BEYOND DISTANCE: LEARNING COURIER BEHAVIOR WITH LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Last-mile delivery is a cornerstone of modern urban logistics, and its efficiency depends heavily on high-quality route planning. Conventional approaches typically optimize for a single geometric objective such as minimizing travel distance or time, as in the classical Traveling Salesman Problem (TSP). Solving such problems to optimality can require substantial computation time as the number of delivery stops increases. However, these methods overlook practical factors that shape real-world delivery, including couriers' behavioral preferences. To address these limitations, we propose the Delivery Optimization and Route Advisor (DORA), a route-planning system built on the compact Qwen3-4B-Instruct model and an adapted Group Relative Policy Optimization (GRPO) algorithm. DORA learns couriers' behavioral patterns from real delivery traces and recommends routes that better reflect couriers' observed delivery patterns. We train and evaluate DORA on LaDe, a publicly available last-mile delivery dataset, and on the Amazon Last Mile Routing Research Challenge (ALMRRC) as well. DORA ranks among the top two evaluated language models in sequence similarity and first in route-length fidelity. DORA is designed to reduce inference costs relative to larger language-model-based approaches, supporting cost-effective deployment for last-mile route prediction.

1 INTRODUCTION

Couriers do not visit stops in the order a distance objective would choose. Real delivery traces from the Amazon Last Mile Routing Research Challenge and LaDe record the visit order couriers follow (Merchán et al., 2024; Wu et al., 2024); on LaDe, a distance-minimizing OR-Tools path agrees with it less than serving stops by acceptance time does (Kendall rank correlation, KRC, 0.132668 versus 0.183315). Our goal is to learn this courier behavior from the traces: given a fixed set of stops, predict the courier's visit order, scored against the recorded route by KRC on LaDe and by the official Amazon Score (AS) on Amazon.

Existing routing systems already capture much of this behavior. Graph2Route and DRL4Route learn delivery-order dependencies and propose complete routes (Wen et al., 2022; Mao et al., 2023), similar past routes expose recurring area orders, and the Amazon challenge winner combines historical ordering with constrained local search (Cook et al., 2024). These sources are useful but disagree with one another, and turning their advice into a better route normally takes hand-designed aggregation or search. Learned combinatorial solvers target objectives such as tour length (Kool et al., 2019; Kwon et al., 2020; Xin et al., 2021), and language-model methods for routing optimize solution quality under a stated objective (Ye et al., 2024; Jiang et al., 2025b;a). None of this work provides a policy that *learns from routing advice* and returns a route that is both executable and consistent with observed behavior.

We introduce DORA (Delivery Optimization and Route Advisor), which learns courier behavior with compact Qwen3-4B language policies (Yang et al., 2025) trained with LoRA (Hu et al., 2022). Each policy reads routing advice and returns executable decisions (Figure 1). On LaDe, the policy reads the task, nine Graph2Route proposals, and two similar earlier routes from the training pool, and generates the complete stop sequence under a remaining-stop mask. Every output is therefore a legal complete permutation without a downstream solver or repair step, and it can differ from every proposal. The policy is first fine-tuned on observed routes, and group-relative reinforcement learning (Shao et al.,

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.

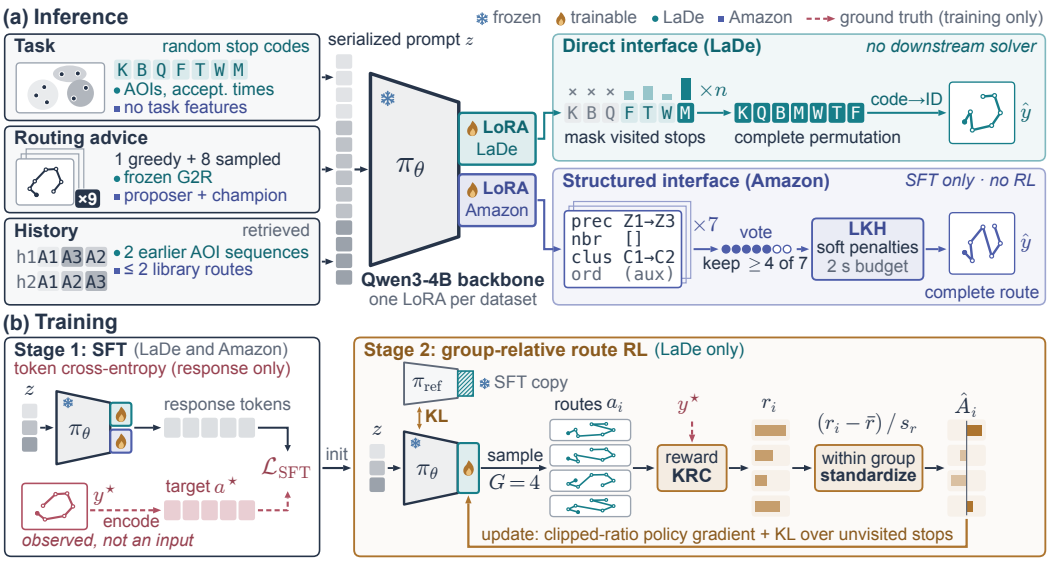


Figure 1: **DORA's two executable interfaces.** (a) On LaDe, the policy reads the task, nine Graph2Route (G2R) proposals, and two similar earlier routes, and generates a complete permutation under a remaining-stop mask. On Amazon, a separate policy reads only candidate zone sequences (nine proposals, up to two retrieved routes, and the champion route) and emits zone and cluster relations; relations backed by at least four of seven draws enter a fixed LKH solver as soft penalties. Each policy is a frozen Qwen3-4B backbone with its own LoRA adapter. (b) SFT trains both policies; LaDe adds group-relative route RL with a whole-route KRC reward. Observed routes $y^\star$ enter only as training targets and rewards.

2024) with a whole-route KRC reward then refines it. On Amazon, a second, supervised policy reads only candidate zone sequences (nine proposals from a fine-tuned zone-order model, up to two retrieved routes from the same station, and the champion route) and emits zone-precedence, neighbor, and cluster-precedence relations. Relations voted by at least four of seven draws enter a fixed LKH solver as soft penalties, so every task again yields a complete route.

On 300 LaDe development tasks, DORA attains the highest mean KRC of all evaluated systems, 0.467738. It is on par with the strongest learned predictor, Graph2Route, and clearly above the historical area-order rule ($+0.091173$, 95% interval $[0.045559, 0.136586]$), distance-based OR-Tools, and local learned-routing baselines. The policy composes new routes instead of selecting among its proposals: across three seeds, 176, 193, and 204 of its 300 routes differ from every one of the nine proposals. The advice accounts for this accuracy: a separately trained task-only adapter reaches KRC 0.00617. Removing sources from a frozen policy at inference shows that it draws primarily on route proposals and uses retrieved history when proposals are missing. Route-level RL gives the best configuration: under the same update schedule it ends above continued token-level supervision at both learning rates, and continued supervision at the native rate lowers all three anchors.

