TAXIGPT: DECOUPLING MOBILITY PRIORS AND ORDER-DERIVED FEEDBACK FOR PASSENGER SEEKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Passenger seeking is central to taxi and autonomous ride-service operations as a vacant vehicle need to decide where to move before its next request is assigned. An effective policy must generate realistic routes and identify movements that lead to passengers. Learning both from a single reward signal is difficult because explicitly modeling both movement and pickup/service dynamics is costly, and coarse approximations can admit unrealistic reward-seeking behavior. We propose TAXIGPT, a knowledge-decoupled framework that separates mobility learning, order-environment reconstruction, and seeking-policy adaptation. A decoder-only Transformer learns a mobility prior from serving trajectories through next-location prediction. An order-derived environment reconstructs passenger availability and joint service outcomes from historical orders, returning rewards and state transitions for variable-length route proposals. Group-relative policy optimization adapts the prior across successive seeking and serving cycles, while locality constraints and reference regularization help preserve movement structure. In Shenzhen simulations, TAXIGPT outperforms five passenger-seeking baselines, increasing cumulative episode profit by 160.7% and reducing average seeking distance by 63.3% relative to the respective best baselines. Applying locality during adaptation further improves profit by 5.6% over inference-only filtering under the same deployment constraint. These results support learning movement priors from real trajectories and seeking preferences from order-derived interaction, enabling effective passenger seeking with less reliance on explicitly modeled urban movement dynamics.

1 INTRODUCTION

Background. After completing a trip, a taxi without an assigned request must decide where to move to find its next passenger. Human drivers can draw on local experience, whereas autonomous taxis require an explicit policy that turns the current spatial–temporal context into a seeking route. These decisions affect empty travel, vehicle utilization, and service income (Han et al., 2016; Shou et al., 2020). Their consequences also extend beyond the next pickup: the passenger's destination and trip duration determine where and when the vehicle becomes available again (Yu et al., 2019). An effective seeking policy must therefore combine plausible urban movement, awareness of passenger demand, and consideration of future opportunities across successive seeking and serving cycles.

State of the art and limitations. Existing methods obtain these capabilities from different sources. Demand-based recommenders use estimated pickup opportunities to rank destinations or plan routes (Wang et al., 2015; Tran et al., 2022); imitation learning derives seeking strategies from demonstrations but centers on behavioral matching (Zhang et al., 2022). Offline RL optimizes reward from logged transitions (Levine et al., 2020), with reliable improvement limited by their coverage (Kumar et al., 2020; Busaranuvong et al., 2023). Simulator-based online RL instead generates policy-dependent experience. Han et al. (2016) learn taxi routing through local grid interactions, while Shi et al. (2019) use a deep Q-network (DQN) with spatiotemporal state and local grid moves for transportation-network-company (TNC) vehicle routing, and O'Keeffe et al. (2022) optimize neighboring road-network moves to reduce taxi idle time. These approaches enable exploration beyond logged actions, but require task-specific simulators to specify the consequences of candidate

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.

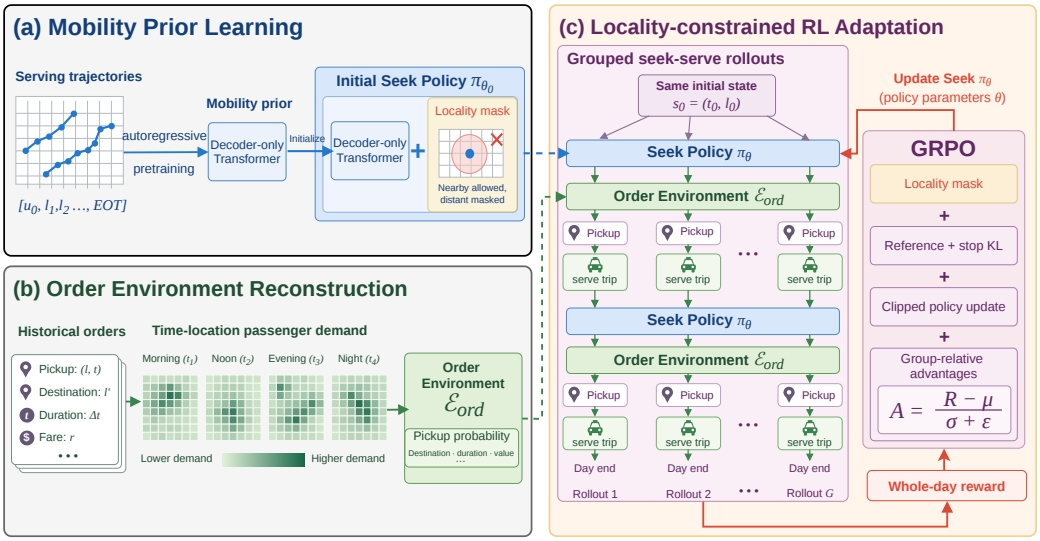

Figure 1: **TaxiGPT overview.** (a) Mobility pretraining initializes the seeking policy. (b) Historical orders reconstruct the order environment $\mathcal{E}_{\text{ord}}$. (c) Shared policy and environment modules generate grouped seek–serve rollouts; remaining-episode returns drive group-relative policy optimization (GRPO), with locality constraints and reference/stopping regularization. The prior p_{θ_0} initializes the same Transformer used by the seeking policy π_θ; order records supply the service distance from which monetary value is constructed.

movements; related fleet-level systems additionally model dispatch and operational interactions (Jin et al., 2020; Liang et al., 2022). This creates a trade-off between simulator fidelity and construction burden, and coarse approximations can expose reward-maximizing policies to unrealistic movement opportunities.

Challenges. These limitations expose a central difficulty in learning passenger-seeking policies through online interaction: the policy should be able to explore decisions beyond historical seeking behavior, yet the environment must provide sufficiently realistic consequences for those decisions. For passenger seeking, evaluating these consequences requires two distinct forms of knowledge. The first is *city-movement knowledge*: how vehicle locations evolve through a city and which movement patterns are consistent with observed travel. The second is *passenger-seeking knowledge*: where and when passengers are likely to be encountered, what service outcomes follow a pickup, and how passenger destinations affect subsequent operating states. Explicitly encoding both forms of knowledge in an interactive simulator imposes a substantial modeling burden, whereas simplifying them too aggressively can introduce mismatches that a reward-maximizing policy may exploit (Pan et al., 2022). The key challenge is therefore to enable policy-dependent exploration while reducing the urban dynamics that must be explicitly specified by the simulator.

Proposed method. We address this challenge with TaxiGPT (Figure 1). Our starting observation is that the knowledge needed for passenger seeking need not come from a single source or be learned in one stage. First, **city-movement knowledge** is represented by a mobility prior learned from serving trajectories. These trips do not demonstrate where a vacant vehicle should search, but do record physically realized movement. Next-location prediction learns movement preferences without requiring labels of successful seeking decisions. Second, **passenger-seeking knowledge** is reconstructed from historical orders: pickup times and locations characterize demand, while passenger destinations, service durations, and trip distances determine subsequent states and modeled service value. The resulting order-derived environment returns pickup outcomes, service transitions, and rewards rather than reproducing the full urban operating system. Third, the **seeking policy** is learned through simulator-based online RL. Starting from the mobility prior, TaxiGPT generates alternative seeking routes, interacts with the order-derived environment, and uses truncated group-relative policy optimization (GRPO) to update route probabilities according to returns from successive seeking and serving cycles.

Pretraining reduces reliance on engineered movement behavior, but reward optimization can erode that prior. We therefore apply locality during adaptation and inference, with reference and stopping regularization to moderate drift. Replay and distance costs provide comparison mechanisms. Locality constrains exploration; the learned prior supplies movement preferences within the allowed neighborhood.

