A CORRECT MASK HIDES THE SOURCE OF FAILURE: TRAJECTORY-RECONSTRUCTIBLE VS. LATENT-REGIME ALIASING IN LEARNED ROUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Learned routing policies act under a correct feasibility mask, yet can still rank feasible actions poorly. We show that such a quality failure has more than one source, and that a correct mask hides which one: the deficit depends on where the decision-relevant information that the decoder observation aliases resides. In controlled experiments with a construction policy implemented independently of any released system, we exhibit a dissociation. When the aliased quantity is a state that accumulates along the route, adding decoder memory reconstructs it and removes 95% of the deficit over fifty-two training seeds. Two comparisons place that recovery: a content-free channel of the same width removes nothing, while the same recurrent module fed a constant removes 48%, mostly because its state is zeroed at each depot visit. Driving that reset from a different instance's depot schedule, matched to about a percent in how deep the state is when it is read, removes nothing (to within 0.07): what the reset supplies is alignment with the instance's own tour, not re-anchoring at the right rate. About half the recovery comes from depth into the tour and half from the route's accumulated state. When the aliased quantity is a per-instance latent regime that leaves no signature in the observed stream, decoder memory cannot reconstruct it and does not help; only an explicit regime signal closes the deficit. The same added memory therefore repairs one failure and leaves the other essentially unchanged, and the separation holds across regime strengths in the latent case and across model widths in the reconstructible case. We then revisit a released multi-task routing model whose mixed-backhaul failure combines both ingredients, an unseen regime and an aliased feasibility state, and where, after fine-tuning on a pool that covers the regime, a one-bit regime signal raises the repair of its $6.2\,\mathrm{pp}$ zero-shot deficit from 0.47 to 0.88. In practice, therefore, the source must be diagnosed before a fix is chosen. The contribution is a mechanism-level dissociation: two hidden sources of failure that a correct mask renders indistinguishable, and that demand different fixes.

1 INTRODUCTION

Neural policies for vehicle routing construct a solution one node at a time, and at each step an environment masks the infeasible actions. A correct mask is a strong guarantee: the policy can only choose feasible next nodes, so it never violates a constraint. It is a guarantee about feasibility, not about quality. The policy still has to rank the feasible actions the mask leaves available, and it can rank them poorly. When a policy is feasible but consistently suboptimal on some family of instances, a correct mask hides where the deficit comes from: capacity, architecture, training, or what the decoder observes.

The source we focus on is the decoder's observation. A construction policy ranks feasible actions from a compact observation of the current state, and that observation can alias states that a correct mask treats identically but whose optimal continuations differ. Two rollouts can reach the same observed state, face the same feasible set, and still call for different next moves; when they do, a policy that sees only the observation cannot consistently rank both correctly. A compact observation is the field's default: the decoder of this model family was given a context that suffices for the traveling-salesman problem it was designed on, where the route already built matters only through

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.

its endpoints (Kool et al., 2019), and modern multi-task decoders inherit that memoryless context, which reduces the route so far to a few running scalars (Liu et al., 2024; Zhou et al., 2024; Berto et al., 2025a).

This failure has more than one source, and the different sources require different fixes. We separate them by a single question: is the aliased, decision-relevant quantity something the policy could in principle reconstruct from the route it has built so far, or a latent property of the instance that the route never reveals? Both cases appear on an aggregate benchmark as the same quality deficit under a correct mask, but they differ in which intervention works. Perceptual aliasing and state abstraction are classical (Whitehead & Ballard, 1991; Li et al., 2006); what we show is that under a correct mask this deficit does not, by itself, reveal which of the two sources produced it, and so which of two fixes, decoder memory or an explicit signal, will work.

We make this concrete with an independently implemented policy on two controlled tasks and observe a dissociation (Sections 3–5). In the first, the aliased quantity accumulates along the route; giving the decoder a memory of the route reconstructs it and removes 95% of the gap, while a content-free channel of the same size does nothing. In the second, the aliased quantity is a per-instance regime the route does not encode; decoder memory now does not help, and only an explicit regime signal closes the gap. The same "add memory" intervention that repairs the first leaves the second essentially unchanged. Memory's efficacy is graded: an extension that exposes the regime through noisy per-step evidence finds recurrent recovery rising with what can be integrated across steps, while memoryless recovery tracks only the single-step signal (Appendix B.3).

Finally we return to a released system (Section 6). A released multi-task routing model exhibits a mixed-backhaul failure that combines both ingredients: a regime its generator never samples, and a decoder scalar that aliases the feasibility-relevant state. The two sources are entangled there, which shows why, in a real system, one must diagnose the source before prescribing a fix.

We draw the distinction between trajectory-reconstructible and per-instance latent observation aliasing under a correct mask (Section 2), and show a controlled dissociation in the effective intervention that holds across regime strength in the latent case and across model capacity and training budget in the reconstructible case (Sections 3–5, Appendix B). We then locate the recurrent recovery: a per-tour reset supplies depth into the tour regardless of the input fed to the memory module, and the route content adds about as much again (Appendix B.4). We analyze a released model in which the two sources co-occur (Section 6), and argue for source diagnosis as a prerequisite to intervention (Section 7).

2 SETUP

We consider construction policies that build a route one node at a time under an environment mask, and we take the mask to be correct: at every step it removes exactly the infeasible next nodes. Feasibility is therefore never the failure. The failure is in how the policy ranks the feasible nodes, and we call a quality deficit that persists under a correct mask a *mask-correct failure*.

A construction policy ranks the feasible nodes from a decoder observation of the current state. The observation *aliases* two states when it maps them to the same value while their cost-optimal continuations differ; if the mask also leaves the same nodes feasible, a policy that depends only on the observation and the mask cannot consistently rank both correctly. We separate aliased quantities by a single property: whether they can, in principle, be reconstructed from the route built so far. A *trajectory-reconstructible* quantity accumulates along the route; a *per-instance latent* is a property of the instance that the route never reveals, which makes the problem a latent MDP, or equivalently a contextual MDP whose context is unobserved (Kwon et al., 2024; Chen et al., 2024). The next two sections hold the constraint and architecture fixed and show that this property determines which intervention repairs the failure.

An organizing observation. Let z be a decision-relevant quantity that the decoder observation o_t aliases: some states share o_t and the feasible set but differ in z and in their cost-optimal continuation, so a policy depending only on (o_t, mask) cannot rank both correctly. Two idealized cases bound the effect of memory.

In the *reconstructible* case the hidden quantity is a measurable function of the route built so far, $z_t = f(\tau_t)$; a policy carrying a sufficient statistic h_t of τ_t then determines z_t and can reproduce the rankings of a policy given z_t directly, so the aliasing deficit is in principle removable by memory. In the *latent* case the hidden quantity is a per-instance r statistically independent of the observable trajectory, so $p(r \mid g(\tau_t)) = p(r)$ for every function g; a policy conditioned only on the trajectory is then confined to prior-mixture decisions and cannot in general attain the regime-aware optimum where the r-conditional optimal decisions differ, and removing that deficit requires access to r rather than more memory.

Both follow from partial observability (Whitehead & Ballard, 1991; Li et al., 2006), and together they name the property, reconstructibility from the trajectory, that decides whether memory or an explicit signal is the effective intervention. The reconstructible case concerns what a sufficient trajectory summary could represent in principle; whether a standard recurrent module actually learns it is the empirical question Family A answers. Family A instantiates that case: the missing quantity accumulates along the route. Family B instantiates the latent case: the regime is drawn per instance and is not recoverable from the route.

3 Family A: Trajectory-Reconstructible State Aliasing

Both families share one base task, one training recipe, and one architecture, a POMO construction policy (Kwon et al., 2020) — an attention encoder–decoder trained by REINFORCE against a baseline formed from several rollouts of the same instance — that shares no code with any released system. They are specified in full in Appendix A, and differ only in where the aliased, decision-relevant quantity resides. In Family A it accumulates along the route, so the route reveals it. The task is a controlled capacitated routing problem: each customer has a forward demand, which a correct capacity mask uses, and a return burden that accumulates along a route and enters a convex route cost: travel length plus $\lambda \sum_{\text{tours}} (\sum_{j \in \text{tour}} b_j)^2$, with $\lambda = 3$, $n = 20$ customers and capacity $C = 30$. The decoder observation exposes the remaining capacity but not the accumulated burden, so two states with the same feasible set can have different cost-optimal continuations, aliasing under a mask that is always correct.

We compare five decoder observations, trained identically for 2000 updates over fifty-two seeds, and report the mean cost of greedy rollouts on a fixed held-out set of 1000 instances (Table 1). Aliasing the accumulated burden raises the cost by 0.17 (about 1.3%) relative to direct access, a deficit about eleven times the reference arm's seed-to-seed standard deviation and six times the aliased arm's. A content-free control that adds a constant channel of the same width leaves the deficit within seed noise, so the deficit stems from missing information rather than input dimensionality. Nor is it about reading burden values at decode time: handing the decoder the demand and burden of the node just served, with no memory over them and in a state vector one channel wider than direct access, is indistinguishable from that constant channel (-0.000, CI $[-0.089, +0.088]$). Two interventions recover it: supplying the burden directly, and giving the decoder a recurrent summary of the route so far, which recovers 95% of the deficit, reaching within $+0.008$ in cost of direct access. About half of that recovery needs no burden content: fed only a constant, the same module, reset at each depot visit, supplies depth into the current tour, and the burden content adds about as much again (Appendix B.4). Across a sixteen-fold range of model width, and again across a sixteen-fold range of training budget, the memoryless deficit shrinks by 50% and 59% respectively without closing under either, while the recurrent summary removes 93% or more of it at every setting of both; the pattern also holds at $n = 50$ (Appendices B.2 and B.5).

4 Family B: Per-Instance Regime Aliasing

Family B keeps the base task, architecture, and training recipe of Family A and changes only where the aliased quantity resides: here it is a per-instance latent the route never reveals. Each instance carries a hidden regime that sets the coefficient of the burden cost (low or high), so the cost-optimal routing depends on the regime. The full state, remaining capacity and accumulated burden, is observed; only the regime is hidden. By construction, the regime cannot be reconstructed from the trajectory: the regime enters the objective alone, setting the coefficient on the burden penalty, and

Table 1: Mean route cost over fifty-two seeds ($\pm$ sd), and the fraction of each family's aliased deficit removed (ρ). Paired 95% intervals appear in Figure 1 and Table 2. The mask is always correct; conditions differ only in the decoder observation. Direct access receives the missing quantity as an input and is the reference. Fractions are within-family: the two deficits are $+0.174$ and $+1.381$.

	Family A (reconstructible)		Family B (latent regime)	
Decoder observation	cost $\pm$ sd	removed	cost $\pm$ sd	removed
Direct access	13.67 ± 0.02	(ref.)	14.41 ± 0.02	(ref.)
Aliased	13.84 ± 0.03	—	15.79 ± 0.02	—
+ content-free channel	13.85 ± 0.05	-0.07	15.80 ± 0.03	-0.007
+ recurrent, noise input	13.76 ± 0.02	$+0.46$	15.79 ± 0.03	$+0.002$
+ recurrent route summary	13.68 ± 0.02	$\mathbf{+0.95}$	15.79 ± 0.03	$\mathbf{-0.001}$

enters neither the transition nor the feasibility mask, so the distribution over trajectories is the same under either regime for any policy.