On 3,052 Amazon evaluation routes, DORA's champion-inclusive structured system, whose learned relations an LKH re-solve executes, improves on a full rerun of the challenge-winning champion (official AS, lower is better: -0.000939, 95% interval $[-0.001174, -0.000704]$), and the gain holds on the 2,752 routes excluded from adapter selection. COPY, a deterministic relational encoding of the champion through the same interface, also improves on the champion and is the strongest instantiation (Section 5.4), so the gain comes from the relational interface. Reading only raw candidates, the learned policy produces constraints identical to COPY's on 2,221 of 3,052 routes.

Our contributions are:

- Two executable interfaces for learning from routing advice: masked direct generation, which always returns a legal complete route and composes routes beyond its proposals, and structured relational decisions that a fixed solver executes.

- A route-level training recipe, supervised behavioral initialization followed by group-relative RL on a whole-route KRC reward, which gives the best LaDe configuration among same-anchor controls.
- Results on two real delivery datasets. On LaDe, DORA reaches the highest mean KRC, on par with Graph2Route, and composes new routes from the advice it reads. On Amazon, relational execution of advice improves on a full rerun of the challenge winner, and the learned policy reproduces the deterministic encoding on most routes.

2 Related Work

Courier route prediction and learned routing. The Amazon challenge scores agreement with observed courier sequences, and its winning method builds historical behavior into constrained local search (Merchán et al., 2024; Cook et al., 2024). Graph2Route and DRL4Route learn spatio-temporal delivery-order dependencies and route-level policies from real pick-up and delivery data (Wen et al., 2022; Mao et al., 2023; Wu et al., 2024). Attention models and POMO learn constructive policies for combinatorial objectives such as tour length (Kool et al., 2019; Kwon et al., 2020), NeuroLKH couples learned guidance with local search (Xin et al., 2021), and GOAL shares representations across optimization problems (Drakulić et al., 2025). DORA takes the outputs of these systems as advice and learns to turn that advice into a route, so its main comparisons include the advisers themselves.

Language models for routing and optimization. Language models have entered routing as heuristic designers (Ye et al., 2024), constraint retrievers (Jiang et al., 2025b), and end-to-end solvers for optimization objectives (Jiang et al., 2025a). These systems optimize solution quality under a stated objective. DORA fits observed behavior and learns from the advice of domain systems rather than from problem statements alone. Supervision can align compact language models (Gao et al., 2025); DORA adapts a compact policy with LoRA (Hu et al., 2022), rewards whole routes with a group-relative objective (Shao et al., 2024), and acts through two interfaces, masked generation and soft relational constraints, that make every output executable.

3 Learning from Routing Advice

A task x contains a fixed set of stops V_x with the attributes available to the evaluation protocol. Let $n = |V_x|$, and let $y^\star$ be the observed courier permutation. DORA predicts a complete route $\hat{y}$ over the same stops.

Fixed providers (frozen routing models and a historical library $\mathcal{H}$) produce the routing advice $A = P_d(x, \mathcal{H})$ for domain d. A renderer builds the policy input z, the language policy emits actions a, and a fixed interpretation map executes them:

$$a \sim \pi_{\theta_d}(\cdot \mid z), \qquad z = d_d(x, A), \qquad \hat{y} = E_d(x, a). \tag{1}$$

In the direct interface, a is a complete route and E_d maps local codes back to stop identifiers. In the structured interface, a is a set of relations that an optimizer executes over the task's stops. On Amazon, d_d renders only the candidate zone sequences in A, and the task's geometry enters through E_d.

Each dataset's route metric scores behavioral agreement: Kendall rank correlation (KRC) on LaDe, where larger is better, and the official sequence-and-travel-cost score (AS) on Amazon, where smaller is better (Merchán et al., 2024). We write the utility

$$q_d(\hat{y}, y^\star) = \begin{cases} \mathrm{KRC}(\hat{y}, y^\star), & d = \mathrm{LaDe}, \\ -\,\mathrm{AS}(\hat{y}, y^\star), & d = \mathrm{Amazon}. \end{cases} \tag{2}$$

Route-level RL on LaDe maximizes q_{LaDe}; the Amazon policy is trained with supervision. The observed route supplies training targets and rewards and never enters the prediction input. LaDe history comes from strictly earlier dates, and Amazon history from a fixed build library (Section 5.1, Appendix A).

4 DORA

DORA trains Qwen3-4B-Instruct-2507 policies (Yang et al., 2025) that turn routing advice into executable routing decisions (Figure 1). Advice construction is fixed within every comparison, so changing the policy never silently changes its upstream source.

4.1 Representing Tasks and Routing Advice

LaDe. The prompt contains the task table (a single-token code, relative coordinates, area of interest (AOI), and acceptance minute for each stop), nine proposals from a Graph2Route (G2R) model trained on the full training pool (one greedy route and eight samples), and two earlier routes retrieved from the 22,301-route training pool. Retrieval considers only routes with strictly earlier dates and ranks them by AOI-set Jaccard similarity, then same-courier status, then recency; when fewer than two earlier routes share an AOI with the task, the same ranking is applied to all earlier routes. Retrieved routes can come from any courier. Each is shown as its AOI sequence, with AOIs absent from the current task marked X, so history shows the order of areas and carries no stop identifiers from outside the task. A randomized bijection assigns stop codes, and the same mapping is used in the target; evaluation mappings are fixed across training seeds.

Amazon. The prompt contains only candidate zone sequences: nine proposals from a separately fine-tuned Qwen3-4B zone-order adapter (one greedy route and eight samples), up to two retrieved build routes from the same station rendered in the current task's zone names, and the champion route, always last. Coordinates, stop lists, and travel times do not appear in the prompt; the executor supplies the task's geometry when it compiles the emitted relations. Retrieval is route-level, since the Amazon data carries no driver identity.

4.2 Direct and Structured Decisions

Direct stop generation. On LaDe, the policy chooses the next unvisited stop from its output-head logits. Identifying each stop with its local code and writing $R_t = V_x \setminus \{a_1, \ldots, a_{t-1}\}$ for the remaining set,

$$\pi_\theta(v \mid z, a_{<t}) = \frac{\exp(\ell_\theta(v \mid z, a_{<t})/T)}{\sum_{u \in R_t} \exp(\ell_\theta(u \mid z, a_{<t})/T)}, \qquad v \in R_t, \tag{3}$$

and zero otherwise. Training rollouts sample from this distribution; evaluation selects the highest-scoring remaining stop. The decoder emits exactly n actions with no end token, and each step removes one stop, so every output is a complete permutation of V_x. No completion step or solver runs after the decoder, and the generated route can differ from every input proposal.