Our main contributions are:

- **Knowledge-decoupled formulation.** We separate mobility learning, order-environment reconstruction, and policy adaptation: serving trajectories provide movement structure, while historical orders provide pickup and service feedback, reducing explicit simulator engineering.
- **Locality-constrained generative RL.** A decoder-only Transformer generates variable-length seeking routes and is adapted with group-relative optimization over seek–serve returns; locality, reference, and stopping regularization preserve movement structure during reward optimization.
- **Shenzhen evaluation.** Comparisons with five passenger-seeking baselines and controlled ablations show gains across all reported operational metrics and isolate the effects of pretraining, order-derived adaptation, reward horizon, and adaptation-time locality.

2 Related Work

Passenger-seeking policies. Demand-based systems recommend hotspots, road clusters, or routes using historical pickups and expected search costs (Yuan et al., 2011; Wang et al., 2015; Qu et al., 2014; Garg and Ranu, 2018; Tran et al., 2022; Liu et al., 2023). Markov decision processes account for downstream service transitions (Shou et al., 2020; Yu et al., 2019), while adversarial imitation learns driver strategies from demonstrations (Ho and Ermon, 2016; Zhang et al., 2022). Offline RL addresses distributional extrapolation (Fujimoto et al., 2019; Kumar et al., 2020), with taxi applications including batch route learning (Yu and Gao, 2022) and CAC (Busaranuvong et al., 2023), but remains limited by logged coverage. Online RL collects policy-dependent experience through grid-based routing (Han et al., 2016; Shi et al., 2019), road-network idle-time reduction (O'Keeffe et al., 2022), and operational taxi simulators (Jin et al., 2020). Related fleet-dispatch systems optimize order assignment (Liang et al., 2022; Zhao and Li, 2026), a distinct decision interface. TaxiGPT assigns movement preferences to serving-trajectory pretraining and pickup/service outcomes to an order-derived environment, reducing movement behavior that must be hand-specified.

Trajectory modeling and constrained reward adaptation. Sequence models learn mobility patterns for prediction and generation (Feng et al., 2018; 2020). In RL, sequence modeling supports return-conditioned policies (Chen et al., 2021), trajectory-based planning (Janner et al., 2021), and online refinement after offline pretraining (Zheng et al., 2022); TrajTok studies discrete vocabularies for driving-behavior generation (Zhang et al., 2026). TaxiGPT learns movement from serving traces without seeking-return supervision, then adapts its grid-token policy using reconstructed order outcomes. Kullback–Leibler (KL) regularization anchors reward adaptation to reference behavior (Teh et al., 2017; Ouyang et al., 2022), while clipped updates (Schulman et al., 2017), critic-free group comparisons (Shao et al., 2024), and action masking (Huang and Ontañón, 2022) provide complementary controls. In TaxiGPT, locality shapes both rollout collection and likelihood-ratio scoring. We compare this placement with inference-only filtering, replay, and distance costs, distinguishing movement plausibility from predictive prior retention. Here "GPT" denotes a causal decoder architecture rather than a natural-language interface.

3 Methodology

We first describe the knowledge decomposition and the seek–serve decision process. We then explain how the mobility prior and order environment are constructed, define the locality-constrained policy, and finally describe its reward-driven adaptation.

3.1 Knowledge decomposition and sequential decision process

TaxiGPT learns city-movement knowledge as a mobility prior p_{θ_0} through next-location and termination prediction on training-date serving trajectories $\mathcal{D}_{\mathrm{mob}}$. Historical orders $\mathcal{D}_{\mathrm{ord}}$ provide

demand and service knowledge through an environment $\mathcal{E}_{\text{ord}}$ that returns pickup outcomes, rewards, and next vacant states. Initialized from θ_0, the same Transformer forms a locality-constrained seeking policy π_θ, which RL adapts using returns from successive seek–serve cycles.

At seeking leg k, the state $s_k = (\ell_k, t_k, w_k)$ records grid location, clock time, and weekday/weekend indicator. The policy proposes $\tau_k = (\ell_{k,1}, \ldots, \ell_{k,L_k}, \text{EOT})$, comprising L_k locations and the route-ending token EOT; execution stops at pickup or the route boundary, followed by service or waiting before the next leg. With M legs, leg reward r_k, and per-leg discount $\gamma \in (0, 1]$, the objective is

$$J(\theta) = \mathbb{E}_{s_1,\, \pi_\theta,\, \mathcal{E}_{\text{ord}}} \left[\sum_{k=1}^{M} \gamma^{k-1} r_k \right]. \tag{1}$$

3.2 Autoregressive mobility pretraining

Trajectory representation. We discretize GPS coordinates into a regular latitude–longitude grid. The input vocabulary contains valid location tokens $\mathcal{V}_{\text{loc}}$, temporal tokens, and EOT. Let $u(t, w)$ encode a 5-minute time bin and the weekday/weekend indicator. A serving trajectory starting at time t_i, with day type w_i and observed locations $\ell_{i,0}, \ldots, \ell_{i,L_i}$, is serialized as $X_i = [u(t_i, w_i),\, \ell_{i,0}, \ell_{i,1}, \ldots, \ell_{i,L_i}, \text{EOT}]$. Segments are packed for training, with EOT marking trajectory boundaries. Each spatial token receives a projected 24-dimensional context vector comprising point-of-interest distances, vehicle presence, pickup volume, and speed. Dynamic statistics are computed only from training dates and frozen for evaluation. Appendix A details the context features and embedding equation.

Autoregressive learning. A decoder-only Transformer (Vaswani et al., 2017) with causal attention, rotary position embeddings, pre-layer normalization, and tied input/output embeddings learns next-token prediction from these sequences. Teacher forcing uses one-token-shifted targets to predict each next location or EOT from the true prefix h. The model produces $p_\theta(\cdot \mid h)$ over valid locations and EOT; a static output mask excludes unobserved cells and temporal tokens, without locality or top-K truncation. For a minibatch $\mathcal{B}_{\text{mob}}$ of packed sequences, let I_X contain location and EOT target positions in sequence X, excluding temporal-token targets. Writing x_j for the target token and $X_{<j}$ for its prefix, we minimize

$$\mathcal{L}_{\text{pre}}(\theta) = -\frac{\sum_{X \in \mathcal{B}_{\text{mob}}} \sum_{j \in I_X} \log p_\theta(x_j \mid X_{<j})}{\sum_{X \in \mathcal{B}_{\text{mob}}} |I_X|}. \tag{2}$$

This token-averaged cross-entropy learns movement and termination preferences without seeking-success labels. The resulting θ_0 initializes the seeking policy, which generates from $[u(t_k, w_k), \ell_k]$ until EOT or the length cap.

3.3 Order environment reconstruction

Demand availability and service outcomes. Historical orders provide pickup time and cell, destination, service duration, and trip distance. The environment represents demand through an effective passenger-waiting window $W = 15$ minutes: an arriving vehicle may match demand that appeared within that window and remains available. Records are indexed by day type w, 15-minute demand bin b, and pickup cell ℓ. Let $N(w, b, \ell)$ be the number of indexed pickups and D_w the number of contributing days of type w. The effective demand count per window and the availability approximation are

$$\lambda(w, b, \ell) = \frac{N(w, b, \ell)}{D_w}, \qquad p_{\text{pick}}(w, b, \ell) = 1 - \exp[-\lambda(w, b, \ell)]. \tag{3}$$

p_{pick} approximates demand availability at vehicle arrival, not new arrivals during a single motion step. Each queried location yields a Bernoulli pickup outcome with this probability. On success, one record from the same index supplies its joint destination, duration, and distance; empty indices return no pickup. The service-value rule below converts the sampled distance into modeled monetary value.

Executing a proposed route. The environment queries the proposed locations in order, advancing the nominal search clock by $\delta = 20$ seconds for each executed location token. Execution stops at the

first pickup or the route boundary. After a pickup, the clock additionally advances by the sampled service duration, and seeking resumes from that trip's destination. If the route ends without a pickup, the vehicle waits for $\omega = 15$ minutes at its terminal location and then starts a new search. This failure-wait interval is distinct from the passenger-availability window W. Only executed location tokens incur the base seeking cost; an unexecuted proposal suffix incurs no travel cost. Let J_k denote the executed token count and $z_k \in \{0, 1\}$ indicate whether leg k obtains a pickup.

