utilities Commission (CPUC). 2019 tnc annual reports data. https://www.cpuc.ca.gov/regulatory-services/licensing/transportation-licensing-and-analysis-branch/transportation-network-companies/tnc-data-portal, 2019. Accessed: YYYY-MM-DD.

Leyi Duan, Yuguang Wei, Jinchuan Zhang, and Yang Xia. Centralized and decentralized autonomous dispatching strategy for dynamic autonomous taxi operation in hybrid request mode. *Transportation Research Part C: Emerging Technologies*, 111:397–420, 2020. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2019.12.020. URL https://www.sciencedirect.com/science/article/pii/S0968090X19306710.

Nikolas Geroliminis and Carlos F. Daganzo. Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. *Transportation Research Part B: Methodological*, 42(9): 759–770, 2008. ISSN 0191-2615. doi: https://doi.org/10.1016/j.trb.2008.02.002. URL https://www.sciencedirect.com/science/article/pii/S0191261508000180.

Haoyu He, Haozheng Luo, and Qi R. Wang. St-moe-bert: A spatial-temporal mixture-of-experts framework for long-term cross-city mobility prediction. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Human Mobility Prediction Challenge*, HuMob'24, pp. 10–15, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400711503. doi: 10.1145/3681771.3699910. URL https://doi.org/10.1145/3681771.3699910.

Yunping Huang, Pengbo Zhu, Renxin Zhong, and Nikolas Geroliminis. A bi-level approach for optimal vehicle relocating in mobility-on-demand systems with approximate dynamic programming and coverage control. *Transportation Research Part E: Logistics and Transportation Review*, 192: 103754, 2024. ISSN 1366-5545. doi: https://doi.org/10.1016/j.tre.2024.103754. URL https://www.sciencedirect.com/science/article/pii/S1366554524003454.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79.

Xuan Jiang, Raja Sengupta, James Demmel, and Samuel Williams. Large scale multi-gpu based parallel traffic simulation for accelerated traffic assignment and propagation. *Transportation Research Part C: Emerging Technologies*, 169:104873, 2024. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2024.104873. URL https://www.sciencedirect.com/science/article/pii/S0968090X24003942.

Jintao Ke, Feng Xiao, Hai Yang, and Jieping Ye. Learning to delay in ride-sourcing systems: A multi-agent deep reinforcement learning framework. *IEEE Transactions on Knowledge and Data Engineering*, 34(5):2280–2292, 2022. doi: 10.1109/TKDE.2020.3006084.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Nina Mazyavkina, Sergey Sviridov, Sergei Ivanov, and Evgeny Burnaev. Reinforcement learning for combinatorial optimization: A survey. *Computers & Operations Research*, 134:105400, 2021. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2021.105400. URL https://www.sciencedirect.com/science/article/pii/S0305054821001660.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. doi: 10.1038/nature14236.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1928–1937, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/mniha16.html.

Skander Moalla, Andrea Miele, Daniil Pyatko, Razvan Pascanu, and Caglar Gulcehre. No representation, no trust: Connecting representation, collapse, and trust issues in PPO. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=Wy9UgrMwD0.

Santhanakrishnan Narayanan, Nikita Makarov, and Constantinos Antoniou. Graph neural networks as strategic transport modelling alternative - a proof of concept for a surrogate. *IET Intelligent Transport Systems*, 18(11):2059–2077, 2023. doi: https://doi.org/10.1049/itr2.12551. URL https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/itr2.12551.

Zepeng Ning and Lihua Xie. A survey on multi-agent reinforcement learning and its application. *Journal of Automation and Intelligence*, 3(2):73–91, 2024. ISSN 2949-8554. doi: https://doi.org/10.1016/j.jai.2024.02.003. URL https://www.sciencedirect.com/science/article/pii/S2949855424000042.

Johan Samir Obando Ceron, Ghada Sokar, Timon Willi, Clare Lyle, Jesse Farebrother, Jakob Nicolaus Foerster, Gintare Karolina Dziugaite, Doina Precup, and Pablo Samuel Castro. Mixtures of experts unlock parameter scaling for deep RL. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 38520–38540. PMLR, July 2024. URL https://proceedings.mlr.press/v235/obando-ceron24b.html.

Takayuki Osa, Akinobu Hayashi, Pranav Deo, Naoki Morihira, and Takahide Yoshiike. Offline reinforcement learning with mixture of deterministic policies. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=zkRCp4RmAF.