Structured decisions. On Amazon, the policy emits explicit zone-precedence, neighbor, and cluster-precedence fields, followed by an auxiliary zone order. Targets come from observed training routes: zone-precedence targets connect zones one or two positions apart, and cluster targets connect adjacent first-visit clusters. A relation enters execution when at least four of seven stochastic draws propose it. The interpretation map compiles admitted relations into an LKH instance with fixed structural penalties and a two-second budget. Relations are soft, so the solver trades them off against the task's travel structure and always returns a complete route. DORA executes relations with a relation-only compiler, specified as the primary configuration before the 3,052-route evaluation; a hybrid compiler that also retains the champion's zone- and super-neighbor transition structure is the secondary configuration. Both compilers are applied identically to the student and to COPY, which encodes the champion route as the same target-style relations without a language model (Appendix D).

4.3 Supervised Behavioral Initialization

For training route $y^\star$, let $a^* = T_d(x, A, y^\star)$ be its direct sequence or structured encoding. We fit a LoRA adapter (Hu et al., 2022) on response tokens so that the policy learns to express observed

behavior through its interface:

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\mathbb{E}_{\mathcal{D}_{\mathrm{tr}}} \frac{\sum_{t=1}^{L} \omega_t \log \pi_\theta(a_t^* \mid z, a_{<t}^*)}{\sum_{t=1}^{L} \omega_t}. \tag{4}$$

Direct actions have unit weights and no end-token action; Appendix B gives the structured output weighting and adapter settings.

4.4 Route-Level Reinforcement Learning

For LaDe, we initialize from the SFT checkpoint and freeze a copy as the reference. Each update samples $G = 4$ complete routes from the current policy and scores each whole route:

$$r_i = \mathrm{KRC}(E_{\mathrm{LaDe}}(x, a_i), y^\star), \qquad \widehat{A}_i = \frac{r_i - \bar{r}}{\max(s_r, \epsilon_A)}, \tag{5}$$

where s_r is the within-group population standard deviation. The reward scores the entire visiting sequence, so every decision receives credit through route-level agreement with observed behavior.

We optimize a GRPO-style group-relative policy-gradient objective with an exact categorical KL penalty to the frozen SFT policy (Shao et al., 2024; Schulman et al., 2017). The KL is computed over the legal support of unvisited stops, which sampling, gradient replay, and the reference distribution share. Losses average over tokens within each route and then over the group, with one optimizer update per sampled group; Appendix B.2 gives the full objective, probability accounting, and pseudocode. Continued-SFT controls start from the same anchors and follow the same update schedule, so they separate the effect of the route reward from that of additional training.

5 Experiments

We first compare complete systems on LaDe (Section 5.2) and analyze where DORA's gains come from (Section 5.3), then evaluate structured decisions on Amazon (Section 5.4).

5.1 Experimental Setup

LaDe. We use static delivery tasks from the Chongqing subset of LaDe (Wu et al., 2024). DORA trains on 512 tasks and is evaluated on a fixed 300-task development cohort covering 186 couriers; its G2R adviser was trained on the 22,301-task pool. We reused this cohort during method development, and 161 of its tasks belonged to the 2,000-task validation probe used to select the adviser's checkpoint. We train three seeds and four branches per supervised anchor: stopping after SFT, route RL, and continued SFT at the RL learning rate (low) and at the original SFT rate (native). Initial SFT runs two epochs (256 updates); every continuation runs 128 updates on a shared task and prompt schedule. References include G2R, proposal aggregation without a language model, historical rules, distance-based routing, and local adaptations of learned routing methods (Appendix E).

Amazon. We use the 6,112 build routes and 3,052 evaluation routes available to us (Appendix A). We select the structured SFT adapter on a 300-route subset and evaluate it frozen; the main comparisons are also reported on the remaining 2,752 routes. Student and COPY share the solver budget and seed within each compiler configuration.

Metrics and uncertainty. We report KRC (↑) on LaDe and official AS (↓) on Amazon. Both measure offline agreement with the recorded courier order, not courier preference or satisfaction. Three-seed LaDe means average the scores of three trained models on the same tasks; LaDe intervals use 10,000 route-paired bootstrap draws, and Amazon intervals use 20,000 (Appendix A).

5.2 DORA Reaches the Top of the LaDe Comparison

Table 1 compares complete systems on the same 300 tasks. DORA SFT + route RL has the highest mean KRC of all evaluated systems, 0.467738. It is clearly above the historical AOI-rank/FIFO rule ($+0.091173$, 95% interval $[0.045559, 0.136586]$), distance-based OR-Tools, and the local learned-routing baselines, and on par with the strongest learned predictor, the saved-full G2R reference ($+0.011965$, $[-0.017591, 0.041090]$).

Table 1: **LaDe route prediction on the same 300 development tasks.** Fixed references have one score; DORA rows average three trained models, and every DORA prediction is a legal complete permutation. Bold marks the highest mean. Baseline provenance appears in Appendix E.

Method	Main information source	KRC ↑
FIFO	Task acceptance times	0.183315
OR-Tools, open-distance path	Geometry, 1 s/route	0.132668
LightGBM, local ranker	Task features	0.164126
DeepRoute / DRL-SCST, local†	Task features	0.138236
Historical AOI rank + FIFO	Task and history	0.376565
DRL4Route, archived local base	Saved prediction	0.226090
G2R, pool greedy	Learned route proposal	0.446406
G2R, saved-full reference	Learned route predictor	0.455773
Borda-of-nine	Nine G2R proposals	0.454727
Medoid-of-nine	Nine G2R proposals	0.446809
DORA SFT	Task, proposals, history	0.461988
DORA SFT + route RL	Task, proposals, history	**0.467738**

†Both local neural adaptations select the same supervised warmup checkpoint.

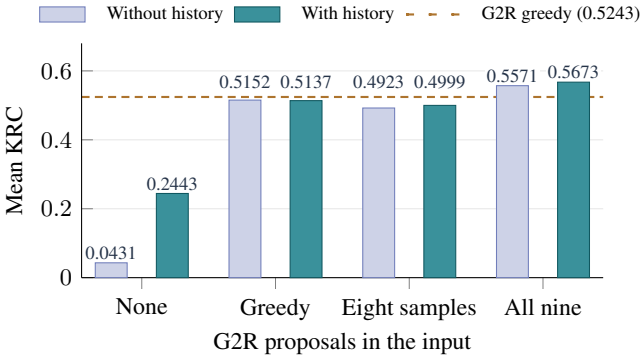


Figure 2: **Proposals drive the policy; history carries signal when they are missing.** Mean KRC of the frozen seed-A route-RL policy on 64 calibration tasks with sources removed at inference; the task table is always present. Appendix C lists every arm, paired intervals, and copy rates.