Reward and modeling scope. For sampled service distance d_k^{srv} (km), the modeled value v_k (Yuan) and training reward are

$$v_k = \max\{b_0 + (p_{\text{km}} - c_{\text{km}})d_k^{\text{srv}}, 0\}, \qquad r_k = z_k \frac{v_k}{s_v} - \alpha J_k. \tag{4}$$

Here b_0 is the base value, p_{km} and c_{km} are per-kilometer revenue and service cost, $s_v > 0$ scales value to reward units, and α penalizes each executed seeking token. All methods share the evaluation reward; coefficients and the distance-cost variant are specified in Appendix B. The environment reuses historical records and advances movement time at a fixed rate per token, without modeling competing vehicles, road occupancy, or endogenous demand.

3.4 Preserving mobility during RL adaptation

To limit drift from pretrained movement, locality constrains exploration and likelihood-ratio updates, while reference and stopping regularizers anchor model probabilities (Section 3.5).

Locality-constrained action support. Valid cells may still lie far apart. Let $g(\ell)$ denote a cell's integer grid coordinates. For radius R, the Chebyshev neighborhood is

$$\mathcal{A}_R(\ell) = \{\text{EOT}\} \cup \{\ell' \in \mathcal{V}_{\text{loc}} : \|g(\ell') - g(\ell)\|_\infty \le R\}, \qquad R = 10. \tag{5}$$

At the 0.001° resolution, this permits at most 441 spatial candidates, including the current cell, before static masking and map-boundary clipping. For a prefix h ending at ℓ, let $\mathcal{A}(h)$ be this set with EOT removed at the first generation step, ensuring a nonempty search.

From model probabilities to the training policy. For a candidate next token a, the constrained action probability is

$$\pi_\theta(a \mid h) = \frac{p_\theta(a \mid h)\mathbf{1}\{a \in \mathcal{A}(h)\}}{Z_\theta(h)}, \qquad Z_\theta(h) = \sum_{a' \in \mathcal{A}(h)} p_\theta(a' \mid h). \tag{6}$$

Here $\mathbf{1}\{\cdot\}$ is the indicator function. RL samples directly from π_θ without top-K truncation; old- and current-policy scores use the same allowed-set rules and normalization (Huang and Ontañón, 2022). Operational evaluation renormalizes the $K = 16$ most probable allowed actions at temperature one. The no-locality variant removes only the spatial restriction, retaining vocabulary validity and the first-step stopping rule. Appendix D details the decoding conventions.

Preserving preferences within the support. For allowed tokens a, a', renormalization preserves $\pi_\theta(a \mid h)/\pi_\theta(a' \mid h) = p_\theta(a \mid h)/p_\theta(a' \mid h)$. Locality bounds movement; pretraining and RL determine preferences within that support. Applying locality during collection changes route prefixes, pickup queries, and subsequent states used for learning. Inference-only filtering leaves that experience unchanged. RQ3 compares these placements under the same operational test-time support.

3.5 Group-relative reward adaptation

Grouped seek–serve rollouts. From each initial state, the frozen behavior policy $\pi_{\theta_{\text{old}}}$ samples G alternative seek–serve rollouts through $\mathcal{E}_{\text{ord}}$. Candidates share the initial state but may diverge thereafter. Within each group, let g index a rollout, M_g its number of legs, and $r_{g,m}$ its leg-m reward. The reward horizon H determines the leg-k return

$$R_{g,k}^{(H)} = \sum_{m=k}^{\min(M_g, k+H-1)} \gamma^{m-k} r_{g,m}. \tag{7}$$

$H = 1$ credits only the current leg; $H = \infty$ includes the remaining operating episode. Longer-horizon returns therefore account for how passenger destinations and service durations affect subsequent seeking opportunities. Returns at a tracked leg are standardized within the initial-state group:

$$A_{g,k} = \frac{R_{g,k}^{(H)} - \mu_k}{\sigma_k + \varepsilon}, \tag{8}$$

where μ_k and σ_k are the group mean and sample standard deviation of the corresponding returns, and $\varepsilon > 0$ stabilizes this critic-free relative signal (Shao et al., 2024).

Reward horizon versus gradient horizon. Only proposals from the first K_{grad} legs receive direct gradient updates. Later legs are sampled without retaining their computation graphs, but their rewards still contribute to Equation (7). Thus H determines how far ahead a proposal is evaluated, whereas K_{grad} determines how many proposals are directly optimized. The default $G = 8$, $H = \infty$, and $K_{\text{grad}} = 1$ credits the first route with the remaining-episode return. All candidates have that first leg and the same initial state, making the default group comparison well defined.

Clipped policy update. The tracked action is the complete route proposal, including its unexecuted suffix if pickup interrupts execution. Let j index a token within leg k of candidate g, with token $x_{g,k,j}$ and preceding prompt/history $h_{g,k,j}$. All scored tokens of that proposal share $A_{g,k}$. Denote the set of scored token positions in the tracked proposals by $\mathcal{I}$, including their termination tokens. Using the constrained probabilities in Equation (6), the clipped surrogate (Schulman et al., 2017) uses

$$\rho_{g,k,j}(\theta) = \exp[\log \pi_\theta(x_{g,k,j} \mid h_{g,k,j}) - \log \pi_{\theta_{\text{old}}}(x_{g,k,j} \mid h_{g,k,j})], \tag{9}$$

$$\mathcal{L}_{\text{clip}} = -\frac{1}{|\mathcal{I}|} \sum_{(g,k,j)\in\mathcal{I}} \min\{\rho_{g,k,j} A_{g,k}, \text{clip}(\rho_{g,k,j}, 1-\epsilon_c, 1+\epsilon_c) A_{g,k}\}. \tag{10}$$

Here $\epsilon_c > 0$ is the clipping width. Updates pool scored tokens across initial-state groups with equal token weighting; old scores and advantages stay fixed during four optimizer passes per collected batch. Later rewards influence tracked proposals through their return. The finite gradient horizon and tokenwise clipping define a truncated surrogate rather than an exact full-episode gradient.

Reference regularization. A sampled likelihood penalty (Shao et al., 2024) anchors location-token probabilities to p_{ref}. For the multiset $\mathcal{C}_{\text{loc}}$ of scored location prefix–token pairs (h, a), define $\Delta(h, a) = \log p_{\text{ref}}(a \mid h) - \log p_\theta(a \mid h)$ and

$$\mathcal{L}_{\text{ref}} = \frac{1}{|\mathcal{C}_{\text{loc}}|} \sum_{(h,a)\in\mathcal{C}_{\text{loc}}} \left[e^{\Delta(h,a)} - \Delta(h, a) - 1\right]. \tag{11}$$

The reference starts from the run's initialization and is periodically refreshed from the current model. It limits drift relative to a recent model, rather than enforcing permanent equality to the original pretrained prior.

Stopping regularization and total objective. A separate model p_{init}, frozen at the run's initialization, anchors the probability of ending a route. Let $\mathcal{H}_{\text{stop}}$ be the multiset of scored decision prefixes, excluding the first forced-nonstop decision. Writing $\text{Bern}(q)$ for the binary stop/continue distribution $(q, 1-q)$, we use the Kullback–Leibler divergence D_{KL} to define

$$\mathcal{L}_{\text{stop}} = \frac{1}{|\mathcal{H}_{\text{stop}}|} \sum_{h\in\mathcal{H}_{\text{stop}}} D_{\text{KL}}(\text{Bern}(p_{\text{init}}(\text{EOT} \mid h)) \,\|\, \text{Bern}(p_\theta(\text{EOT} \mid h))). \tag{12}$$

Unlike the refreshed reference, this fixed anchor retains the initial stopping preference throughout adaptation. Both auxiliary penalties operate on vocabulary-valid model probabilities p_θ, before locality and top-K renormalization; the clipped policy ratio operates on π_θ. With nonnegative weights β_{ref} and β_{stop}, the overall objective is

$$\mathcal{L}_{\text{TaxiGPT}} = \mathcal{L}_{\text{clip}} + \beta_{\text{ref}}\mathcal{L}_{\text{ref}} + \beta_{\text{stop}}\mathcal{L}_{\text{stop}}. \tag{13}$$

The default run initializes all three models from θ_0; the scratch control initializes the actor and its references randomly. Appendix B gives the training procedure, coefficients, and replay and distance-cost alternatives.