We compare the same decoder observations over fifty-two seeds, with a noise-fed recurrent arm alongside them (Table 1). Hiding the regime raises the mean cost by 1.38 (about 9.6%): with no way to tell which regime is active, the policy is trained against conflicting objectives and settles for a prior-mixture compromise, the latent case of Section 2. A content-free control removes nothing; unlike Family A, the recurrent route summary does not help either, and is statistically indistinguishable from that control: there is nothing in the route to summarize. The same recurrent module fed $\mathcal{N}(0, 1)$ noise in place of the served pair yields an equivalent outcome, removing $+0.002$ of the deficit (CI $[-0.005, +0.008]$); across the three conditions of Table 1 that lack the regime signal, none recovers more than 0.9% of the 1.381 deficit at fifty-two seeds (simultaneous one-sided 95% bounds). Only supplying the regime directly recovers the deficit; that condition receives the latent as an input and is the regime-aware reference. With the regime unobservable and drawn independently of everything the trajectory contains, a prior-mixture policy is the best any history-dependent policy can do, and the regime-aware reference solves a better-observed process (Singh et al., 1994): the deficit measures the cost of the observation gap. The comparison measures the flip: holding architecture, base task and recipe fixed and moving only where the decision-relevant quantity resides changes which intervention works. The pattern holds across regime strengths: the aliased deficit grows monotonically with regime strength and vanishes when the regime is a no-op, while recurrent memory recovers essentially none of it at every strength with a nonzero deficit, staying indistinguishable from the content-free control; the pattern persists at $n = 50$ (Appendix B). Memory's failure here is conditional: a further extension keeps the regime latent but makes it partially inferable through noisy per-step evidence, and recurrent recovery then rises with what can be integrated across steps while a memoryless decoder tracks only the single-step signal (Appendix B.3).

5 The Controlled Dissociation

Families A and B produce the same surface symptom, a quality deficit under a correct mask, but respond differently to the same interventions (Figure 1). In both, supplying the missing quantity directly closes the deficit, and a content-free channel of the same width does not, so in both the deficit is about missing information rather than input width. The two families diverge on memory: a recurrent route summary removes 95% of the Family A deficit and almost none of the Family B deficit, which instead needs an explicit regime signal. Over fifty-two seeds, paired 95% intervals (t, df=51) support the split. In Family A the recurrent memory closes 95% of the deficit, reaching within $+0.008$ in cost of the reference (CI $[+0.001, +0.015]$ once evaluation-set variance is included; Appendix B.4). In Family B it removes none of the 1.381 deficit: recovery is -0.001, CI $[-0.009, +0.007]$, so memory recovers at most 0.7% of it. The content-free control closes neither deficit (Family A: -0.07, CI $[-0.17, +0.02]$; Family B: -0.007, CI $[-0.015, +0.001]$).

The flip is the same in cost units: Family A's memory arm removes $+0.166$ while Family B's removes nothing (-0.001, CI $[-0.012, +0.010]$), across a 7.9$\times$ difference in deficit size. Across Family B deficits from $+0.14$ to $+2.02$, one of them smaller than Family A's, the recurrent arm's recovered fraction stays near zero (Appendix B.1).

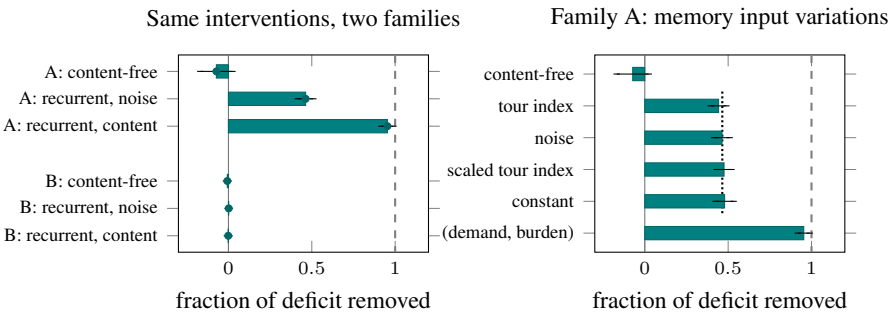


Figure 1: Fraction of each family's aliased deficit removed by each decoder observation, over fifty-two training seeds on one fixed set of 1000 held-out instances. Bars are means of per-seed ratios, whiskers paired 95% t intervals (df=51); the dashed line at 1 is the direct-access reference, which receives the missing quantity as an input. Fractions are within-family: the deficits are $+0.174$ (A) and $+1.381$ (B) in cost units. Right: the input to the Family A recurrent module is varied with everything else held fixed, the top row is the content-free channel, the same constant input without recurrence, and the dotted line is the recovery pooled over the four burden-independent inputs, 0.465.

The families share the constraint and the architecture, and differ only in where the aliased decision-relevant quantity resides. When it accumulates along the route, the trajectory carries it and a memory of the trajectory recovers it. When it is a per-instance latent, the trajectory does not carry it and memory cannot recover it; the information has to be supplied. A correct mask is silent about which case one is in, so the same observed deficit calls for a different fix depending on a property the benchmark does not expose.

6 A Real-World Case Study: RouteFinder Mixed Backhauls

The controlled families isolate one source at a time; in a released system the two sources can be entangled, which is what makes diagnosis necessary. We show this in RouteFinder, a released multi-task routing model (Berto et al., 2025a), on its mixed-backhaul (class-2) family.

In the classic (class-1) problem with backhauls (VRPB), deliveries precede pickups (Goetschalckx & Jacobs-Blecha, 1989); RouteFinder's class-2 setting instead lets linehaul and backhaul load coexist, so a class-2 linehaul customer with demand d is feasible when

$$d \leq C - \max(u_\ell, u_b), \tag{1}$$

with C the capacity and u_ℓ, u_b the used linehaul and backhaul loads. Two ingredients coincide. The released decoder context exposes a single available-load scalar that switches from linehaul to backhaul load once pickups begin, aliasing states whose class-2 feasibility differs; and the released generator samples class-2 instances with probability zero, so the regime is also unseen in training. The observed failure, a cost gap to the benchmark's best-known solutions 6.23 percentage points above that of the corresponding seen variant on the same 1000 instances (Appendix C.5), is an entangled sum of the two effects; the ablations below decompose measured repairs on a matched covered pool, not that original deficit.

Separating them requires holding one fixed. On a common pool that does contain class-2 instances, fine-tuning alone recovers about half the deficit (repair ratio $\mathcal{R} = 0.472$ for the base context, where $\mathcal{R}$ is the fraction of the pre-exposure deficit removed, Appendix C): the coverage share. On top of it, a five-seed ablation isolates the added signal from added capacity: a one-bit class-2 regime signal raises the repair to $\mathcal{R} = 0.879$, while a content-free control of the same form adds nothing over that base (0.434 versus 0.472): the Family B pattern, that a regime needs a signal. Re-running these three conditions on a rebuilt software stack reproduces each one to within its interval (Appendix D.4). The released system's adapter recipe targets this regime: it adds a mixed-backhaul flag to the encoder's global attributes and a class-2-gated headroom term to the decoder context, through zero-padded projections, and then fine-tunes. It does not separate them, so which addition carries the repair is not reported. Under that released recipe the contrast barely moves here ($\mathcal{R} = 0.018$, CI

$[-0.006, +0.042]$). That is an unmatched reference rather than a component comparison. Over the same fine-tuning the recipe left the mixed-backhaul gap essentially where it started (10.18% to 10.08%) while degrading by 3.19 pp the seen class-1 case it already handled (3.95% to 7.15%). The recipe supplies the class-2 signal, so its failure implicates its training rather than the signal. Full ablations (Appendix D), a family map (Appendix E), a multi-depot boundary case (Appendix F), a cross-backbone reproduction on a shared decoder code path (Appendix C.3), and what the fine-tuning costs on variants the pool does not cover (Appendix D.5) are detailed in the appendix.

In a real system the two sources co-occur and an aggregate benchmark reports only their sum; the mechanism is established by the controlled families. The two sources do not contribute equally here: the measured repair is dominated by the regime/coverage component (the class-2 signal), while the reconstructible-state component contributes little (Appendix D). Recurrent memory lowers the base-context gap (-0.56 pp, CI $[-0.71, -0.41]$, five seeds), and the same module fed noise lowers it nearly as far (-0.52 pp, CI $[-0.66, -0.38]$), which five seeds cannot separate (-0.04 pp, CI $[-0.17, +0.09]$). Removing the per-tour reset returns the noise arm to base context ($+0.02$ pp) and removes most of the content-fed arm's improvement, leaving -0.19 pp. Both recurrent arms sit about 0.6 pp below the content-free control: recurrence improves performance here, yet it cannot be attributed to the switched load, since noise achieves a comparable reduction. The Family A memory mechanism is therefore established in the controlled tasks rather than directly in this system. The coverage gap and the observation aliasing in this failure require distinct remedies.

7 Diagnosing Mask-Correct Failures in Practice

The practical consequence of the dissociation is a diagnostic one: under a correct mask, the source of a quality deficit should be identified before a fix is chosen, because the fixes differ and the benchmark does not reveal the source. A deficit that is really trajectory-reconstructible state aliasing is repaired by giving the decoder access to the route it has built — explicit state closes it and a recurrent summary removes 95% of it — and not by more of the same observation. A deficit that is really per-instance regime aliasing is closed by supplying the regime as an explicit signal, and not by memory. In Family B the wrong intervention, memory, leaves the deficit untouched.

For a mask-correct deficit of unknown source, we propose two diagnostic checks. Detecting that a representation is non-Markovian at all is a well-studied question: the λ-discrepancy (Allen et al., 2024) does it model-free, without the hidden state or a reference policy that observes it. The checks below start after that answer and ask which of the two sources the deficit comes from, which decides the fix. First, widen the memoryless decoder and lengthen its training: if the deficit shrinks under both and closes under neither, capacity and training do not substitute for the missing quantity over that range. Second, add a recurrent summary of the route and compare it against the same recurrent module fed burden-independent input; a non-recurrent content-free channel matches the explicit-signal arms in width but not the memory in parameters. In Family A the noise-fed recurrent arm removes 0.46 of the deficit against the content-fed arm's 0.95 (Appendix B.4). What the counterpart is fed does not matter — a constant, a step index and noise are indistinguishable — so a simple constant input suffices. The counterpart is noise-fed in both families. In Family B content and noise coincide (-0.003 in ρ, CI $[-0.012, +0.006]$), while the families' content-minus-noise contrasts differ by $+0.494$, CI $[+0.436, +0.551]$ (Welch, $\mathrm{df}=54$): the content-over-noise gap is present in Family A and absent in Family B. On the two families, whose sources are known by construction, the second check returns the right answer for both. If the content-fed memory removes more of the deficit than its own noise-fed counterpart, the quantity is inferable from the route and memory is the fix; if both sit at the content-free level while the deficit persists, the failure is consistent with a per-instance latent regime, which memory cannot recover and which calls for an explicit signal. Intermediate outcomes are informative too: when the latent is partially inferable from accumulated evidence, recurrent recovery rises with what can be integrated (Appendix B.3), so a partial content-over-noise gap points to a partially inferable quantity. The width and budget sweeps and the memory-versus-control contrast of Appendix B are these two checks.

One of the two sources is architectural. A decoder that reduces the route it has already built to a few running scalars is the field's default, for a reason — sufficiency for the traveling-salesman problem — that does not bear on quantities that accumulate along the route (Section 8). A trajectory-

reconstructible deficit is therefore a possibility created by a design pattern, not by a particular checkpoint, and the checks above apply to any policy built on that design.

A mask-correct failure need not indicate that the architecture is too weak: in both families the same architecture reaches the reference once the missing information is available. And it need not indicate an unlearnable representation barrier: in Family A the information is present in the trajectory and a small memory recovers it.