Routes beyond the proposal set. Across the three seeds, 176, 193, and 204 of DORA's 300 RL outputs differ from every one of the nine proposals it reads. The policy composes these routes from its advice and matches the best proposal-only aggregator, Borda-of-nine, which consumes the same nine proposals without a language model ($+0.013010$, $[-0.013978, 0.039222]$).

5.3 Where the Gains Come From

The policy learns from its advice. A separately trained task-only adapter (same 512 tasks, no proposals or history) reaches KRC 0.00617 on the development cohort (Appendix G). To find which sources the trained policy draws on, we remove them at inference from one frozen seed-A route-RL policy on 64 calibration tasks inside the adviser's training period (Figure 2). With all inputs, the policy reaches 0.5673, above the greedy G2R proposal it receives on the same tasks (0.5243; $+0.0430$, $[0.0101, 0.0791]$). Removing all proposals lowers it by 0.3230 ($[0.2347, 0.4183]$). Removing the eight samples while keeping the greedy proposal lowers it by 0.0535 ($[0.0172, 0.0955]$), so the policy uses the diversity of the proposal set beyond the greedy proposal. Without proposals, history lifts the task-only score from 0.0431 to 0.2443 ($+0.2012$, $[0.0965, 0.3035]$); alongside all nine proposals, its effect is $+0.0102$ ($[-0.0132, 0.0375]$).

Route-level RL gives the best configuration. Table 2 lists each supervised anchor with its continuation branches. From the same anchors and update schedule, route RL raises the mean KRC over SFT by $+0.005750$ (route-paired interval $[-0.000982, 0.012548]$, which includes zero; two of three seeds rise) and ends above continued supervision at the low rate ($+0.007939$, $[-0.000155, 0.016172]$) and at the native rate ($+0.018298$, $[0.002496, 0.033547]$). Continued token-level supervision moves the policy the other way: it lowers the mean at both learning rates and lowers all three anchors at the native rate.

Table 2: **Same-anchor post-training on LaDe (KRC ↑).** A/B/C are training seeds. Every continuation starts from its seed's SFT anchor and runs 128 updates on the same task and prompt schedule.

Training branch	A	B	C	Mean
SFT anchor	0.457743	0.478039	0.450182	0.461988
SFT + route RL	0.470946	0.473336	0.458931	**0.467738**
Continued SFT, low LR	0.454704	0.469661	0.455033	0.459799
Continued SFT, native LR	0.446651	0.457566	0.444103	0.449440

Table 3: **Amazon structured decisions on the 3,052 evaluation routes** (official AS, lower is better). Differences are paired against the full champion rerun, with 95% route-bootstrap intervals. The student is the supervised structured policy; COPY encodes the champion route as the same relations without a language model. Both run with each compiler under the same two-second LKH budget; DORA uses the relation-only compiler.

System	AS ↓	vs. champion	95% interval
Full champion rerun	0.024871	—	—
DORA student + relation-only compiler	0.023932	-0.000939	$[-0.001174, -0.000704]$
COPY + relation-only compiler	**0.023681**	-0.001190	$[-0.001365, -0.001024]$
DORA student + hybrid compiler	0.025123	$+0.000252$	$[0.000089, 0.000423]$
COPY + hybrid compiler	0.024935	$+0.000064$	$[-0.000033, 0.000172]$

5.4 Amazon: Relational Execution of Structured Decisions

Table 3 evaluates structured decisions on the 3,052 Amazon evaluation routes. The champion-inclusive structured system, in which an LKH re-solve executes the student's relations, improves on a full rerun of the challenge-winning champion (-0.000939 AS, $[-0.001174, -0.000704]$). The gain holds on the 2,752 routes excluded from adapter selection (-0.000984, $[-0.001236, -0.000735]$) and against an earlier, independent run of the champion pipeline (-0.000920, $[-0.001153, -0.000685]$). It comes from the relational interface: COPY, which executes the champion route through the same relations without a language model, improves by -0.001190 and scores better than the student. On the 2,752 selection-excluded routes, student minus COPY is $+0.000231$ ($[0.000064, 0.000404]$), and on all routes it is $+0.000251$ ($[0.000093, 0.000423]$). Reading only raw candidates, the student emits the same normalized constraints as COPY on 2,221 of 3,052 routes. These gains are measured with the relation-only compiler that DORA uses; with the hybrid compiler, COPY is on par with the champion ($+0.000064$, $[-0.000033, 0.000172]$) and the student is $+0.000252$ ($[0.000089, 0.000423]$).

6 Conclusion

DORA learns courier behavior from recorded delivery traces. Its compact language policies read the outputs of existing routing systems as advice and turn that advice into executable routes. On LaDe, masked direct generation makes every output a legal complete route. After supervised initialization and route-level RL, the policy reaches the highest mean KRC among evaluated systems, and most of its routes differ from every proposal it reads. It draws primarily on route proposals and uses retrieved history when proposals are missing. On Amazon, executing the advice as soft relational constraints in a fixed solver improves on a full rerun of the challenge winner, and the learned policy reproduces the deterministic encoding of the champion from raw candidates on most routes.

AI USE STATEMENT

Generative AI tools, including Codex and Claude, assisted with research-idea development, methodology and implementation review, code generation, mathematical formulation, literature discovery, result interpretation, diagrams, and manuscript drafting and editing. Every reported number is computed from recorded experiment outputs. The tools are not authors; the human authors verified the AI-assisted content and take full responsibility for the submission.

ETHICS STATEMENT

This study analyzes released delivery datasets under their access conditions and reports aggregate metrics only. Location traces can reveal work routines, so any reuse of these data should protect individual trajectories. A route predictor learns from observed courier behavior. Any operational use should be assessed together with the couriers whose work it models, and its outputs are not intended as worker quotas.

REPRODUCIBILITY STATEMENT

Appendices A–G document data splits, advice construction and prompts, both action interfaces, training objectives and settings, same-anchor controls, the frozen-policy source analysis, uncertainty estimation, and cost measurement. All results are computed from archived predictions and training records, and the appendix states the model revision, software versions, training seeds, evaluation nonce, and bootstrap settings.

REFERENCES

William Cook, Stephan Held, and Keld Helsgaun. Constrained local search for last-mile routing. *Transportation Science*, 58(1):12–26, 2024. doi: 10.1287/trsc.2022.1185. URL https://pubsonline.informs.org/doi/10.1287/trsc.2022.1185. Published online November 8, 2022.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0441. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1feb87871436031bdc0f2beaa62a049b-Abstract.html.

