SELECTION REGRET: LOCAL ACCURACY MISRANKS CLOSED-LOOP RELIABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Local next-transition accuracy is a tempting way to choose components for a system that will eventually run in a loop. This paper shows that the metric can pick the wrong component once that system is evaluated end to end. In a controlled frozen-state routing setup that holds state access and termination fixed while varying transition scorers, teacher-forced edge accuracy is nearly identical across scorers—about 50–53%—but exact constrained-route accuracy differs by 36–88 points. Across 1,800 scorer-choice contexts, choosing by teacher-forced edge accuracy matches the closed-loop winner only 62.4% of the time. Mean selection regret is 9.84 points, compared with 13.37 for uniform-random selection; the 95th-percentile regret is 86.56 points, and 40.7% of contexts have tied top local scores. The mismatch persists when rank correlation is high: at $L{=}4$, correlation is 0.874 while selection agreement is 53.6%. Gold-prefix controls show that some failures precede generated-prefix feedback. Matched length-four continuation supervision brings $L{=}4$ accuracy from near zero to 75–82%; STOP-only data do not help, and margin-guided selection does not outperform random selection. On 90 AppWorld tasks and five backbones, a route-only metric selects a model with 0% task success under one official scaffold and 7.8% under a second; the best model reaches 13.3% and 23.3%, respectively. An offline metric that also checks call arguments selects that best model in both settings. Thus local next-step accuracy alone does not reliably select a component for closed-loop use. On the controlled surface, tied top local scores account for 64.2% of selection regret and can be detected without rollout.

1 INTRODUCTION

Suppose a system is assembled from learned components, each chosen by a local score. How often is the component that scores best on its own the one that works best once the system runs? We first met this question in a frozen-state routing setup: a frozen decoder supplies hidden states, a supervised selector reads them, and a learned relation function scores which skill follows which. On held-out triples, two scorers have nearly identical teacher-forced edge accuracy—about 50%—yet their exact closed-loop route accuracy is 97.5% versus 9.8% on Gemma3, with large gaps on Qwen3 and Llama3.1 as well. Teacher forcing hides the difference because it supplies the gold predecessor; once the model's own prediction becomes the next input, the two systems stop behaving alike. A single route score cannot tell whether the failure comes from state access, transition composition, or termination.

That transitions can fail when composed is not news; the question is narrower: does the local score still tell you which components to deploy? At the horizon where its rank correlation with closed-loop accuracy is highest, it still disagrees with the closed-loop winner in 46.4% of decisions; on the AppWorld agent benchmark, its regret is similar to uniform-random selection. We force a gold prefix and inspect the first non-gold decision: a negative correct-target margin there localizes a transition error before a generated action can feed back. For a route $A \to B \to C$, pairwise training can score both edges well in isolation while C is poorly scored after the B context that composition produced. This localizes an error without identifying its cause.

We ask whether local evaluation can misrank the closed-loop reliability of transition scorers. State access is held fixed, prefix-conditioned and generated-prefix decisions are measured separately, and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

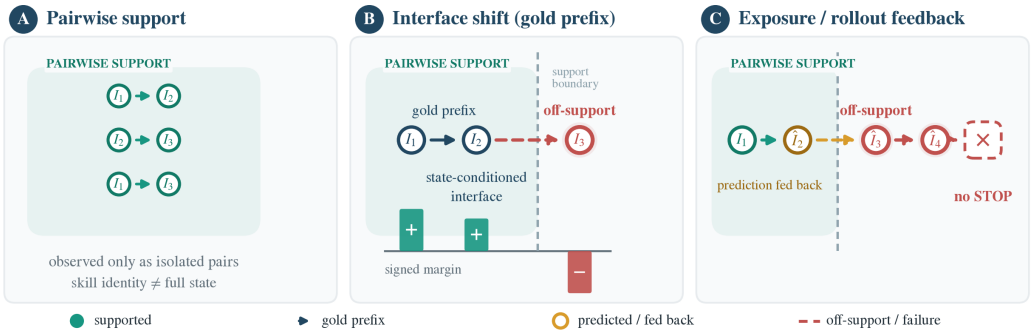


Figure 1: **Why local accuracy can miss a conditional transition error.** (A) Pairwise adaptation trains isolated transitions. (B) A gold prefix can still expose a conditional transition with negative correct-target margin. (C) Generated-prefix feedback can then amplify an earlier error. The example route is `rotate_left` → `mask_vowels` → `duplicate`.

matched continuation supervision is compared against a STOP-only control. The decoder stays frozen: a supervised soft selector reads its hidden sequence and a relation function maps adjacent states to transitions. Four parameter-matched scorer families share the same features, data, optimization budget, and evaluation; adaptation uses atoms and directed pairs only, and the main test is exact closed-loop execution of held-out triples. Figure 1 summarizes the setup.