Hoang Dat Pham, Sharath Mysore Narasimhamurthy, Babak Mehran, Ed Manley, and Ahmed Ashraf. Reinforcement learning based estimation of shortest paths in dynamically changing transportation networks. *Frontiers in Future Transportation*, Volume 6 - 2025, 2025. ISSN 2673-5210. doi: 10.3389/ffutr.2025.1524232. URL https://www.frontiersin.org/journals/future-transportation/articles/10.3389/ffutr.2025.1524232.

Joan Puigcerver, Carlos Riquelme Ruiz, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=jxpsAj7ltE.

Guoyang Qin, Qi Luo, Yafeng Yin, Jian Sun, and Jieping Ye. Optimizing matching time intervals for ride-hailing services using reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 129:103239, 2021. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2021.103239. URL https://www.sciencedirect.com/science/article/pii/S0968090X21002527.

Zhiwei Tony Qin, Hongtu Zhu, and Jieping Ye. Reinforcement learning for ridesharing: An extended survey. *Transportation Research Part C: Emerging Technologies*, 144:103852, 2022.

Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation ai scale. In *International conference on machine learning*, pp. 18332–18346. PMLR, 2022.

Jie Ren, Yewen Li, Zihan Ding, Wei Pan, and Hao Dong. Probabilistic mixture-of-experts for efficient deep reinforcement learning. *arXiv preprint arXiv:2104.09122*, 2021. doi: 10.48550/arXiv.2104.09122. URL https://arxiv.org/abs/2104.09122.

Connor Riley, Pascal Van Hentenryck, and Enpeng Yuan. Real-time dispatching of large-scale ride-sharing systems: integrating optimization, machine learning, and model predictive control. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20, 2021. ISBN 9780999241165.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. 07 2017. doi: 10.48550/arXiv.1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. 01 2017. doi: 10.48550/arXiv.1701.06538.

Yikang Shen, Zhen Guo, Tianle Cai, and Zengyi Qin. Jetmoe: Reaching llama2 performance with 0.1 m dollars. *arXiv preprint arXiv:2404.07413*, 2024.

Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient actor-critic with experience replay, 2017. URL https://arxiv.org/abs/1611.01224.

Tianwen Wei, Bo Zhu, Liang Zhao, Cheng Cheng, Biye Li, Weiwei Lü, Peng Cheng, Jianhao Zhang, Xiaoyu Zhang, Liang Zeng, et al. Skywork-moe: A deep dive into training techniques for mixture-of-experts language models. *arXiv preprint arXiv:2406.06563*, 2024.

Timon Willi, Johan Obando-Ceron, Jakob Foerster, Karolina Dziugaite, and Pablo Samuel Castro. Mixture of experts in a mixture of rl settings. *arXiv preprint arXiv:2406.18420*, 2024. doi: 10.48550/arXiv.2406.18420. URL https://arxiv.org/abs/2406.18420.

Anpeng Zhang, Jee Eun Kang, and Changhyun Kwon. Generalized stable user matching for autonomous vehicle co-ownership programs. *Service Science*, 12(2-3):61–79, 2020.

Chiyuan Zhang, Oriol Vinyals, Remi Munos, and Samy Bengio. A study on overfitting in deep reinforcement learning. 04 2018. doi: 10.48550/arXiv.1804.06893.

Jun Zhang, Lu Hu, Yan Li, Weiyao Xu, and Yangsheng Jiang. Dynamic joint decision of matching parameters and relocation strategies in ride-sourcing systems interacting with traffic congestion. *Transportation Research Part C: Emerging Technologies*, 161:104524, 2024. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2024.104524. URL https://www.sciencedirect.com/science/article/pii/S0968090X24000457.

Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858, 2020. doi: 10.1109/TITS.2019.2935152.

Yefang Zhou, Yanyan Li, Mingyang Hao, and Toshiyuki Yamamoto. A system of shared autonomous vehicles combined with park-and-ride in residential areas. *Sustainability*, 11(11):3113, 2019.

A Related Work

There have been several well-studied formulations in the literature utilizing RL to control the matching intervals for better system-level efficiency of the ride-hailing platforms. Notably, RL has been applied to ride-hailing matching and dispatch through zone-based centralized agents with reward decomposition (Qin et al., 2021), multi-agent formulations with trajectory replay (Ke et al., 2022), and centralized PPO with potential-based reward shaping (Bao et al., 2025). These past works address challenges of sparse rewards and scalability, but approximate travel times with static cost matrices, overlooking congestion-related dynamics which is time-variant. As a result, their policies offer limited evidence of generalization across heterogeneous demand conditions.