4 EXPERIMENTS

We examine three questions. **RQ1: Overall effectiveness.** How does TAXIGPT compare with passenger-seeking baselines? **RQ2: Component contributions.** What do mobility pretraining, order-derived adaptation, and the reward horizon contribute? **RQ3: Mobility preservation.** How do locality during RL, inference-only filtering, replay, and distance costs trade off utility and mobility?

4.1 EXPERIMENTAL PROTOCOL

Data and comparisons. All experiments use Shenzhen taxi data. Mobility pretraining uses only serving trajectories from training dates; dynamic context covariates are also constructed exclusively from those dates. Validation and test order environments use July 23–25 and July 29–31, 2016, respectively. Test initial states are real seeking starts from the test dates. We run five independent training seeds and report mean results under shared test dates, initial states, and scenario counts. Appendix A specifies the data representation.

Table 1 compares CAC (Busaranuvong et al., 2023), cGAIL (Zhang et al., 2022), local-action PPO (Schulman et al., 2017), DQN-TNC (Shi et al., 2019), and RL-Idle-A2C (O'Keeffe et al., 2022). DQN-TNC and RL-Idle-A2C add domain-specific value-based and actor–critic online-RL comparisons under the shared grid and order environment. All methods use the same monetary rules; domain baselines retain their adapted local action interfaces. Appendix E details these adaptations.

Operational metrics. **Profit** (Yuan/episode) averages cumulative modeled profit from each start to the operating-episode boundary. **Mean profit** (abbreviated **M. profit**, Yuan/leg) averages profit per attempted seeking leg. **EE** (Yuan/h) divides total profit by seeking plus serving time, excluding failure waiting. **SR** (%), **Seek** (min), and **Dist.** (km) measure pickup success, seeking time, and executed distance, pooling successful and failed legs. Transformer policies use top-K sampling during operational evaluation.

Mobility metrics and evaluation. **NLL** (nats/token) is the held-out negative log-likelihood of real next locations given real prefixes; **ΔNLL** (nats/token) measures its change from pretraining. **JSD** (Jensen–Shannon divergence, nats) (Lin, 1991) compares empirical relative grid-displacement distributions, while **Step-W1** (m) (Vallender, 1974) compares single-step haversine-distance distributions. **Jump** (%) measures the fraction of test-decoded transitions exceeding the common locality-equivalent bound of 1,512.97 m per token. Higher Profit, Mean profit, EE, and SR are preferred; lower values are preferred for all other metrics. Bold values indicate the best result in each column, including ties.

NLL is evaluated on 500 held-out serving trajectories, always without locality or top-K filtering. JSD and Step-W1 use 200 held-out starts with five free continuations per start and no order-environment interaction. The complete TAXIGPT model uses its native $R = 10$ generation support in both Tables 2 and 3; the other reported free-generation diagnostics use no locality. In Table 3, Test-only locality reuses the no-locality checkpoint and its unmasked diagnostic results, although its operational evaluation applies locality. Appendix C gives the metric definitions and aggregation rules.

4.2 RQ1: OVERALL PASSENGER-SEEKING EFFECTIVENESS

Table 1: Shenzhen passenger-seeking performance. All methods use the shared order environment and monetary rules.

Method	Profit ↑	Mean profit ↑	EE ↑	SR ↑	Seek ↓	Dist. ↓
CAC	207.48	11.02	36.98	43.34	12.74	3.759
cGAIL	186.42	13.15	33.51	47.25	18.04	3.464
PPO	31.62	8.93	3.47	54.52	143.40	43.211
DQN-TNC	87.80	17.49	9.93	72.51	95.79	29.272
RL-Idle-A2C	33.73	8.88	3.70	55.00	137.43	8.331
TAXIGPT	**540.91**	**21.93**	**117.53**	**75.84**	**2.90**	**1.272**

Strong utility across baseline families. TAXIGPT is best on all six operational metrics in Table 1. It raises full-day profit from the strongest baseline value of 207.48 Yuan (CAC) to 540.91 Yuan (+160.7%) and EE from 36.98 to 117.53 Yuan/hour (+217.8%). It also reduces average seeking time from CAC's 12.74 minutes to 2.90 minutes (-77.2%) and seeking distance from the strongest baseline value of 3.464 km (cGAIL) to 1.272 km (-63.3%).

The result is not driven only by faster termination. DQN-TNC is the strongest baseline in Mean Profit and SR, with 17.49 Yuan/leg and 72.51%, whereas TAXIGPT reaches 21.93 Yuan/leg and 75.84% while completing searches much more efficiently. DQN-TNC and RL-Idle-A2C require 95.79 and 137.43 minutes of seeking per leg, respectively. Together with the CAC, cGAIL, and PPO comparisons, these results support RQ1: combining a learned mobility prior with order-derived adaptation produces efficient repeated seek–serve operation across optimization-, imitation-, value-based-, and actor–critic baseline families. Because the baselines retain different action interfaces, Table 1 compares complete passenger-seeking methods rather than isolating the RL optimizer.

4.3 RQ2: PRETRAINING AND REWARD ADAPTATION

Table 2 evaluates three changes to the complete method: *w/o RL* keeps the pretrained prior frozen, *w/o pretraining* initializes RL randomly, and $H = 1$ uses immediate-leg rather than remaining-episode returns. The complete TAXIGPT uses $H = \infty$.

Table 2: Component contributions and mobility-prior quality on Shenzhen.

	Operational performance						Mobility diagnostics			
Variant	Profit ↑	M. profit ↑	EE ↑	SR ↑	Seek ↓	Dist. ↓	Jump ↓	NLL ↓	JSD ↓	Step-W1 ↓
w/o RL	435.80	21.12	102.85	67.38	5.04	1.435	**0.00**	**1.523**	**0.0139**	76.25
w/o pretraining	96.03	**27.05**	27.08	**95.77**	50.63	117.69	0.68	11.800	0.5851	14565.04
$H = 1$	526.03	22.12	115.63	73.59	**2.17**	1.415	0.01	4.066	0.0509	150.96
TAXIGPT	**540.91**	21.93	**117.53**	75.84	2.90	**1.272**	**0.00**	3.032	0.0169	**72.14**

Mobility pretraining supplies movement structure that task feedback alone does not recover. The frozen pretrained prior (*w/o RL*) achieves a held-out NLL of 1.523, Transition JSD of 0.0139, and Step-W1 of 76.25 m. In contrast, random-initialized RL (*w/o pretraining*) under the same locality rule remains far from real mobility statistics (NLL 11.800, JSD 0.5851, and Step-W1 14,565.04 m). Its high SR (95.77%) and Mean Profit (27.05 Yuan/leg) come with 50.63 minutes and 117.69 km of seeking per leg, yielding only 96.03 Yuan in full-day profit. This directly supports the first part of RQ2: locality bounds the action support, but serving-trajectory pretraining supplies the fine-grained movement and stopping preferences needed for efficient operation.

Order-derived adaptation turns the mobility prior into a seeking policy. Starting from the frozen prior, full TAXIGPT raises full-day profit from 435.80 to 540.91 Yuan (+24.1%), EE from 102.85 to 117.53 Yuan/hour (+14.3%), and SR from 67.38% to 75.84%, while reducing seeking time from 5.04 to 2.90 minutes and distance from 1.435 to 1.272 km. Teacher-forced NLL increases from 1.523 to 3.032, showing that reward adaptation changes the original next-location distribution. At the same time, the complete policy's deployment-matched free rollouts remain close to real movement statistics, with Transition JSD $= 0.0169$ and Step-W1 $= 72.14$ m. The combination is consistent with the intended decomposition: pretraining provides city-movement structure, whereas interaction with the order-derived environment changes which plausible searches are preferred for passenger-seeking return without discarding the movement structure needed at deployment.