8 Related Work

Multi-task neural routing. Attention-based neural routing policies (Kool et al., 2019) and POMO-style training and inference (Kwon et al., 2020) underpin many modern neural solvers. RouteFinder, MTPOMO, and MVMoE extend this line to multiple VRP variants through shared representations or mixture-of-experts designs (Berto et al., 2025a; Liu et al., 2024; Zhou et al., 2024). CaDA and MoSES study related forms of cross-problem representation and policy reuse (Li et al., 2025; Pan et al., 2025). We ask when such shared policies fail on a constraint regime they were not trained to cover, and which signals repair it under fixed coverage.

Constraint-aware policy inputs. Constraint-aware routing methods modify how a policy receives or uses evolving information. CARM conditions a decoder on constraint-relevant variables (Yu et al., 2026), and CCL reformulates context during route construction (Gui et al., 2026). These works motivate the distinction between reweighting an observed signal and supplying a signal that separates decision-relevant regimes; the class-2 flag in our case study is a regime indicator rather than a task identifier.

Decoder recurrence in neural routing. The mechanism our reconstructible family relies on is one this line already had and discarded. Nazari et al. (2018) removed the recurrent *encoder* as unnecessary for an unordered input set, arguing it "adds extra complication to the encoder but is actually not necessary", while keeping a recurrent decoder whose state "summarizes the information of previously decoded steps". Kool et al. (2019) then removed decoder recurrence as well, on the ground that "by using attention instead of recurrence (LSTMs) we introduce invariance to the input order of the nodes", a comparison drawn against the recurrent pointer-network line (Vinyals et al., 2015; Bello et al., 2016). That argument concerns the encoder's input set, not whether the decoder's context carries a quantity that accumulates along the route, and modern multi-task decoders inherit the memoryless context (Berto et al., 2025a; Liu et al., 2024; Zhou et al., 2024). Designs that depart from it re-derive the residual problem at each step (Drakulic et al., 2023; Luo et al., 2023) or enlarge the decoder (Huang et al., 2025), and a recurrent *encoder* has been used to reuse computation across steps (Dernedde et al., 2026); these target generalization and latency rather than a decision-relevant quantity the context omits. Our Family A condition therefore measures what that removal costs when the aliased quantity is trajectory-accumulated.

Instance-level conditioning. Instance-level indicators are standing practice in neural routing: multi-task decoders already carry a per-instance open-route flag directly in the decoder context (Liu et al., 2024; Zhou et al., 2024), and RouteFinder's own adapter recipe already prescribes, for the class-2 regime we study, a mixed-backhaul flag on the encoder's global attributes together with a class-2-gated headroom term on the decoder context, introduced through zero-padded projection weights and then fine-tuned. Our case study therefore separates that fix: the recipe adds these terms and further state in one move and does not report which component carries the repair, whereas the ablation of Section 6 holds coverage fixed on a matched pool, varies the added signal against a content-free channel of the same form, and attributes the repair.

Backhaul feasibility and state abstraction. The VRPB traditionally enforces delivery-before-pickup precedence (Goetschalckx & Jacobs-Blecha, 1989). RouteFinder's class-2 variant relaxes this order and requires capacity checks while linehaul and backhaul loads coexist. Work on proactive infeasibility prevention likewise targets difficult constraint structure (Bi et al., 2024). Our analysis differs in that the class-2 action mask is already correct. The question is whether the decoder observation distinguishes the states that the mask treats differently. This relates the diagnosis to perceptual aliasing and state abstraction (Whitehead & Ballard, 1991; Li et al., 2006).

Hidden states recoverable versus unrecoverable from history. The distinction between trajectory-reconstructible state and per-instance latent parameters has long-established foundations in reinforcement learning. The information-state formalism defines state as a function of the history sufficient to predict reward (Subramanian et al., 2022), so a quantity absent from the history lies outside it by construction, and hidden-parameter MDPs and task inference in meta-reinforcement learning draw the per-episode case explicitly (Doshi-Velez & Konidaris, 2016; Rakelly et al., 2019; Zintgraf et al., 2020). Each remedy is known separately too. Chrisman (1992) already separates aliasing that more immediate input resolves from aliasing that does not, observing that incomplete perception "cannot be overcome by obtaining additional immediate perceptual input". That axis is immediate perception against memory, and the case drawn there is one memory recovers: our Family A, not the contrast between the families. Utile-distinction methods then ask how much history a distinction requires (McCallum, 1995), and recurrent policies became the standard aggregator under partial observability (Hausknecht & Stone, 2015). In the other direction, weakly revealing side information about a per-episode latent context restores tractability (Kwon et al., 2024), and separation conditions put a nearly sharp statistical threshold on the horizon length at which such a latent becomes efficiently learnable (Chen et al., 2024). What has not been established is the comparison, because the prevailing expectation is that memory is the general remedy: augmenting model-free RL with recurrence is, as Ni et al. (2022) put it, in theory "a general approach to solving all types of POMDPs", and their experiments find recurrent baselines strong on many POMDPs rather than all. We hold base task, architecture, and training recipe fixed, move only where the decision-relevant quantity resides, and control for input width with a content-free channel; the effective intervention then flips. Our latent-case claim is confined to the non-identifiable regime, drawn independently of the route: meta-reinforcement learning shows recurrence succeeds to the extent that the latent leaves a signature in the observed stream, and Appendix B.3 traces that dependence.

Conditional side paths. FiLM provides feature-wise affine conditioning (Perez et al., 2018). Zero-initialized control paths and factors are used in ControlNet and LoRA to preserve a base mapping when introducing new controls (Zhang et al., 2023; Hu et al., 2022). We use this established device on the encoder initialization and decoder context to compare candidate signals without replacing RouteFinder's backbone.

9 LIMITATIONS

Task design and effect sizes. The Family B regime is intentionally consequential, so its large deficit is a property of the construction; the claim we draw from it is the dissociation pattern (memory fails, a signal succeeds), not the magnitude of the gap.

Scope of the distinction. The controlled experiments instantiate two families on opposite sides of the reconstructibility distinction, one for each case; other sources of mask-correct failure, such as genuine capacity limits, are outside what we test. The two families are two points rather than a measured continuum: one quantity is fully reconstructible from the trajectory and the other is statistically independent of it. The evidence-noise extension (Appendix B.3) varies the signal-to-noise ratio of an added observation channel within the latent case, which shows that recurrent recovery is graded in observation informativeness.

What the controls establish. The robustness checks of Appendix B vary overall model width and, separately, the training budget, and neither capacity nor budget substitutes for recurrence; the Family B sweep establishes a range-wide regularity across the regime strengths we test. Both vary one setting across all components at once, so they bound what substitutes for recurrence without isolating any single one, and the regularity is a property of this family rather than of every latent-regime problem. The content-free control rules out a same-width added channel as the source of the Family A recovery, and a recurrent-content control splits it: the same recurrent module recovers about 0.47 of the deficit whatever burden-independent input it is fed, including a constant, because its per-tour reset makes the hidden state a function of depth into the instance's own tour. Deleting that reset eliminates three quarters of the improvement achieved by the constant-fed module; driving it instead from a neighboring instance's depot schedule — matched to about a percent in how deep the state is when the decoder reads it — eliminates the entire recovery, while the burden content provides an additional increment above it (Appendix B.4). In Family A, recovery is stepped rather than graded

in the information reaching the memory, and the steps are not nested: a reset at the right rate on the wrong tour leaves the memory no better than a static channel. The recurrent arm recovers about 95% of the deficit, leaving $+0.008$ in cost: the repair in Family A is not complete. The misaligned-reset arms drive the reset from outside the policy, so they measure alignment together with the policy's ability to shape its own reset cadence.

Real-world evidence. The real-world evidence is from one ecosystem. RouteFinder is one released model, and the cross-backbone reproduction we report shares a decoder code path across its backbones, a shared-implementation effect, not evidence across independent architectures. The case study shows that the two sources can co-occur in a released system, but the generality of the distinction beyond these settings rests on the controlled experiments. The one-bit signal also does not transfer across the family without training: mapped over seven mixed-backhaul variants it recovers 0.96 of the deficit when only a route-duration limit is added, about 0.42 on the two open-route variants without time windows and only 0.01–0.04 on the four with them, so it repairs the regime it was measured on rather than the family (Appendix E). Finally, all released-model gaps use the benchmark's best-known-solution references, which are time-limited heuristic values rather than proofs of optimality, and the study uses synthetic and benchmark instances rather than operational routing data.

10 Conclusion

A correct feasibility mask guarantees feasibility, not ranking quality, and when quality fails it does not identify the cause. We separated two causes by a single property of the aliased, decision-relevant quantity (whether the route reconstructs it) and showed a controlled dissociation: a recurrent memory recovers a trajectory-accumulated state but not a per-instance regime, which needs an explicit signal instead. A released multi-task model in which the two sources co-occur shows why the distinction matters in practice. The source of a mask-correct failure determines which fix works, so it has to be diagnosed before a fix is chosen.

AI Use Statement

We used a large language model solely for language editing and polishing the text of this manuscript.

References

Cameron Allen, Aaron Kirtland, Ruo Yu Tao, Sam Lobel, Daniel Scott, Nicholas Petrocelli, Omer Gottesman, Ronald Parr, Michael L. Littman, and George Konidaris. Mitigating partial observability in sequential decision processes via the Lambda discrepancy. In *Advances in Neural Information Processing Systems*, 2024.

Irwan Bello, Hieu Pham, Quoc V. Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940*, 2016.

Federico Berto, Chuanbo Hua, Nayeli Gast Zepeda, André Hottung, Niels A. Wouda, Leon Lan, Junyoung Park, Kevin Tierney, and Jinkyoo Park. RouteFinder: Towards foundation models for vehicle routing problems. *Transactions on Machine Learning Research*, 2025a. URL `https://openreview.net/forum?id=QzGLoaOPiY`.

Federico Berto, Chuanbo Hua, Junyoung Park, Laurin Luttmann, Yining Ma, Fanchen Bu, Jiarui Wang, Haoran Ye, Minsu Kim, Sanghyeok Choi, Nayeli Gast Zepeda, André Hottung, Jianan Zhou, Jieyi Bi, Yu Hu, Fei Liu, Hyeonah Kim, Jiwoo Son, Haeyeon Kim, Davide Angioni, Wouter Kool, Zhiguang Cao, Qingfu Zhang, Joungho Kim, Jie Zhang, Kijung Shin, Cathy Wu, Sungsoo Ahn, Guojie Song, Changhyun Kwon, Kevin Tierney, Lin Xie, and Jinkyoo Park. RL4CO: An extensive reinforcement learning for combinatorial optimization benchmark. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5278–5289, 2025b. doi: 10.1145/3711896.3737433.

Jieyi Bi, Yining Ma, Jianan Zhou, Wen Song, Zhiguang Cao, Yaoxin Wu, and Jie Zhang. Learning to handle complex constraints for vehicle routing problems. In

Advances in Neural Information Processing Systems, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a9d2a5fd12d34250c21b5e4fa8d906b0-Abstract-Conference.html.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8(1):6085, 2018. doi: 10.1038/s41598-018-24271-9.

Fan Chen, Constantinos Daskalakis, Noah Golowich, and Alexander Rakhlin. Near-optimal learning and planning in separated latent MDPs. In *Proceedings of the 37th Annual Conference on Learning Theory (COLT)*, volume 247 of *Proceedings of Machine Learning Research*, pp. 995–1067, 2024.

Lonnie Chrisman. Reinforcement learning with perceptual aliasing: The perceptual distinctions approach. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, pp. 183–188. AAAI Press, 1992.

Tim Dernedde, Daniela Thyssens, and Lars Schmidt-Thieme. Recurrent state encoders for efficient neural combinatorial optimization. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2026.