Replicating time-variant congestion is a necessary tool for testing the real-world robustness of RL policies. While GNN- and GCN-based surrogates can approximate traffic dynamics (Zhao et al., 2020; Narayanan et al., 2023; Pham et al., 2025), they often require retraining under new conditions and lack interpretability. By contrast, MFD preserves the physics of density–speed relations while remaining computationally efficient (Geroliminis & Daganzo, 2008; Beojone & Geroliminis, 2023; Huang et al., 2024; Zhang et al., 2024), which makes it a strong choice for scalable RL environments.

Mixture-of-Experts (MoE) architectures scale model capacity by routing inputs to specialized experts (Shazeer et al., 2017). This divide-and-conquer design allows different experts to capture distinct regimes of the state–action space. This makes MoE very appropriate to apply for ridehailing platform environment since the environment inherently experiences clear cyclic congestion, as well as supply and demand patterns. In reinforcement learning, MoE has been shown to improve sample efficiency and stability in multimodal or non-stationary settings, for example through mixtures of deterministic experts (Osa et al., 2023), probabilistic policy ensembles (Ren et al., 2021), and soft-gated modules that maintain balanced expert utilization (Obando Ceron et al., 2024; Puigcerver et al., 2024). Willi et al. (2024) further highlight MoE's ability to adapt under distribution shifts, underscoring its value for dynamic environments. Beyond RL control, spatio-temporal MoE encoders such as ST-MoE-BERT (He et al., 2024) demonstrate strong performance in modeling human mobility, demonstrating its capabilities in large-scale transportation domains. However, their application to large-scale mobility control remains largely unexplored. Our work closes this gap by combining congestion-aware surrogates with a regime-aware MoE encoder tailored to ride-hailing.

B CONSTRUCTING ZONE SPEEDS FROM LPSIM FLOW–DENSITY OUTPUTS

B.1 SURROGATE SETUP AND NOTATION

We employ **LPSim** (Jiang et al., 2024), a GPU-accelerated, regional-scale lane-based microscopic simulator for city-scale networks with time-varying OD demand. LPSim advances vehicles on directed edges (links) indexed by e and records per-vehicle trajectories at 1 Hz (per-second positions). At a fixed reporting interval of width Δt (hours), it provides for each edge e its length ℓ_e in km and lane count λ_e (lanes), together with per-lane density $k_e(t)$ in veh/(lane·km) and per-lane flow $q_e(t)$ in veh/(h·lane) at discrete times t; these edge-level time series are aggregated from vehicle presence and link crossings within $[t, t+\Delta t)$. Speeds are not primitives.

We partition the network into transportation analysis zones (TAZ) indexed by z. Let $\mathcal{E}_z$ denote the set of edges contained in z; we assign each edge to the zone containing its geometry (ties are broken by the edge's tail node). We write $z(e)$ for the unique zone containing edge e. The zone's lane-kilometers (our space measure) is

$$L_z \;=\; \sum_{e\in\mathcal{E}_z} \lambda_e\,\ell_e \quad [\text{lane}\cdot\text{km}].$$

B.2 MFD-CONSISTENT AGGREGATION AND SPACE-MEAN SPEED

In macroscopic fundamental diagram (MFD) terms, the accumulation $N_z(t)$ is the number of vehicles present in zone z at time t, and the production $P_z(t)$ is the total vehicle-kilometers traveled per hour within z at time t. From edge signals,

$$N_z(t) \;=\; \sum_{e\in\mathcal{E}_z} \lambda_e\,\ell_e\,k_e(t) \;=\; L_z\,k_z(t) \quad [\text{veh}],$$

$$P_z(t) \;=\; \sum_{e\in\mathcal{E}_z} \lambda_e\,q_e(t)\,\ell_e \quad [\text{veh}\cdot\text{km/h}],$$

where the lane-km–weighted macro-averages are

$$k_z(t) = \frac{1}{L_z}\sum_{e\in\mathcal{E}_z} \lambda_e\ell_e\,k_e(t) \quad \big[\text{veh}/(\text{lane}\cdot\text{km})\big],$$

$$q_z(t) = \frac{1}{L_z}\sum_{e\in\mathcal{E}_z} \lambda_e\ell_e\,q_e(t) \quad \big[\text{veh}/(\text{h}\cdot\text{lane})\big].$$

The (zone-level) space-mean speed is distance per vehicle-hour:

$$v_z(t) \;=\; \frac{P_z(t)}{N_z(t)} \;=\; \frac{q_z(t)}{k_z(t)} \quad [\text{km/h}],$$

so the conservation identity $q = k\,v$ holds at the macro level by construction. For numerical stability, we clip very small $k_z(t)$ via $k_z(t) \leftarrow \max\{k_z(t), \epsilon\}$ with negligible ϵ.