Long-horizon feedback improves cumulative operation and reduces myopic drift. Relative to $H = 1$, the full remaining-episode horizon increases full-day profit from 526.03 to 540.91 Yuan (+2.8%), SR from 73.59% to 75.84%, and EE from 115.63 to 117.53 Yuan/hour, while reducing seeking distance from 1.415 to 1.272 km. The myopic variant terminates searches faster (2.17 versus 2.90 minutes) and has slightly higher Mean Profit per leg (22.12 versus 21.93 Yuan), but its teacher-forced NLL is higher (4.066 versus 3.032), indicating greater predictive drift from the pretrained prior. Thus RQ2 supports using downstream seek–serve consequences when adapting the prior: the

full horizon improves cumulative operation and better preserves next-location predictive structure than immediate-leg feedback.

4.4 RQ3: MOBILITY PRESERVATION DURING ADAPTATION

Table 3 compares TAXIGPT with *w/o locality*, which omits the constraint throughout; *Test-only locality*, which adds it only at inference; and *Replay* and *Distance penalty*, which use these preservation mechanisms instead of locality. All variants start from mobility pretraining.

Table 3: Locality and alternative preservation mechanisms.

	Operational performance						Mobility diagnostics			
Variant	Profit ↑	M. profit ↑	EE ↑	SR ↑	Seek ↓	Dist. ↓	Jump ↓	ΔNLL ↓	JSD ↓	Step-W1 ↓
w/o locality	**629.47**	**24.95**	**125.56**	79.59	2.89	1.670	0.78	+1.815	0.0277	**52.35**
Test-only locality	512.18	21.49	125.08	68.65	2.70	1.365	0.02	+1.815	0.0277	**52.35**
Replay	355.20	24.50	87.38	**89.17**	6.95	1.852	0.09	**+0.216**	0.0335	104.40
Distance penalty	544.43	21.68	122.83	77.34	**2.07**	1.273	0.60	+1.605	0.0312	68.31
TAXIGPT	540.91	21.93	117.53	75.84	2.90	**1.272**	**0.00**	+1.509	**0.0169**	72.14

Learning within the deployment constraint. The unconstrained policy earns the highest simulator profit, but adding locality only at inference lowers profit and pickup success. The central comparison is therefore whether the policy learns under the support used at deployment. Under identical test-time locality, adaptation-time locality improves profit by 5.6% and pickup success by 7.19 percentage points over Test-only locality, while reducing seeking distance. Test-only locality retains shorter seeking time and higher EE. Together, these results support exposing the policy to locality during adaptation: the mobility prior supplies movement preferences, and order-derived RL learns how to use them within the deployment constraint.

Mobility preservation is multi-faceted. Mobility replay best preserves teacher-forced likelihood ($\Delta\text{NLL} = +0.216$), but its full-day profit falls to 355.20 Yuan and its native-support free rollouts have the largest Transition JSD and Step-W1 among the preservation variants (0.0335 and 104.40 m). Distance-cost adaptation retains similar utility to TAXIGPT (544.43 versus 540.91 Yuan) and has Step-W1 = 68.31 m, but still permits 0.60% large transitions. In contrast, locality during RL drives Jump to 0.00%, attains the lowest Transition JSD in Table 3 (0.0169), and keeps Step-W1 at 72.14 m. Thus the diagnostics capture complementary notions of preservation: replay most strongly anchors teacher-forced next-token likelihood, whereas adaptation-time locality provides explicit transition control while maintaining realistic deployed rollout statistics.

5 CONCLUSION

TAXIGPT separates passenger seeking into three learning roles: serving trajectories provide a mobility prior, historical orders provide pickup and service feedback, and locality-constrained RL adapts the prior for seeking return. On Shenzhen, the resulting policy outperforms five passenger-seeking baselines across all reported operational metrics. RQ2 shows why the decomposition matters: random-initialized RL under the same locality constraint does not recover realistic movement, order-derived adaptation substantially improves the frozen prior's operating utility, and the full seek–serve reward horizon improves cumulative outcomes while showing less teacher-forced predictive drift than myopic feedback. RQ3 further shows that locality is more useful when it shapes the experience used for adaptation than when it is added only at inference. Under its native deployment support, the complete policy has a reported large-transition rate of 0.00% and achieves the lowest Transition JSD among the preservation variants while maintaining competitive step-length fidelity. These results support learning city movement from realized trajectories and passenger-seeking preferences from order-derived interaction, reducing the amount of urban movement behavior that must be explicitly engineered into the simulator.

REPRODUCIBILITY STATEMENT

Section 3 specifies tokenization, the order-derived environment, normalized training policy, and optimization objective. Appendices A–C describe the training-date data construction, model configuration, and metric definitions. Appendices D–E provide evaluation conventions and baseline adaptations. Training uses five independent seeds and the paper reports their mean results. All operating columns for a given run are computed from its chained seek–serve evaluation.

ETHICS STATEMENT

Urban trajectories can reveal sensitive locations and routines (de Montjoye et al., 2013). This work evaluates aggregate behavior in simulation, not decisions deployed to passengers or drivers. Any release should exclude personally identifying trajectory records. Operational deployment would require privacy safeguards, geographic service-equity evaluation, and controls on how profit optimization changes access to transportation. Aggregate profit and movement plausibility alone are insufficient deployment criteria.

AI USE STATEMENT

Generative AI assisted with manuscript drafting and polishing, code implementation, and debugging of experimental code. The authors are responsible for the final paper and the reported results.

REFERENCES

Palawat Busaranuvong, Xin Zhang, Yanhua Li, Xun Zhou, and Jun Luo. CAC: Enabling customer-centered passenger-seeking for self-driving ride service with conservative actor-critic. In *IEEE International Conference on Data Mining*, 2023.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision Transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems*, 2021.

Yves-Alexandre de Montjoye, Cesar A. Hidalgo, Michel Verleysen, and Vincent D. Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific Reports*, 3:1376, 2013.

Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. DeepMove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 World Wide Web Conference*, pp. 1459–1468, 2018. doi:10.1145/3178876.3186058.

Jie Feng, Zeyu Yang, Fengli Xu, Haisu Yu, Mudan Wang, and Yong Li. Learning to simulate human mobility. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 3426–3433, 2020. doi:10.1145/3394486.3412862.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2052–2062, 2019. https://proceedings.mlr.press/v97/fujimoto19a.html.

Nandani Garg and Sayan Ranu. Route recommendations for idle taxi drivers: Find me the shortest route to a customer! In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1425–1434, 2018. doi:10.1145/3219819.3220055.

Miyoung Han, Pierre Senellart, Stéphane Bressan, and Huayu Wu. Routing an autonomous taxi with reinforcement learning. In *ACM International Conference on Information and Knowledge Management*, pp. 2421–2424, 2016. https://doi.org/10.1145/2983323.2983379.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016. https://proceedings.neurips.cc/paper/2016/hash/cc7e2b878868cbae992d1fb743995d8f-Abstract.html.

Shengyi Huang and Santiago Ontañón. A closer look at invalid action masking in policy gradient algorithms. In *The International FLAIRS Conference Proceedings*, volume 35, 2022. https://doi.org/10.32473/flairs.v35i.130584.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. In *Advances in Neural Information Processing Systems*, volume 34, pp. 1273–1286, 2021. https://proceedings.neurips.cc/paper/2021/hash/099fe6b0b444c23836c4a5d07346082b-Abstract.html.

Kun Jin, Wei Wang, Xuedong Hua, and Wei Zhou. Reinforcement learning for optimizing driving policies on cruising taxis services. *Sustainability*, 12(21):8883, 2020. https://doi.org/10.3390/su12218883.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020. https://arxiv.org/abs/2005.01643.

Enming Liang, Kexin Wen, William H. K. Lam, Agachai Sumalee, and Renxin Zhong. An integrated reinforcement learning and centralized programming approach for online taxi dispatching. *IEEE Transactions on Neural Networks and Learning Systems*, 33(9):4742–4756, 2022.