Darko Drakulić, Sofia Michel, and Jean-Marc Andreoli. Goal: A generalist combinatorial optimization agent learner. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 52465–52488, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/826aea2253363fe04e8c4991b2a8869e-Paper-Conference.pdf.

Shiping Gao, Fanqi Wan, Jiajian Guo, Xiaojun Quan, and Qifan Wang. Advantage-guided distillation for preference alignment in small language models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 18916–18938, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/2f891d026c7ba978168621842bc6fe73-Paper-Conference.pdf.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Xia Jiang, Yaoxin Wu, Minshuo Li, Zhiguang Cao, and Yingqian Zhang. Large language models as end-to-end combinatorial optimization solvers. In *Advances in Neural Information Processing Systems*, volume 38, 2025a. doi: 10.52202/085713-5495. URL https://papers.neurips.cc/paper_files/paper/2025/hash/f0c6c93e1dbff84daf6082fef7e1a094-Abstract-Conference.html.

Xia Jiang, Yaoxin Wu, Chenhao Zhang, and Yingqian Zhang. DRoC: Elevating large language models for complex vehicle routing via decomposed retrieval of constraints. In *International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/73d6c3e4b214deebbbf8256e26d2cf45-Abstract-Conference.html.

Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ByxBFsRqYm.

Yeong-Dae Kwon, Jinho Choo, Byoungjip Kim, Iljoo Yoon, Youngjune Gwon, and Seungjai Min. POMO: Policy optimization with multiple optima for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://papers.nips.cc/paper_files/paper/2020/hash/f231f2107df69eab0a3862d50018a9b2-Abstract.html.

Xiaowei Mao, Haomin Wen, Hengrui Zhang, Huaiyu Wan, Lixia Wu, Jianbin Zheng, Haoyuan Hu, and Youfang Lin. DRL4Route: A deep reinforcement learning framework for pick-up and delivery route prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2023. URL https://arxiv.org/abs/2307.16246.

Daniel Merchán, Jatin Arora, Julian Pachon, Karthik Konduri, Matthias Winkenbach, Steven Parks, and Joseph Noszek. 2021 Amazon Last Mile Routing Research Challenge: Data set. *Transportation Science*, 58(1):8–11, 2024. doi: 10.1287/trsc.2022.1173. URL https://pubsonline.informs.org/doi/10.1287/trsc.2022.1173. Published online September 14, 2022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

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. URL https://arxiv.org/abs/2402.03300.

Haomin Wen, Youfang Lin, Xiaowei Mao, Fan Wu, Yiji Zhao, Haochen Wang, Jianbin Zheng, Lixia Wu, Haoyuan Hu, and Huaiyu Wan. Graph2Route: A dynamic spatial-temporal graph neural network for pick-up and delivery route prediction. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2022. doi: 10.1145/3534678.3539084. URL https://doi.org/10.1145/3534678.3539084.

Lixia Wu, Haomin Wen, Haoyuan Hu, Xiaowei Mao, Yutong Xia, Ergang Shan, Jianbin Zheng, Junhong Lou, Yuxuan Liang, Liuqing Yang, Roger Zimmermann, Youfang Lin, and Huaiyu Wan. LaDe: The first comprehensive last-mile express dataset from industry. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5991–6002. ACM, 2024. doi: 10.1145/3637528.3671548. URL https://doi.org/10.1145/3637528.3671548.

Liang Xin, Wen Song, Zhiguang Cao, and Jie Zhang. NeuroLKH: Combining deep learning model with Lin-Kernighan-Helsgaun heuristic for solving the traveling salesman problem. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://papers.nips.cc/paper_files/paper/2021/hash/3d863b367aa379f71c7afc0c9cdca41d-Abstract.html.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Haoran Ye, Jiarui Wang, Zhiguang Cao, Federico Berto, Chuanbo Hua, Haeyeon Kim, Jinkyoo Park, and Guojie Song. ReEvo: Large language models as hyper-heuristics with reflective evolution. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1381. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/4ced59d480e07d290b6f29fc8798f195-Abstract-Conference.html.

A Data, Information, and Evaluation

A.1 LaDe Task Construction and Advice

The processed Chongqing training/history pool contains 22,301 closed-wave tasks from May 1–September 24. The 512 adaptation tasks and a 64-task calibration set were selected by fixed hash within the training dates and cover June 1–September 24. The 300 development tasks cover September 25–October 12. Task identifiers are disjoint across these splits, while couriers are shared. All calendar ranges refer to the released dataset.

G2R was trained on the full processed pool, so its proposals for adaptation and calibration tasks are in-sample. Of the development tasks, 161 overlap the 2,000-task validation probe used to select its checkpoint. We used the calibration set for input-length checks in the primary comparison, and it is also the cohort for the frozen-policy source analysis (Appendix C). Because it lies inside the training dates, an earlier calibration route can legitimately appear as history for an adaptation task.

Each LaDe prompt contains one greedy G2R route and eight samples generated at temperature 0.7, followed by exactly two earlier routes. The two routes are retrieved from the 22,301-route training pool among routes with strictly earlier dates, ranked by AOI-set Jaccard similarity to the current task, then same-courier status, then recency. When fewer than two earlier routes share an AOI with the task, the same ranking is applied to all earlier pool routes. Retrieved routes may come from any courier; same-courier status acts only after AOI similarity. Each route is rendered as its AOI sequence, and AOIs absent from the current task are shown as `X`. The candidate cache is fixed across the four post-training branches. The saved-full G2R comparator is a separate prediction artifact that never enters the prompt; it differs from pool greedy on 14 of the 300 tasks. Borda sums candidate ranks, breaking ties by greedy position and then stop ID; the medoid minimizes integer pairwise inversion distance, breaking ties by the earliest candidate slot.

The prompt excludes the query's true order, completion times, and candidate truth scores. Preprocessing is retrospective, which makes the evaluation an offline, retrospective study. Station estimates use acceptance GPS across dates, operational-day filtering uses completion dates, location cleaning uses delivery GPS and scan-order diagnostics, and closed-wave construction uses the day's maximum wave. G2R also receives day-level package/wave counts.

A.2 Amazon Route Sets and Prompts

The available build and evaluation sets contain 6,112 and 3,052 routes, with disjoint route identifiers; the challenge published 3,072 evaluation routes (Merchán et al., 2024). Historical retrieval uses the fixed build library without a past-only filter. The structured supervised pool contains 1,431 targets after separating 68 validation targets from a 1,499-target pool. The all-route comparison uses a frozen adapter. The 2,752-route subset excludes the 300 evaluation routes that were used in adapter selection and inspected descriptively beforehand.

