# WHEN CAN NEURAL ROUTING POLICIES DISCARD REDUNDANT INPUTS?

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Fixing an executed route prefix in capacitated vehicle routing leaves a residual optimization problem, yet neural solvers can continue to use inputs that no longer determine its feasible continuations or costs. When can such task-redundant inputs be removed while preserving routing quality? We connect residual-task formulation to computational interventions and controlled retraining. We select POMO and ReLD for their contrasting mechanisms of information reuse: cross-token normalization and cached readouts from a normalization-free encoder. In both, local masking leaves served-customer influence in shared statistics or retained embeddings. Closing the identified paths removes this influence but worsens fixed-weight route quality, separating task sufficiency from the dependencies of a trained representation. Complementary retraining experiments on progress inputs show that, under mixed-size training, deleting dedicated step inputs maintains quality comparable to normalized retention, while normalized progress better supports transfer from a single small training size. These findings motivate a state-aware graph-pointer that retains observed context, represents unfinished customers beyond current feasibility, and uses normalized progress. Arrivals and demand realizations update this same observation interface, allowing the solver to transfer zero-shot to dynamic and stochastic tasks from a static supervised initialization and improve through complete-route adaptation. These adaptation gains recur across training lineages. The study establishes a basis for simplification that accounts for both residual-task sufficiency and learned computational dependence, with deletion and re-encoding evaluated by the quality of the resulting routes.

## 1 INTRODUCTION

Constructive vehicle routing progressively reduces the problem that remains to be solved. Once an executed prefix is fixed, static $N$-customer capacitated vehicle routing (CVRP, denoted C$N$) becomes an optimization over future continuations: past travel adds a constant, while the depot, current location, remaining capacity and unserved customers determine subsequent feasibility and cost. This state-based view underlies exact subset dynamic programming (Bellman, 1962; Held & Karp, 1962) and resource-constrained path algorithms for routing (Feillet et al., 2004), and motivates neural state reduction (Kool et al., 2019; Drakulic et al., 2023). It identifies task-redundant inputs, whose variation preserves the feasible continuations and their future costs.

A neural solver can nevertheless use these inputs through attention, normalization and cached representations. This creates two distinct design questions: what information defines the optimization problem, and what information supports the computation learned by a particular policy? Conflating them can lead to removing a useful computational dependency or to retaining an encoding that transfers poorly across instance sizes. Resolving their relationship is therefore central to simplifying neural routing representations while preserving solution quality.

Existing work supplies complementary strategies. BQ-NCO constructs policies on reduced states (Drakulic et al., 2023); MDAM and LEHD adapt embeddings to the remaining construction process (Xin et al., 2021; Luo et al., 2023); ReLD studies irrelevant context and improves the use of static embeddings (Huang et al., 2025). CS-Probing connects embedding structure, interventions and generalization (Zhang et al., 2025). We build on these advances to ask: through which computa-



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.

tional paths do task-redundant inputs affect routing, and when can a changed representation preserve quality after learning?

Our study makes this connection in two complementary settings. Served coordinates expose dependence in frozen solvers. POMO's attention policy encodes the full instance once and reuses node embeddings throughout construction; its cross-token normalization couples retained features to served customers (Kwon et al., 2020). ReLD also retains static embeddings but strengthens their decoder readout, using a normalization-free encoder (Huang et al., 2025). The contrasting normalization and readout designs of POMO and ReLD let us test exclusion predictions across two distinct mechanisms of information reuse. Absolute progress then provides a controlled input for learning with deletion or alternative encodings. The findings motivate a solver that retains observed computational context, uses normalized progress and updates its state as the remaining work changes. Dynamic arrivals and stochastic demand test whether this representation supports execution and adaptation under changing residual problems.

Our contributions are:

- **A predictive account of task-redundant computation.** Local masks can leave an input's influence in shared statistics or cached embeddings. Coordinated closure and readout reopening identify the effective exclusion paths, with all sixteen prospective ReLD predictions confirmed. The resulting route-cost increases show why task-level redundancy alone does not determine the effect of editing a frozen solver.
- **Quality-preserving representation choices.** Controlled progress retraining separates the presence of an input from its encoding. Mixed-source deletion maintains route quality within 1% of normalized retention at C20/C50/C100; single-size training favors normalized retention for C100 transfer. This establishes both a setting where the dedicated counter channels are dispensable and an effective retained encoding. A separate compact-C20 configuration also maintains comparable quality with reduced training cost.
- **A solver derived from the representation findings.** Observed context, normalized progress and unfinished-customer support form a common graph-pointer interface. Classical-tour supervision initializes its decisions; complete-route adaptation addresses arrivals and realized demand. Paired studies establish repeatable gains in both processes, providing a constructive application of the representation analysis.

## 2 Related Work

**Residual tasks and neural context design.** Pointer and attention policies establish the constructive routing framework (Vinyals et al., 2015; Kool et al., 2019), with POMO introducing multiple-start training (Kwon et al., 2020). BQ-NCO formalizes residual-state reduction through bisimulation and uses it to build efficient neural solvers (Drakulic et al., 2023). MDAM's embedding glimpse excludes visited nodes in the top attention layer (Xin et al., 2021); LEHD dynamically encodes available-node relationships (Luo et al., 2023); DR-DRL uses visited-node removal and dynamic decoder context (Guan et al., 2025). ReLD examines irrelevant context in static embeddings and strengthens their decoder readout (Huang et al., 2025). These works connect residual structure to effective architectures. Our study makes the implemented dependence explicit: coordinated closure and individual path reopening test prospective exclusion predictions, and complete routes measure the consequences of those changes.

**Representation analysis and size generalization.** CS-Probing studies encoded routing knowledge, neuron ablations, quality under embedding-dimension reduction and sparsity-based generalization (Zhang et al., 2025). INViT uses invariant nested views to generalize across geometry and scale (Fang et al., 2024). Our interventions are defined by residual-task semantics, which allow the input to change while the optimization problem stays fixed. The progress study then connects seven input encodings to complete-route quality under paired retraining. It establishes when deletion preserves quality across tested sizes and when normalized retention gives better transfer, providing concrete representation choices beyond the detection of learned dependence.

**Constraint context and online execution.** Dynamic attention updates representations during construction (Peng et al., 2020). CARM represents unvisited customers with constraint-conditioned

modulation (Yu et al., 2026); CCL propagates constraint context through re-embedding (Gui et al., 2026); DGL supplies dynamic distance statistics for local candidates (Xiao et al., 2025). OSLA within MUSLA evaluates successor features after a hypothetical action (Chen et al., 2025). Routing reinforcement learning also supports stochastic arrivals and demands (Nazari et al., 2018). Building on these approaches, our online solver combines observed context, normalized progress and complete-route adaptation. Paired dynamic and stochastic studies test reuse of this representation as the observed residual problem changes. Capacity-conditioned readouts receive a separate controlled comparison on the common backbone.

# 3 METHOD

## 3.1 RESIDUAL PROBLEM FORMULATION AND TASK REDUNDANCY

Consider static deterministic CVRP with depot 0, coordinates $x_i$, known demands $d_i$ and replenishable capacity $Q$. Fix an executed prefix $\sigma_t$ ending at $c_t$. Let $\mathcal{F}(s_t)$ contain the continuations $\rho$ that start there, serve all remaining customers subject to capacity, and return to the depot. Euclidean travel cost decomposes as

$$L(\sigma_t \oplus \rho) = L(\sigma_t) + L(\rho; s_t), \qquad V^*(s_t) = \min_{\rho \in \mathcal{F}(s_t)} L(\rho; s_t). \tag{1}$$

Here $\oplus$ concatenates routes; $L(\rho; s_t)$ includes replenishment travel and the final return. Since $L(\sigma_t)$ is fixed, optimizing the complete route conditional on its prefix is exactly the residual problem $V^*(s_t)$.

Let $q_t$ be remaining capacity and $U_t$ the unserved customer set. The information determining this residual problem is

$$\phi(s_t) = \big(x_0, c_t, x_{c_t}, q_t, Q, \{(i, x_i, d_i) : i \in U_t\}\big). \tag{2}$$

Remaining demands and capacity determine legal service actions; the depot restores capacity to $Q$. Future edges use the current, depot and unserved coordinates. Legal transitions and costs therefore depend on $\phi(s_t)$, giving

$$\phi(s) = \phi(s') \implies \mathcal{F}(s) = \mathcal{F}(s'), \quad L(\rho; s) = L(\rho; s') \ \ \forall \rho \in \mathcal{F}(s). \tag{3}$$

An exact residual dynamic program can therefore recurse on $\phi$: its Bellman update uses the same legal actions, edge costs and successor residual states. Optimal future cost factors through $\phi$, providing the state-sufficiency basis for neural reduction (Kool et al., 2019; Drakulic et al., 2023). Appendix A.1 gives the proof.

**Task redundancy in learned solvers.** We define an input variation as *task-redundant* when it preserves the feasible continuations and their future costs. For the static task above, Equation 3 supplies a sufficient condition: keep $\phi$ fixed. Changes to noncurrent served coordinates or the absolute action counter satisfy that condition. Currently infeasible unserved customers remain in $\phi$ because replenishment makes them serviceable; Appendix A.3 gives an optimal-action reversal caused by moving one such customer.

A learned policy can depend on task-redundant inputs through its internal computation. Learning with a changed input interface can preserve quality, with the outcome depending on encoding and training regime. We study the existing dependence with frozen-weight interventions and the ability to learn a different representation with controlled progress retraining.

## 3.2 TASK-PRESERVING DEPENDENCY ANALYSIS

Our aim is to distinguish a response to a changed routing problem from a response to its neural representation. Write an implemented policy as $\pi_\theta(\cdot \mid s) = \mathrm{softmax}_{\mathcal{A}(s)} D_\theta(E_\theta(P(s)), s)$, where $P, E_\theta, D_\theta$ denote preprocessing, encoding and decoding, including cached inputs; $\mathcal{A}(s)$ is the legal-action set. For $\kappa$ with $\phi(\kappa s) = \phi(s)$, the residual task is unchanged. Probability changes with fixed weights and decoder controls therefore detect dependence on the varied input. We then predict and test which coordinated restrictions remove that response, reopen individual paths, and evaluate complete routes. These steps distinguish dependence, effective exclusion and the quality consequence of exclusion.

We select POMO and ReLD for contrasting mechanisms in attention-based routing (Kwon et al., 2020; Huang et al., 2025). POMO's cross-token InstanceNorm can couple retained features to served nodes even after attention restriction. ReLD has a normalization-free encoder but reuses mixed static embeddings through several decoder readouts. It tests whether dependence persists through encoded memory when the normalization path is absent. Their comparison examines a common design issue: an input can be masked at one operation while remaining active elsewhere in the computation. Native-interface checks of BQ, LEHD and INViT provide additional contrasts (Appendix A.6).

### 3.3 Progress Simplification and Re-encoding

A quality penalty from intervening on a frozen policy leaves open whether the input is needed to learn a good policy. The absolute action counter provides a controlled setting for this complementary question because it is task-redundant in the static formulation and its numerical range changes with problem size. The comparison asks whether quality depends on providing the counter, its encoding, or the training regime.

Seven interfaces cover lookup, support-capped lookup, NoStep, scalar, sinusoidal, MLP and size-normalized scalar. NoStep uses a learned constant and zeros both absolute-step scalar channels. Normalized retention keeps the same constant and supplies $t/(2N)$, where $t$ is the action counter and $N$ the original customer count. All interfaces retain visited indicators, current location, capacity and graph context. Paired retraining and fixed quality tolerances thus test the value of dedicated counter channels given the remaining observations.

### 3.4 State-Aware Graph-Pointer Solver

Frozen-weight exclusion changes route quality, whereas counter retraining preserves quality in the mixed regime and favors normalized progress for single-size transfer (Sections 4.1 and 4.2). These findings motivate retained observed context and normalized progress; while the residual formulation motivates representing all unfinished customers.

**Observed state and graph representation.** The dependence finding motivates keeping served customers in the observed graph as computational context. The residual formulation additionally keeps every known unfinished customer, including those currently beyond capacity, because they can affect later decisions. With $R_t, U_t$ denoting revealed and unserved sets, this graph combines observed coordinates and demands with visited/revealed indicators, current node $c_t$, capacity $q_t/Q$ and observed clock. The feasible-action mask governs service now; graph support retains information for subsequent decisions. Unrevealed coordinates, demands and incident distances are zeroed, and hidden nodes are excluded from attention and pooling. Release-time features are disabled; the total customer count $N$ is public.