Jianhua Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991. https://doi.org/10.1109/18.61115.

Junming Liu, Mingfei Teng, Weiwei Chen, and Hui Xiong. A cost-effective sequential route recommender system for taxi drivers. *INFORMS Journal on Computing*, 35(5):1098–1119, 2023. doi:10.1287/ijoc.2021.0112.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. https://openreview.net/forum?id=Bkg6RiCqY7.

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*, 518:529–533, 2015. https://doi.org/10.1038/nature14236.

Kevin O'Keeffe, Sam Anklesaria, Paolo Santi, and Carlo Ratti. Using reinforcement learning to minimize taxi idle times. *Journal of Intelligent Transportation Systems*, 26(4):498–509, 2022. https://doi.org/10.1080/15472450.2021.1897803.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. In *International Conference on Learning Representations*, 2022. https://openreview.net/forum?id=JYtwGwIL7ye.

Meng Qu, Hengshu Zhu, Junming Liu, Guannan Liu, and Hui Xiong. A cost-effective recommender system for taxi drivers. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 45–54, 2014. doi:10.1145/2623330.2623668.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Dian Shi, Jiahao Ding, Sai Mounika Errapotu, Hao Yue, Wenjun Xu, Xiangwei Zhou, and Miao Pan. Deep Q-network-based route scheduling for TNC vehicles with passengers' location differential privacy. *IEEE Internet of Things Journal*, 6(5):7681–7692, 2019. https://doi.org/10.1109/JIOT.2019.2902815.

Zhenyu Shou, Xuan Di, Jieping Ye, Hongtu Zhu, Hua Zhang, and Robert Hampshire. Optimal passenger-seeking policies on e-hailing platforms using Markov decision process and imitation learning. *Transportation Research Part C*, 111:91–113, 2020. https://doi.org/10.1016/j.trc.2019.12.005.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced Transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. https://arxiv.org/abs/2104.09864.

Yee Teh, Victor Bapst, Wojciech M. Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. https://proceedings.neurips.cc/paper/2017/hash/0abdc563a06105aee3c6136871c9f4d1-Abstract.html.

Duy Hoang Tran, Pieter Leyman, and Patrick De Causmaecker. Adaptive passenger-finding recommendation system for taxi drivers with load balancing problem. *Computers & Industrial Engineering*, 169:108187, 2022. https://doi.org/10.1016/j.cie.2022.108187.

S. S. Vallender. Calculation of the Wasserstein distance between probability distributions on the line. *Theory of Probability & Its Applications*, 18(4):784–786, 1974. https://doi.org/10.1137/1118101.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Ran Wang, Chi-Yin Chow, Yan Lyu, Victor C. S. Lee, Sam Kwong, Yanhua Li, and Jia Zeng. TaxiRec: Recommending road clusters to taxi drivers using ranking-based extreme learning machines. In *ACM SIGSPATIAL*, 2015. https://doi.org/10.1145/2820783.2820843.

Xinlian Yu and Song Gao. A batch reinforcement learning approach to vacant taxi routing. *Transportation Research Part C: Emerging Technologies*, 139:103640, 2022. doi:10.1016/j.trc.2022.103640.

Xinlian Yu, Song Gao, Xianbiao Hu, and Hyoshin Park. A Markov decision process approach to vacant taxi routing with e-hailing. *Transportation Research Part B: Methodological*, 121:114–134, 2019. doi:10.1016/j.trb.2018.12.013.

Jing Yuan, Yu Zheng, Liuhang Zhang, Xing Xie, and Guangzhong Sun. Where to find my next passenger? In *Proceedings of the 13th International Conference on Ubiquitous Computing*, pp. 109–118, 2011. doi:10.1145/2030112.2030128.

Xin Zhang, Yanhua Li, Xun Zhou, and Jun Luo. cGAIL: Conditional generative adversarial imitation learning – an application in taxi drivers' strategy learning. *IEEE Transactions on Big Data*, 8(5):1288–1300, 2022.

Zhiyuan Zhang, Xiaosong Jia, Guanyu Chen, Qifeng Li, Zuxuan Wu, Yu-Gang Jiang, and Junchi Yan. TrajTok: What makes for a good trajectory tokenizer in behavior generation? In *International Conference on Learning Representations*, 2026.

Zijian Zhao and Sen Li. Triple-BERT: Do we really need MARL for order dispatch on ride-sharing platforms? In *International Conference on Learning Representations*, 2026.

Qinqing Zheng, Amy Zhang, and Aditya Grover. Online Decision Transformer. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27042–27059, 2022. https://proceedings.mlr.press/v162/zheng22c.html.

A SHENZHEN DATA REPRESENTATION AND EVALUATION

Distinct data roles. Serving trajectories provide mobility-pretraining targets; historical passenger-trip records parameterize the order-derived environment; and real seeking starts initialize evaluation episodes. Mobility pretraining does not use vacant trajectories. Serving movements describe physically observed routes without serving as demonstrations of a vacant driver's seeking choices. The components may share underlying training-date trips, but optimize different objectives.

Spatial and temporal representation. The Shenzhen grid uses 0.001° cells and spans latitude 22.479883–22.864067 and longitude 113.783216–114.364555. Grid indices are row-major, with 224,070 grid positions before the observed-cell mask. The vocabulary additionally includes 576 time tokens and one boundary token. Time tokens use 5-minute bins with separate weekday and weekend ranges; passenger demand is indexed in 15-minute bins. Geographic distances are computed from decoded grid-cell coordinates using the haversine formula.

Data partitions. July 23–25, 2016 and July 29–31 define validation and test dates, respectively; the remaining July dates form training. Mobility pretraining uses only serving trajectories from training dates, and training demand is reconstructed only from those dates' trip records. Validation and test records parameterize their respective evaluation environments. The departure-cell histogram, observed-location vocabulary, and dynamic context statistics are estimated from training data and remain fixed at evaluation. Serving sequences reserved for the mobility holdout are excluded from pretraining and replay; the diagnostics use a common set of 500 held-out sequences and 200 held-out start contexts.

Context covariates. For the context vector $f(\ell, t, w) \in \mathbb{R}^{24}$ at location ℓ, time t, and day type w, we concatenate 21 Manhattan grid distances to points of interest with mean run-start presence, mean pickup count, and mean segment speed at the cell, 5-minute bin, and day type. Presence is aggregated from training-date run-start observations. Pickup and speed statistics likewise use only training dates. Segment speeds above 40 m/s are excluded during covariate construction. The decoder-only Transformer uses causal self-attention (Vaswani et al., 2017), rotary position embeddings (Su et al., 2021), pre-layer normalization, and tied input/output embeddings. For a spatial token, the context is injected as

$$e_j = E[x_j] + \mathbf{1}\{x_j \in \mathcal{V}_{\mathrm{loc}}\}\, W_f f(x_j, t_j, w_j). \tag{14}$$

Here x_j is the input token at position j, t_j and w_j are its associated time and day type, e_j is its input embedding, E is the learned token-embedding matrix, and W_f is a learned linear projection. The context contribution is zero for temporal and boundary tokens.

Start states and order-environment interaction. Operating episodes sample real seeking-start states from July 29–31 within 08:00–20:00. Training prompts sample locations from the training departure histogram and times from the operating interval. Validation uses a fixed set of 512 prompts. Each demand query samples a pickup outcome and, upon success, the associated destination, duration, and service distance from the indexed historical orders.

B OPTIMIZATION PROCEDURE AND CONFIGURATION

Table 4 lists the training configuration. Here B is the number of initial states collected per processing rank, each replicated into a group of G candidate rollouts.

One adaptation iteration.

1. Sample B initial states and replicate each G times. Freeze a behavior checkpoint for collection and initialize a candidate-specific environment clock.
2. Autoregressively sample a route from $\pi_{\theta_{\mathrm{old}}}$ using the shared valid-action and locality normalization, without top-K truncation. Execute its prefix until pickup or route termination; record reward and advance through the serving or waiting phase.
3. Continue seek–serve cycles to the operating boundary. Retain route tensors and masks only for the first K_{grad} legs; retain scalar rewards for all legs included by H.