Our contributions are:

- a controlled finding that teacher-forced edge accuracy can select the wrong transition scorer for closed-loop use: across 1,800 scorer-choice contexts it agrees with the closed-loop winner only 62.4% of the time and incurs mean selection regret of 9.84 points versus 13.37 for uniform-random selection, even when the local metric has rank correlation 0.874 with route accuracy;
- a real-agent replication across two officially released scaffolds: on the full 90-task AppWorld pool, selecting among five backbones by route-only accuracy has regret similar to uniform-random selection under both, while offline scoring of call arguments as well selects the closed-loop winner;
- a locus for the failure: on the controlled surface, contexts whose top local scores tie carry 64.2% of the selection regret while being 40.7% of contexts, so two thirds of the damage is identifiable without any rollout; and a factorized protocol that separates state access, transition selection, and termination using gold-prefix margins;
- a horizon intervention that restores the measured composition horizon with matched continuation supervision, separating the value of sequence coverage from the unresolved choice among continuation examples.

The experiments expose the metric gap, localize the conditional error, test horizon-matched supervision, and then check the protocol elsewhere. The external checks are deliberately narrow: SCAN gives natural-language action programs, ToolBench keeps exact API identities in one repeated family, and AppWorld supplies both frozen-prefix API-call contexts and a real 90-task agent evaluation of the same selection question.

2 Related Work

Our framing connects three usually separate literatures: systematic compositional generalization, train–test mismatch under autoregressive rollout, and distribution shift at the interfaces a composed system consumes. Scheduled sampling and dataset aggregation address the first by exposing a learner to its induced states (Bengio et al., 2015; Ross et al., 2011), and self-recovery work shows the resulting errors need not accumulate monotonically (He et al., 2021). Distributional compositional generalization and compositional-interface analyses likewise emphasize that unseen combinations

depend on both support and the representation exchanged between modules (Fu & Zheng, 2025; Luketina et al., 2025), and the geometry of that representation has been used to predict when compositional generalization succeeds (An & Du, 2026). We use these ideas to motivate an interface-level diagnostic rather than a new scorer architecture.

Composition and horizons. Compositional generalization has been a target since probabilistic program induction (Lake et al., 2015), and SCAN, COGS, CFQ, and grounded extensions now provide complementary tests of novel combinations and structured compositions (Lake & Baroni, 2018; Kim & Linzen, 2020; Keysers et al., 2020; Ruis et al., 2020); simple architectural and decoding choices already move these numbers a great deal (Csordás et al., 2021), which is why we hold them fixed and vary only the interface. ReCOGS and related work study how incidental surface detail in logical forms inflates apparent errors, and how grounding affects generalization (Wu et al., 2023; Wold et al., 2024; 2025). Longer chains also expose depth-dependent failures (Dziri et al., 2023): formal verification above 58% on single functions collapses on two-to-five-function chains (Xu et al., 2025), and individually solvable reasoning steps lose roughly thirty points when combined (Alazraki et al., 2025). Length- and horizon-limited behavior is also observed in deterministic reasoning (Ebrahimi et al., 2026; Guo et al., 2026). Our protocol exposes the intermediate frozen-state route and evaluates exact closed-loop transitions.

Exposure bias and prefix-conditioned diagnostics. Exposure-bias analyses attribute rollout failure to feeding a model its own erroneous prefixes (Bengio et al., 2015; Ross et al., 2011; He et al., 2021); teacher-forced evaluation can also inflate success by leaking a gold prefix (Yang et al., 2025). Our gold-prefix control asks whether a local transition error is already present before generated-prefix feedback, without treating that error as a unique causal mechanism.

Agent loops. Interactive agents interleave reasoning with tool calls (Yao et al., 2023b), and search over intermediate states has been proposed to recover from local mistakes (Yao et al., 2023a). Our question is upstream of both: whether the local scores such loops consume can rank the components the loop is built from.