The encoder uses three distance-biased graph-attention blocks of width 128 with four heads. It is evaluated on the current route state at each step. Following the encoding result, progress $p_t = t/(2N)$ enters both node features and the pointer query; the step-embedding channel uses a learned constant $e_0$. The observed clock is scaled by $10.98$. If $h_i$ is the resulting node embedding and $\bar{h}_t$ the mean over observed nodes, the base pointer computes

$$z_t = g_\theta([h_{c_t}, h_0, \bar{h}_t, q_t/Q, p_t]), \qquad s_i^{\text{base}} = z_t^\top W h_i/\sqrt{128}, \quad i \in \mathcal{A}_t. \tag{4}$$

The environment's legal action set $\mathcal{A}_t$ controls selection. Greedy execution chooses the highest-scoring legal action, updates the physical state and repeats until every request is served and the vehicle returns to the depot. Figure 2 summarizes this loop.

**From classical continuations to an adaptable policy.** A classical tour supplies a feasible continuation and next-action label at each prefix. Behavior cloning (BC) turns these residual decisions into supervision under a common teacher: 2,048 C20 and 2,048 C50 LKH tours (Helsgaun, 2017). This initializes constructive routing behavior while keeping action labels and training prefixes shared across representation variants. BC thus provides a common starting policy for studying representation choices; reinforcement learning then optimizes complete-route cost under the target transition process.

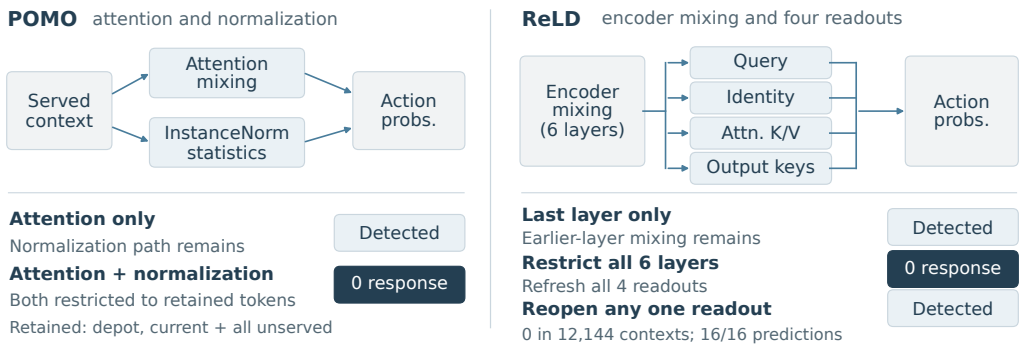


Figure 1: Dependency paths. A response is an action-distribution change after perturbing only noncurrent served coordinates, with weights and the residual task fixed. "Detected" means at least one sampled comparison changes a feasible-action probability by more than $10^{-6}$; "0 response" means all sampled probability vectors remain bitwise identical.

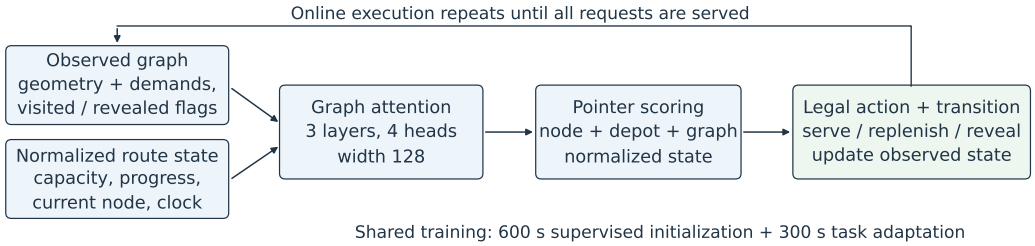


Figure 2: Observed context and normalized state feed a graph-attention pointer. Legal actions update the physical state and observations; supervision and complete-route adaptation train this execution loop.

Arrivals change the observed unfinished set; realized demands change load and can trigger replenishment. Updating the graph and resource state lets the same policy execute these processes from its static BC initialization. We test this zero-shot execution and then adapt to the changed transition process. Sampled complete trajectories use a greedy self-critical baseline (Rennie et al., 2017) and a $0.005$ KL penalty to the BC policy. The objective includes replenishment travel and request revelations during travel or waiting. Public mean demands enter the stochastic policy; realizations are revealed on arrival and charged through physical recourse. Source validation selects checkpoints separately for dynamic and stochastic-specific adaptation.

**Controlled readout extensions.** The normalized-progress backbone is the normalized control. Three readout extensions test additional continuation context. A full-context attention/FFN adapter updates query and pointer-key readouts as a generic expansion control. Two matched relation heads compare current and after-action capacity, both aggregating successors $J_i = (R_t \cap U_t) \setminus \{0, c_t, i\}$ for each legal candidate. The afterstate model uses $q_i = q_t - d_i$ for a customer and $q_0 = Q$ for the depot. It groups successors by $d_j \le q_i + 10^{-6}$ into direct-service and replenishment channels, using direct/via-depot distances, demands and capacity features. Channel summaries and candidate-local features produce an additive score $s_i = s_i^{\text{base}} + \Delta_i$. The readout adds 1,729 parameters; its zero-initialized output starts from the base scores.

The current-capacity control shares architecture and initialization, replacing $q_i$ with $q_t$ for customer actions while retaining the depot reset. All four configurations retain the same backbone context. This comparison tests after-action conditioning against current-capacity relations and generic expansion. Under stochastic demand, $d_i$ is the public mean, so $q_i$ is a nominal conditioning feature; the environment handles realized-demand recourse. Appendix C.1 gives the full pair features, networks and aggregation equations.

# 4 EXPERIMENTS

## 4.1 DOES LOCAL EXCLUSION REMOVE COMPUTATIONAL DEPENDENCE?

To isolate dependence on task-redundant inputs, we perturb only noncurrent served coordinates while fixing the residual task, published checkpoint and decoder controls. POMO compares native execution, encoder-attention restriction, joint attention and normalization restriction, and physical packing. ReLD compares eight predeclared arms: native and refreshed-full encoding, final-layer restriction, all-layer restriction with refreshed readouts, and individual readout reopening. Predictions are fixed before C50/C100 targets are generated. Probability responses test effective exclusion; complete rollouts test its consequence for route cost. Appendices A.4 and A.5 give the contexts and all declared arms.

**Exclusion depends on the complete computational path.** In POMO, restricting encoder attention still leaves served tokens in normalization statistics. Joint attention and normalization restriction eliminates the sampled probability response; physical packing provides a second implementation of exclusion. In ReLD, restricting only the last encoder layer leaves earlier mixing active. Restricting all six layers and refreshing the query, current-embedding residual, attention keys/values (K/V) and scoring keys removes probability changes in all 12,144 sampled comparisons, including bitwise equality. Reopening any individual readout restores detected sensitivity at both sizes. All sixteen prospective intervention-by-size predictions hold (Figure 1). Figure 3 quantifies these action-distribution responses: coordinated exclusion yields zero, and each reopened readout restores sensitivity.

Together, the contrasting architectures expose a common mechanism: removing a token from one operation can leave its information in statistics or embeddings consumed elsewhere. The transferable diagnostic principle is to follow the input's paths to active logits and test their coordinated exclusion. Native checks give further contrasts: BQ is invariant in the sampled tests, while LEHD and INViT expose memory and padding paths (Appendix A.6).

| Model | Intervention | Responding instances C50 | Responding instances C100 | C50 route cost change |
|---|---|---|---|---|
| POMO | Attention restriction | 128/128 | 128/128 | $+2.593$ |
| POMO | Attention + normalization | 0/128 | 0/128 | $+4.306$ |
| ReLD | Last encoder layer | 127/128 | 128/128 | $+0.10\%$ |
| ReLD | All layers + all readouts | 0/128 | 0/128 | $+2.59\%$ |
| ReLD | Reopen identity readout | 128/128 | 127/128 | $+6.77\%$ |

Table 1: Different implementations require different exclusion locations. An instance responds if any eligible sampled context changes a probability by more than $10^{-6}$; zero entries also passed bitwise checks. Route changes are within-model comparisons with the native checkpoint: POMO uses absolute cost units and ReLD percentages, as marked. These are selected mechanism contrasts; all declared arms are in Appendices A.4 and A.5.

**A frozen-policy penalty measures reliance on its representation.** Effective exclusion increases ReLD cost by 2.59% on C50 and 1.28% on C100, and POMO C50 cost by 4.306 distance units (Table 1). The optimization problem has stayed fixed while the computation constructing its solution has changed. The penalty therefore identifies reliance of these trained solvers on their existing representations. Task sufficiency leaves room for such algorithmic reliance: the same residual problem can be solved through different internal computations. This motivates the next study, which asks whether a policy can learn with a reduced input interface.

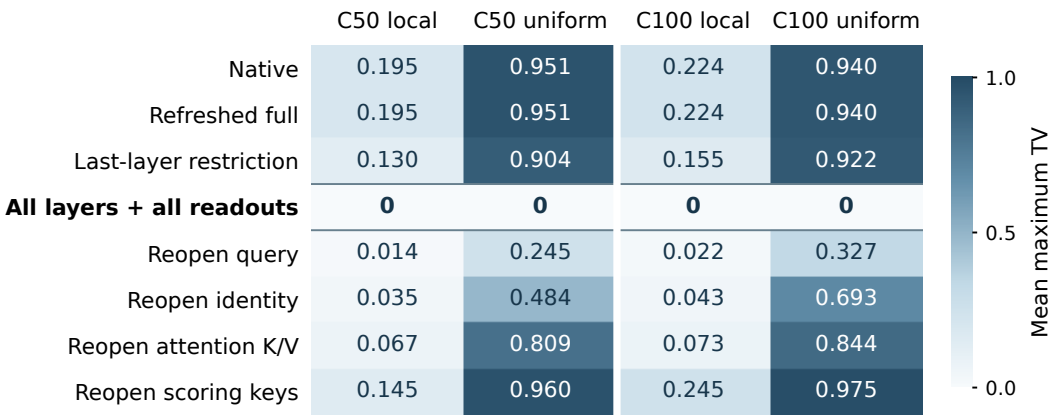


Figure 3: ReLD sensitivity to noncurrent served coordinates. Within each row's encoding/readout configuration, compare action distributions $p, p'$ before and after perturbation using the total-variation distance $\mathrm{TV}(p, p') = \frac{1}{2}\sum_{a\in\mathcal{A}(s)} |p_a - p'_a|$. Each cell averages 128 instance-level maxima over eligible states and four perturbation draws. Local adds Gaussian noise (SD $0.01$) and clips coordinates to $[0, 1]^2$; uniform redraws coordinates in $[0, 1]^2$. Larger values indicate stronger action-distribution responses.

### 4.2 Can Retraining Preserve Quality with Less Explicit Input?

To study quality after learning a changed interface, we compare the seven progress interfaces from Section 3.3 using sixteen fresh paired initializations per source regime: C20 and mixed C20/C50. Pairing fixes backbone initialization and sampling order; checkpoints use source validation. The regimes change size coverage, capacity and source-state composition. Each supplies 512 common C100 targets, alongside source-size checks. We compare route cost with lookup and normalized retention, reporting one-sided 97.5% paired upper bounds to assess whether deletion keeps cost within 1% of its reference. Fixed-weight support-index interventions separately test dependence on lookup indices. The complete comparison rule and all outcomes are reported in Appendices B and B.5.

**Encoding changes the value of explicit progress.** The counter study addresses learning with deletion, complementing the served-coordinate study's intervention on frozen computation. Removing absolute-step channels reduces C100 cost by 16.45% after C20 training and 10.85% after mixed training relative to lookup. Yet normalized retention gives the lowest C100 mean in each regime, 19.136 and 17.900 (Table 2). Thus the lookup comparison alone cannot determine whether explicit progress is useful: its encoding matters. The support-index interventions identify dependence on indices outside training support (Appendix B). Normalized progress expresses stage relative to instance size, offering a scale-aligned alternative to those absolute indices. The MLP results also show that a continuous encoding alone does not ensure good transfer.

**Comparable quality establishes that dedicated channels can be omitted.** Against normalized retention, mixed-source NoStep costs differ by $+0.14\%$, $+0.15\%$ and $+0.03\%$ at C20, C50 and C100. All three one-sided 97.5% paired upper bounds place the relative cost increase below 1%. Because NoStep retains visited indicators, location, capacity and graph context, this result identifies a dispensable explicit counter interface while the policy still observes the remaining work. It provides positive quality-preservation evidence, including transfer beyond training sizes.

After C20 training, NoStep remains within a 1% relative cost difference on C20, but its C100 mean cost is 2.83% above normalized retention. Comparable quality therefore depends on the training regime: mixed-size training supports counter removal across C20/C50/C100, while single-size training benefits from retaining relative progress for C100 transfer. The online solver adopts this normalized interface.