Finale Doshi-Velez and George Konidaris. Hidden parameter Markov decision processes: A semi-parametric regression approach for discovering latent task parametrizations. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

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*, volume 36, 2023.

Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in Markov decision processes. *Artificial Intelligence*, 147(1–2):163–223, 2003.

Marc Goetschalckx and Charlotte Jacobs-Blecha. The vehicle routing problem with backhauls. *European Journal of Operational Research*, 42(1):39–51, 1989. doi: 10.1016/0377-2217(89)90057-X.

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://openreview.net/forum?id=AhE6aSlz5g.

Matthew Hausknecht and Peter Stone. Deep recurrent Q-learning for partially observable MDPs. In *AAAI Fall Symposium on Sequential Decision Making for Intelligent Agents*, 2015.

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

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.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134, 1998. doi: 10.1016/S0004-3702(98)00023-X.

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

Jeongyeol Kwon, Yonathan Efroni, Shie Mannor, and Constantine Caramanis. Prospective side information for latent MDPs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 25755–25783, 2024.

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

Han Li, Fei Liu, Zhi Zheng, Yu Zhang, and Zhenkun Wang. CaDA: Cross-problem routing solver with constraint-aware dual-attention. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 35438–35456, 2025. URL https://proceedings.mlr.press/v267/li25bi.html.

Lihong Li, Thomas J. Walsh, and Michael L. Littman. Towards a unified theory of state abstraction for MDPs. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematics*, 2006. URL https://thomasjwalsh.net/pub/aima06Towards.pdf.

Fei Liu, Xi Lin, Zhenkun Wang, Qingfu Zhang, Xialiang Tong, and Mingxuan Yuan. Multi-task learning for routing problem with cross-problem zero-shot generalization. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1898–1908, 2024. doi: 10.1145/3637528.3672040.

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*, volume 36, 2023.

R. Andrew McCallum. Instance-based utile distinctions for reinforcement learning with hidden state. In *Machine Learning: Proceedings of the Twelfth International Conference (ICML)*, pp. 387–395. Morgan Kaufmann, 1995.

Mohammadreza Nazari, Afshin Oroojlooy, Lawrence V. Snyder, and Martin Takáč. Reinforcement learning for solving the vehicle routing problem. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pp. 9861–9871, 2018.

Tianwei Ni, Benjamin Eysenbach, and Ruslan Salakhutdinov. Recurrent model-free RL can be a strong baseline for many POMDPs. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16691–16723, 2022.

Yuxin Pan, Zhiguang Cao, Chengyang Gu, Liu Liu, Peilin Zhao, Yize Chen, and Fangzhen Lin. Multi-task vehicle routing solver via mixture of specialized experts under state-decomposable MDP. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/2ee1c87245956e3eaa71aaba5f5753eb-Abstract-Conference.html.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. doi: 10.1609/aaai.v32i1.11671.

Kate Rakelly, Aurick Zhou, Deirdre Quillen, Chelsea Finn, and Sergey Levine. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

Satinder P. Singh, Tommi Jaakkola, and Michael I. Jordan. Learning without state-estimation in partially observable Markovian decision processes. In *Proceedings of the Eleventh International Conference on Machine Learning*, pp. 284–292, 1994.

Jayakumar Subramanian, Amit Sinha, Raihan Seraj, and Aditya Mahajan. Approximate information state for approximate planning and reinforcement learning in partially observed systems. *Journal of Machine Learning Research*, 23(12):1–83, 2022.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, volume 28, pp. 2692–2700, 2015.

Steven D. Whitehead and Dana H. Ballard. Learning to perceive and act by trial and error. *Machine Learning*, 7(1):45–83, 1991. doi: 10.1023/A:1022619109594.

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.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Zhang_Adding_Conditional_Control_to_Text-to-Image_Diffusion_Models_ICCV_2023_paper.html.

Jianan Zhou, Zhiguang Cao, Yaoxin Wu, Wen Song, Yining Ma, Jie Zhang, and Xu Chi. MVMoE: Multi-task vehicle routing solver with mixture-of-experts. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61804–61824, 2024. URL https://proceedings.mlr.press/v235/zhou24c.html.

Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. VariBAD: A very good method for Bayes-adaptive deep RL via meta-learning. In *International Conference on Learning Representations (ICLR)*, 2020.

A Controlled Families: Task, Cost, and Architecture

This appendix specifies the two controlled families of Sections 3–5. The families share a task backbone, a policy architecture, and a training recipe, and differ only in which decision-relevant quantity is hidden from the decoder. The implementation is self-contained and shares no code with the released system of Section 6.

Task backbone. Each instance is a capacitated routing problem with $n = 20$ customers and a depot. Customer coordinates are drawn uniformly on the unit square; each customer j has an integer forward demand d_j and an integer return burden b_j, both drawn uniformly from $\{1, \dots, 9\}$, with $d_0 = b_0 = 0$ at the depot. The capacity is $C = 30$, and demands enter the observation and the mask normalized by C (burdens likewise). A solution is a sequence of tours from the depot; the feasibility mask allows a customer only if the forward load used in the current tour plus its demand stays within capacity, so *the mask is always correct*. The objective is

$$\text{cost} \;=\; \text{travel length} \;+\; \lambda \sum_{\text{tours}} \Big(\textstyle\sum_{j \in \text{tour}} b_j \Big)^2, \tag{2}$$

a travel term plus a convex (quadratic) penalty on the burden accumulated within each tour. The quadratic makes how burden is grouped across tours decision-relevant, so the accumulated tour burden B is a genuine part of the cost-to-go.

Family A (trajectory-reconstructible state). The coefficient is fixed at $\lambda = 3$. The hidden quantity is the burden B accumulated in the current tour, which grows deterministically along the route and resets at each depot visit. The direct-access observation exposes the remaining capacity and B; the aliased observation exposes only the remaining capacity, so two states with the same feasible set but different B are aliased. Every customer's own burden b_j is a node feature of the encoder in all conditions (Architecture, below), so what the aliased observation withholds is not the burdens but their running sum over the nodes already served in the current tour. Because B is a function of the route so far, it is trajectory-reconstructible.

Family B (per-instance latent regime). The full state, remaining capacity and accumulated burden B, is observed by every condition. Each instance additionally carries a hidden binary regime $r \in \{0, 1\}$, drawn with probability $1/2$ and independent of the coordinates, demands, and burdens, which sets the coefficient in Equation 2 to λ_r with $\lambda_0 = \lambda_{\text{lo}} = 0.5$ and $\lambda_1 = \lambda_{\text{hi}} = 8$ (the sweep of Appendix B.1 varies λ_{hi}). Because r is drawn per instance and influences no observed quantity along the route, it is not reconstructible from the trajectory; only an explicit signal carries it.

Decoder observations. In both families the four conditions differ only in the state vector supplied to the decoder, all under the same correct mask. *Direct access* (Family A) / *regime signal* (Family B) supplies the missing quantity directly (the burden B, respectively the regime r) and is the reference. *Aliased* omits it. *Content-free* appends a constant channel of the same width, matching the added input dimension and parameter count without adding information. *Recurrent* instead appends a memory of the route (below).

Architecture. The policy is a POMO-style encoder–decoder (Kwon et al., 2020). The encoder maps each node's features (coordinates, demand, burden) to width d_m and applies $L = 2$ transformer blocks (multi-head attention with 8 heads, a width-512 feed-forward sublayer, layer normalization, and residual connections). At each step the decoder concatenates the graph embedding, the current-node embedding, a state embedding, and a memory vector, projects the result to a query, and scores actions by scaled dot products of the query with node keys, clipped by $10\tanh(\cdot)$ and masked. The state embedding is a two-layer MLP on the observation vector. In the *recurrent* condition the memory is a GRU cell of width d_m that consumes the (demand, burden) pair of each chosen node and is reset to zero at every depot visit; its hidden state is the memory vector, which in Family A can reconstruct the accumulated tour burden and in Family B carries no information about r. The other conditions pass a zero memory vector. Main-text runs use $d_m = 128$ (the width sweep of Appendix B.2 varies d_m).

Training and evaluation. Each condition is trained from scratch by REINFORCE with a POMO shared baseline: $R = 8$ parallel rollouts per instance, with the per-instance mean return as the baseline. We use Adam with learning rate 3×10^{-4}, batch size 128 instances, gradient-norm clipping at 1.0, and 2000 update steps; Appendix B.2 varies that budget sixteen-fold. Evaluation is greedy decoding on a fixed held-out set of 1000 instances; training and evaluation instances are disjoint. The controlled-family tables report fifty-two training seeds; the sweeps and the evidence-noise extension in Appendix B use five seeds per point.

B Robustness and Extensions of the Controlled Dissociation

The main text reports each controlled family at a single operating point. This appendix varies the strength of the Family B regime and the width of the Family A model; the dissociation holds across both. Both sweeps reuse the main-text task and training recipe unchanged, over five seeds per point. For each point we report the aliased deficit (aliased minus the direct-access reference) and the fraction of that deficit removed by a recurrent route summary (ρ_{mem}) and by a same-width content-free control (ρ_{cf}), each normalized by that point's own deficit. A difference is *resolvable* when its paired 95% interval excludes zero. Where the deficit is within seed noise of zero the fraction is undefined and is omitted. The extensions then follow the same protocol: an evidence-noise sweep inside the latent case, a recurrent-content control, a coarse-signal control, a training-budget sweep, and a scale check.

B.1 Family B: regime-strength sweep

We vary the regime strength α, the gap between the two burden-cost coefficients that the hidden regime selects; $\alpha = 0$ makes the regime a no-op. Figure 2 sweeps α over six settings with the four Family B conditions of Section 4. The aliased deficit is within seed noise of zero at $\alpha = 0$ and rises monotonically as the regime strengthens, so the deficit tracks the regime rather than the setup. At every setting with a non-negligible deficit, the recurrent summary removes essentially none of it and is indistinguishable from the content-free control; the regime-signal condition, which receives the latent directly, is the reference. The Family B pattern, that memory cannot recover a per-instance latent while an explicit signal can, therefore holds across the whole range. The setting used in Section 4 ($\lambda_{\mathrm{hi}} = 8$) sits at the high end of this range. That point is a five-seed run; over fifty-two seeds the deficit is 1.381.

B.2 Variations that do not change the conclusion

Three axes were varied with everything else held at the Section 3 setting. None of them substitutes for a recurrent trajectory summary.

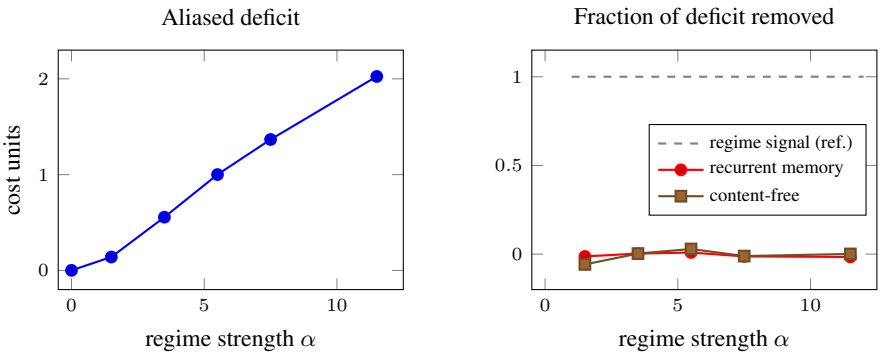


Figure 2: Family B regime-strength sweep (five seeds per point, λ_{lo} fixed). Left: the aliased deficit is near zero when the regime is a no-op ($\alpha = 0$) and rises monotonically with regime strength. Right: the fraction of the deficit removed by recurrent memory and by the content-free control, both near zero at every strength, against the regime-signal reference (dashed). Memory does not recover a per-instance latent at any strength. Fractions are means of per-seed ratios; the ratio of means reverses the sign of the recurrent point at $\alpha = 1.5$.