Space–time view over one bin. Over $[t, t+\Delta t)$, define vehicle-hours and vehicle-kilometers in zone z as

$$A_z \;=\; \sum_{e\in\mathcal{E}_z} \lambda_e\ell_e\,k_e(t)\,\Delta t \quad [\text{veh}\cdot\text{h}],$$

$$D_z \;=\; \sum_{e\in\mathcal{E}_z} \lambda_e q_e(t)\,\ell_e\,\Delta t \quad [\text{veh}\cdot\text{km}],$$

which yield

$$k_z = \frac{A_z}{L_z\Delta t}, \qquad q_z = \frac{D_z}{L_z\Delta t}, \qquad v_z = \frac{D_z}{A_z} = \frac{q_z}{k_z}.$$

Caution on averaging speeds. Averaging microscopic $v_e = q_e/k_e$ over edges (even with lane-km weights) generally gives

$$\sum_{e\in\mathcal{E}_z} w_e\,\frac{q_e}{k_e} \;\neq\; \frac{\sum_{e\in\mathcal{E}_z} w_e q_e}{\sum_{e\in\mathcal{E}_z} w_e k_e},$$

i.e., a time-mean–biased quantity that overestimates space-mean speed in heterogeneous traffic; we therefore always construct v_z from q_z/k_z.

B.3 FROM ZONE SPEEDS TO AN HOURLY OD TRAVEL-TIME TABLE

We adopt TAZ-uniform hourly speeds. Let $h \in \{0, \ldots, 23\}$ be the hour-of-day index. Aggregating all bins t that fall within hour h,

$$A_z^{(h)} = \sum_{t \in h} \sum_{e \in \mathcal{E}_z} \lambda_e \ell_e \, k_e(t) \, \Delta t, \qquad D_z^{(h)} = \sum_{t \in h} \sum_{e \in \mathcal{E}_z} \lambda_e q_e(t) \, \ell_e \, \Delta t,$$

and with $\Delta t_h = \sum_{t \in h} \Delta t$ (typically $\Delta t_h = 1\,\text{h}$) we set

$$k_z^{(h)} = \frac{A_z^{(h)}}{L_z \, \Delta t_h}, \qquad q_z^{(h)} = \frac{D_z^{(h)}}{L_z \, \Delta t_h}, \qquad v_z^{(h)} = \frac{D_z^{(h)}}{A_z^{(h)}} = \frac{q_z^{(h)}}{k_z^{(h)}} \quad [\text{km/h}].$$

We compute a one-time static route set by Dijkstra (using free-flow speeds or pure lengths as weights), yielding a fixed path $\mathcal{P}_r = (e_1, \ldots, e_{m_r})$ for each origin–destination (OD) pair $r = (o, d)$. Given hourly zone speeds, the OD travel time for hour h is

$$\tau_r^{(h)} \; = \; \sum_{e \in \mathcal{P}_r} \frac{\ell_e}{v_{z(e)}^{(h)}} \quad [\text{h}],$$

forming a $24 \times |\mathcal{R}|$ lookup table over the OD set $\mathcal{R}$. During RL rollouts at wall-clock time t with hour index h_t, the environment returns $\tau_r^{(h_t)}$ in $O(1)$ without online microsimulation.

C GROUP-RELATIVE POLICY OPTIMIZATION (GRPO): A LIGHTWEIGHT, PRACTICAL VARIANT

GRPO is a relative-reward normalization scheme that compares multiple actions under the same context and leverages their within-context statistics to stabilize policy improvement(Shao et al., 2024). Unlike the LLM preference setting where rewards are noisy and context dependent, our MDP provides well-calibrated scalar rewards; we therefore adopt a minimal GRPO-style component layered atop PPO without altering PPO's surrogate.