Table 2: Progress encoding and size transfer. C100 route cost is the mean ± SD of sixteen paired initialization means on 512 common targets per source regime. All checkpoints are selected on source validation. Support cap shares lookup training; normalized scalar uses $t/(2N)$.

| Progress interface | C20 training | C20/C50 training |
|---|---|---|
| Absolute-step lookup | $23.554 \pm 3.258$ | $20.084 \pm 0.991$ |
| Support-capped lookup | $22.588 \pm 9.617$ | $18.869 \pm 1.316$ |
| NoStep | $19.678 \pm 0.975$ | $17.905 \pm 0.183$ |
| Scalar | $19.266 \pm 0.945$ | $18.124 \pm 0.228$ |
| Sinusoidal | $20.252 \pm 0.884$ | $18.012 \pm 0.197$ |
| MLP | $34.887 \pm 13.292$ | $18.427 \pm 0.484$ |
| Size-normalized scalar | $19.136 \pm 0.473$ | $17.900 \pm 0.175$ |

**Simplification can have a resource benefit at preserved quality.** A separate seven-seed C20 study retrains a compact representation containing only depot, current and unserved customers. Against full-graph training at matched updates, it reduces total training time by 22.325% and keeps C20 cost within 1%, with a one-sided 97.5% upper bound of $+0.523\%$ on the relative increase (Appendix B.6). This links a concrete reduction in represented nodes to lower training cost at comparable C20 route quality.

### 4.3 Does the State-Aware Solver Transfer and Adapt?

Dynamic arrivals extend the observed unfinished set; stochastic demand changes remaining capacity and replenishment needs as service reveals actual consumption. Both processes therefore update the graph and resource state already used by the method, giving a natural extension of its decision interface. We compare zero-shot BC policies with complete-route adaptation, and test added readouts on the common observation interface. Four configurations receive 600-second BC and 300-second adaptation caps on the same scheduled GPU. Caps include setup, validation and checkpoint writes; update counts vary by architecture. Shared teacher preprocessing and resource records are in Appendix C.3.

D50 has capacity 40, half the customers initially revealed and release horizon 10.98. Eight fresh paired training lineages are evaluated on 1,024 common D50 instances, generated after source validation selects checkpoints. Paired-$t$ intervals and exact sign-flip tests compare afterstate with each control, with Holm correction across the three contrasts. Table 3 reports means and SDs across lineage means; adaptation changes are descriptive.

A separate exploratory follow-up adapts the same 32 BC starts directly to dynamic stochastic demand, using 512 fresh DVRPSD50 targets and 256 D50 retention instances. The D50-adapted policies also receive static and size-transfer checks. Saved routes are replayed for legal completion, distance, event timing and replenishment. Complete protocols appear in Appendices C.2 and C.3.2.

**Dynamic adaptation uses the same decision interface.** D50 adaptation improves every one of the eight lineages in all four configurations. Mean reductions are 0.99–1.13%, yielding costs of 13.6942–13.7229. Arrivals alter the observed unfinished set, while travel advances the clock; both enter the policy's state at the next decision. The static BC policy can therefore execute the process before adaptation, and complete-route training aligns its decisions with the cost of that process. The repeated gains support reuse and adaptation of this interface under changing observations.

Table 3: D50 route cost before and after adaptation (mean ± SD). BC transfers directly; adapted checkpoints use source validation. Parameters are trainable/total, and stage times include validation and checkpointing.

| Policy | Params (train./all) | BC cost | Adapted cost | BC (s) | Adapt. (s) |
|---|---|---|---|---|---|
| Normalized GNN | 695,320 / 744,985 | $13.856 \pm 0.102$ | $13.710 \pm 0.097$ | $598.6 \pm 0.6$ | $298.7 \pm 0.3$ |
| Full-context adapter | 893,600 / 943,265 | $13.849 \pm 0.112$ | $13.694 \pm 0.100$ | $598.6 \pm 0.5$ | $298.3 \pm 0.6$ |
| Current-capacity relation | 697,049 / 746,714 | $13.858 \pm 0.095$ | $13.721 \pm 0.095$ | $598.3 \pm 0.7$ | $298.6 \pm 0.7$ |
| Action-afterstate relation | 697,049 / 746,714 | $13.879 \pm 0.108$ | $13.723 \pm 0.115$ | $598.4 \pm 0.6$ | $298.5 \pm 0.6$ |

The gains extend to the dynamic stochastic-demand transfer panel: DVRPSD50 cost falls by 0.36–0.42%, with negative descriptive paired intervals in every arm. VRPSD20 changes are mixed: the full adapter's mean rises by 0.12%, and all four change intervals span zero. Complete results on C20, C50, C100, D20 and VRPSD20, with 256 instances per secondary panel, appear in Appendix C.3.

**Stochastic adaptation responds to realized resource changes.** In the separate follow-up, mean DVRPSD50 cost falls by 0.53–0.82%, with each arm improving in all eight lineages. D50 retention cost also falls by 0.56–0.83%; all eight descriptive change intervals are below zero. Here the policy observes public mean demands before service, while actual consumption changes remaining capacity and may trigger replenishment. The representation exposes these observed resource changes, and the complete-route objective charges their recourse cost. Improvement under this information pattern shows that adaptation can use the existing state interface without access to future demand realizations. The mean retention gains additionally show that the adapted policy remains useful on deterministic arrivals. These are paired within-panel results from stochastic source selection (Appendix C.3.2).

**The common state interface supports gains across readout designs.** All four configurations improve through adaptation, including the normalized backbone. Explicit after-action grouping gives no resolved advantage over the backbone, full adapter or current-capacity relations: D50 mean cost increases are $0.097\%$, $0.210\%$ and $0.012\%$, with all paired intervals spanning zero (Table 18). Stochastic comparisons also leave its incremental benefit unresolved. The repeatable finding is adaptation across readout choices, including the backbone. This provides a constructive application of the common state-aware interface under changing observations and resources.

**Complete-route quality against classical references.** On the full 1,024-instance D50 panel, the adapted policies cost 3.10–3.31% less than the specified bounded rolling-LKH reference and 11.57–11.75% less than rolling nearest neighbor. Every arm's eight lineage means lie below both references. Appendix C.3.3 provides comparator budgets and fallback accounting.

## 5 Discussion and Conclusion

The study distinguishes the information defining a residual routing task from the information used by a trained computation. POMO and ReLD expose two implementations of the same phenomenon: local masking leaves influence in shared statistics or retained embeddings. Coordinated path tests make that dependence predictable, and complete routes show the cost of changing an existing representation.

Controlled retraining establishes the complementary design result. Dedicated absolute-progress channels can be omitted at comparable quality under mixed-source training; normalized retention better supports transfer from a single small training size. Quality preservation holds with graph context, visited indicators, location and capacity retained. An independent compact-C20 configuration also maintains comparable C20 route quality with reduced training cost.

These findings motivate the observed-context, normalized-progress solver. Its interface updates with revealed work and realized resources, supporting zero-shot execution from static supervision and repeatable dynamic and stochastic adaptation gains. The added afterstate readout has no statistically established incremental benefit. The constructive contribution is the state-aware policy that translates the representation findings into complete routing behavior.

## Reproducibility Statement

The appendices specify the architecture, observation interface, training budgets, checkpoint selection and statistical analyses. Source and code hashes identify the data, model checkpoints and complete evaluation panels. Saved routes and per-step traces support independent checks of action legality, completion, physical cost, event times and replenishment. The accompanying records include all configurations, seeds and outcomes, with resource accounting and deployment measurements for each study.

## AI Use Statement

Generative AI assisted literature review, hypothesis and protocol design, implementation, debugging, analysis, and preparation of text, code, tables and figures. Seeded numerical generators produced the synthetic instances. The authors remain responsible for reviewing and validating the submission.

## Ethics Statement

The study uses synthetic routing instances and contains no personal or human-subject data.

## References

Richard Bellman. Dynamic programming treatment of the travelling salesman problem. *Journal of the ACM*, 9(1):61–63, 1962. doi: 10.1145/321105.321111.

Jingxiao Chen, Ziqin Gong, Lvda Chen, Minghuan Liu, Jun Wang, Yong Yu, and Weinan Zhang. Looking ahead to avoid being late: Solving hard-constrained traveling salesman problem. In *Proceedings of the 2024 6th International Conference on Distributed Artificial Intelligences*, DAI '24, pp. 1–12. Association for Computing Machinery, 2025. doi: 10.1145/3719545.3759088. URL https://doi.org/10.1145/3719545.3759088.

Darko Drakulic, Sofia Michel, Florian Mai, Arnaud Sors, and Jean-Marc Andreoli. BQ-NCO: Bisimulation quotienting for efficient neural combinatorial optimization. In *Advances in Neural Information Processing Systems*, 2023.

Han Fang, Zhihao Song, Paul Weng, and Yutong Ban. Invit: A generalizable routing problem solver with invariant nested view transformer. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 12973–12992, 2024. URL https://proceedings.mlr.press/v235/fang24c.html.

Dominique Feillet, Pierre Dejax, Michel Gendreau, and Cyrille Gueguen. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44(3):216–229, 2004. doi: 10.1002/net.20033.

Qingshu Guan, Hui Cao, Tiansen Niu, Lixin Jia, Shuangsi Xue, and Badong Chen. Diverse route-driven deep reinforcement learning for vehicle routing problems. *Neurocomputing*, 653: 131099, 2025. doi: 10.1016/j.neucom.2025.131099. URL https://doi.org/10.1016/j.neucom.2025.131099.

Shuangchun Gui, Suyu Liu, Xuehe Wang, and Zhiguang Cao. Chain-of-context learning: Dynamic constraint understanding for multi-task VRPs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/1a7a22152cd21f0ca3c0f8139bb32905-Abstract-Conference.html.

Michael Held and Richard M. Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, 10(1):196–210, 1962. doi: 10.1137/0110015.

Keld Helsgaun. An extension of the Lin-Kernighan-Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems. *Roskilde University Technical Report*, 2017.

Ziwei Huang, Jianan Zhou, Zhiguang Cao, and Yixin Xu. Rethinking light decoder-based solvers for vehicle routing problems. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2503.00753.

Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *International Conference on Learning Representations*, 2019.

Yeong-Dae Kwon, Jinho Choo, Byoungjune Kim, Iljoo Yoon, Youngjune Gwon, and Seungjai Min. POMO: Policy optimization with multiple optima for reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.

Fu Luo, Xi Lin, Fei Liu, Qingfu Zhang, and Zhenkun Wang. Neural combinatorial optimization with heavy decoder: Toward large scale generalization. In *Advances in Neural Information Processing Systems*, 2023.

Mohammadreza Nazari, Afshin Oroojlooy, Lawrence V. Snyder, and Martin Takac. Reinforcement learning for solving the vehicle routing problem. In *Advances in Neural Information Processing Systems*, 2018.

Bo Peng, Jiahai Wang, and Zizhen Zhang. A deep reinforcement learning algorithm using dynamic attention model for vehicle routing problems. *arXiv preprint arXiv:2002.03282*, 2020.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7008–7024, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Rennie_Self-Critical_Sequence_Training_CVPR_2017_paper.html.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, 2015.

Yubin Xiao, Yuesong Wu, Rui Cao, Di Wang, Zhiguang Cao, Xuan Wu, Peng Zhao, Yuanshu Li, You Zhou, and Yuan Jiang. DGL: Dynamic global-local information aggregation for scalable VRP generalization with self-improvement learning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 8669–8677. International Joint Conferences on Artificial Intelligence Organization, 2025. doi: 10.24963/ijcai.2025/964. URL https://doi.org/10.24963/ijcai.2025/964.

Liang Xin, Wen Song, Zhiguang Cao, and Jie Zhang. Multi-decoder attention model with embedding glimpse for solving vehicle routing problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Canhong Yu, Changliang Zhou, Rongsheng Chen, Zhenkun Wang, and Yu Zhou. Rethinking constraint awareness for efficient state embedding of neural routing solver. *arXiv preprint arXiv:2605.10122*, 2026. URL https://arxiv.org/abs/2605.10122.

Zhiqin Zhang, Yining Ma, Zhiguang Cao, and Hoong Chuin Lau. Probing neural combinatorial optimization models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/becf96dcee66f9df08169e4c96bbbd8c-Abstract-Conference.html.