Routing, modularity, and frozen adaptation. Neural module networks, adapter systems, neural-symbolic recursors, and mixture-of-experts models study reusable components and routing (Andreas et al., 2016; Jiang & Bansal, 2020; Gai et al., 2021; Ponti et al., 2023; Caccia et al., 2023; Li et al., 2024; Shazeer et al., 2017; Fedus et al., 2022). Recent work shows that tool-catalog scale, pre-inference routing, and intermediate-output uncertainty affect deployed route reliability (Gillespie & Perry, 2026; Singla et al., 2026; Liu et al., 2026). Our setting studies the same failure class one level down: a frozen decoder does not change (Hu et al., 2022), and parameter-matched scorers are evaluated on rollout-conditioned interfaces absent from adaptation (Liu et al., 2024; Guo et al., 2017).

3 FACTORIZED FROZEN-STATE ROUTING

Let a frozen decoder produce $H = (h_1, \ldots, h_T)$ for an ordered skill program $z = (z_1, \ldots, z_L)$, with $L \in \{1, 2, 3\}$. The skill inventory contains five deterministic transformations. Router adaptation observes all five atoms and twenty directed pairs; ten directed triples are held out.

Protocol boundary label. Let $\mathcal{D}_{\text{pair}}$ be the pairwise adaptation set. At step t, let P_t denote the selector-produced slot probabilities and σ_t the current source node. The consumed interface is $\mathcal{I}_t = (u_t, z_{t+1})$, where $u_t = (P_t, \sigma_t)$ and z_{t+1} is the gold next skill. The notation $r_\theta(u_t, y)$ denotes the score obtained after constructing the implementation's candidate-specific relational features for target y. We label an interface *held-out* when its declared route context is absent from $\mathcal{D}_{\text{pair}}$, even though its individual directed transitions may each occur there. The label comes from a deterministic split rule and carries no claim about population support, latent mechanism, or causality. The gold-prefix margin below measures the associated conditional transition error.

The skill identity z_t is not assumed to be sufficient to characterize the continuation state: the next decision may depend on the rollout-conditioned representation u_t induced by the preceding composition.

Factorization. We write exact-route success as the conjunction of three events,

$$\mathcal{E}_{\text{route}} = \mathcal{E}_{\text{access}} \cap \mathcal{E}_{\text{transition}} \cap \mathcal{E}_{\text{stop}}. \tag{1}$$

In the notation above, $\mathcal{E}_{\text{access}}$ holds when the ordered selector returns the gold slot identities, $\mathcal{E}_{\text{transition}}$ holds when r_θ ranks every gold successor above its admissible competitors, and $\mathcal{E}_{\text{stop}}$ holds when the presence controller emits the gold program length. The equality in Equation 1 is operational: each event has a dedicated intervention and metric, which keeps accurate length prediction from being credited to the transition model and oracle state access from hiding inside the router.

Selection regret. Let $\mathcal{C}$ be a finite set of candidate components compared in one decision context, let $\ell(c)$ be the local teacher-forced score used for model selection, and let $u(c)$ be exact closed-loop route success. The local winner and the closed-loop oracle are

$$\hat{c}_\ell = \arg\max_{c \in \mathcal{C}} \ell(c), \qquad c^\star = \arg\max_{c \in \mathcal{C}} u(c), \tag{2}$$

with ties broken by a fixed deterministic rule. We define selection regret as

$$R(\mathcal{C}; \ell, u) = u(c^\star) - u(\hat{c}_\ell). \tag{3}$$

Equation 3 is the quantity we audit; selection agreement is the indicator $1[R = 0]$. Regret is the decision-theoretic quantity that matters when a local metric is used to choose a component for closed-loop deployment.

Rank correlation does not bound selection regret. Spearman correlation is invariant to strictly increasing transformations of either score and measures global rank association. Regret instead depends on the utility gap between the top two candidates and on tie-breaking near the top. The two quantities carry different information. For any $\epsilon > 0$ there is a candidate set with Spearman correlation at least $1 - \epsilon$ and positive regret arbitrarily close to the utility range: order n candidates locally as $a_1 \succ a_2 \succ \cdots \succ a_n$ and assign closed-loop utilities so that a_2 is best, a_1 is second, and the remaining order is unchanged. The local metric selects a_1, the oracle selects a_2, and the regret is $u(a_2) - u(a_1)$; adding enough lower-ranked candidates drives the rank correlation arbitrarily close to one while leaving this top-two reversal intact. Thus a high local correlation, by itself, cannot certify that a local metric is safe for selection.