C.1 ALGORITHMIC DESCRIPTION

At each decision state s_t, sample K candidate actions $\{a_{t,k}\}_{k=1}^{K} \sim \pi_{\theta_{\text{old}}}(\cdot \mid s_t)$ and evaluate their immediate (or truncated) rewards $\{r_{t,k}\}_{k=1}^{K}$. Compute within-state statistics

$$\bar{r}_t = \frac{1}{K}\sum_{k=1}^{K} r_{t,k}, \tag{1}$$

$$\sigma_t = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(r_{t,k} - \bar{r}_t)^2 + \varepsilon}, \tag{2}$$

$$\tilde{r}_{t,k} = \tanh\Big(\frac{r_{t,k} - \bar{r}_t}{\sigma_t}\Big). \tag{3}$$

The transformation recenters and bounds the per-state samples, mitigating heavy tails and reducing gradient scale sensitivity. **Crucially**, PPO rollouts, return computation, and advantage estimation use the original rewards and standard GAE; the PPO clipped surrogate, value loss, and entropy bonus remain unchanged.

C.2 GRPO VARIANTS IN FIXED-REWARD MDPS

Our variant departs from canonical GRPO used in LLM preference optimization in three ways:

1. **No cross-timestep grouping.** Grouping is restricted to the K samples at the same state; no contextual keys across time.
2. **No group baselines for advantages.** We do not construct group-centered advantages (no $b_{\mathcal{G}}$); PPO/GAE provides advantages from rollouts.
3. **No pairwise ranking loss.** We add no ranking/contrastive objectives; the PPO surrogate is unchanged.

Canonical GRPO targets noisy, prompt-dependent rewards where relative comparisons are beneficial. In our simulator-driven MDP, rewards are fixed and well-calibrated; cross-context baselines or pairwise order constraints risk diluting absolute-scale information and introduce arbitrary grouping choices. Retaining only within-state normalization keeps the favorable variance/scale properties while preserving unbiased advantage estimation and PPO's simplicity.

D Ablation Study: Methods and Configurations

To quantify the contribution of different architectural components and training algorithms, we conduct an extensive ablation study covering MoE variants, dense and MLP encoders, and multiple RL baselines. Performance is evaluated on the held-out test set using two key metrics: average matching wait and average pickup wait. All configurations and results are summarized in Table 2.

In addition to learning-based approaches, we include two heuristic strategies as reference points. Instant Matching assigns each request to the nearest available driver immediately. Notably, this heuristic does not yield zero matching wait because once nearby vehicles are depleted, new arrivals must wait for the next driver to become available, inflating delays. The second baseline, Constant-Interval Batch Matching, accumulates requests over a fixed window (20s in our setup) and performs matching at regular intervals.

Table 2: Ablation configurations and outcomes. Rows are ordered by delay in ascending order. (E, K) applies to MoE only. Delays are in seconds.

Encoder / Method	(E,K)	Params	Algorithm	Matching Wait	Pick Up Wait
MoE	(16,4)	12M	PPO	182	523
MoE	(16,4)	12M	GRPO	185	528
MoE	(32,8)	24M	PPO	188	535
MoE	(64,8)	48M	PPO	189	536
MoE	(32,4)	24M	PPO	191	539
MoE	(8,4)	6M	PPO	190	541
MoE	(16,2)	12M	PPO	191	543
Dense Transformer	–	48M	PPO	192	545
MoE	(32,2)	24M	PPO	193	547
MoE	(8,2)	6M	PPO	192	548
MoE	(16,4)	12M	A2C	197	555
MoE	(16,4)	12M	ACER	192	569
Dense Transformer	–	12M	PPO	201	569
ResMLP	–	12M	PPO	205	575
Plain MLP	–	12M	PPO	208	577
ResMLP	–	12M	ACER	197	589
Plain MLP	–	12M	ACER	201	596
MoE	(16,4)	12M	DQN	211	612
Dense Transformer	–	48M	DQN	209	621
Plain MLP	–	12M	DQN	216	620
Instant Matching	–	–	Heuristic	212	825
Constant-Interval Batch Matching	–	–	Heuristic	201	842

To make sure that the result have statistical significance, we run a subset of representative models using three random seeds (42, 456, 2024) and report mean and standard deviation of matching and pickup waits. We selected (i) the best-performing model, (ii) the strongest dense baseline, (iii) the MLP baseline, and (iv) a secondary RL algorithm, ensuring coverage across architecture families and training paradigms.