# APPENDIX GUIDE


| Block | Contents | Page |
|---|---|---|
| A: Dependencies | Residual-state proof, task-relevant customer example, POMO/ReLD protocols and complete outcomes, native-interface checks. | 12 |
| B: Representation | Seven progress interfaces, paired training and all quality comparisons; compact-C20 training-cost evidence. | 16 |
| C: Online solver | Readout specification, training and selection, complete dynamic/stochastic comparisons, classical references and resource summary. | 21 |
| D: Reproduction | Statistical units, physical replay, inference settings and source records. | 27 |


# A COMPUTATIONAL DEPENDENCIES AND TASK-PRESERVING INTERVENTIONS

## A.1 RESIDUAL-TASK SUFFICIENCY

We spell out the static CVRP reduction used to define our interventions. The executed prefix is fixed, demands are known and deterministic, capacity is replenished at the depot, and completion requires serving every remaining customer and returning to the depot. Cost is the sum of future Euclidean travel distances; the task has no absolute-step deadline. Let $\phi$ be the state in Equation 2.

**Proposition 1.** *If $\phi(s) = \phi(s')$, then $s$ and $s'$ admit the same feasible continuations with the same remaining cost. In particular, for any legal action $a$, the optimal future cost satisfies $Q^*(s,a) = \bar{Q}^*(\phi(s), a)$ for a function $\bar{Q}^*$ on residual states.*

*Proof.* The unserved set, demands, current node and remaining capacity determine which customers can be served and whether a depot return is legal. For a customer action $i$, the edge cost is $\|x_{c_t} - x_i\|_2$, the remaining capacity becomes $q_t - d_i$, and $i$ is removed from $U_t$. A depot return costs $\|x_{c_t} - x_0\|_2$ and restores capacity to $Q$. Each action therefore has the same legality, cost and successor residual state from $s$ and $s'$. Induction along an action sequence gives the same feasible continuations and edge costs, including the final depot return. Minimizing over continuations with first action $a$ gives the stated factorization of $Q^*$. □

Noncurrent served coordinates and the absolute action counter do not occur in $\phi$, so changing either leaves the residual problem fixed. Already incurred prefix costs can differ but add a constant to every continuation. The result establishes equality of optimal action sets; a learned policy can still use omitted inputs in its own computation. The dynamic and stochastic solver in Section 3.4 additionally observes the process state, including revealed requests and the clock. The static reduction specifies the interventions used here.

## A.2 EFFECTIVE COMPUTATIONAL EXCLUSION

During served-coordinate interventions we also fix decoder controls. For a deterministic policy computation, unroll preprocessing, encoder, caches and decoder into a directed acyclic graph. If every input to every ancestor of an active feasible logit is fixed or exactly excluded before entering that ancestor, the feasible-action probabilities agree. The proof is induction in topological order: equal inputs give equal deterministic outputs, including the final softmax on a fixed support. The intervention experiments test which remaining paths actually produce a response.

For token feature $k$, retained-support InstanceNorm must compute

$$\mu_k = |S_t|^{-1} \sum_{i \in S_t} z_{ik}, \qquad \sigma_k^2 = |S_t|^{-1} \sum_{i \in S_t} (z_{ik} - \mu_k)^2.$$

Reducing over excluded tokens instead allows their coordinates to change active features even after attention masking. Tokenwise LayerNorm has no such cross-token reduction. Full-instance caches and preprocessing statistics supply other paths. The POMO/ReLD restrictions use support $S_t = \{0, c_t\} \cup U_t$, including currently infeasible unserved customers. Episode controls, including POMO's forced first customer, stay fixed during each intervention.

### A.3 WHY CURRENTLY INFEASIBLE CUSTOMERS REMAIN TASK-RELEVANT

Let the depot be $(0.5, 0.5)$, the current served customer $(0.5, 0)$, vehicle capacity 3 and remaining capacity 1. Two unserved demand-1 customers are $A = (0, 0)$ and $B = (1, 0)$. A third demand-2 customer $H$ is currently infeasible. Compare $H = (0, 0.1)$ and $H = (1, 0.1)$ while fixing every currently feasible feature. Serving the demand-2 current customer from the depot reaches the stated resource state. Exact remaining costs conditional on the first action are:

| Position of $H$ | Return to depot | First $A$ | First $B$ |
|---|---|---|---|
| $(0, 0.1)$ | 3.361633 | 3.559514 | **2.654526** |
| $(1, 0.1)$ | 3.361633 | **2.654526** | 3.559514 |

Enumeration over all customer orders and optional depot returns agrees with an independent dynamic program to $10^{-12}$. The unique optimal first action reverses. A representation containing only the current feasible customers, depot and resource state cannot distinguish these problems. The artifact's `future_customer_certificate.json` records the coordinates, demands, completions and exact costs.

### A.4 PROSPECTIVE POMO PLACEMENT TEST

The placement protocol is fixed after the initial native-model audit and source inspection, before generating its new targets. C50 and C100 each contain 128 instances, seeds 9285050 and 9285100, capacities 40 and 50. Geometry checks exclude prior source, validation and target sets; separate pilot instances are used for technical acceptance.

Four modes share the published checkpoint: native; attention restriction at every encoder layer; attention plus retained-support InstanceNorm at both normalization sites in every layer; and physical packing before the native encoder. All use one forced customer-1 start, greedy decoding, unchanged feasibility/load inputs and no augmentation or retraining. The prediction is that attention alone leaves normalization dependence, whereas joint restriction and packing remove it.

All modes use the same nearest-feasible roll-in states, at the first eligible decision after 25%, 50% and 85% service, separately at customer and depot locations. Eligibility requires an unfinished task and at least two feasible actions. Four local contexts add clipped Gaussian coordinate noise of SD 0.01; four uniform contexts redraw coordinates in $[0, 1]^2$. Only noncurrent served coordinates change. Contexts are indexed by original instance and shared across its states. The primary response is a maximum feasible-action probability change above $10^{-6}$ across that instance's eligible states and draws within a perturbation mode. Bitwise and first-action changes are separate outcomes. The full grid comprises 6,108 state records and 48,864 context comparisons.

All 1,024 complete routes pass independent coverage, capacity, closure and float64 distance replay, with maximum cost discrepancy $3.6 \times 10^{-15}$. Joint restriction and packing each worsen all 128 C50 routes. Their two C50 comparisons against native form the primary Holm family; C100 and attention-only route comparisons are secondary.

Dense restriction and packing agree on all corresponding first actions and complete routes. Their maximum float32 probability discrepancies are 0.0065974 on C50 and 0.0001026 on C100; float64 reduces the discrepancies at those states to $4.91 \times 10^{-12}$ and $5.51 \times 10^{-14}$.

### A.5 PROSPECTIVE RELD CLOSURE AND REOPENING

The official ReLD implementation (Huang et al., 2025) is fixed at commit `a8a3421bce6f01c884d567cd557eb731286473f2`, with the public C100 checkpoint and no retraining. Static source inspection identifies six attention/FF encoder layers, width 128, eight heads, no encoder normalization or global pooling, and four decoder readouts: attention query, current-encoding identity residual, K/V and output-scoring keys. The query attends to the key/value memory; the identity residual adds the current-node encoding directly to the attention output. Output keys turn the refined decoder representation into action scores. The native negative-log-distance term, capacity mapping, clipping at 50 and feasibility rules are retained.

| Set | Intervention | Context | Prob. | Action | Action 95% CI (%) |
|---|---|---|---|---|---|
| C50 | Native decoder mask | local | 128/128 | 87/128 | [59.15, 75.94] |
| C50 | Native decoder mask | uniform | 128/128 | 127/128 | [95.72, 99.98] |
| C50 | Encoder attention masks | local | 128/128 | 56/128 | [35.00, 52.79] |
| C50 | Encoder attention masks | uniform | 128/128 | 127/128 | [95.72, 99.98] |
| C50 | Attention + retained normalization | local | 0/128 | 0/128 | [0.00, 2.84] |
| C50 | Attention + retained normalization | uniform | 0/128 | 0/128 | [0.00, 2.84] |
| C50 | Pack before encoder | local | 0/128 | 0/128 | [0.00, 2.84] |
| C50 | Pack before encoder | uniform | 0/128 | 0/128 | [0.00, 2.84] |
| C100 | Native decoder mask | local | 128/128 | 86/128 | [58.33, 75.22] |
| C100 | Native decoder mask | uniform | 128/128 | 128/128 | [97.16, 100.00] |
| C100 | Encoder attention masks | local | 128/128 | 43/128 | [25.49, 42.48] |
| C100 | Encoder attention masks | uniform | 128/128 | 128/128 | [97.16, 100.00] |
| C100 | Attention + retained normalization | local | 0/128 | 0/128 | [0.00, 2.84] |
| C100 | Attention + retained normalization | uniform | 0/128 | 0/128 | [0.00, 2.84] |
| C100 | Pack before encoder | local | 0/128 | 0/128 | [0.00, 2.84] |
| C100 | Pack before encoder | uniform | 0/128 | 0/128 | [0.00, 2.84] |

Table 4: Instance-level probabilities and first-action responses over all eligible stages/locations. Intervals are marginal exact binomial intervals. These are paired checkpoint interventions, not independent training replicas.

| Set | Intervention | Cost $\Delta$ | 95% instance CI | Primary Holm $p$ |
|---|---|---|---|---|
| C50 | Encoder attention masks | +2.593 | [2.256, 2.939] | – |
| C50 | Attention + retained normalization | +4.306 | [4.040, 4.579] | 0.00004 |
| C50 | Pack before encoder | +4.306 | [4.045, 4.585] | 0.00004 |
| C100 | Encoder attention masks | +2.996 | [2.656, 3.349] | – |
| C100 | Attention + retained normalization | +5.201 | [4.874, 5.522] | – |
| C100 | Pack before encoder | +5.201 | [4.883, 5.522] | – |

Table 5: All frozen POMO route effects. The two C50 closed-boundary contrasts are the primary family. Paired bootstrap intervals use 5,000 draws and mean sign-flip tests use 50,000 Monte Carlo draws with plus-one correction. C100 and attention-only results are secondary. Learned parameters, start and decoding rule are fixed.

The protocol, source, checkpoint and scoring code are frozen before targets. Source-only checks establish native probability/path parity and exact agreement of the all-retained encoder before target generation. The declared matrix is:

| Intervention | Fixed operation and prediction |
|---|---|
| Native | Full encoding; detected response. |
| Refresh full cache | Recompute full encoding; native parity and response. |
| Last layer only | Restrict layer six, refresh all readouts; earlier mixing retains response. |
| All layers / all ports | Restrict all six layers and refresh all four readouts; zero response. |
| Reopen query | Query alone reads full encoding; response returns. |
| Reopen identity | Identity residual alone reads full encoding; response returns. |
| Reopen K/V | Decoder K/V alone read full encodings; response returns. |
| Reopen output key | Scoring keys alone read full encodings; response returns. |

An open arm predicts at least one responding original instance per size; the closed arm predicts no above-threshold response. All sixteen arm-by-size predictions match this locally declared criterion.

C50/C100 have 128 targets each, seeds 9321950/9322000 and capacities 40/50. Coordinates are uniform in $[0, 1]^2$ and integer demands are uniform on 1–9, normalized by capacity. All geometries are distinct and absent from 26,784 registered historical instances; the original pretraining corpus is unavailable. The synthetic loader is used. All arms share batch size 32, one forced customer-1 start, greedy decoding, no augmentation and capacity tolerance $10^{-6}$.

Audit states are native-policy prefixes at the same three completion stages and customer/depot strata as above, requiring at least two feasible actions and one noncurrent served customer. There are 750 C50 and 768 C100 eligible states; eighteen late-depot C50 cells lack an eligible prefix and receive no replacement. Every instance still contributes other states and every route arm. Four local and four uniform contexts per state use the POMO perturbation construction. Depot/current/unserved coordinates, demands, load, visited set and feasible actions remain fixed. Original instances are the statistical units; draws and stages are repeated measurements.