A calibrated local score does bound regret. The converse condition is informative. If the local score is a δ-calibrated proxy, $|\ell(c) - u(c)| \le \delta$ for every candidate, then

$$R(\mathcal{C}; \ell, u) \le 2\delta. \tag{4}$$

Equation 4 states a sufficient condition for safe selection, although the deviation δ requires closed-loop measurements and is not estimated by the local score alone. To see this, let c_ℓ be the local winner and $c^\star$ the closed-loop oracle. Since $\ell(c_\ell) \ge \ell(c^\star)$, $u(c_\ell) \ge \ell(c_\ell) - \delta \ge \ell(c^\star) - \delta \ge u(c^\star) - 2\delta$. The observed regret shows that any uniform calibration bound for these scores must be at least half the observed regret. Estimating such a bound for a new selector would require additional closed-loop evidence; correlation alone cannot certify it.

Ordered state selection. For slot j, a learned query q_j attends to every non-padding position:

$$a_{j,i} = \text{softmax}_i(q_j^\top W h_i / \tau), \qquad s_j = \sum_{i=1}^{T} a_{j,i} h_i. \tag{5}$$

The selector predicts a skill label and binary presence variable p_j; both read out from the pooled slot state, with P_t a distribution over the skill vocabulary and p_j a scalar presence gate. It receives program labels during selector fitting, but no token IDs, offset mappings, or program spans; the decoder remains frozen. Thus it is a controlled, supervised state-access component rather than an unsupervised retriever. Replacing it with unsupervised retrieval or attention would introduce an additional access failure mode; our transition-support diagnostic applies conditionally to whatever states that selector supplies.

Transition function. A relation function scores candidate target y from the interface features, $r_\theta(u_t, y)$. We use four parameter-matched scorer families as diagnostic instruments, keeping features, data, optimization, and evaluation fixed. Their implementation details and parameter counts are deferred to Appendix A; the claim concerns unsupported rollout interfaces, whatever the scorer architecture.

Operational criterion. Let $\mathcal{Y}_t$ be the admissible skill-only candidate set after the implementation's START, STOP, and self-transition masks. We measure the definition above with the correct-target margin of Equation 6

$$m_t = r_\theta(u_t, z_{t+1}) - \max_{y \in \mathcal{Y}_t \setminus \{z_{t+1}\}} r_\theta(u_t, y). \tag{6}$$

For gold-prefix evaluation the interface u_t is built from START at the first position and the gold previous skill thereafter; rollout evaluation instead uses the previous predicted skill. A negative gold-prefix margin therefore identifies a local failure before rollout feedback.

Termination regimes. Constrained routing uses only the selector's predicted presence prefix to determine the number of transitions. Learned-stop adds the factorized presence controller to the learned transitions. Raw greedy routing must infer the next edge and STOP directly. All-length global routing fixes nothing: it scores every candidate program length against the same transition scores and takes the length with the highest total, so transition and length errors enter the same comparison. These regimes separate transition errors from length-control errors.

4 Experimental protocol

We use frozen Qwen3-8B, Gemma-3-4B-it, and Llama-3.1-8B decoders, reading the complete non-padding hidden sequence of layer -8 (the eighth layer counting back from the final output) in bfloat16. The independent task uses `rotate_left`, `mask_vowels`, `duplicate`, `sort_chars`, and `trim_edges`, and all strings are disjoint across decoder fitting, selector fitting, adaptation, and evaluation. The main replication uses input seeds 26031/29047, optimization seeds 60–69, and phase orders 4301/7701, which fix the order in which the two adaptation phases are applied. Scorer families are labeled A–D here to keep the narrative architecture-neutral; their implementations are listed only in Appendix A.

We report selector length, teacher-forced edge, constrained full-route, proper-stop, raw greedy, and all-length accuracy. Teacher forcing is diagnostic only and full-route accuracy is strictly closed-loop. The independent-task teacher metric includes STOP, and the long-horizon skill-only metric supplies the approximately 66.7% values in Appendix Table 15; the two use different splits, seeds, and candidate semantics, so they are not comparable. Optimization seeds are replicates while input seeds and phase orders are paired factors, so pooled summaries are exploratory. In constrained routing only the selector-predicted length is controlled: the scorer still selects every skill and never receives the gold route.

5 Results

Four questions run through the locked protocol: does the gap exist, does it survive longer horizons, does a gold prefix localize it, and does matched supervision repair it? The answers are yes, yes with the ordering itself moving, partly, and yes at the horizon it covers.