Proposal sources. Amazon prompts contain only candidate zone sequences, with no coordinates, stop lists, or travel times. Each prompt lists (i) nine proposals from a separately fine-tuned Qwen3-4B zone-order SFT adapter, one greedy decode and eight samples at temperature 0.8 and top-p 0.95; (ii) up to two retrieved build routes from the same station (from all stations when the station has no build route), rendered in the current task's zone names; and (iii) the champion route, always last. Each retrieved route is projected onto the zones it shares with the task, and the task's remaining zones are inserted by nearest travel-time insertion. A retrieval slot with no overlapping build route is dropped from the prompt. Of the 3,052 prompts, 2,656, 223, and 173 carry two, one, and zero retrieved routes, for twelve, eleven, and ten candidates. The Amazon data has no driver identity, so retrieval is route-level. Source-name text is removed from training prompts; candidate positions are not shuffled, and deployment prompts include source names.

A.3 Metrics and Statistical Units

For two complete permutations of the same $n \geq 2$ stops, with N_d discordant pairs,

$$\mathrm{KRC}(y, y^\star) = 1 - \frac{2N_d}{\binom{n}{2}}. \tag{6}$$

Amazon scores use the official sequence-and-travel-cost implementation. Position-tolerance diagnostics (the historical LaDe H_3 and HR@3) and the local-baseline top-three set recall are distinct quantities. We never combine them, and no claim rests on any of them.

All primary means retain the full evaluation cohort. LaDe direct outputs are validated as complete, duplicate-free permutations, with no hidden completion or route deletion. Amazon parse failures contribute empty constraints and remain in evaluation. Auxiliary zone-order legality is reported separately from final stop-route validity.

For LaDe, we average each route's score across the three trained models before paired route bootstrap, with 10,000 draws and bootstrap seed 20260921. Courier-cluster resampling over the 186 couriers is a sensitivity check; for route RL minus SFT it gives $[-0.001192, 0.012732]$, against the route-paired $[-0.000982, 0.012548]$. Amazon uses 20,000 paired route-bootstrap draws with seed 423. Intervals describe the retained models on their evaluation cohorts and do not include training-seed uncertainty.

B Model and Optimization Details

B.1 Backbone and Supervised Adapters

Both instantiations use the backbone `Qwen/Qwen3-4B-Instruct-2507`. LaDe training and evaluation, and the current Amazon inference, pin revision `cdbee75f17c01a7cc42f958dc650907174af0554`; the historical Amazon SFT notebook did not pin a revision. The adapters, numeric precision, response formats, and training pools differ between the two instantiations. Backbone size is the total parameter count and differs from the trainable LoRA parameter count.

Table 4: Supervised and route-reward settings.

Setting	LaDe direct policy	Amazon structured policy
Base loading	NF4, double quantization (Dettmers et al., 2023); BF16 compute	BF16; no quantized loading
LoRA rank / alpha	16 / 32	32 / 64
Adapted projections	q, k, v, o	q, k, v, o, gate, up, down
SFT tasks / epochs	512 / 2	1,431 / 2
SFT learning rate	5×10^{-5}	10^{-5}
Gradient accumulation	4	16, batch size 1
Response objective	Unit-weight stop actions; no EOS action	Response-only; EOS weight 50
Maximum sequence length	4,096 (prompt and actions)	8,704 training; 12,288 inference
Route-reward stage	128 updates; four sampled routes/task	Not used
RL rate / KL / clip	2×10^{-6} / 0.02 / 0.2	Not applicable
Evaluation	Greedy masked stop selection	Seven samples, $T = 0.8$, top-$p = 0.95$

LaDe uses zero LoRA dropout. The pinned training stack is Torch 2.11.0+cu128, Transformers 4.57.1, PEFT 0.17.1, Accelerate 1.14.0, and bitsandbytes 0.50.2. The Amazon seed is 423; structured inference allows up to 4,608 new tokens and also records a greedy diagnostic draw.

B.2 Group-Relative Objective and Probability Accounting

For generated route i, let $h_{it} = (z, a_{i,<t})$ and let $\mathcal{A}(h_{it})$ be the remaining-stop support. The behavior policy is the current policy snapshotted before sampling. With $\rho_{it} = \pi_\theta(a_{it} \mid h_{it})/\pi_{\text{old}}(a_{it} \mid h_{it})$,

the implemented surrogate is

$$\mathcal{J}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{L_i}\sum_{t=1}^{L_i}\left\{\min\left(\rho_{it}\widehat{A}_i, \mathrm{clip}(\rho_{it}, 1-\epsilon, 1+\epsilon)\widehat{A}_i\right) - \beta D_{it}\right\}\right], \tag{7}$$

$$D_{it} = \sum_{v\in\mathcal{A}(h_{it})} \pi_\theta(v \mid h_{it}) \log \frac{\pi_\theta(v \mid h_{it})}{\pi_{\mathrm{ref}}(v \mid h_{it})}. \tag{8}$$

The KL runs from the current policy to the frozen SFT reference and is computed over the full legal categorical distribution rather than with a sampled-token estimator. Advantages use population standard deviation and a floor $\epsilon_A = 10^{-6}$. Sampling temperature is 1.0, and sampling, replay, and reference scoring use the same prefix, support, and temperature. Prefixes are recomputed from the full sequence rather than read from a detached cache. Per-token backpropagation releases activations and still preserves route/group averaging. With exact legal-action KL, full-prefix replay, zero dropout, and shared sampling temperatures, the probability accounting is exact.

The implementation makes one optimizer update per sampled group, so replay ratios are approximately one before that update. The replay check rejects updates with a selected-token absolute log-probability difference above 0.15. Accepted ratios therefore lie in $[e^{-0.15}, e^{0.15}]$, inside the clipping interval $[0.8, 1.2]$, and the clipped branch is inactive in accepted updates. The operative learning signal is the group-relative policy-gradient term plus reference KL, and we call the objective GRPO-style for that reason. Groups with equal rewards have zero reward advantages, while the KL term can remain nonzero. The mask is represented numerically by a large negative value, and implementation assertions check that decoded routes remain legal.

Algorithm 1 LaDe supervised initialization and route-level adaptation

Require: Training tasks and observed routes; fixed advice cache; Qwen policy
1: Fit the response-only supervised policy for the prescribed schedule
2: Freeze its reference copy π_{ref}
3: **for** each of 128 post-training updates **do**
4: Select a task and render its fixed advice under the prescribed code mapping
5: Snapshot $\pi_{\mathrm{old}} \leftarrow \pi_\theta$
6: Sample four complete routes; retain behavior log-probabilities
7: Score KRC using the training reference and compute Equation 5
8: Replay identical prefixes, masks, and temperature with gradients enabled
9: Reject inconsistent replay; otherwise make one update using Equation 7
10: **end for**
11: Freeze the final adapter; greedily decode all evaluation tasks before scoring

B.3 Same-Anchor Controls

Training seeds A/B/C are 20260911, 20260922, and 20260923. Seed A recovers the archived SFT256 checkpoint, reproduces its predictions, and reuses verified historical RL128 predictions. B/C independently repeat initial SFT and all continuation branches. All seeds use evaluation nonce 29260911, so a seed change leaves the evaluation code maps and table order unchanged.