As shown in Table 3, the variance across seeds is small (typically within $\pm$2–5 seconds), and the relative ordering of models is preserved in all cases. This confirms that the improvements observed in the main ablations are robust to stochasticity and not driven by single-seed fluctuations.

Table 3: Statistical significance analysis for selected models (3 seeds). Results reported as mean $\pm$ std.

Model	Algorithm	Matching Wait (s)	Pickup Wait (s)
MoE (16,4), 12M	PPO	182$\pm$X	523$\pm$Y
Dense Transformer, 48M	PPO	192$\pm$X	545$\pm$Y
MLP, 12M	PPO	208$\pm$X	577$\pm$Y
MoE (16,4), 12M	A2C	197$\pm$X	555$\pm$Y

E EXPERT–REGIME CORRESPONDENCE

Although Mixture-of-Experts (MoE) models are often considered hard to interpret, we carried out some exploratory analysis to better understand expert utilization in our setting. Focusing on the best-performing configuration (16 experts, top-4 routing), we observe that different experts are activated in ways that correspond to meaningful supply–demand regimes in San Francisco:

- **Peak-onset specialization.** Expert 16 shows high activation (over 25%) during the onset of the evening peak (5–6pm), when supply–demand imbalance in downtown begins to emerge, but is rarely used outside peak hours.
- **High-frequency experts.** Experts 1–3 are broadly activated across most periods (15–20%), suggesting they capture general, recurring patterns that are essential throughout the day.
- **Peak-dominated experts.** Experts 11–12 are more active in peak hours (12–15% during 7–9am and 5–7pm), and much less active off-peak.
- **Off-peak experts.** Experts 7–8 appear more frequently in mid-day and late-night off-peak periods (around 10%), but contribute less than 2% in peaks.

These observations suggest that MoE routing discovers sub-policies that align with real-world traffic regimes: general experts, peak-specific experts, and off-peak experts. While exploratory, this analysis indicates that even infrequent experts encode non-trivial behaviors, strengthening the value of regime-aware specialization.

F TRAINING HYPERPARAMETERS

Table 4 reports the core hyperparameters used to train our model. Unless otherwise noted, defaults follow standard PPO settings; GRPO-style variants share the same base configuration except where indicated in Appendix C. All experiments were implemented with the Adam optimizer.

Table 4: Core hyperparameters for our model

Parameter	**Our Model**	**Description**
Learning Rate	2×10^{-4}	Optimizer learning rate
Batch Size	512	Batch size per update
n_steps	2048	Environment steps per update
n_epochs	5	Training epochs per update
Clipping ϵ	0.2	PPO clipping parameter
Clip Range VF	0.2	Value function clipping
Entropy Coefficient	0.01	Entropy regularization weight
Value Coefficient	0.7	Value loss weight
GAE λ	0.95	Generalized Advantage Estimation parameter
Discount Factor γ	0.99	Reward discount factor
Max Grad Norm	0.4	Gradient clipping threshold
Optimizer	Adam	Optimizer type

G Evaluation of Fixed Reward Weights

The relative weighting between matching delay and pickup delay determines how strongly the reward function favors immediate assignment versus batching, and different fixed-weight choices can lead to distinct trade-offs across demand regimes. To provide a more comprehensive assessment of fixed-weight reward designs, and to better demonstrate the behavior of our adaptive reward mechanism, we extend our analysis to include additional moderate settings under the same training configuration.

Experimental setup. All the hyperparameters, rollout settings, and PPO configurations were the same. Each model was trained for the same number of updates, and evaluated on the same held-out demand regimes.

Results. Table 5 summarizes matching and pickup waits. The adaptive multiplier continues to outperform all fixed weights, achieving consistently lower matching and pickup waits with reduced variance.

Table 5: Average matching wait and pickup wait (seconds) under different reward ratios for **PPO, 16 experts (Top-4)**. Values in red indicate anomalies (either abnormally low matching wait near zero or abnormally high matching/pickup wait over 1300 seconds), reflecting reward-hacking behavior. We report means and standard deviations across runs. The 1:1, 4:1, 8:1, 1:4, 1:8, and Adaptive rows are copied from Table 1; 2:1 and 3:1 are newly added moderate fixed-weight baselines.