Model width. Sweeping d_m sixteen-fold from 16 to 256, with the feed-forward sublayer fixed at 512, leaves the memoryless aliased deficit between 0.137 and 0.277 in cost units at every width, with no approach to zero and no monotone descent across the five points at five seeds each. At every width the recurrent summary removes 0.93 or more of the deficit, while the same-width content-free control stays between -0.13 and $+0.16$ of it and is not resolvably different from zero at four of the five widths. Five seeds per width.

Training budget. At 2000, 8000 and 32000 steps the deficit falls monotonically, 0.197 to 0.120 to 0.081 (CI $[+0.072, +0.090]$), and stays resolvably above zero at every budget level; a single power law, $\propto \text{steps}^{-0.32}$, fits all three to within 4%. Three budget levels cannot separate that law, which decays to zero, from a decay to a small positive asymptote, and we claim neither. What the budget does not change is the substitution: the recurrent summary removes 0.930, 1.005 and 1.042 at the three budgets, and the change across them is within noise ($+0.113$, CI $[-0.144, +0.369]$).

A coarse explicit signal. Quantizing the accumulated burden to a single bit, $\mathbf{1}\{B > 0.3\}$, and supplying it through the same channel as the content-free control removes 0.367 of the deficit at fifty-two seeds (CI $[+0.315, +0.419]$). The bit has to carry information to do this: it is resolvably better than the same network with a constant in its place (paired -0.073 in cost, CI $[-0.088, -0.058]$, $t = -9.9$), and resolvably worse than the recurrent summary (paired $+0.099$ in cost, CI $[+0.090, +0.109]$, $t = 21.7$). A single return burden is at most 0.300, so the bit cannot fire before the second customer of a tour and is on with probability $0.00, 0.56, 0.88, 0.98, 1.00$ after $1, \ldots, 5$ of them: it is a soft indicator of depth into the tour rather than a graded reading of the burden. A step-count control isolates that. Replacing the bit with $\mathbf{1}\{k > M\}$, where k counts customers since the last depot, at the threshold whose firing rate matches, recovers 0.196 (CI $[+0.133, +0.259]$) against the burden flag's 0.367, and the two are resolvable (paired -0.028 in cost, CI $[-0.038, -0.019]$, $t = -5.9$). The other threshold the match admits recovers 0.264 and also falls short (paired $+0.103$ in ρ, CI $[+0.049, +0.157]$, $t = 3.9$). Both rules read depth through one bit, so the flag's lead over both excludes a single bit of depth as its source, not depth in general. Sweeping the threshold τ over $\{0.10, 0.15, 0.30, 0.50, 0.75, 0.90\}$ gives a plateau over $0.15 \le \tau \le 0.75$ with resolvable falloff at both ends; we make no claim that 0.30 is the maximum.

B.3 Partially inferable latents: an evidence-noise sweep

The two families place the missing quantity at opposite poles: fully reconstructible from the route, or statistically independent of it. This extension stays inside the latent case and varies how much an added observation channel reveals about the regime, which grades what a memory has to work with without changing where the regime comes from. Each decoding step now emits noisy evidence

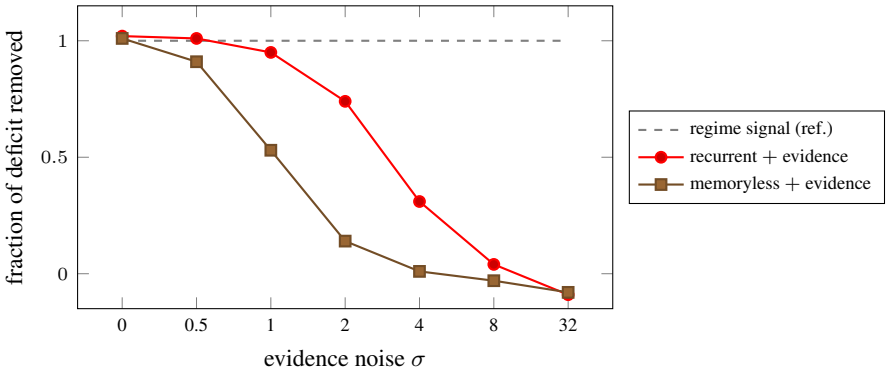


Figure 3: Partially inferable latents (five seeds per point). Each step emits evidence of the hidden regime with noise σ. With exact evidence both conditions reach the regime-aware reference (dashed); with nearly uninformative evidence ($\sigma = 32$) both fall to the Family B null. In between, the memoryless condition tracks only the single-step signal while the recurrent condition integrates evidence across steps, removing 74% of the deficit at $\sigma = 2$ where the memoryless condition removes 14%. Recovery by memory tracks online inferability.

of the hidden regime, $s_t = (2r - 1) + \sigma\,\varepsilon_t$ with $\varepsilon_t \sim \mathcal{N}(0, 1)$ drawn independently per step, so the regime is partially inferable at a rate set by σ: at $\sigma = 0$ each step reveals it exactly, and as σ grows the task approaches Family B, where only the prior is available. Two conditions receive this evidence on top of the full observed state: a memoryless decoder, which can use only the current step's evidence, and a recurrent decoder whose memory consumes the evidence along with the served demands and burdens and can therefore integrate it across steps. Because the latent is a property of the episode rather than of a tour, this recurrent condition carries its memory across the episode, without the per-tour reset used in the main-text conditions. The regime-aware and aliased conditions of Section 4 were retrained alongside as references and reproduce the five-seed deficit to within 0.008 (1.360 versus 1.367; the fifty-two-seed level is 1.381); everything else, task, architecture, and recipe, is unchanged, with five seeds per condition.

Figure 3 reports the fraction of the aliased deficit each condition removes. The two ends anchor as expected: with exact per-step evidence both conditions reach the regime-aware reference within seed noise, and at $\sigma = 32$, where the evidence is nearly uninformative, both fall to within a few points of the Family B null. Between them the curves separate. The memoryless condition tracks only the single-step signal-to-noise ratio and has lost most of the recovery by $\sigma = 2$ (removing 14%), whereas the recurrent condition still removes 74% there and 31% at $\sigma = 4$, the gap peaking at 0.60. This is online latent inference in the sense of hidden-parameter and meta-reinforcement-learning formulations (Doshi-Velez & Konidaris, 2016; Rakelly et al., 2019; Zintgraf et al., 2020), with the recurrent policy aggregating weak evidence across steps (Hausknecht & Stone, 2015). The regime is drawn per instance independently of the route at every σ, so this sweep sits inside Family B throughout, and its $\sigma = 0$ end coincides with Family B's own regime-signal reference rather than with Family A. Recurrence recovers what the observation stream makes inferable, so a graded stream yields graded recovery. This sweep also required an episode-level memory without the per-tour reset used in the main text, which indicates that how much is recoverable and over what horizon are separate axes. On its own that horizon contributes almost nothing: the same memory with the reset removed, given no evidence channel, removes 0.017 of the Family B deficit over fifty-two seeds (CI $[+0.011, +0.023]$), against -0.001 with the reset in place (paired $+0.019$, CI $[+0.010, +0.027]$, $t = 4.4$). The sweep's recovery is therefore what the evidence channel makes inferable, not what the longer horizon supplies. Neither arm can read the regime in any case: it enters the objective alone and reaches neither the observation, the mask, nor the memory's input, so a rollout is identical under either regime and only its cost differs.

B.4 A RECURRENT-CONTENT CONTROL IN FAMILY A

A fifth condition separates recurrent computation from the burden content the memory carries. It repeats the recurrent condition exactly, same width, parameters, and per-tour reset, but feeds the

memory per-step $\mathcal{N}(0,1)$ noise in place of the served (demand, burden) pair, keeping the recurrent structure while removing the burden content. The conditions of Table 2 and the one-bit flag were trained under one recipe in a single deterministic run, so every contrast is within-run and paired by seed. The noise-fed module removes 0.46 of the deficit (95% CI $[+0.42, +0.51]$, t, df=51), resolvably more than the same-width static channel (paired $+0.090$ in cost, CI $[+0.078, +0.102]$, $t = 15.4$) and resolvably less than the full-content condition (paired $+0.083$ in cost, CI $[+0.074, +0.091]$, $t = 19.4$).

A constant-fed counterpart settles what the recurrence supplies. Feeding the same module a constant, an exact step index, a rescaled step index, or Gaussian noise gives 0.479, 0.443, 0.475 and 0.463. The four are not separable ($F(3, 153) = 0.79$, $p = 0.50$) and differ by at most 0.09 of the deficit pairwise, and 0.11 simultaneously over the six contrasts, against a common recovery of 0.465 (CI $[+0.429, +0.501]$); this design resolves about 0.07 in ρ, so they are equal to within a quarter of the effect each produces. With a constant input the hidden state is an exact function of steps since the last depot and of nothing else, so the reset structure acts as a clock of depth into the instance's own tour, and the input contributes nothing beyond it. Deleting the reset and changing nothing else leaves that hidden state a function of the absolute decode step, and with a constant input that makes it the same vector for every instance in the batch: the channel stops saying anything about the particular route being built. Recovery falls to 0.127 (CI $[+0.069, +0.184]$), a paired drop of 0.352 in ρ (CI $[+0.282, +0.421]$, $t = 10.1$). Absolute position in the episode is an unsuitable temporal reference for this task by construction: the burden resets at every depot visit, so only depth into the current tour tracks it.

Zeroing the hidden state at the depot does three things at once. It re-anchors the temporal progression to the current tour, it holds the state away from the fixed point an unreset recurrence on a constant input tends toward, and it shortens the path gradients travel back through. Two further arms separate them, both keeping the reset and changing only when it fires. The first fires on a fixed period of four decode steps. The second fires on a real depot schedule belonging to a neighboring instance in the same batch: each rollout is reset on the depot visits of the same-numbered rollout of the adjacent instance.

Instrumenting the trained policies shows how closely the borrowed schedule matches the arm's own. Over the steps at which a decision is still being made, the borrowed schedule fires on 0.274 of steps against 0.254 for the arm's own depot visits, its intervals average 3.59 steps against 3.75, and the depth of the state when the decoder reads it averages 1.57 with standard deviation 1.46 against 1.56 and 1.44. That last quantity is the one that fixes both how far the state has moved from its fixed point and how far gradients travel, and it is matched to about one percent. Permuting the schedule eliminates temporal alignment between the clock and the route: the correlation between the depth the decoder reads and this instance's own accumulated burden falls from 0.90 to 0.15. Recovery falls with it, to -0.038 for the periodic reset (CI $[-0.103, +0.028]$) and $+0.001$ for the borrowed one (CI $[-0.067, +0.068]$), paired drops of 0.516 (CI $[+0.451, +0.582]$, $t = 15.9$) and 0.478 (CI $[+0.409, +0.547]$, $t = 13.9$). The depot reset therefore supplies alignment with this instance's own tour, not re-anchoring at the right rate. What the decoder reads from the aligned reset is depth at a precision one bit cannot reach: the within-tour step index at full precision, given to the same module, is not separable from the noise-fed control ($+0.020$ in ρ, CI $[-0.025, +0.065]$, $p = 0.37$), while the one-bit step-count flag is resolvably worse than that control (paired $+0.267$ in ρ, CI $[+0.212, +0.322]$, $t = 9.8$; the control recovers about 2.4 times as much) and falls short of the full recurrent summary ($+0.757$ in ρ, CI $[+0.687, +0.828]$). How finely the step count is read is what separates them. The policies trained with the two clocks also read them differently: the mean norm of the hidden state at read time is 3.41 under the arm's own schedule and 0.52 under the borrowed one, so a policy given a clock it cannot use learns to down-weight the channel.