Each continuation starts from its own seed's SFT adapter and a fresh optimizer. The task and prompt schedule is shared across branches: update u selects a task with seed $s + 100000 + u$ and a rendering nonce $s + 1000 + u$. Low-rate continued SFT uses 2×10^{-6} and native-rate continued SFT uses 5×10^{-5}. RL samples four routes and includes reference KL; continued SFT replays the target route without KL. Equal update counts therefore compare adaptation recipes at unequal compute (Appendix F). The recorded A schedule visits 108 distinct tasks during its 128 updates.

Artifact checks cover configurations, predictions, mappings, legality, and score consistency. Some archived bundles omit raw remote model/optimizer tensors and per-update diagnostic fields.

C FROZEN-POLICY SOURCE ANALYSIS

Protocol. The analysis uses the seed-A route-RL adapter of Table 2 (SFT256 on 512 tasks for two epochs, then RL128 with group size 4) on `Qwen/Qwen3-4B-Instruct-2507` at the pinned revision. The policy was trained once with all sources. Each arm removes sources at inference by whitelist projection before prompt construction, with no retraining per arm. The arms combine the task table (T: stop codes, relative coordinates, AOI, and acceptance minute), the cached G2R greedy route (G, candidate slot 0), the eight cached G2R samples (R, slots 1–8), and the two retrieved earlier routes as AOI sequences (H). Decoding is greedy over the remaining stops, with exactly n steps and no solver, repair, or edge guidance; evaluation nonce 29260911 fixes the stop-code map and display order across arms.

Tasks. The 64 calibration tasks cover 57 couriers with 5–24 stops per task and dates June 1–September 24. They are drawn by fixed hash from the training split, so they lie inside the adviser's training period: G2R proposals are in-sample, and history is not information-isolated. The analysis therefore characterizes how the trained policy uses its sources, and the system comparison uses the 300-task development cohort.

Identity. The adapter weight hash is identical in all four inference jobs. In every job, eight anchor replays exactly match the saved route-RL predictions and the legacy rollout, and the expected training configuration matches the original run's configuration. The anchors are the first eight task IDs on which the route-RL output differs from G2R greedy, and the saved SFT and RL predictions differ on one of them. No original weight hash exists for the training run, so identity is established behaviorally. An adjacent training-configuration file was found and matched in two of the four jobs and was absent in the other two.

Uncertainty. Paired intervals use 10,000 route-level percentile bootstrap draws from one shared 10000×64 index matrix (seed 20260925). Ties count equal KRC; for TGRH − TGR there are 43 ties and 41 identical routes, since two tasks have different permutations with equal KRC. Position-tolerance HR@3 is recorded but not used.

Table 5: Frozen-policy arms and aggregation references on the 64 calibration tasks. Copy columns count tasks whose route equals G2R greedy, equals any of the nine proposals, or differs from all nine.

Arm	Proposals	History	KRC ↑	= greedy	= any of nine	Differs from all
T	None	No	0.0431	0	3	61
TH	None	Yes	0.2443	0	2	62
TG	Greedy	No	0.5152	46	47	17
TGH	Greedy	Yes	0.5137	43	45	19
TR	Eight samples	No	0.4923	8	20	44
TRH	Eight samples	Yes	0.4999	6	18	46
TGR	All nine	No	0.5571	25	32	32
TGRH	All nine	Yes	**0.5673**	28	34	30
G2R greedy	Candidate slot 0		0.5243			
Borda-of-nine	Nine proposals		0.5394			
Medoid-of-nine	Nine proposals		0.5288			

Table 6: Paired frozen-policy comparisons on the 64 calibration tasks (left minus right, mean KRC).

Comparison	Difference	95% interval	Wins / ties / losses	Identical routes
TGRH − TGR	+0.0102	$[-0.0132, 0.0375]$	13 / 43 / 8	41
TGRH − TH	+0.3230	$[0.2347, 0.4183]$	47 / 8 / 9	0
TH − T	+0.2012	$[0.0965, 0.3035]$	38 / 9 / 17	5
TGRH − G2R greedy	+0.0430	$[0.0101, 0.0791]$	20 / 32 / 12	28
TGRH − Borda-of-nine	+0.0279	$[-0.0158, 0.0704]$	27 / 16 / 21	9
TGRH − TGH	+0.0535	$[0.0172, 0.0955]$	21 / 32 / 11	30
TGRH − TRH	+0.0673	$[-0.0039, 0.1398]$	30 / 16 / 18	11

D AMAZON STRUCTURED EXECUTION

D.1 TARGETS, PARSING, AND VOTING

The evaluated adapter corresponds to the archived structured SFT configuration `dnotag_v4_g2eos`. Zone precedences are extracted at gaps one and two in the true non-missing-zone order. Cluster targets follow the adjacent first-visit cluster chain, and neighbor targets are empty. Training relations use the "must" confidence label. An auxiliary complete zone order is also supervised, but the executor reads only the explicit relation fields and never converts the auxiliary order into relations.

Every prompt includes the champion route (Appendix A). Each of seven stochastic draws contributes at most one vote per relation kind and endpoint pair. The admission threshold stays at four votes even when some parses fail, and an unrecoverable parse contributes an empty constraint set. Among supporting draws, normalized confidence labels determine the retained tier: if at least half mark the relation "must", it enters the strong tier, and otherwise it is advisory. Duplicates and self-loops are normalized, and the tiering rules are fixed with no learned component.

The separate greedy diagnostic parses all 3,052 tasks with salvage; 3,036 parse cleanly, and auxiliary order legality is 98.034%. This legality is separate from final stop-route validity, and a malformed auxiliary order removes no task.

D.2 COMPILERS AND THE COPY ENCODING

Every injected LKH solve uses a two-second time limit and seed 1. Relations enter the solver as soft penalties. Zone precedences use weights 10 and 1 for the two tiers, and cluster relations expand into cross-cluster zone pairs with weight capped at 3. The relation-only configuration, which DORA uses, disables cluster/super-path structure. The hybrid configuration retains zone/super-neighbor transitions while replacing the two historical relation sources with injected relations. Neither injected configuration is identical to the complete champion pipeline.

COPY constructs the same target-style relation fields from the champion candidate, then applies the same normalization and solver as the student, with no student forward pass. It therefore measures what the relational interface yields from the champion route alone.

Table 7: Additional paired Amazon comparisons. Negative AS differences favor the first system. All-route comparisons include the adapter-selection subset; the 2,752-route subset excludes it.