Table 6: ReLD context audits by perturbation mode. Each count is out of 128 original instances, taking a maximum over its declared eligible states and four draws. Probability response uses $10^{-6}$; action change uses argmax disagreement. Repeated states and draws are not independent replicas.

| | | Probability response | | Action change | |
|---|---|---|---|---|---|
| Size | Intervention | Local | Uniform | Local | Uniform |
| C50 | Native | 117 | 128 | 33 | 122 |
| C50 | Refresh full cache | 117 | 128 | 33 | 122 |
| C50 | Last layer only | 111 | 127 | 20 | 119 |
| C50 | All layers / all ports | 0 | 0 | 0 | 0 |
| C50 | Reopen query | 127 | 128 | 2 | 32 |
| C50 | Reopen identity | 128 | 128 | 6 | 63 |
| C50 | Reopen K/V | 128 | 128 | 15 | 106 |
| C50 | Reopen output key | 128 | 128 | 24 | 124 |
| C100 | Native | 124 | 128 | 34 | 121 |
| C100 | Refresh full cache | 124 | 128 | 34 | 121 |
| C100 | Last layer only | 121 | 128 | 27 | 120 |
| C100 | All layers / all ports | 0 | 0 | 0 | 0 |
| C100 | Reopen query | 127 | 128 | 4 | 35 |
| C100 | Reopen identity | 125 | 127 | 11 | 92 |
| C100 | Reopen K/V | 128 | 128 | 17 | 112 |
| C100 | Reopen output key | 128 | 128 | 49 | 128 |

Table 7: Complete-route ReLD outcomes. Intervals are descriptive paired-instance 95% $t$ intervals for intervention minus native cost; no arm was selected using these targets. The sham refresh has identical complete paths.

| Size | Intervention | Mean cost | Difference | 95% interval |
|---|---|---|---|---|
| C50 | Native | 10.661 | +0.000 | Reference |
| C50 | Refresh full cache | 10.661 | +0.000 | [0.000, 0.000] |
| C50 | Last layer only | 10.672 | +0.011 | [-0.017, 0.038] |
| C50 | All layers / all ports | 10.937 | +0.276 | [0.219, 0.334] |
| C50 | Reopen query | 10.955 | +0.294 | [0.234, 0.354] |
| C50 | Reopen identity | 11.383 | +0.722 | [0.634, 0.809] |
| C50 | Reopen K/V | 11.107 | +0.446 | [0.378, 0.515] |
| C50 | Reopen output key | 11.067 | +0.406 | [0.344, 0.469] |
| C100 | Native | 16.122 | +0.000 | Reference |
| C100 | Refresh full cache | 16.122 | +0.000 | [0.000, 0.000] |
| C100 | Last layer only | 16.118 | -0.003 | [-0.040, 0.033] |
| C100 | All layers / all ports | 16.329 | +0.207 | [0.148, 0.265] |
| C100 | Reopen query | 16.440 | +0.318 | [0.260, 0.376] |
| C100 | Reopen identity | 17.728 | +1.606 | [1.480, 1.732] |
| C100 | Reopen K/V | 16.856 | +0.734 | [0.644, 0.825] |
| C100 | Reopen output key | 16.844 | +0.722 | [0.652, 0.792] |

Native and refreshed-full probabilities and routes agree exactly. Complete restriction gives zero probability difference in all 12,144 comparisons; each individual reopened readout restores detected

responses. Full restriction raises mean route cost by 2.5914% on C50 and 1.2818% on C100. Descriptive paired-bootstrap intervals are [2.073%,3.126%] and [0.925%,1.639%] (10,000 original-instance resamples). Individual reopening can worsen route quality more than full restriction, as the complete eight-arm table shows.

All 2,048 routes pass independent physical replay; the maximum difference from official float32 cost is $2.255 \times 10^{-6}$. The 97,152-comparison audit and all route paths reproduce exactly on the original GPU and batch shape. Raw probabilities and source/checkpoint bindings accompany the results.

### A.6 Additional native interfaces and the INViT padding path

The preceding native audit uses published BQ, LEHD, POMO and INViT checkpoints with their native decision interfaces. BQ and LEHD preserve direct/via-depot macro-actions; intermediate committed depot visits are not new decisions. POMO uses one forced customer-1 start. INViT uses action-view size 15, state-view sizes 35/50 and its native strict demand-less-than-capacity rule. All use greedy decoding without augmentation or target adaptation.

Targets contain 128 C20, 512 C50 and 256 C100 instances, seeds 9265020/9265050/9265100. The first 128 per size supply both shared nearest-feasible states and each policy's own states at the three completion stages, with customer/depot strata. The same four local and four uniform served-coordinate interventions preserve the remaining task. The saved audit contains 17,963 states, 143,704 contexts and 4,397 exact-cost states with at most eight remaining customers. Counts below use eligible original instances; the own-policy table includes exact binomial intervals.

Source inspection predicts an INViT preprocessing path: at a depot, an unused kNN slot can contain index $-1$, gathering the last customer before coordinate normalization. All 4,870 detected probability responses occur at depot states where that customer is served. Canonical padding maps only already-masked slots to the current coordinate before normalization, retaining learned weights, valid nodes, candidate rules and masks. This removes all sampled probability responses, with exact float32 agreement between perturbed contexts. The complete-route comparison is:

All 4,480 routes pass physical replay (maximum cost discrepancy $1.5 \times 10^{-14}$). Canonical-minus-native C50/C100 mean-cost tests form the two-comparison primary family: 50,000 paired sign-flip draws with a plus-one correction and paired instance-bootstrap intervals. The intervals span zero; the corrected C50 mean-cost $p$ is 0.36811. The padding repair therefore removes the observed dependency without an established mean-route improvement. Perturbations are indexed by original instance across models.

## B Progress Encoding: Matched Confirmation and Complete Results

This study compares removing explicit absolute progress with retaining it through alternative encodings. A three-initialization development study selected NoStep for independent confirmation. Confirmation uses sixteen new paired initialization/order seeds in each of two source regimes, new source-validation instances and new targets, with a shared teacher corpus. Its primary endpoint is complete-route cost on C100.

### B.1 Seven input interfaces

The backbone has three attention layers of width 128, four heads and zero dropout. Every arm retains full graph context, visited indicators, current location, capacity and the same other state features. The interfaces differ in the designated step embedding and the two counter scalars, one in the node features and one in the decoder query.

NoStep replaces the indexed embedding signal with the learned constant $e_0$ and sets both absolute-step scalars to zero; it does not remove the other state variables. Size-normalized scalar also uses $e_0$, but supplies $t/(2N)$ to both scalar channels, where $N$ is the original customer count. The actual environment counter continues to advance in all interfaces. Sinusoidal encoding uses base 10,000 and a learned $128 \times 128$ projection initialized at zero. The MLP has a 126-unit GELU hidden layer

| Model | Customer local | Customer uniform | Depot local | Depot uniform |
|---|---|---|---|---|
| BQ published | 0/128 | 0/128 | – | – |
| LEHD published | 0/128 | 4/128 | – | – |
| POMO published | 25/128 | 82/128 | 25/116 | 93/116 |
| INViT published | 0/128 | 0/128 | 0/116 | 6/116 |
| INViT + canonical padding | 0/128 | 0/128 | 0/116 | 0/116 |

Table 8: Changed first executed action / eligible instances on shared late C50 states, using four contexts per mode. Local coordinate SD is 0.01. A depot visit inside a BQ/LEHD macro is not a new native decision (dashes). These are fixed published checkpoints, with native decision rules and no augmentation; POMO uses one forced start. Full intervals and own-policy results are in Appendix A.6.

| Model / location | Local $k/N$ | 95% CI (%) | Uniform $k/N$ | 95% CI (%) |
|---|---|---|---|---|
| BQ published / customer | 0/128 | [0.00, 2.84] | 0/128 | [0.00, 2.84] |
| LEHD published / customer | 0/128 | [0.00, 2.84] | 0/128 | [0.00, 2.84] |
| POMO published / customer | 16/128 | [7.32, 19.50] | 47/128 | [28.38, 45.69] |
| POMO published / depot | 13/122 | [5.80, 17.53] | 76/122 | [53.07, 70.91] |
| INViT published / customer | 0/128 | [0.00, 2.84] | 0/128 | [0.00, 2.84] |
| INViT published / depot | 2/111 | [0.22, 6.36] | 31/111 | [19.83, 37.24] |
| INViT + canonical padding / customer | 0/128 | [0.00, 2.84] | 0/128 | [0.00, 2.84] |
| INViT + canonical padding / depot | 0/112 | [0.00, 3.24] | 0/112 | [0.00, 3.24] |

Table 9: Late C50 states on each policy's own native trajectories. Intervals are marginal Clopper–Pearson intervals over eligible independent instances. Different policies can reach different eligible state sets; causal paired comparisons use the shared states in Table 8. A perturbed prefix is feasible, not necessarily generated by that policy.

and a zero-initialized, bias-free output projection. Both retain their common constant component and the two $t/128$ scalars. The inherited embedding-input cap is $u = \min(t, 128)$, including for sinusoidal and MLP; the scalar channels use the actual counter.

Support cap shares lookup training and changes the embedding index to $\min(u, K)$, where $K$ is the maximum supervised counter. It selects its own checkpoint on source validation. A separate fixed-weight comparison applies this cap to lookup states without changing the checkpoint or either scalar channel. In confirmation, all 5,932 supported C20-source states and 14,831 supported mixed-source states retain identical probabilities. Outside support, actions change in 35.10% of 22,312 C20-source states and 30.21% of 13,495 mixed-source states. This directly identifies an index-dependent computation; the route tables below measure the separately trained or source-selected interfaces.

### B.2 Paired training and source-only checkpoint selection

C20 training uses 4,096 demonstrations at capacity 30. Mixed training uses the first 2,048 demonstrations from that same C20 corpus and 2,048 C50 demonstrations at capacity 40. Both corpora use one pinned LKH labeling procedure. Their maximum supervised counters are $K = 23$ and $K = 58$. The two regimes vary instance size, capacity and source-state composition together.

Seeds 68200–68215 pair backbone initialization, $e_0$ and state-sampling order across interfaces. Each fit completes 40 blocks of 173 updates with batch size 512: 6,920 updates and 3,543,040 state presentations. Mixed training alternates sizes, assigning half the updates to each. AdamW uses learning rate $5 \times 10^{-4}$, weight decay 0.01 and gradient clipping at 1; TF32 is disabled. Five main training arms across two regimes and sixteen seeds yield 160 fits. Support cap reuses lookup fits, and size-normalized scalar adds 32 fits under the same training settings.

Source validation contains 256 C20 instances (seed 9821020) and 256 C50 instances (seed 9821050). C20-source selection minimizes C20 cost; mixed selection averages the two costs after division by their respective nearest-neighbor references. Initialization and every fifth block are eligible, with ties retaining the earlier checkpoint. The target tables use the source-selected checkpoint within the full 40-block budget.

| Set | Instances | Cost change | 95% instance CI | Primary Holm $p$ | Better/worse/tied |
|---|---|---|---|---|---|
| C20 | 128 | -0.0054 | [-0.0152, 0.0017] | – | 7/6/115 |
| C50 | 512 | -0.0031 | [-0.0078, 0.0014] | 0.36811 | 70/40/402 |
| C100 | 256 | -0.0045 | [-0.0109, 0.0021] | 0.36811 | 34/26/196 |

Table 10: Same-weight INViT canonical padding minus native padding on fresh complete greedy routes. C50/C100 form the two-comparison primary family; mean-cost sign-flip tests use 50,000 Monte Carlo draws and a plus-one correction. Paired instance bootstrap intervals are unadjusted. Sign tests, reported in the artifact, test direction counts rather than the primary mean-cost question. The average quality improvement remains unresolved.

Table 11: Exact progress interfaces in the new matched study. Here $t$ is the actual environment counter, $u = \min(t, 128)$, $N$ is the original number of customers, and $K$ is the maximum counter in supervised source states. The scalar appears in both node and query features. The learned constant $e_0$ is retained; NoStep does not fix the embedding vector to zero. All rows retain full graph context and the same other state features.

| Interface | Step embedding component | Both scalars |
|---|---|---|
| Lookup | $e_u$ | $t/128$ |
| Support cap | $e_{\min(u,K)}$ | $t/128$ |
| NoStep | $e_0$ | 0 |
| Scalar | $e_0$ | $t/128$ |
| Sinusoidal | $e_0 + W(\mathrm{PE}(u) - \mathrm{PE}(0))$ | $t/128$ |
| MLP | $e_0 + f(u/128) - f(0)$ | $t/128$ |
| Size-normalized scalar | $e_0$ | $t/(2N)$ |

**The added normalized control.** Size-normalized scalar was declared while the original confirmation was still training on source data, before any confirmation target generation. Its two scalars change from $t/128$ to $t/(2N)$ without a denominator search or additional module. All 32 added fits use the same seeds, data, initial tensors, optimizer and exposure schedules as scalar. The added five quality conditions below supplement the original ten-condition rule.

### B.3 Fresh source ranking and fixed comparison types

Development source validation fixed the continuous comparator $B$ as MLP for C20 training and sinusoidal for mixed training. On the fresh confirmation validation set, ordinary scalar has the lowest mean source score among scalar, sinusoidal and MLP in both regimes. This new ranking does not change $B$: validation selects a checkpoint within each fixed type. All seven interfaces remain in target evaluation, and the size-normalized guard remains part of the final decision.

### B.4 Targets, paired inference and the complete decision rule

All interfaces share 512 instances each at C20, C50 and C100, and 256 each at C200 and clustered C100. Their seeds are respectively 9822020, 9822050, 9822100, 9822200 and 9822150; capacities are 30, 40, 50, 50 and 50. Uniform targets use coordinates in the unit square and integer demands 1–9. Clustered customers use three uniform centers, equiprobable cluster assignments and Gaussian noise with standard deviation 0.07, clipped to the square; the depot is uniform. Geometry checks exclude registered historical, source and validation instances, allowing for customer permutations. Targets are generated after source selection freezes. Decoding is autonomous greedy, without augmentation or search.