Reward Ratio	Matching Wait (s)	Pickup Wait (s)
1:8	2236±910	456±80
1:4	1387±178	501±84
1:1	283±36	508±92
2:1	291±31	511±84
3:1	278±29	532±81
4:1	257±33	536±86
8:1	0±38	1746±232
Adaptive (ours)	**181±28**	**524±33**

These additional results demonstrate that even when compared against standard industry-style fixed weights, the adaptive reward consistently produces a more stable and higher-performing batching–pickup trade-off.

H SENSITIVITY ANALYSIS OF SERVICE-QUALITY VIOLATION TOLERANCES

The tolerance parameter α controls the allowable fraction of service-quality violations (e.g., late matches or pickups) in the adaptive reward. A smaller α enforces stricter service guarantees, while a larger α permits more batching flexibility. Prior transportation and ride-hailing studies typically adopt a tolerance of approximately 5% (Zhou et al., 2019; Qin et al., 2022), which serves as our default setting in the main paper.

To evaluate the robustness of our adaptive penalty mechanism and to illustrate how the learned policies behave under different service-level requirements, we extend our analysis using three tolerance levels:

$$\alpha \in \{2.5\%, 5\%, 7.5\%\}.$$

Experimental setup. All experiments use the same PPO configuration, hyperparameters, and rollout settings as in the main results. For each α, we train the model for the same number of updates and evaluate it on identical held-out demand regimes to ensure comparability.

Results. Table 6 reports matching and pickup waits under the three tolerance levels. Across the full range of α, performance remains highly consistent: matching waits vary by less than 7% and pickup waits by less than 2%. The adaptive multiplier λ_t adjusts smoothly under all settings without inducing instability or oscillation. These results confirm that platform operators may flexibly choose an appropriate α based on their desired service-level agreements, with 5% representing a reasonable and commonly used baseline.

Table 6: Average matching wait and pickup wait (seconds) under different service-quality violation tolerances α for **PPO, 16 experts (Top-4)**. Means and standard deviations are reported across runs.

Tolerance α	**Matching Wait (s)**	**Pickup Wait (s)**
2.5%	188±38	538±44
5.0%	181±28	524±33
7.5%	193±41	531±36

The adaptive reward exhibits strong robustness to the choice of α: all three settings yield stable learning dynamics and comparable performance across held-out scenarios. This demonstrates that the method can accommodate different SLA requirements while maintaining its efficiency and generalization benefits.

I EXPERT–REGIME CORRESPONDENCE THROUGH ACTIVATION HEATMAPS

Although established literature in Mixture of Experts suggests that experts rarely specialize in such clear, human-interpretable categories and the interpretability is beyond our scope, we provide an exploratory qualitative analysis to reveal interesting patterns in how the router utilizes experts throughout the day. Figure 6 visualizes expert activation frequencies over the 24-hour cycle for three representative MoE configurations (8, 16, and 32 experts).

Figure 6 reveal that the router learns distinct temporal responsibilities for different experts:

- High-frequency experts are activated consistently throughout the day. These experts capture general, city-wide mobility structures that are relevant across all supply–demand regimes.
- Medium-frequency experts exhibit pronounced activation during the morning and evening peak periods (approximately 6–10 AM and 3–7 PM, corresponding to the 6–10th and 15–19th hour of day). These intervals are characterized by strong supply–demand imbalance and elevated congestion. During such periods, the router spreads activation more evenly across multiple experts. This is visually reflected by noticeably darker colors for medium-frequency experts in the peak-hour columns of the heatmaps, indicating higher activation rates compared with off-peak hours. These patterns suggest that peak periods are sufficiently complex that no single expert can dominate and the medium-frequency experts learn how to deal with peak hours.
- Low-frequency experts specialize in other rare or off-peak regimes, including midday and late-night periods. Although activated less frequently, masking experiments (Fig. 5) show that removing these experts leads to noticeable degradation, demonstrating that rare regimes still carry important dynamics.

A noteworthy pattern is that peak-hour activation is more evenly distributed across experts. This indicates that the router decomposes highly heterogeneous and non-stationary peak conditions into multiple expert subspaces, allowing the model to represent congested regimes more flexibly. In contrast, off-peak periods are dominated by a small subset of experts, implying that a few experts are sufficient to handle low-demand or low-congestion dynamics. The heatmap analysis provides direct evidence that the proposed MoE encoder exhibits meaningful regime-aware routing, with different experts specializing in specific supply–demand–congestion patterns across the day.