The borrowed schedule is a counter on 97% of decisions and clamped on the rest. An episode ends when its customers are served and a finished rollout is forced to the depot at every remaining step, so once the donor's episode ends its indicator is true throughout: on 2.8% of the steps at which the recipient is still deciding the clock remains clamped near zero rather than counting, which is what puts the borrowed reset rate slightly above the arm's own. The periodic arm has no such tail and yields an indistinguishable result. Both misaligned arms drive the reset from outside the policy, so they measure alignment together with the policy's ability to shape its own reset cadence; the correspondence they destroy is mechanical rather than learned, because the depth counted from a

Table 2: Recurrent-content control in Family A (fifty-two seeds). Four middle conditions keep the recurrent structure of the full condition and differ only in what the memory consumes: a constant, the step index within the current tour, that index rescaled to the served pair's range, and Gaussian noise. They are not separable from one another. The no-reset row instead keeps the constant input and deletes the per-tour reset, so its hidden state is a function of the absolute decode step. The next two keep both the constant input and the reset and change only when it fires: on a fixed four-step period, and on the depot schedule of a neighboring instance in the same batch.

Decoder condition	Mean route cost $\pm$ sd	Deficit removed	95% CI
Direct access	13.67 ± 0.02	(ref.)	—
Aliased	13.84 ± 0.03	—	—
Aliased + content-free channel	13.85 ± 0.05	-0.07	$[-0.17, +0.02]$
Aliased + recurrent, constant input	13.76 ± 0.03	$+0.48$	$[+0.42, +0.53]$
Aliased + recurrent, scaled tour index	13.76 ± 0.02	$+0.48$	$[+0.43, +0.52]$
Aliased + recurrent, noise input	13.76 ± 0.02	$+0.46$	$[+0.42, +0.51]$
Aliased + recurrent, tour index	13.76 ± 0.02	$+0.44$	$[+0.40, +0.49]$
Aliased + recurrent, constant, no reset	13.82 ± 0.03	$+0.13$	$[+0.07, +0.18]$
Aliased + recurrent, constant, periodic reset	13.85 ± 0.03	-0.04	$[-0.10, +0.03]$
Aliased + recurrent, constant, neighbor's reset	13.84 ± 0.03	$+0.00$	$[-0.07, +0.07]$
Aliased + recurrent, (demand, burden)	13.68 ± 0.02	$+0.95$	$[+0.92, +0.99]$

rollout's own last depot visit fixes how many customers have been added since its burden was last zeroed.

Recurrence is what makes that depth available, not the extra channel: the same constant without the recurrence removes nothing, and the two differ by 0.091 in cost (CI $[+0.076, +0.107]$, $t = 12.0$). The burden content is worth a further 0.489 above the pooled burden-independent level (CI $[+0.444, +0.533]$). The summand on its own accounts for none of it: the served demand and burden handed to the decoder without a memory sit at -0.073, within noise of the content-free channel's -0.072 (paired -0.000, CI $[-0.089, +0.088]$, $t = -0.01$), while a memory over that same input is worth 1.026 more (CI $[+0.949, +1.103]$, $t = 26.7$). What the decoder lacks is the running sum, not the values entering it. Recovery therefore comes in steps rather than gradually, but the steps are not a nested ladder. Recurrence without the per-tour reset is worth 0.199 over the static channel (CI $[+0.110, +0.289]$, $t = 4.5$) and the aligned reset a further 0.352; in Family B, where nothing accumulates within a tour, deleting the same reset is worth 0.019. A reset at the right rate on the wrong tour removes the first step rather than building on it: the periodic and borrowed-schedule arms sit below the unreset arm by 0.164 (CI $[+0.105, +0.223]$) and 0.126 (CI $[+0.066, +0.187]$), and they are not separable from the static channel ($F(2, 102) = 1.61$, $p = 0.21$; pooled -0.036, CI $[-0.098, +0.025]$).

Seeding is per (arm, seed), so every smaller seed set is a subset of the fifty-two. At eight seeds the recurrent and coarse-signal arms are close to their five-seed values (0.909, CI $[+0.79, +1.03]$ for the full-content condition; 0.408, CI $[+0.29, +0.52]$ for the one-bit flag of Appendix B.2). The content-free control is the exception. One seed beyond the first five finishes 0.27 *worse* than its own aliased baseline, widening that arm to -0.101, CI $[-0.60, +0.40]$. Its training trace shows no fault. At fifty-two seeds the arm removes -0.072 of the deficit, CI $[-0.169, +0.024]$, consistent with zero under the t test ($t = -1.5$), the Wilcoxon signed-rank test ($p = 0.32$), and a 10% trimmed mean (-0.043). Its separation from the one-bit flag is -0.073 in cost at fifty-two seeds, CI $[-0.088, -0.058]$, $t = -9.9$. This arm's five-seed interval, $[-0.02, +0.14]$, is the outlier: its five-seed spread (0.067) is one fifth of its fifty-two-seed spread (0.345), a draw in the lowest 0.3% of the sampling distribution of the sample standard deviation, so that interval is spuriously narrow and does not contain the fifty-two-seed estimate. Seed 5 is the arm's only genuine excursion: removing it alone returns the cost residuals to near-normality (skew $-1.56 \to +0.22$, excess kurtosis $+6.77 \to -0.38$) and the arm to -0.043, CI $[-0.121, +0.035]$. Four further seeds have ρ below -0.5 with ordinary cost residuals; they hold the first, second, fourth and tenth smallest aliased deficits of the fifty-two, where the normalization magnifies an ordinary residual. The conclusion this control supports, that it recovers nothing, holds across sample sizes; the five-seed precision does not.

The full-content arm moves the other way. At five and ten seeds its interval contains full recovery (0.930, CI $[+0.72, +1.14]$; 0.934, CI $[+0.83, +1.04]$); at fifty-two seeds it is 0.954,

Table 3: Scale check at $n = 50$ (five seeds, matched budget): mean route cost $\pm$ sd. Each family's four conditions are trained in one run, so every contrast is within-run and paired by seed.

Decoder condition	Family A	Family B
Direct access (reference)	32.40 ± 0.07	34.23 ± 0.03
Aliased	32.94 ± 0.07	37.81 ± 0.05
Aliased + content-free channel	32.88 ± 0.06	37.80 ± 0.06
Aliased + recurrent memory	32.35 ± 0.05	37.85 ± 0.07

CI $[+0.918, +0.989]$, and the residual to the direct-access reference is $+0.008$ in cost, CI $[+0.002, +0.014]$ over seeds, so both exclude full recovery. Those intervals are over training seeds on one evaluation set, and for this contrast we can say what that costs. Per-instance costs for the reference, aliased and recurrent arms at five seeds (`results/per_instance/` in the supplement) split the paired difference into an instance component that the shared evaluation set never averages away ($\hat{\sigma} = 0.048$) and a seed-by-instance component that does (0.313); only the first fails to shrink with seed count, and the second is already inside the seed interval. Carrying the first through widens the residual interval to $[+0.001, +0.015]$, which still excludes zero, and the recovery bound to 0.993, which still excludes full recovery. The residual stops resolving at an instance component of 0.087 and the recovery bound at 0.082; that component is itself estimated from five seeds; and we apply no family-wise correction across the arm comparisons. The reset arms carry the same term, measured the same way on the same evaluation set. Their instance components are 0.088 for the aligned-against-borrowed contrast and 0.088 for the borrowed-against-unreset one, which widen those intervals from $[+0.409, +0.547]$ to $[+0.402, +0.554]$ and from $[-0.187, -0.066]$ to $[-0.194, -0.058]$. Neither approaches zero, so the shared evaluation set is not what carries them.

B.5 Scale check at $n = 50$

Both families were rerun with $n = 50$ customers and every other setting unchanged, including the training budget (Table 3). The deficits grow with instance size, to 0.54 in Family A and 3.58 in Family B, and the dissociation is unchanged: the recurrent summary recovers the Family A deficit (its interval spans full recovery) and none of Family B's (indistinguishable from zero), with the content-free control recovering nothing in either. Under the matched budget the absolute costs at $n = 50$ are not converged benchmarks; the comparison is internally matched.

C Diagnosing the Mixed-Backhaul Failure

This appendix gives the full analysis behind the case study of Section 6: the feasibility rule in the released system, the observation aliasing it induces, the released adapter, and the cross-backbone reproduction.

C.1 The feasibility rule and the decoder observation

The RouteFinder class-2 branch checks a conjunction of capacity constraints. For a linehaul customer, the feasibility condition in Equation 1 depends on both u_ℓ and u_b; for a backhaul customer, the branch checks $u_b + d \le C$. The released decoder context instead exposes

$$a = C - \begin{cases} u_\ell, & u_b = 0, \\ u_b, & u_b > 0. \end{cases} \tag{3}$$

When $u_b > 0$, states with different u_ℓ can have the same a but different linehaul feasibility boundaries. The observation in Equation 3 therefore aliases states relevant to the class-2 choice.

This decoder-context alias is local: it does not rule out route-state information elsewhere in the policy. It is an instance of perceptual aliasing, where distinct underlying states induce the same observation even though their decision consequences differ (Whitehead & Ballard, 1991). Relative to this observation, the decision is partially observable (Kaelbling et al., 1998); it also resembles a state abstraction that merges states which need not be behaviorally interchangeable (Li et al., 2006), which is where an aggregation fails the bisimulation condition (Givan et al., 2003).

Table 4: Mixed-backhaul diagnosis. The two official test sets are the same 1000 instances: every stored field is identical except the backhaul class, which selects the feasibility rule rather than the draw, so the class-2 contrast is within-instance. The EAL row describes its released recipe.

Finding	Detail	Implication
Class-2 linehaul mask	Requires $d \leq C - \max(u_\ell, u_b)$	Feasibility depends jointly on two loads.
Released decoder scalar	Equation 3	The decoder observation aliases class-2 states.
Default-generator composition	0% class-2; EAL recipe about 25%; common pool about 50%	Coverage and input representation cannot be conflated.
Initial official-set contrast	$\Delta_{\text{pre}} = 6.23\,\text{pp}$	MB has a substantial zero-shot deficit relative to the seen VRPB variant.
Released EAL recipe	$\mathcal{R} = 0.018$	The released configuration does not close the MB contrast under its reported recipe.
Zero-training coupled-state substitution (Appendix I)	$+0.54\,\text{pp}$ on MB; no VRPB change	The coupled quantity is decision-relevant, but exposure without learning is not the full repair.

C.2 Coverage and regime signaling are separate factors

The decoder alias alone cannot explain the complete zero-shot deficit. The default generator contains 0% class-2 instances, whereas the released Efficient Adapter Layers (EAL) recipe contains about 25% and the common mixed-backhaul pool used for the ablation contains about 50%. The initial mixed-backhaul (MB) minus VRPB contrast therefore combines a changed regime with absent coverage. The experiments below keep this common pool fixed and ask which added signals matter after exposure.

The released EAL modification replaces both embeddings. On the initialization side, it adds the class-2 indicator $\mathbf{1}\{\text{class}{=}2\}$. On the context side, it appends $\mathbf{1}\{\text{class}{=}2\}(C - u_b)$ and two coordinates of the current depot. These broader embedding changes make regime information available, but they do not explicitly supply the coupled headroom in Equation 1. In the matched ablation of Appendix D, the repair is carried almost entirely by the regime signal: the flag-only condition reaches $\mathcal{R} = 0.879$ without exposing the coupled state, and adding the load signals changes the result little. The released recipe's $\mathcal{R} = 0.018$ is an unmatched contextual reference, reflecting its own data composition and injection form, not a component-level comparison.