Table 4: **Shenzhen model and adaptation configuration.** The complete method uses mobility pretraining, locality, and the full remaining-episode reward horizon.

Quantity	Value	Quantity	Value
Transformer layers	12	Hidden width	768
Attention heads	12	Head dimension	64
Feed-forward expansion	4	Dropout	0
Pretraining context	4,096	Grid resolution	0.001°
Initial states per rank, B	8	Candidates per state, G	8
Maximum generated length	1,000	Test top-K	16
RL top-K truncation	None	Training seeds	5
Temperature	1.0	Locality radius, R	10 cells
Tracked legs, K_{grad}	1	Reward discount, γ	0.95
Optimizer passes per batch	4	Clipping width, ϵ_c	0.2
Reference coefficient	0.15	Stop coefficient	0.3
Reference refresh	200 batches	RL learning rate	4×10^{-5}
AdamW weight decay	0.1	Gradient-norm cap	1.0
Microbatch per rank	4	Reward horizon, H	∞
Validation prompts	512	Nominal token duration	20 s

4. Compute reward-to-go and group-normalized advantages. Cache normalized old-policy log-probabilities on the collected prefixes, together with the auxiliary reference scores.
5. Reuse the trajectories for four optimizer passes. Recompute current-policy log-probabilities with the same valid-action and locality rules as collection, apply the clipped surrogate and auxiliary penalties, and update model parameters. Replay variants additionally perform a separate next-token step after an RL step.
6. Evaluate periodically on the fixed validation prompts. The general reference is refreshed every 200 collection batches; the stopping reference remains frozen at the run's initialization.

Training details. Pretraining uses AdamW (Loshchilov and Hutter, 2019) with peak learning rate 3×10^{-4}, minimum 3×10^{-5}, weight decay 0.1, a 200-step warmup, and a 50,000-step decay schedule. Spatial and boundary prediction losses are pooled over valid targets. RL uses peak learning rate 4×10^{-5}, 40-step warmup, and an 18,000-step decay schedule. The moving-reference refresh interval is counted in collection batches, not optimizer steps. Log-likelihood differences in the surrogate and reference penalty are clamped to $[-10, 10]$ for numerical stability. The random-stop augmentation is disabled for RL collection.

Route scoring and stopping. The tracked action is the route proposal, including the unexecuted suffix when pickup occurs early. Only its executed prefix incurs seeking cost; scored proposal tokens receive the route advantage. The first-step prohibition of EOT is part of the normalized action distribution for both collection and scoring. At the length cap, generation appends a final boundary marker; this forced closure is distinct from an EOT sampled by the policy. The auxiliary reference penalty and stop regularizer use full valid-vocabulary model probabilities, while the clipped ratio uses normalized behavior-policy probabilities. Scratch runs initialize the actor and both reference models randomly.

Reward coefficients. Equation (4) uses $b_0 = 10$, $p_{\mathrm{km}} = 2.7$, $c_{\mathrm{km}} = 0.05$, $s_v = 15$, and $\alpha = 0.0008$. These define distance-derived service value rather than observed fares. The base seeking charge is αJ_k in normalized reward units. Distance-cost training substitutes the distance charge described below; operational evaluation uses the common token-based charge.

Alternative preservation mechanisms. The replay variant interleaves RL updates with next-token updates on pretraining trajectories outside the mobility holdout, using replay weight 0.3 and no distance penalty or locality constraint. The distance-cost variant uses a coefficient of 0.01 per kilometer in normalized reward, without replay or locality. The distance charge sums consecutive generated-location displacements, excluding the prompt-to-first-generated-location segment. All compared policies use the same base token-cost rule during operational evaluation.

C METRIC DEFINITIONS

Notation and aggregation. Let episode $n \in \{1, \dots, N\}$ contain M_n seeking legs. For leg k, let $u_{n,k}$ be monetary profit in Yuan under the common evaluation reward, $z_{n,k} \in \{0, 1\}$ indicate pickup, $T^{\text{seek}}_{n,k}$ be executed seeking time in seconds, and $T^{\text{srv}}_{n,k}$ the serving duration in seconds, set to zero for unmatched legs. Let $D^{\text{seek}}_{n,k}$ be executed seeking distance in kilometers. Write $M_{\text{all}} = \sum_n M_n$. All per-leg statistics pool legs across episodes before division.

Full-day and per-leg profit. We report

$$\text{FullDayProfit} = \frac{1}{N} \sum_{n,k} u_{n,k}, \qquad \text{MeanProfit} = \frac{1}{M_{\text{all}}} \sum_{n,k} u_{n,k}. \tag{15}$$

The first measures cumulative profit per simulated operating episode; the second measures average profit per attempted search. Both use the monetary scale in Equation (4). An episode starts at its sampled state and chains seeking, serving, and failure waiting until the clock passes 20:00. Full-day Profit and Mean Profit describe different operating scales: a policy can complete more legs while earning less per leg.

Pickup success, seeking time, and distance. The pooled operational metrics are

$$\text{SR} = 100 \frac{\sum_{n,k} z_{n,k}}{M_{\text{all}}}, \qquad \text{Seek} = \frac{\sum_{n,k} T^{\text{seek}}_{n,k}}{60 M_{\text{all}}}, \qquad \text{Distance} = \frac{\sum_{n,k} D^{\text{seek}}_{n,k}}{M_{\text{all}}}. \tag{16}$$

All three include unsuccessful attempts in the denominator. Seeking time excludes passenger service and the extra waiting interval following a failed search. Executed seeking distance sums haversine distances between consecutive generated grid-cell locations, truncated at the first pickup or at route termination. Displacements after a pickup are not counted as executed search. Distance complements time by distinguishing short spatial movements over many tokens from larger movements over fewer tokens.

Earning efficiency. The rate over seeking and serving time is

$$\text{EE} = 3600 \, \frac{\sum_{n,k} u_{n,k}}{\sum_{n,k} \left(T^{\text{seek}}_{n,k} + T^{\text{srv}}_{n,k} \right)}. \tag{17}$$

Profit and time are summed before taking the ratio; this is not an average of leg-specific hourly rates. Serving time is included only for successful pickups. The failure-wait interval is excluded from the EE denominator, although it advances the operating clock in the chained episode. EE therefore measures earnings per hour of seeking plus serving, while Full-day Profit additionally reflects the number of service opportunities completed within the operating episode.

Large-transition rate. For a generated route τ_i with L_i spatial tokens and token duration $\delta = 20$ seconds, let $d_\star = 1{,}512.97$ meters denote the displacement of a ten-cell diagonal move at the map's center latitude (approximately $22.672°$). The common locality-equivalent speed threshold is $v_\star = d_\star / \delta \approx 75.65$ m/s. We compute

$$\text{Jump} = 100 \, \frac{\sum_i \sum_{j=2}^{L_i} \mathbf{1}\{d_{\text{geo}}(\ell_{i,j-1}, \ell_{i,j}) > d_\star\}}{\sum_i \max(L_i - 1, 0)}. \tag{18}$$

Here d_{geo} is haversine distance in meters. The same threshold is used for all preservation variants. Jump pools transitions of complete generated routes, rather than only the executed prefixes; boundary tokens and prompt-to-first-generated-location transitions are excluded. It measures the frequency of transitions exceeding the shared locality-scale bound, whereas Distance measures actual search travel per attempt.

Held-out NLL and predictive retention. Let $\mathcal{H}$ be the same 500 held-out serving trajectories for all models and I_X^{eval} the scored spatial-token positions. With real prefixes (teacher forcing),

$$\text{NLL}(p_\theta) = -\frac{\sum_{X\in\mathcal{H}}\sum_{j\in I_X^{\text{eval}}}\log p_\theta(x_j \mid X_{<j})}{\sum_{X\in\mathcal{H}}|I_X^{\text{eval}}|}, \tag{19}$$

$$\Delta\text{NLL} = \text{NLL}(p_\theta) - \text{NLL}(p_{\theta_0}), \qquad \text{PPL} = \exp[\text{NLL}(p_\theta)]. \tag{20}$$

For this held-out probe, the evaluator prepends a boundary token and temporal context, scores spatial positions including the initial location, excludes temporal targets, and does not append a terminal boundary target. Locality and top-K filtering are disabled; the static valid-vocabulary mask remains. Logarithms are natural, giving nats per spatial token; PPL denotes perplexity. Absolute NLL evaluates predictive quality; positive ΔNLL denotes deterioration after RL. The absolute NLL values for the untrained model, pretrained prior, and complete method are 11.786, 1.523, and 3.032, respectively. Inference-only locality does not change model parameters and therefore does not change this diagnostic.