The original joint claim requires at least 1% C100 improvement over lookup and at most 1% loss against $B$ in both source regimes. Source-size guards additionally require at most 1% loss against lookup and $B$: at C20 after C20 training, and at both C20 and C50 after mixed training. These give four primary conditions and six source guards. For each comparator $b$, define the paired initialization contrast

$$X_s = \frac{1}{m}\sum_{i=1}^{m}\left(C_{s,i}^{\mathrm{NoStep}} - qC_{s,i}^{b}\right), \qquad U = \overline{X} + t_{0.975,15}\frac{\mathrm{sd}(X_s)}{\sqrt{16}}, \tag{5}$$

where $q = 0.99$ for lookup C100 comparisons and $q = 1.01$ for quality guards. A condition passes only if $U < 0$; the original joint claim requires all ten conditions. This is an intersection–union decision at one-sided $\alpha = 0.025$. The additional normalized-control guard requires the same 1% noninferiority condition at C100 in both regimes, C20 in both regimes, and mixed-source C50: all five must also pass for the complete NoStep recommendation.

The statistical units are sixteen paired initialization/order seeds, conditional on the shared corpora, validation and target panels. Crossed seed/instance resampling with 10,000 draws supplies descriptive 95% intervals for the same contrasts. All conditions and every secondary target remain below; individual target instances are not training replicates.

### B.5 Complete route outcomes

Table 12: C20 source: every confirmation mean across 16 initialization seeds. C20, C50 and C100 each contain 512 instances; C200 and clustered C100 each contain 256. C100 is primary, source sizes supply quality guards, and other cells are secondary. No target cell selects an interface or checkpoint.

| Interface | C20 | C50 | C100 | C200 | C100 cluster |
|---|---|---|---|---|---|
| Lookup | 6.863 | 12.940 | 23.554 | 66.109 | 22.266 |
| Support cap | 6.862 | 12.519 | 22.588 | 49.874 | 19.115 |
| NoStep | 6.762 | 11.920 | 19.678 | 39.978 | 19.618 |
| Scalar | 6.755 | 11.869 | 19.266 | 44.020 | 17.680 |
| Sinusoidal | 6.783 | 12.209 | 20.252 | 44.127 | 16.751 |
| MLP | 6.761 | 13.093 | 34.887 | 93.730 | 31.912 |
| Normalized scalar | 6.766 | 11.872 | 19.136 | 41.446 | 18.700 |

Table 13: C20 + C50 source: every confirmation mean across 16 initialization seeds. C20, C50 and C100 each contain 512 instances; C200 and clustered C100 each contain 256. C100 is primary, source sizes supply quality guards, and other cells are secondary. No target cell selects an interface or checkpoint.

| Interface | C20 | C50 | C100 | C200 | C100 cluster |
|---|---|---|---|---|---|
| Lookup | 6.866 | 11.882 | 20.084 | 46.145 | 18.487 |
| Support cap | 6.866 | 11.882 | 18.869 | 38.234 | 17.212 |
| NoStep | 6.784 | 11.745 | 17.905 | 32.616 | 15.683 |
| Scalar | 6.772 | 11.727 | 18.124 | 49.006 | 16.411 |
| Sinusoidal | 6.792 | 11.762 | 18.012 | 33.955 | 15.741 |
| MLP | 6.778 | 11.709 | 18.427 | 57.690 | 17.324 |
| Normalized scalar | 6.775 | 11.728 | 17.900 | 32.684 | 15.688 |

Table 14: All 10 original conditions. $L = \overline{C_{\mathrm{NoStep}} - fC_{\mathrm{control}}}$; $f = 0.99$ for the lookup C100 superiority conditions and $f = 1.01$ otherwise. $U$ is the one-sided 97.5% paired-initialization $t$ upper bound. A condition passes only if $U < 0$. The crossed seed/instance 95% interval is descriptive. $\Delta\%$ is the change in grand mean route cost, not the linear contrast.

| Source / target | Control | $\Delta\%$ | $L$ | $U$ | Crossed interval | Pass |
|---|---|---|---|---|---|---|
| C20 / C100 | Lookup | -16.45 | -3.640 | -1.831 | [-5.291, -2.082] | Yes |
| C20 / C100 | MLP | -43.60 | -15.558 | -8.428 | [-22.262, -9.527] | Yes |
| C20 / C20 | Lookup | -1.47 | -0.169 | -0.145 | [-0.198, -0.141] | Yes |
| C20 / C20 | MLP | +0.01 | -0.067 | -0.051 | [-0.089, -0.046] | Yes |
| Mixed / C100 | Lookup | -10.85 | -1.978 | -1.467 | [-2.465, -1.534] | Yes |
| Mixed / C100 | Sinusoidal | -0.59 | -0.287 | -0.184 | [-0.380, -0.190] | Yes |
| Mixed / C20 | Lookup | -1.20 | -0.151 | -0.124 | [-0.180, -0.120] | Yes |
| Mixed / C20 | Sinusoidal | -0.12 | -0.076 | -0.055 | [-0.102, -0.050] | Yes |
| Mixed / C50 | Lookup | -1.15 | -0.256 | -0.211 | [-0.300, -0.209] | Yes |
| Mixed / C50 | Sinusoidal | -0.14 | -0.135 | -0.097 | [-0.175, -0.095] | Yes |

Table 15: All 5 additional normalized-control conditions. $L = \overline{C_{\text{NoStep}} - fC_{\text{control}}}$; $f = 0.99$ for the lookup C100 superiority conditions and $f = 1.01$ otherwise. $U$ is the one-sided 97.5% paired-initialization $t$ upper bound. A condition passes only if $U < 0$. The crossed seed/instance 95% interval is descriptive. $\Delta\%$ is the change in grand mean route cost, not the linear contrast.

| Source / target | Control | $\Delta\%$ | $L$ | $U$ | Crossed interval | Pass |
|---|---|---|---|---|---|---|
| C20 / C100 | Normalized | +2.83 | +0.351 | +0.759 | [+0.028, +0.749] | No |
| C20 / C20 | Normalized | -0.06 | -0.072 | -0.047 | [-0.098, -0.044] | Yes |
| Mixed / C100 | Normalized | +0.03 | -0.174 | -0.076 | [-0.263, -0.082] | Yes |
| Mixed / C20 | Normalized | +0.14 | -0.058 | -0.037 | [-0.083, -0.034] | Yes |
| Mixed / C50 | Normalized | +0.15 | -0.100 | -0.068 | [-0.135, -0.063] | Yes |

All ten original conditions pass. NoStep reduces C100 mean cost relative to lookup by 16.45% after C20 training and 10.85% after mixed training. The normalized comparison provides the additional design result: C20-source C100 cost is 19.136 with normalized progress and 19.678 with NoStep, an observed 2.83% increase for deletion. The margin-adjusted upper bound is $U = +0.759 > 0$ distance units (Equation 5), so this condition fails; the other four normalized conditions pass. The mixed-source C100 means are nearly the same, 17.900 and 17.905. Thus normalization provides an effective retained encoding. The complete NoStep recommendation fails its fixed C20-source C100 quality guard.

The full target tables also preserve outcomes beyond C100. For example, C20-source C200 costs are 39.978 for NoStep and 41.446 for normalized scalar; on clustered C100, sinusoidal reaches 16.751 compared with 18.700 for normalized scalar. These target results show how the preferred encoding changes with instance size and spatial distribution.

**Verification and retained records.** Checks cover paired initialization, optimizer counters, exposures and earliest eligible source optima, including 576 original and 96 added source-budget references. Independent physical scorers verify all 393,216 original and 65,536 normalized-control target routes for coverage, capacity, depot closure and cost. The artifact retains per-seed means, source curves, checkpoint hashes and execution records. The separate compact-training study follows below.

### B.6 Compact representation and training cost

A separate seven-seed experiment (64100–64106) tests the resource effect of physically packing depot, current and all unserved customers, including those currently beyond capacity. Full/random, full/bucketed and compact residual/bucketed arms share backbone initialization, optimizer settings and 4,096 C20 LKH demonstrations containing 88,171 nontrivial states. Each completes 40 epochs of 173 updates with batch size 512, giving 6,920 matched updates. Bucketing stably sorts a randomized state order by retained size, forms batches, then randomizes their order; the full bucketed control uses the same schedule.

Training uses one RTX 3090, four CPU threads, float32 and disabled TF32. Synchronized timing includes initialization, optimization, source validation and writes; teacher generation is reused. Source validation uses 512 C20 instances (seed 9401020) every five epochs and selects the best feasible checkpoint available by each budget. All checkpoints freeze before fresh shared C20/C50/C100 targets: 1,024/1,024/256 instances, seeds 9405020/9405050/9405100. Greedy routes receive independent physical replay.

The primary estimands average seven paired seed ratios, each formed after averaging over common target instances. Compact training reduces total time by 22.325%, with a one-sided 95% lower bound of 21.691%. C20 cost changes by $-0.403\%$, with one-sided 97.5% upper bound $+0.523\%$, within a 1% difference. C50 changes by $+0.665\%$ with upper bound $+3.239\%$, so the joint C20/C50 quality criterion fails. C100 is secondary. These results support the matched-update C20 saving. At the separate 20-epoch budget, full/random takes 48.23 seconds and its secondary one-sided 95% bounds against 40-epoch full/random are $+0.809\%$ on C20 and $+0.669\%$ on C50. That shorter reference separates the matched-update comparison from minimum time to reach a given quality. Complete budget curves and per-seed records remain with the experiment data.

| Training scheme | Time (s) | C20 | C50 | C100 |
|---|---|---|---|---|
| Full / random | 96.15 | 6.7609 | 11.9796 | 19.4437 |
| Full / bucketed | 96.16 | 6.7894 | 11.9021 | 18.7447 |
| Residual / packed + bucketed | 74.69 | 6.7333 | 12.0535 | 19.1020 |

Table 16: Complete training-pipeline time and mean geometric route cost at 40 epochs (seven paired seeds). Time includes initialization, optimization, source validation and checkpoint writes. C20/C50 have 1,024 shared targets; C100 has 256 and is secondary. Noninferiority is decided from paired-seed upper bounds, not these rounded means.

# C Online Solver: Specification and Complete Evaluation

## C.1 Full Candidate–Successor Readout Specification

**Legal observations and successor support.** The graph-pointer backbone observes the current routing state and retains observed served context. Let 0 denote the depot, $c_t$ the current node, $Q$ capacity, $q_t$ remaining capacity, $d_i$ observed demand, and $R_t, U_t$ the revealed and unserved sets. For candidate $i$, the relation support is

$$J_i = (R_t \cap U_t) \setminus \{0, c_t, i\}. \tag{6}$$

It includes currently capacity-infeasible customers, which can be served after replenishment. Only candidates in the environment's legal action set can be selected; the action mask is reapplied after the dense score correction. All four arms sanitize unrevealed coordinates, demands, associated distances, future release fields and unobserved realized demands before rollout, anchor and gradient-recomputation calls. The head also masks hidden demands and pair distances. Total customer count and the current observed state are available to every arm.

**Capacity-conditioned pair features.** For a legal customer action, $q_i = q_t - d_i$; the depot resets $q_0 = Q$. With direct distance $\ell_{ij}$ and depot-mediated distance $v_{ij} = \ell_{i0} + \ell_{0j}$, define

$$r_{ij} = [\ell_{ij}, v_{ij}, v_{ij} - \ell_{ij}, d_j/Q, q_i/Q, (q_i - d_j)/Q]. \tag{7}$$

Distances use native coordinate units. A shared $6 \to 16 \to 16$ network $\psi$, with GELU after each linear layer, embeds every supported pair. The two capacity channels are

$$J_i^{\mathrm{dir}} = \{j \in J_i : d_j \le q_i + 10^{-6}\}, \qquad J_i^{\mathrm{rep}} = J_i \setminus J_i^{\mathrm{dir}}. \tag{8}$$

They distinguish direct service from service requiring replenishment while retaining both distances. Each successor enters one channel; the descriptor aggregates possible continuations rather than prescribing the next customer.