C.3 The alias is shared across backbones

The switched-load observation in Equation 3 is not specific to one model. The released RouteFinder transformer, MVMoE, and MTPOMO draw their decoder context from two classes in one repository, `MTVRPContextEmbeddingRouteFinder` for the RouteFinder checkpoint and `MTVRPContextEmbedding` for MTPOMO and MVMoE, and the two carry byte-identical copies of this computation, so none of the three computes the coupled quantity in Equation 1; Appendix I gives the code paths. The zero-shot deficit recurs accordingly: the mixed-backhaul minus VRPB contrast is $6.23\,\text{pp}$ for the RouteFinder transformer, $6.41\,\text{pp}$ for MTPOMO, and $6.44\,\text{pp}$ for MVMoE; the two further RouteFinder backbones we measured, its mixture-of-experts and POMO variants, give $6.34\,\text{pp}$ and $6.44\,\text{pp}$. Because all five share this code path, the deficit is a shared-implementation effect rather than evidence across independent architectures; Appendix D asks whether the repair transfers as well.

C.4 Repair metric

For the RouteFinder MB contrast, we report the repair ratio

$$\mathcal{R} = \frac{\Delta_{\text{pre}} - \Delta_{\text{post}}}{\Delta_{\text{pre}}}, \qquad \Delta_{\text{post}} = g_{\text{MB}} - g_{\text{VRPB,base}}, \tag{4}$$

where g denotes mean benchmark gap and $\Delta_{\text{pre}} = 6.23\,\text{pp}$. We use the base VRPB gap as the reference because it is the corresponding seen class-1 variant in RouteFinder's benchmark. Thus $\mathcal{R} = 1$ denotes equality to that reference.

C.5 What the class-2 reference does and does not carry

Δ_{pre} is the difference of two gaps to time-limited heuristic references. Two measurements bound what the references contribute to it.

The contrast is within-instance. The two official test sets are the same 1000 instances and differ only in the field that selects the feasibility rule, so nothing here compares two draws. Relaxing the precedence on those instances creates room for cost reduction that an optimization solver can exploit: under the same time limit the benchmark reference lowers cost by 0.836 on average (95% CI $[+0.818, +0.855]$), improving on 999 of 1000 instances. The released model lowers it by 0.054, improving on 617 instances and worsening on 362. Given strictly more freedom, its response is close to noise with a small positive drift, which is the failure the class-2 flag repairs.

One comparison removes the class-2 reference from the argument entirely. The model's class-2 solutions cost 3.57% more than the reference's *class-1* solutions (95% CI $[+3.50, +3.64]$, worse on 999 of 1000), and those solutions satisfy the stricter precedence, so they are feasible for class 2.

The references carry no optimality certificate, so the exact magnitude of the deficit is open. Reference error accounts for all of Δ_{pre} only if the reference's class-1 solutions exceed the optimum by approximately 6% while its class-2 solutions coincide with it.

A second solver is beaten by the benchmark reference on 100% of instances in both classes, but by 3.85% on class 1 and 10.45% on class 2. The widening is attributable to the second solver, which does not respond to the relaxation at all (mean change $+0.004$, CI $[-0.004, +0.012]$; 646 of 1000 costs unchanged across the two classes) and whose class-2 solution is beaten by its own class-1 solution on 173 instances. The class-2 reference therefore rests on one solver, as does the cost-reduction figure above.

D Ablating Candidate Signals

This appendix details the signal ablation summarized in Section 6: the conditional side paths and the five-seed ablation of Table 5, a 2×2 design over load signals and a class-2 flag with an added content-free control.

D.1 Conditional side paths

To compare candidate signals without changing RouteFinder's attention blocks or routing objective, we add conditional side paths. Each candidate slot supplies a numeric value and an availability bit. The latter distinguishes an inapplicable slot from a numeric zero, following the general value-plus-indicator pattern used for missing data (Che et al., 2018). In this setting, however, zero availability means that a configuration-level concept does not apply; it does not represent an unobserved measurement.

The side paths act on both parts of the policy input: an encoder initialization embedding and the decoder context. Given a slot vector s, the decoder path has the form

$$\begin{aligned} z &= \text{MLP}(s), \\ (\gamma, \beta) &= \text{split}(W_0 z + b_0), \\ \widetilde{c} &= c \odot (1 + \gamma) + \beta, \end{aligned} \tag{5}$$

where c is the original decoder context and W_0, b_0 are zero-initialized. The encoder path uses the same FiLM construction (Perez et al., 2018) on its initialization embedding. Zero initialization preserves the base outputs at initialization (Zhang et al., 2023; Hu et al., 2022).

For MB, every condition uses the same encoder-side path driven by a node-level is-depot slot. The experimental profiles vary only the decoder-side slots. Among these decoder profiles, flag-only contains only the class-2 flag and its availability bit. The constant-bit control uses the same one-bit

Table 5: Five-seed ablation on the common pool. All conditions use the same encoder-side is-depot slot, dual-side FiLM architecture, full fine-tuning, matched budgets, and the same seeds. The constant-bit control uses a fixed one-bit value with the same decoder-side form as flag-only. $\mathcal{R}$ is defined in Equation 4.

Condition	Added signal(s)	$\mathcal{R}$
Base context	None beyond base context	0.472
Constant-bit control	Fixed bit, no regime information	0.434
Loads-only	Separate linehaul and backhaul load signals	0.525
Flag-only	One-bit class-2 flag	0.879
Loads+flag	Separate load signals and class-2 flag	0.907
Full profile	Seven-slot profile	0.805

input form but supplies a fixed value for every instance, without regime information. The full profile has seven scalar dimensions before availability bits (Appendix G), including the two loads and coupled headroom. Flag-only and the full profile have nearly identical side-path parameter budgets ($\approx 30{,}000$), so the contrast is not one of capacity. All conditions use the same side-path architecture.

D.2 Evaluation protocol

The primary evaluation uses the official RouteFinder 100-customer test sets, with 1000 instances per variant. Each instance is evaluated using dihedral-8 augmentation and 100 greedy starts per augmentation. We compare models on the same official instances whenever possible. Gap is percentage distance to the benchmark's best-known-solution (BKS) reference, which is a time-limited heuristic reference rather than an optimality certificate.

The 2×2 MB ablation uses the common pool, whose instances are about 50% class-2, and five independently trained seeds. All conditions use full fine-tuning, the same training budget, the same side-path form, and matched random seeds. The cross-backbone comparison uses the same pool, with four seeds on MTPOMO and a single configuration on MVMoE. Appendix H provides the implementation environment.

D.3 Separating regime and load signals

Table 5 separates two candidate additions to a base-context side path: separate linehaul and backhaul load signals, and the class-2 flag. The base-context condition feeds only quantities already present in the released decoder context. The joint condition combines the load signals and the flag.

The content-free constant-bit control reaches $\mathcal{R} = 0.434$ and does not reproduce the flag-only gain.

Every row uses the same one-pass recipe, and three passes do not let base context bridge the gap. At three passes it stands at 6.677% against flag-only's 3.586% at the same budget, 3.09 pp behind (CI $[+2.94, +3.25]$, no seed overlap), and at the rate it improved from one to three passes it would need about twelve passes to reach that. Both conditions improve with budget — base context by -0.691 pp (CI $[-0.886, -0.495]$), flag-only by -1.075 pp ($[-1.177, -0.974]$) — and the gap between them widens rather than closes (difference in differences -0.385 pp, CI $[-0.579, -0.191]$; intervals here are unpaired, since the seeds set initialization only and the pool order is fixed). Both also pay more on the seen class at three passes, base context $+0.133$ pp and flag-only $+0.042$ pp, and netting that out moves the difference in differences to -0.294 pp without closing it. Two budget levels cannot locate an asymptote, so this bounds the budget over which the advantage persists rather than establishing converged quality.

Measured against the benchmark reference's class-1 solutions, which satisfy the stricter precedence and are therefore class-2 feasible, the ordering does not depend on the class-2 reference at all: at three passes flag-only's class-2 solutions cost 2.59% less than those (CI $[-2.63, -2.54]$, better on 882 of 1000 instances), against 1.58% less at one pass (757 of 1000) and 0.30% more for base context at three passes (384 of 1000).

Table 6: Three conditions from Table 5, re-run on a rebuilt software stack. Five seeds, varying side-path initialization and rollout sampling on a fixed pool order.

Condition	$\mathcal{R}$ (re-run)	95% CI	$\mathcal{R}$ (Table 5)
Base context	0.452	[0.430, 0.474]	0.472
Constant-bit control	0.439	[0.423, 0.455]	0.434
Flag-only	**0.886**	[0.876, 0.896]	**0.879**

D.4 Reproducibility across software environments

The ablation of Table 5 was run under one software environment. We re-ran three of its conditions end to end under an independently rebuilt software build, holding the released checkpoint, the mixed-backhaul test file, the common pool and the ablation code fixed and changing only the software build and the initialization seeds (Table 6).

All three estimates remain within about 0.02 of their original values, and both contrasts drawn from them are unchanged: the constant bit is indistinguishable from base context ($+0.08$ pp in gap, seed ranges overlap), so an added decoder channel by itself recovers nothing, and flag-only beats it by 2.79 pp with no seed overlap, so the repair is the bit's content and not the channel.

Adding the flag to the base-context condition lowers the MB gap by 2.53 pp. Adding the separate load signals after the flag lowers it by a further 0.18 pp. The load signals without the flag improve over the base-context condition by 0.33 pp. The coupled-state substitution in Table 4 confirms that the missing load quantity is decision-relevant. The alias therefore identifies a local mismatch in the released observation. In the common-pool ablation, the flag yields a larger measured improvement than the separate load signals.

Under the reported recipes, flag-only has a higher aggregate repair ratio than the full profile (0.879 versus 0.805).

D.5 Released EAL, cross-backbone portability, and retention

As described in Appendix C, the released EAL configuration replaces the initialization and context embeddings. Under its released MB recipe, which differs from the common-pool ablation in data composition and injection form, its MB repair ratio is $\mathcal{R} = 0.018$.

Retention off the rehearsed class. Performance degradation on the rehearsed class, VRPB, is comparable across all conditions: base context $+0.676$ pp (CI $[+0.594, +0.758]$), constant bit $+0.696$ pp ($[+0.582, +0.809]$), flag-only $+0.625$ pp ($[+0.563, +0.687]$), five seeds each on the official VRPB set following the evaluation protocol of all other conditions. The three are not separable there: the largest pairwise gap is 0.071 pp (CI $[-0.042, +0.184]$; $F = 1.32$, $p = 0.30$, and the per-seed rank sums are exactly tied). A run with no side path at all, at the same learning rate on the same pool, costs $+0.646$ pp. Seeds here set initialization only — the pool order is fixed, so the first batch is identical across arms and seeds — and the intervals are unpaired for that reason.

Off the rehearsed class the cost is smaller and the conditions stop coinciding. On the four seen variants this pool does not cover, CVRP, VRPTW, OVRP and VRPL, at the same protocol and from the same five checkpoints, base context costs $+0.409$ pp (CI $[+0.333, +0.484]$) and flag-only $+0.504$ pp ($[+0.412, +0.596]$). Flag-only's $+0.095$ pp excess is not separable from zero pooled over the four ($[-0.004, +0.195]$) and survives a four-way Holm correction on one variant, VRPTW ($+0.201$ pp, $[+0.070, +0.333]$, adjusted $p = 0.03$); CVRP ($+0.099$ pp), VRPL ($+0.077$ pp) and OVRP ($+0.004$ pp) do not. What is separable is the comparison between the two settings: differencing each checkpoint against itself, flag-only's cost relative to base context is $+0.147$ pp higher off the rehearsed class than on it ($[+0.016, +0.277]$). The bulk of what fine-tuning on this pool costs is a property of the pool, within 0.1 pp of identical across conditions on the rehearsed class and incurred in full by a run with no side path at all. A residual of order 0.1 pp off the rehearsed class is not.