Matched free-generation protocol. Free-generation diagnostics use 200 held-out serving-trajectory start contexts, with five autoregressive continuations per context. Start location, time token, and day type match the real trajectory context, and no passenger-matching environment is invoked. In the RQ3 preservation comparison, generation follows each checkpoint's native spatial support: the no-locality, inference-only, replay, and distance-cost checkpoints are generated without locality, while the complete TAXIGPT checkpoint uses the same $R = 10$ locality support under which it was trained and deployed. Because inference-only locality reuses the no-locality checkpoint, their model-level free-generation diagnostics are identical. The complete-method Transition JSD and Step-W1 reported in Table 2 are synchronized with this RQ3 protocol. The five continuations per start are sampling repetitions, distinct from the independent training seeds. Transition JSD pools displacements across the matched contexts; it is not conditioned on time or region.

Transition JSD. For grid coordinates $g(\ell_j) = (r_j, c_j)$, let

$$\mathbf{d}_j = (\Delta r_j, \Delta c_j) = (r_j - r_{j-1},\ c_j - c_{j-1}). \tag{21}$$

Pool consecutive spatial transitions within the real and generated trajectories separately. If $C_{\text{real}}(\mathbf{d})$ and $C_\theta(\mathbf{d})$ count displacement $\mathbf{d}$, their empirical probability masses are

$$P(\mathbf{d}) = \frac{C_{\text{real}}(\mathbf{d})}{\sum_{\mathbf{d}'} C_{\text{real}}(\mathbf{d}')}, \qquad Q_\theta(\mathbf{d}) = \frac{C_\theta(\mathbf{d})}{\sum_{\mathbf{d}'} C_\theta(\mathbf{d}')}. \tag{22}$$

Following Lin (1991), with the union of observed displacement categories and the mixture $U(\mathbf{d}) = [P(\mathbf{d}) + Q_\theta(\mathbf{d})]/2$,

$$\text{TransitionJSD} = \frac{1}{2}\sum_{\mathbf{d}} P(\mathbf{d})\log\frac{P(\mathbf{d})}{U(\mathbf{d})} + \frac{1}{2}\sum_{\mathbf{d}} Q_\theta(\mathbf{d})\log\frac{Q_\theta(\mathbf{d})}{U(\mathbf{d})}. \tag{23}$$

Zero-mass terms contribute zero. Natural logarithms bound JSD in $[0, \log 2]$. Smaller values indicate closer empirical relative-movement distributions. The displacement categories encode both grid direction and grid magnitude. This marginal statistic does not by itself validate road connectivity, conditional next-step decisions, or all long-range trajectory dependencies.

Step-length Wasserstein distance. For every consecutive pair, compute the haversine displacement in meters:

$$d_j = d_{\text{geo}}(\ell_{j-1}, \ell_j). \tag{24}$$

The generated sequence includes the prompt anchor before extracting steps, so the anchor-to-first-generated-location transition is counted. Let F_{real} and F_θ be the empirical cumulative distributions of real and generated step lengths. The one-dimensional Wasserstein-1 distance (Vallender, 1974) is

$$\text{Step-W1} = \int_0^\infty |F_{\text{real}}(d) - F_\theta(d)|\ \mathrm{d}d. \tag{25}$$

For sorted union support $0 \le u_1 < \cdots < u_m$, it can equivalently be computed as

$$\sum_{k=1}^{m-1} |F_{\text{real}}(u_k) - F_\theta(u_k)|\,(u_{k+1} - u_k). \tag{26}$$

Lower values indicate closer single-step movement scales. The unit is meters; unlike Jump, this measure compares the entire distance distribution rather than the frequency of crossing one threshold. Transitions are formed within trajectories and not across unrelated route boundaries. Both excessively large and excessively small movement scales can increase Step-W1.

D EVALUATION CONFIGURATION

Operational protocol. The shared test setup uses real seeking starts from July 29–31, within 08:00–20:00. Each episode chains seeking, serving, and failure waiting to the operating boundary. Full-day Profit, Mean Profit, SR, EE, Seek, and Distance for a configuration use the same chained operating trajectories. Model training is repeated with five independent seeds, and results are reported as means. The complete-method entries in Tables 1–3 refer to the same method configuration.

Training and test decoding. Training samples the normalized valid-action policy π_θ in Equation (6), with no top-K truncation. For operational evaluation, let $S_{\theta,K}(h)$ be the top-K set among the allowed actions. The test decoder is

$$\pi^{\text{test}}_{\theta,K}(a \mid h) = \frac{\pi_\theta(a \mid h)\mathbf{1}\{a \in S_{\theta,K}(h)\}}{\sum_{a' \in S_{\theta,K}(h)} \pi_\theta(a' \mid h)}, \qquad K = 16. \tag{27}$$

The first action cannot be EOT. All Transformer operating variants use temperature one; locality is enabled or disabled according to the variant. Top-K is a test-time decoding choice, not the distribution used in the RL likelihood ratio. Domain baselines instead execute their specified local-action policies.

Preservation comparisons and diagnostic samples. No-locality RL and inference-only locality share model parameters; only operational filtering differs. Replay uses next-token training without a distance cost or locality, whereas distance-cost adaptation uses neither replay nor locality. The complete method applies locality during RL collection and operational decoding. Jump uses 300 generated routes under the respective test decoder. NLL uses 500 real serving trajectories without locality or top-K filtering. Transition JSD and Step-W1 use 200 held-out start contexts with five continuations each and no order interaction; for RQ3 they follow each checkpoint's native spatial support, with $R = 10$ only for the complete locality-trained model. These diagnostic protocols are distinct from pickup-truncated operational evaluation.

E BASELINE ADAPTATIONS AND CONTROLLED COMPARISON

Domain online-RL baselines. DQN-TNC and RL-Idle-A2C extend the overall comparison with value-based and actor–critic approaches developed for taxi routing or idle-time reduction. Their reported variants use the shared Shenzhen grid, order-derived interaction, monetary objective, and chained operating evaluation. They are adaptations rather than full reproductions of their original demand or road-network simulators.

DQN-TNC (adapted). The method is based on the DQN routing approach of Shi et al. (2019). Its state contains normalized position, temporal information, and the 24-dimensional spatial context. A Q-network selects one of four cardinal adjacent-grid actions with exploration during learning, experience replay, and a target network (Mnih et al., 2015). No stay or explicit stop action is included. The original competing-vehicle state component is omitted because the shared environment does not simulate competing vehicles. The adaptation uses the common monetary reward. The full-day evaluator resumes seeking after the sampled service trip.

RL-Idle-A2C (adapted). The baseline instantiates the A2C family evaluated for taxi idle-time reduction by O'Keeffe et al. (2022). An actor–critic uses position, time, and shared context to select among eight adjacent-grid directions. Grid adjacency replaces the original street-graph node interface, and the reported version uses the common monetary objective rather than an idle-time-specific reward. It therefore tests a domain policy-based learner under the same operating task, without claiming to reproduce the original road graph or fleet dynamics.

Comparison scope. DQN-TNC and RL-Idle-A2C preserve their local grid-action interfaces, whereas TAXIGPT generates variable-length routes over its locality-constrained Transformer policy. Table 1 therefore compares complete passenger-seeking methods under a common operating objective rather than identical action spaces; the component controls in Table 2 isolate the contributions of mobility pretraining, order-derived adaptation, and reward horizon within TAXIGPT itself.