**Aggregation and additive scoring.** For $g \in \{\mathrm{dir}, \mathrm{rep}\}$ and $n_i^g = |J_i^g|$, compute

$$m_i^g = \frac{\sum_{j \in J_i^g} \psi(r_{ij})}{\max(1, n_i^g)}, \quad f_i^g = \frac{n_i^g}{\max(1, |J_i|)}, \quad p_i^g = \mathbf{1}[n_i^g > 0]. \tag{9}$$

Empty channels return zero mean, fraction and presence flag. Concatenating the two 16-dimensional means, two fractions, two indicators and $[\ell_{c_t i}, \ell_{i0}, d_i/Q, \mathbf{1}[i = 0]]$ gives 40 features. A $40 \to 32 \to 1$ network, with GELU after its first layer, produces $\Delta_i$ and the legal-action score $s_i = s_i^{\mathrm{base}} + \Delta_i$. The final weight and bias start at zero; other layers use ordinary initialization. Initial logits therefore equal the normalized backbone's, with gradients reaching earlier head layers after the output becomes nonzero. Dense pair construction has quadratic node-count cost.

**Controls.** The current-capacity head has the same architecture and initialization, substituting $q_i = q_t$ for customer candidates in both pair features and channel membership; it retains the depot reset $q_0 = Q$. The normalized control is the backbone. The full adapter adds attention and feed-forward query/pointer-key readouts over all observed context, including served nodes. It tests generic expansion with its own parameter count. All arms share legal observations, source data and training-time caps; the resource ledger records their different update and exposure counts.

**Stochastic demands.** The environment reveals realized demand on arrival and executes automatic recourse. Policies use public means $\bar{d}_i$ in the same head, so $q_t - \bar{d}_i$ and its capacity partition are nominal features, not realized or expected post-recourse capacity. The original D50-trained models are evaluated for stochastic transfer; the follow-up in Appendix C.3.2 trains and selects on stochastic demand through this same interface.

### C.2 Frozen Action-Conditioned Study Protocol

**Endpoint, models and task.** The primary endpoint is complete greedy strict-D50 route cost after source-selected dynamic adaptation. The normalized, full-adapter, current-relation and afterstate-relation policies have respectively 695,320, 893,600, 697,049 and 697,049 trainable parameters. The relation variants differ only in customer-capacity conditioning. D50 uses capacity 40, independent release times over horizon 10.98, half the customers initially revealed and zero service time. Total customer count is known; future coordinates, demands and release fields are removed from policy and recomputation inputs. Hidden realizations are used only for environment transitions.

**Training and resource accounting.** Every fit uses the same verified LKH corpus of 2,048 C20 and 2,048 C50 demonstrations, a 600-second supervised cap and a 300-second D50 adaptation cap on the same scheduled GPU. Caps include model/optimizer setup, validation and checkpoint writes. Shared teacher verification, state construction and resident-data preparation are itemized separately. The allowed overrun is the greater of 2% of the cap and the final validation/checkpoint unit; larger overruns prevent stage finalization. The ledger retains all actual durations, updates, exposures and unused budgets. Supervised learning uses AdamW, batch 512 and learning rate $5 \cdot 10^{-4}$. Self-critical adaptation uses 128 fresh instances per update, learning rate $3 \cdot 10^{-6}$ and KL weight $0.005$ to the frozen supervised checkpoint. Complete sampled/greedy trajectories determine costs; illegal or incomplete trajectories fail the run. Chunked sampled-state recomputation accumulates the full-trajectory gradient before one optimizer update.

**Source selection and target generation.** Initialization and eight equally spaced resource checkpoints are eligible per stage, with the earliest strict minimum selected. The supervised score averages C20/C50 validation costs normalized by their nearest-neighbor costs; adaptation uses separate strict-D50 validation. Each validation task has 256 instances. Every source selection in a phase is frozen before that phase's targets are generated. The saved registry checks overlap among historical, source, validation and target geometries.

**Evaluation units and inference.** Two development lineages precede eight fixed confirmation lineages, completed regardless of development direction. The common confirmation panels contain 1,024 D50 instances and 256 each for C20, C50, C100, D20, VRPSD20 and dynamic VRPSD50. Both supervised and adapted checkpoints are evaluated on every panel. The eight paired complete training lineages are the primary units, conditional on the teacher and target sets. Afterstate-minus-control differences use two-sided paired-$t$ 95% intervals, exact paired sign-flip tests and Holm correction across the three primary comparisons. Crossed seed/instance resampling provides sensitivity intervals. Secondary tasks retain all outcomes; stochastic panels measure transfer from deterministic adaptation using public mean-demand features.

**Physical replay and classical references.** Saved primitive inputs and per-decision traces support independent NumPy replay of legality, completion, capacity, replenishment and each transition. Replay computes distance in float64 and reconstructs the float32 event clock. Fresh-study distances use direct float64 coordinate differences converted to float32; historical teacher caches retain their recorded convention. After training, rolling nearest-neighbor and LKH run on frozen D50 panels. Comparator budgets and fallback accounting are specified in Appendix C.3.3.

### C.3 Complete Action-Conditioned Solver Results

All declared arms and tasks retain source-selected supervised and adapted checkpoints. Inference uses eight paired complete training lineages, conditional on the shared teacher and target sets.

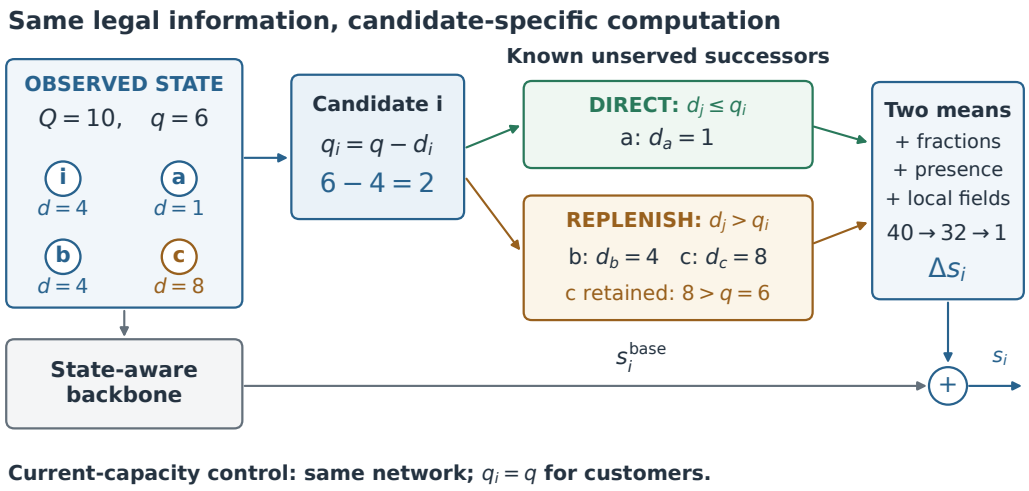


Figure 4: Candidate-conditioned continuation readout using the same legal observation as the state-aware backbone. With $Q = 10$, $q = 6$ and candidate demand $d_i = 4$, the after-action capacity is $q_i = 2$. Known unserved customer $c$ has demand 8: it is currently infeasible, but remains in the replenishment channel because a later depot visit can restore capacity. Unrevealed customer attributes are excluded. Each channel supplies a mean pair embedding, a fraction and a presence flag; these are combined with candidate-local features to produce an additive score correction. The same-architecture current-capacity control substitutes $q_i = q$ for customer candidates; both variants reset capacity for a depot action.

Table 17: Complete confirmation retention/transfer results, mean ± sample SD across all eight lineages; $n$ is the number of shared target instances per lineage. All seven tasks and both stages are retained. Lower cost is better. BC uses static behavioral-cloning weights; adapted means source-selected strict D50 adaptation. The stochastic tasks receive no stochastic-demand training; capacity relations use public demand means, not realized future demand. Secondary results are descriptive and receive no primary Holm label.

| Task | Stage | $n$ | Normalized | Full adapter | Current relation | Afterstate relation |
|---|---|---|---|---|---|---|
| D50 | BC | 1024 | $13.856 \pm 0.102$ | $13.849 \pm 0.112$ | $13.858 \pm 0.095$ | $13.879 \pm 0.108$ |
| D50 | Adapted | 1024 | $13.710 \pm 0.097$ | $13.694 \pm 0.100$ | $13.721 \pm 0.095$ | $13.723 \pm 0.115$ |
| C20 | BC | 256 | $6.829 \pm 0.061$ | $6.768 \pm 0.046$ | $6.813 \pm 0.039$ | $6.805 \pm 0.022$ |
| C20 | Adapted | 256 | $6.796 \pm 0.053$ | $6.760 \pm 0.047$ | $6.805 \pm 0.042$ | $6.770 \pm 0.040$ |
| C50 | BC | 256 | $11.848 \pm 0.074$ | $11.758 \pm 0.062$ | $11.802 \pm 0.071$ | $11.867 \pm 0.057$ |
| C50 | Adapted | 256 | $11.797 \pm 0.107$ | $11.741 \pm 0.059$ | $11.769 \pm 0.058$ | $11.787 \pm 0.077$ |
| C100 | BC | 256 | $18.016 \pm 0.198$ | $17.864 \pm 0.145$ | $17.854 \pm 0.126$ | $18.006 \pm 0.082$ |
| C100 | Adapted | 256 | $17.967 \pm 0.205$ | $17.785 \pm 0.125$ | $17.824 \pm 0.090$ | $17.927 \pm 0.108$ |
| D20 | BC | 256 | $8.020 \pm 0.045$ | $8.027 \pm 0.059$ | $8.027 \pm 0.032$ | $8.032 \pm 0.038$ |
| D20 | Adapted | 256 | $7.987 \pm 0.061$ | $7.986 \pm 0.038$ | $7.993 \pm 0.043$ | $7.994 \pm 0.044$ |
| VRPSD20 | BC | 256 | $6.935 \pm 0.057$ | $6.917 \pm 0.052$ | $6.913 \pm 0.035$ | $6.939 \pm 0.047$ |
| VRPSD20 | Adapted | 256 | $6.921 \pm 0.054$ | $6.926 \pm 0.058$ | $6.913 \pm 0.044$ | $6.932 \pm 0.057$ |
| DVRPSD50 | BC | 256 | $14.168 \pm 0.113$ | $14.135 \pm 0.103$ | $14.184 \pm 0.087$ | $14.199 \pm 0.106$ |
| DVRPSD50 | Adapted | 256 | $14.109 \pm 0.102$ | $14.076 \pm 0.101$ | $14.132 \pm 0.068$ | $14.146 \pm 0.094$ |

### C.3.1 Changes under dynamic adaptation

Supervised-to-adapted changes pair checkpoints within each lineage and are descriptive. Training uses deterministic D50; the stochastic panels measure cross-task transfer.

### C.3.2 Exploratory stochastic-specific adaptation

A separate protocol, frozen before follow-up training and target generation, adapts all 32 supervised starts: four arms and eight reused lineages 92240–92247. Each starts from its supervised checkpoint,

Table 18: Prespecified D50 contrasts after adaptation: action-afterstate relation minus comparator; negative values favor the candidate. Confidence intervals use the paired $t$ distribution over eight complete training lineages. Two-sided exact sign-flip tests use all 256 sign assignments; Holm adjustment covers exactly these three contrasts. The 95% intervals are individual, not multiplicity-adjusted. Inference is conditional on the shared teacher and common test panel.