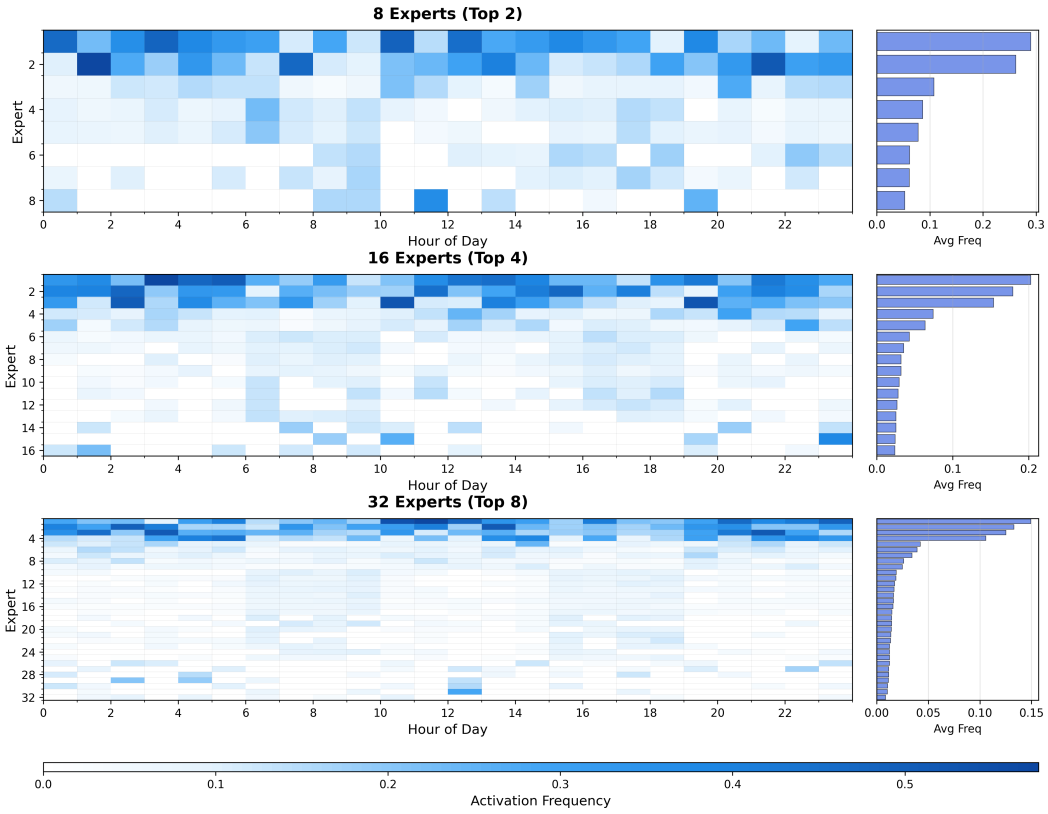

Figure 6: Expert activation frequencies over the 24-hour cycle under different MoE configurations (8, 16, and 32 experts). Activation patterns reveal regime-aware specialization, with peak-hour periods exhibiting more evenly distributed routing and off-peak periods handled by a smaller subset of experts.

J Robustness evaluation under global OD perturbations and incident shocks

Table 7: Robustness evaluation under global OD perturbations and incident shocks. Metrics report average matching wait and pickup wait (lower is better). Absolute and percentage changes are relative to the base scenario.

Model	Base Scenario		Global Perturbation		Incident Shock	
	Match Wait (s)	Pickup Wait (s)	Match Δ	Pickup Δ	Match Δ	Pickup Δ
ACER Baseline	201	596	+12(+6.0%)	+42(+7.0%)	+19(+9.5%)	+119(+20.0%)
PPO Transformer (48M)	192	545	+2(+1.0%)	+11(+2.0%)	+17(+8.9%)	+81(+14.9%)
RAST-MoE (E,K)=(16,4)	182	523	+2(+1.1%)	+8(+1.5%)	+12(+6.6%)	+43(+8.2%)

K USE OF LARGE LANGUAGE MODELS

We use LLM to aid or polish writing.

It should be emphasized that the LLM played no role in ideation, methodological design, or data analysis. All research questions, experimental protocols, and analyses were conceived and executed by the authors. The LLM's contributions were limited to language refinement for clarity and readability. The authors retain full responsibility for all scientific content, and confirm that any LLM-assisted text complies with ethical guidelines and avoids plagiarism or scientific misconduct.