The parameter-matched control isolates that residual. The constant-bit control carries flag-only's $29{,}536$ side-path parameters against base context's $29{,}792$, and off the rehearsed class it costs $+0.424$ pp, landing with base context's $+0.409$ pp rather than flag-only's $+0.504$ pp ($+0.015$ pp

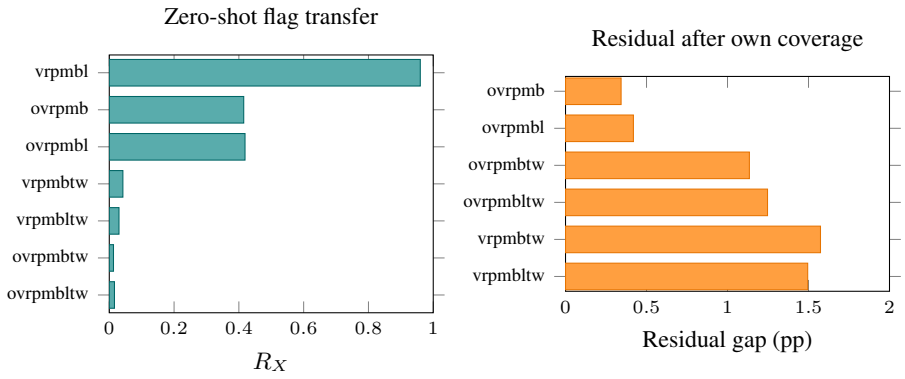


Figure 4: Mixed-backhaul transfer varies across variants. The left panel uses each variant's own initial deficit as the denominator; the right panel is an absolute residual relative to the base model on the relevant seen counterpart. Variant names follow RouteFinder: `vrpmb` is the mixed-backhaul problem, and `o`, `l` and `tw` add open routes, a route-duration limit and time windows.

apart, CI $[-0.081, +0.111]$). Side-path capacity does not produce the excess. Neither is the excess resolvable against that control: flag-only exceeds it by $+0.080$ pp pooled over the four variants ($[-0.025, +0.186]$), one variant survives a four-way Holm correction (VRPTW, $+0.212$ pp, $[+0.084, +0.340]$, adjusted $p = 0.03$), and OVRP reverses sign (-0.088 pp, $[-0.303, +0.126]$). What reproduces against both comparators is the cross-setting shift: moving off the rehearsed class, flag-only's cost rises $+0.147$ pp more than base context's ($[+0.016, +0.277]$) and $+0.151$ pp more than the constant bit's ($[+0.028, +0.274]$), while the two comparators do not separate from each other on that quantity (-0.005 pp, $[-0.094, +0.085]$); a Holm correction over the two contrasts leaves both at $p = 0.05$. The residual attaches to the flag-only profile, not to side-path capacity. Attributing it to the bit's content would require a control whose slot varies during fine-tuning and is clamped off the rehearsed class; the constant bit is clamped throughout.

The residual is an excess over the matched condition rather than an absolute cost of leaving the rehearsed class. Measured against flag-only's own cost on the rehearsed class it is -0.121 pp, and against its cost averaged over all five seen variants measured here, -0.024 pp: flag-only costs less off the rehearsed class than on it, and the excess exists only relative to a condition that differs from it by one channel. And the released recipe's 3.19 pp class-1 cost was measured under its own protocol — a different software stack, the full multi-task pool, a learning rate of 3×10^{-4} against this ablation's 10^{-4}, and one run rather than five. Plain full fine-tuning at that learning rate on this pool already costs $+1.11$ pp.

Because the aliased state-embedding path is shared across backbones (Appendix C), we test whether the added state signal transfers. Relative to a matched adaptation without it, the added signal lowers the MTPOMO mixed-backhaul gap by 1.92 pp (four seeds, 95% CI $[1.71, 2.12]$) and produces a smaller positive effect of 0.71 pp on MVMoE ($[0.64, 0.78]$).

E Mapping the Mixed-Backhaul Family

The case study of Section 6 is one point in a broader family of variants, which this appendix maps. We map three quantities for seven mixed-backhaul variants. First, D_{pre} is the base-model gap difference between a variant and its seen counterpart. Second, R_X is the fraction of that particular variant's deficit recovered by flag-only without further training:

$$R_X = \frac{D_{\text{pre}}(X) - D_{\text{flag}}(X)}{D_{\text{pre}}(X)}. \tag{6}$$

It is not the RouteFinder MB repair ratio in Equation 4. Third, for variants that receive their own class-2 coverage, we report the residual gap relative to the base model on the seen counterpart.

The family map (Figure 4) has three implications. First, the initial deficit is widespread, ranging from about 5.3 pp to 14.7 pp. Second, zero-shot flag transfer is strong only for vrpmbl: elsewhere it

Table 7: Signals in the full mixed-backhaul profile. The depot-coordinate signal contributes two scalar dimensions, for seven scalar dimensions before availability bits.

Signal	Definition
Used linehaul load	u_ℓ
Used backhaul load	u_b
Coupled headroom	$C - \max(u_\ell, u_b)$
Class-2 flag	$\mathbf{1}\{\text{class}{=}2\}$
Current depot coordinates	Coordinates of the current depot
Normalized depot count	Number of depots divided by 3

recovers $R_X \approx 0.42$ on the two open-route variants without time windows and only 0.01–0.04 on the four with them. A one-bit regime signal is consequently not a family-wide zero-shot solution.

Third, own-coverage training reduces the residuals on those two variants to about 0.3–0.4 pp. Across the four time-window variants, residuals remain between about 1.1 and 1.6 pp. Own coverage therefore reduces the initial deficits without fully repairing every variant. Compared with fine-tuning without an added conditional path, the flag has a visible incremental effect on the open-route axes and little incremental effect on the time-window axes. This pattern motivates the interpretation of the class-2 bit as a *regime signal*: it is useful in a mixed distribution in which the policy must distinguish whether the class-2 constraint regime is active.

F A Multi-Depot Boundary Case

This appendix reports a boundary case adjacent to the case study of Section 6. Multi-depot CVRP (MD) is a contrasting setting rather than a second instance of the coverage failure: all evaluated MD instances have three depots, so there is no within-distribution regime switch for a signal to identify. The question here is only whether released structural inputs help beyond the effect of adding a conditional path. Over five seeds, a content-free constant-channel control already lowers the MD gap by 1.04 pp relative to no-signal fine-tuning, and released EAL structural embeddings lower it by a further 0.85 pp beyond that control. Structural inputs therefore contribute beyond the added path. This setting does not identify why the constant channel also helps, and the conclusion is specific to the three-depot configuration.

G Profile and Side-Path Details

Each profile value is paired with an availability bit. A zero availability bit denotes that the corresponding configuration-level concept is inapplicable, rather than that the observed value happens to be zero. Table 7 lists the full mixed-backhaul profile. The flag-only profile activates only the class-2 flag and its availability bit. The constant-bit control uses the same signal and availability bit with a fixed value for every instance.

The profiles share the same side-path form and hidden widths, differing only in the width of their first projection; flag-only and the full profile have nearly identical parameter budgets ($\approx 30{,}000$).

For a decoder context c, Equation 5 applies FiLM conditioning to the original context embedding. An analogous path conditions the encoder initialization embedding. Both output projections are zero-initialized so that the added paths begin from the base mapping.

Table 8 reports the direct matched gap comparisons underlying the mixed-backhaul ablation and the multi-depot boundary case.

Table 8: Direct matched gap comparisons. For MB, negative values favor the first named condition. For MD, the reported positive value favors released EAL.

Setting	Comparison	Difference	95% CI
MB	Flag-only minus base context	-2.53 pp	$[-2.72, -2.35]$
MB	Flag-only minus loads+flag	$+0.18$ pp	$[+0.10, +0.26]$
MB	Loads-only minus base context	-0.33 pp	$[-0.57, -0.10]$
MD	Released EAL over constant channel	$+0.85$ pp	$[+0.65, +1.06]$

H Experimental Details

All experiments were run with PyTorch 2.5.1 and CUDA 12.4 on a single NVIDIA A100 80GB, enforcing determinism (`cudnn.deterministic=True`, `cudnn.benchmark=False`, TF32 disabled, `torch.use_deterministic_algorithms(True)`, `CUBLAS_WORKSPACE_CONFIG=:4096:8`). Repeated runs of the same configuration reproduce bit-for-bit. The released-model (RouteFinder) experiments use the RL4CO stack (Berto et al., 2025b); the controlled experiments (Sections 3–5) use a self-contained POMO implementation with no shared code.

The MB ablation conditions use full fine-tuning with Adam (learning rate 10^{-4}, batch size 64, gradient-norm clipping at 1.0) for one pass through the common mixed-backhaul pool, over five seeds. The constant-bit control shares the flag-only side-path form but supplies a fixed value for every instance. Training and evaluation instances are disjoint, and all gaps use the official RouteFinder best-known-solution references.

The released EAL configuration uses `eal-full` with `adapter_only=False`. It replaces the initialization and context embeddings with `MTVRPInitEmbeddingFull` and `MTVRPContextEmbeddingFull`. The added initialization feature is $\mathbf{1}\{\texttt{backhaul_class} = 2\}$; the added context features are $\mathbf{1}\{\texttt{backhaul_class} = 2\}(C - u_b)$ and two coordinates of the current depot. Both EAL and the side-path conditions use full fine-tuning. In the MB ablation, candidate signals, including the class-2 flag, enter through decoder-side profiles; every condition shares the same encoder-side is-depot path.

Code reproducing the controlled families and the robustness studies, together with the result files they produce, is provided as anonymous supplementary material.

I Implementation Details for the Diagnosis

This appendix details the class-2 feasibility calculation and the decoder observation in the released RouteFinder system.

MB feasibility mask. The RouteFinder MTVRP environment implements the relevant class-2 branch in `get_action_mask`. For a class-2 linehaul customer, it conjoins $d \le C - u_\ell$ and $d \le C - u_b$, equivalently $d \le C - \max(u_\ell, u_b)$; for a backhaul customer, it checks $u_b + d \le C$.

Base decoder context. The decoder context derives a phase-switched `available_load`, current time, and remaining distance.

Released EAL embeddings. The released EAL configuration adds the class-2 indicator $\mathbf{1}\{\texttt{backhaul_class} = 2\}$ to `MTVRPInitEmbeddingFull`. It appends $\mathbf{1}\{\texttt{backhaul_class} = 2\}(C - u_b)$ and two coordinates of the current depot to `MTVRPContextEmbeddingFull`.

Coupled-state substitution and side paths. For the zero-training substitution, we replace the first context slot with coupled headroom on class-2 instances. The side paths use availability-aware slots and FiLM modules on both the decoder context and encoder initialization embedding.

Shared state-embedding path across backbones. The phase-switched `available_load` is produced by `MTVRPContextEmbedding` in the MTPOMO and MVMoE decoders and by

`MTVRPContextEmbeddingRouteFinder` in the RouteFinder decoders. The two classes are distinct but carry byte-identical copies of the computation. The coupled quantity $C - \max(u_\ell, u_b)$ is therefore absent from the decoder observation in all three, which is why the zero-shot deficit recurs across backbones.