| Comparator | $\Delta$ cost | Relative | Paired 95% CI | Exact $p$ | Holm $p$ |
|---|---|---|---|---|---|
| Normalized GNN | +0.0133 | +0.10% | [-0.1134, +0.1400] | 0.8125 | 1.0000 |
| Full-context adapter | +0.0287 | +0.21% | [-0.0499, +0.1072] | 0.3984 | 1.0000 |
| Current-capacity relation | +0.0017 | +0.01% | [-0.1568, +0.1601] | 0.9922 | 1.0000 |

| Task | Arm | BC | Adapted | $\Delta$ [descriptive 95% CI] | Relative $\Delta$ [95% CI] |
|---|---|---|---|---|---|
| D50 | Normalized | 13.856 | 13.710 | -0.147 [-0.203, -0.090] | -1.06% [-1.46, -0.65] |
| D50 | Full adapter | 13.849 | 13.694 | -0.155 [-0.199, -0.110] | -1.11% [-1.43, -0.80] |
| D50 | Current relation | 13.858 | 13.721 | -0.137 [-0.211, -0.062] | -0.99% [-1.52, -0.45] |
| D50 | Afterstate relation | 13.879 | 13.723 | -0.157 [-0.233, -0.080] | -1.13% [-1.68, -0.58] |
| C20 | Normalized | 6.829 | 6.796 | -0.033 [-0.067, +0.001] | -0.48% [-0.98, +0.01] |
| C20 | Full adapter | 6.768 | 6.760 | -0.008 [-0.021, +0.005] | -0.12% [-0.31, +0.08] |
| C20 | Current relation | 6.813 | 6.805 | -0.008 [-0.030, +0.013] | -0.12% [-0.44, +0.20] |
| C20 | Afterstate relation | 6.805 | 6.770 | -0.035 [-0.052, -0.017] | -0.51% [-0.77, -0.25] |
| C50 | Normalized | 11.848 | 11.797 | -0.052 [-0.102, -0.002] | -0.44% [-0.86, -0.01] |
| C50 | Full adapter | 11.758 | 11.741 | -0.017 [-0.042, +0.008] | -0.14% [-0.36, +0.07] |
| C50 | Current relation | 11.802 | 11.769 | -0.033 [-0.066, -0.001] | -0.28% [-0.56, -0.01] |
| C50 | Afterstate relation | 11.867 | 11.787 | -0.080 [-0.124, -0.035] | -0.67% [-1.05, -0.30] |
| C100 | Normalized | 18.016 | 17.967 | -0.049 [-0.135, +0.037] | -0.27% [-0.75, +0.20] |
| C100 | Full adapter | 17.864 | 17.785 | -0.079 [-0.113, -0.045] | -0.44% [-0.63, -0.25] |
| C100 | Current relation | 17.854 | 17.824 | -0.030 [-0.077, +0.017] | -0.17% [-0.43, +0.09] |
| C100 | Afterstate relation | 18.006 | 17.927 | -0.079 [-0.155, -0.003] | -0.44% [-0.86, -0.02] |
| D20 | Normalized | 8.020 | 7.987 | -0.033 [-0.052, -0.014] | -0.41% [-0.64, -0.18] |
| D20 | Full adapter | 8.027 | 7.986 | -0.041 [-0.076, -0.006] | -0.51% [-0.95, -0.08] |
| D20 | Current relation | 8.027 | 7.993 | -0.034 [-0.051, -0.017] | -0.42% [-0.63, -0.21] |
| D20 | Afterstate relation | 8.032 | 7.994 | -0.038 [-0.060, -0.017] | -0.48% [-0.74, -0.21] |
| VRPSD20 | Normalized | 6.935 | 6.921 | -0.013 [-0.034, +0.008] | -0.19% [-0.49, +0.11] |
| VRPSD20 | Full adapter | 6.917 | 6.926 | +0.009 [-0.011, +0.029] | +0.12% [-0.16, +0.41] |
| VRPSD20 | Current relation | 6.913 | 6.913 | -0.001 [-0.029, +0.028] | -0.01% [-0.42, +0.41] |
| VRPSD20 | Afterstate relation | 6.939 | 6.932 | -0.007 [-0.039, +0.025] | -0.10% [-0.57, +0.36] |
| DVRPSD50 | Normalized | 14.168 | 14.109 | -0.059 [-0.104, -0.015] | -0.42% [-0.73, -0.11] |
| DVRPSD50 | Full adapter | 14.135 | 14.076 | -0.059 [-0.104, -0.014] | -0.41% [-0.73, -0.10] |
| DVRPSD50 | Current relation | 14.184 | 14.132 | -0.051 [-0.100, -0.003] | -0.36% [-0.71, -0.02] |
| DVRPSD50 | Afterstate relation | 14.199 | 14.146 | -0.053 [-0.103, -0.003] | -0.37% [-0.72, -0.02] |

Table 19: All four arms and seven tasks (confirmation; 8 paired training lineages). Changes are adapted minus BC, so negative values favor adaptation. Both stages are source-selected. Intervals are unadjusted paired-$t$ intervals across lineages; relative intervals average each lineage's percentage change. Only deterministic strict D50 is used for adaptation. VRPSD20/DVRPSD50 measure zero-shot transfer to stochastic demands, not stochastic-specific training. Static intervals do not establish noninferiority. These descriptive contrasts do not replace the prespecified between-arm D50 comparisons.

receives a fresh 300-second inclusive strict-dynamic-VRPSD50 budget, and retains the original D50 primary comparisons. Inference is conditional on these shared starts, not independent full-pipeline replications.

**Task and source selection.** Capacity is 40, half the customers are initially revealed, release horizon is 10.98 and service time is zero. Capped independent Poisson demands are revealed on arrival.

Policy, fixed anchor and gradient-recomputation calls exclude unrevealed actual demands and future customer attributes; the relation head uses public mean demands as nominal capacity features. Self-critical updates use 128 fresh instances, learning rate $3 \times 10^{-6}$, the original advantage normalization and KL weight $0.005$ to the supervised checkpoint. The cap includes setup, loading, validation and writes with the original atomic-unit tolerance. A 256-instance validation panel (seed 292309501) selects the earliest strict minimum among initialization and eight scheduled validations. All 32 selections are frozen before target generation.

**Resources and evaluation.** All fits satisfy the tolerance, taking 297.47–299.62 seconds and using over 99.1% of their caps, with all nine validations retained. Update counts are 183–188 for normalized, 158–165 for full adapter, 145–152 for current relations and 146–151 for afterstate relations. Both supervised and adapted checkpoints run on 512 fresh DVRPSD50 instances (seed 292309502) and 256 fresh D50 retention instances (seed 292309503): 128 cells and 49,152 routes. Geometry checks exclude source, validation, historical and streaming training instances. Independent replay verifies legal completion and physical cost on every saved route.

**Statistics and outcomes.** The three adapted DVRPSD50 afterstate-minus-control contrasts use the eight paired starting lineages, individual paired-$t$ 95% intervals, exact two-sided sign flips over all 256 assignments and Holm correction within these three exploratory comparisons. Adaptation changes and D50 retention are descriptive. Arm-level stochastic cost reductions are 0.53–0.82%, and D50 retention reductions are 0.56–0.83%, computed relative to each arm's supervised mean. All eight absolute-change intervals lie below zero. Every lineage improves on DVRPSD50; on D50, eight improve for normalized and seven for each other arm. Afterstate cost exceeds normalized, full adapter and current relations by 0.0551, 0.0748 and 0.0214. Their intervals include zero and Holm $p$-values are 0.7344, 0.3984 and 0.7344: the study supports stochastic adaptation without establishing an incremental afterstate benefit.

### C.3.3 Classical route-quality references

Rolling nearest-neighbor and rolling LKH use the same full 1,024-instance D50 panel as the neural configurations. LKH uses one trial, one run and a 50-ms cap per subprocess, with up to three vehicle-count attempts per decision and fallback retained. Its mean cost is 14.1626, compared with 15.5177 for nearest neighbor and 13.6942–13.7229 for the neural policies. All eight lineage means of every adapted arm lie below both reference means, with negative descriptive intervals. These intervals measure training variation conditional on the common panel and fixed classical executions.

LKH fallback occurs on 7,995/60,348 decisions (13.248%) or 7,995/59,324 residual-solver calls (13.477%). Among 176,704 vehicle-count attempts, 37,934 time out (21.468%), 1,272 have another error and 137,498 succeed. The 50-ms limit applies per subprocess; later attempts and fallback remain in complete-route cost and feasibility summaries.

### C.3.4 Training resources and source selection

The common caps include initialization, optimization, validation and checkpoint writes. Table 23 summarizes all 64 fit-stage records; Table 3 reports elapsed times. A BC update samples 512 supervised states, and a D50 adaptation update samples 128 instances. Thus matched time permits different update and exposure counts across architectures.

Fits run serially on the scheduled GPU. Shared teacher verification and state preparation take 2.756 seconds once, outside all per-fit caps. Every stage retains initialization and eight resource-spaced validation checks, selecting the earliest minimum under the rule in Appendix C.2. One BC stage resumes its model and optimizer with prior work and restoration charged to the same 600-second cap. Its complete charge, validation history and checkpoint identity remain in the per-fit records with all other stages.

Table 20: Exploratory stochastic-specific adaptation from the same eight BC lineages. Lower cost is better. Changes are descriptive; D50 measures retention on a separate fresh panel.

| Task | Arm | BC | Adapted | Δ [paired 95% CI] |
|---|---|---|---|---|
| DVRPSD50 | Normalized | 14.217 | 14.100 | -0.116 [-0.169, -0.064] |
| DVRPSD50 | Full adapter | 14.161 | 14.081 | -0.080 [-0.146, -0.014] |
| DVRPSD50 | Current relation | 14.209 | 14.134 | -0.075 [-0.110, -0.040] |
| DVRPSD50 | Afterstate relation | 14.250 | 14.156 | -0.095 [-0.147, -0.042] |
| D50 | Normalized | 13.838 | 13.760 | -0.078 [-0.119, -0.037] |
| D50 | Full adapter | 13.817 | 13.703 | -0.114 [-0.178, -0.051] |
| D50 | Current relation | 13.865 | 13.777 | -0.088 [-0.147, -0.029] |
| D50 | Afterstate relation | 13.827 | 13.743 | -0.084 [-0.144, -0.024] |

Table 21: Supplementary adapted DVRPSD50 comparisons: afterstate relation minus each control over eight paired lineages. Exact two-sided sign flips enumerate all 256 assignments; Holm correction covers these three exploratory comparisons only.

| Control | Δ [paired 95% CI] | Exact $p$ | Holm $p$ |
|---|---|---|---|
| Normalized | +0.055 [-0.082, +0.192] | 0.3672 | 0.7344 |
| Full adapter | +0.075 [-0.033, +0.183] | 0.1328 | 0.3984 |
| Current relation | +0.021 [-0.079, +0.122] | 0.6328 | 0.7344 |

Table 22: External quality references on all 1,024 frozen D50 instances. All four adapted policies retain all eight source-selected training lineages; the same complete classical routes are used for each. Negative neural-minus-reference differences favor the policy. Descriptive 95% paired-t intervals vary the eight training lineages conditional on the target set and classical run. These comparisons do not replace the three primary component contrasts or carry their Holm-adjusted decision rule. Classical and neural training/decoding resources are different.

| Policy | Reference | Neural cost | Reference cost | Δ | 95% interval |
|---|---|---|---|---|---|
| Normalized GNN | Nearest | 13.7096 | 15.5177 | -1.8081 | [-1.8890, -1.7271] |
| Normalized GNN | Rolling LKH | 13.7096 | 14.1626 | -0.4530 | [-0.5340, -0.3721] |
| Full-context adapter | Nearest | 13.6942 | 15.5177 | -1.8234 | [-1.9074, -1.7395] |
| Full-context adapter | Rolling LKH | 13.6942 | 14.1626 | -0.4684 | [-0.5524, -0.3845] |
| Current-capacity relation | Nearest | 13.7212 | 15.5177 | -1.7964 | [-1.8754, -1.7174] |
| Current-capacity relation | Rolling LKH | 13.7212 | 14.1626 | -0.4414 | [-0.5204, -0.3624] |
| Action-afterstate relation | Nearest | 13.7229 | 15.5177 | -1.7947 | [-1.8906, -1.6989] |
| Action-afterstate relation | Rolling LKH | 13.7229 | 14.1626 | -0.4397 | [-0.5356, -0.3438] |

Table 23: Completed update ranges over eight paired training lineages under the common 600-second BC and 300-second D50 adaptation caps.

| Policy | BC updates | D50 adaptation updates |
|---|---|---|
| Normalized backbone | 30,386–31,760 | 186–193 |
| Full-context adapter | 23,863–24,586 | 160–164 |
| Current-capacity relation | 24,532–25,000 | 145–149 |
| Afterstate relation | 24,234–24,964 | 145–150 |

# D REPRODUCTION AND VERIFICATION

**Units and physical replay.** POMO/ReLD contrasts use original instances conditional on fixed checkpoints; progress comparisons use sixteen paired initialization/order means per source regime; online comparisons use eight paired training lineages. Contexts and target instances are paired measurements within those units. Independent scorers reconstruct routes from original instances and saved actions, checking customer coverage, capacity, depot closure and geometric distance. Online replay additionally checks legality, release times, waiting and stochastic replenishment.

**Inference reproduction.** Replay of saved routes and regeneration by model inference are separate checks. The portable progress scorer verifies 485,632 routes, including development records, with maximum cost discrepancy below $2.85 \times 10^{-14}$; all 260 selected checkpoint hashes match. At the recorded execution shape, CPU/CUDA inference reproduces all 11,648 paths in the first original batch of each checked study/regime/interface/target cell: four CPU threads, batch 64 for $N \geq 100$ and 128 otherwise. Use those batch shapes when comparing exact paths, since floating-point changes can alter near-tied decisions. ReLD's full inference grid reproduces at its recorded GPU batch shape (Appendix A.5).

**Source records.** Protocols and hash manifests bind source/target instances, selected checkpoints, route traces and numerical tables. Per-fit records contain update/exposure counts, elapsed time and source-validation history. These records, the complete budget curves, deployment logs and earlier development studies remain with the research artifact. The appendix retains the methods and full comparison panels needed to interpret the current claims; numerical rows are copied from the verified exports.