Comparison	n	Mean difference	95% interval
COPY relation-only − champion	3,052	-0.001190	$[-0.001365, -0.001024]$
Student relation-only − COPY relation-only	3,052	$+0.000251$	$[0.000093, 0.000423]$
Student hybrid − champion	3,052	$+0.000252$	$[0.000089, 0.000423]$
Student hybrid − COPY hybrid	3,052	$+0.000188$	$[0.000057, 0.000328]$
COPY hybrid − champion	3,052	$+0.000064$	$[-0.000033, 0.000172]$
Student relation-only − champion	2,752	-0.000984	$[-0.001236, -0.000735]$
Student relation-only − COPY relation-only	2,752	$+0.000231$	$[0.000064, 0.000404]$
Student relation-only − earlier champion run	3,052	-0.000920	$[-0.001153, -0.000685]$

The primary comparison (student relation-only minus champion rerun on all 3,052 routes), the student-versus-COPY test on the 2,752 selection-excluded routes, and the earlier-run champion reference were fixed in a written protocol before any evaluation route was scored. The student, COPY, and champion means on the selection-excluded subset are 0.024039, 0.023808, and 0.025023. Student and COPY emit identical normalized injected relations on 2,221 of 3,052 tasks. On the post-hoc subset of 831 tasks with different relations, student minus COPY is $+0.000921$, with interval $[0.000336, 0.001549]$; this subset is descriptive and does not define a deployable routing rule.

E ADDITIONAL BASELINES AND DIAGNOSTICS

The local baseline study uses 22,301 training tasks and 2,000 validation tasks. LightGBM is a next-stop ranker with at most 200,000 ranking states and selects 300 trees from checkpoints at 50,

100, 200, and 300. Both neural adaptations complete 30 epochs. DeepRoute selects epoch one. DRL-SCST uses five supervised warmup epochs followed by 25 self-critical policy-gradient epochs and also selects epoch one, so its selected weights and all 300 predictions equal the supervised baseline; its best RL epoch has validation KRC 0.13385, against 0.13400 for the selected supervised checkpoint. These baselines are local adaptations trained to their full budget and do not reproduce each paper's best reported setting.

Table 8: Additional LaDe references. All scores use the same 300 tasks.

Reference	KRC	Interpretation
Station-start nearest neighbor	0.136520	No true first-stop initialization
Distance LKH	−0.015687	Native orientation, no truth-based reversal
Historical AOI rank + LKH	0.262963	Fixed historical guidance
Archived local DRL4Route, augmented	0.225050	Different historical implementation
New-stream G2R standalone greedy	0.446638	Distinct from prompt greedy

The archived local DRL references use a different implementation with supervised loss and an LSD-based policy-gradient term; their alignment is reconstructed from ordered manifests and export code, whereas the new baselines have committed per-task outputs. Historical AOI-rank rules were selected during earlier development. Distance-LKH optimizes its own geometric objective, which does not involve observed order.

DORA adds its 512-task adapter budget to pretrained Qwen, full-pool G2R training, and a historical library, so the primary comparisons are against the strongest retained advice and against same-anchor post-training controls.

E.1 Changes to Direct Predictions

Across A/B/C, RL changes 49, 74, and 71 complete LaDe permutations. Exact matches to any of the nine proposals move from 120/100/89 after SFT to 124/107/96 after RL, so most RL outputs remain new compositions. There are 4/1/1 zero-variance groups among 128 updates.

For the 214 tasks with at most ten stops, RL minus SFT is $+0.017230$, -0.012090, and $+0.009279$ for A/B/C. These strata are post hoc and descriptive and do not replace the primary endpoint.

F Cost Measurement

Table 9: Recorded local CPU baseline components, in seconds, with two threads and serial execution. Prediction totals cover all 300 development tasks, include startup, and are not per-request latency quantiles.

Method	Training component	Prediction component
FIFO	Not applicable	0.053
Nearest neighbor	Not applicable	0.066
OR-Tools, 1 s/route	Not applicable	300.594
LightGBM	43.113	0.559
DeepRoute, local	2698.153	0.983
DRL-SCST, local	5899.345	0.862
Frozen G2R greedy	Historical cost unmeasured	3.146

The CPU study used Apple M5 with 24 GiB unified memory, macOS 27.0, Python 3.11.15, Torch 2.7.1, LightGBM 4.6.0, OR-Tools 9.14.6206, SciPy 1.17.1, and NumPy 2.2.6. MPS was available but not used in the reported CPU workflow. The complete baseline training workflow took 8,641.72 seconds and the seven-method prediction workflow 307.06 seconds; workflow totals include overhead beyond component sums.

The recorded LaDe B/C SFT stages took approximately 11.4–11.5 minutes per model on A100; route RL took about 95.2 minutes, and continued SFT about 1.55–1.61 minutes. Downstream 300-task decoding took about 10.6–11.1 minutes per model and stage. These exclude complete loading,

historical retrieval, and G2R proposal generation. The historical seed-A training cost is not inferred from B/C. CPU baseline totals and A100 language-model totals come from different configurations and are not a hardware-normalized comparison.

A separate Amazon CPU sample measured approximately 42.8 seconds per route for full champion generation before merging and about 2.10 seconds for a two-second constrained solve. Champion-inclusive refinement incurs both stages, and the student adds language-model inference and voting. A GPU throughput of 4.40 seconds per route, observed in a different structured experiment, does not measure latency for the evaluated adapter. We do not combine these records into an end-to-end figure.

A deployment measurement would include retrieval, proposal generation, policy inference, aggregation, execution, failed attempts, and relevant I/O. Sequential stage times can be summed; overlapping stages require an outer wall-clock measurement.

G Alternative Interfaces and Additional Probes

These studies use separately trained models or separate protocols and are not part of the primary systems.

Task-only LaDe policy. A separately trained task-only Qwen adapter used 512 tasks, two epochs, and no G2R/history advice. It obtained KRC 0.00617 on the development cohort, compared with 0.45577 for the saved-full G2R reference. A 32-task fitting diagnostic raised fixed-mapping training KRC from 0.0247 to 0.9332; remapping codes and display order on the same tasks gave 0.2242, so task-only fitting is tied to the code mapping and does not transfer to new tasks.

LaDe relation policy. An exploratory relation-output model greedily abstained after both SFT and RL, returning the cached G2R greedy route with KRC 0.4464, although 126 of 128 sampled RL groups had nonzero reward variation. In a frozen-executor intervention on the 64 calibration tasks, empty relations scored 0.5673 (the same routes as the TGRH arm of Appendix C), corrective true relations 0.6805, and all selected true relations 0.7026; random and flipped directions scored 0.5129 and 0.1816. Because the oracle rows use target information, they measure executor sensitivity and do not estimate deployable accuracy.