5.1 Triples expose a gap hidden by teacher forcing

Table 1 shows the gap that motivated the paper. Teacher-forced all-edge accuracy stays in the 50–53% range for both relation models, while constrained triple accuracy separates them by 87.7, 62.9, and 36.4 points on Gemma 3, Qwen3, and Llama 3.1. The separately locked long-horizon protocol reports a skill-only teacher metric near 66.7%; because its split and seeds differ, we report it separately in Appendix Table 15. Local edge metrics can therefore conceal route failure, and the scorer ordering itself varies by backbone and horizon.

Table 1: **Closed-loop accuracy separates diagnostic scorers that local edge metrics do not predict.** Independent held-out-triple replication (percent); means aggregate ten paired optimization seeds and two input-seed runs. "Teacher edge" is the artifact's all-valid-edge metric and includes the terminal STOP edge; the long-horizon skill-only metric is reported separately in Appendix Table 15. Scorer A/B are deliberately anonymized here; implementations and architecture sensitivity are in the appendix.

Backbone	Metric	Scorer A	Scorer B	Δ
Gemma 3	length / STOP	100.0 / 100.0	100.0 / 100.0	0.0
	teacher edge (all)	53.1	51.3	-1.9
	constrained route	9.8	**97.5**	**+87.7**
Qwen3	length / STOP	100.0 / 100.0	100.0 / 100.0	0.0
	teacher edge (all)	52.5	50.0	-2.5
	constrained route	9.7	**72.6**	**+62.9**
Llama 3.1	length / STOP	98.8 / 99.3	98.8 / 99.3	0.0
	teacher edge (all)	52.6	50.8	-1.8
	constrained route	9.3	**45.7**	**+36.4**

Table 2: **Teacher-forced selection does not reliably choose the closed-loop winner.** For each choice dimension, candidates are ranked by teacher-forced edge accuracy; regret is the closed-loop route-accuracy loss relative to the best candidate in that context. Random is the expected regret of uniform candidate selection. Cluster intervals resample input–optimization clusters (15 clusters on the main surface), 5,000 resamples, seed 48173.

Surface	Choice	Contexts	Agreement	Local regret	Random regret	Cluster 95% CI	Tie rate
Main	scorer	1,800	62.4	9.84	13.37	8.74–10.82	40.7
Main	backbone	2,400	76.0	2.07	4.76	1.68–2.49	36.6
Main	condition	1,440	82.1	0.85	17.72	0.66–1.03	48.1
Agent-like	scorer	360	90.3	3.24	15.08	2.30–4.18	7.5
Agent-like	backbone	240	82.1	2.24	9.31	0.73–3.13	46.2
Agent-like	condition	144	88.9	1.05	10.90	0.16–1.70	52.8

Paired exact two-sided sign-flip tests on the constrained-route differences yield $p = 0.00195, 0.00781$, and 0.00586 for Gemma, Qwen, and Llama, respectively; Holm correction across the three pre-specified backbone contrasts remains below 0.012.

Selection audit across all locked route units. To test whether the local metric would support model selection rather than merely correlate with an aggregate table, we audit the full continuation matrix: three backbones, three input seeds, five optimization seeds, two phase orders, and four scorer kinds. The main surface contains 1,800 route units, which yield 7,200 unit-length records; grouping the four scorer candidates within each horizon yields 1,800 scorer-choice contexts. Within each context, candidates are ranked by teacher-forced edge accuracy with ties broken by a stable identifier, not by closed-loop success. The teacher-forced winner agrees with the closed-loop winner in only 62.4% of contexts (cluster 95% CI $[59.8, 64.8]$); mean closed-loop regret is 9.84 points (cluster 95% CI $[8.74, 10.82]$), with a 95th-percentile regret of 86.56 points. Uniform-random selection has mean regret 13.37 points (Figure 2), so the local metric is useful on average but still far from selection-safe. Appendix Table 24 gives the same quantities by route length. Top local scores are tied in 40.7% of contexts; switching to random tie-breaking gives mean regret 10.21 and leaves agreement at 62.4%, so the qualitative result is not an artifact of the tie rule. The ties are also where the metric fails, which we did not anticipate: they carry mean regret 15.55 points against 5.93 (95% CI on the difference $[6.82, 12.32]$), so the 40.7% of contexts with indistinguishable top local scores account for 64.2% of the total regret (Appendix M). Table 2 reports the other choice dimensions, and Appendix Table 24 the same ordering by route length.

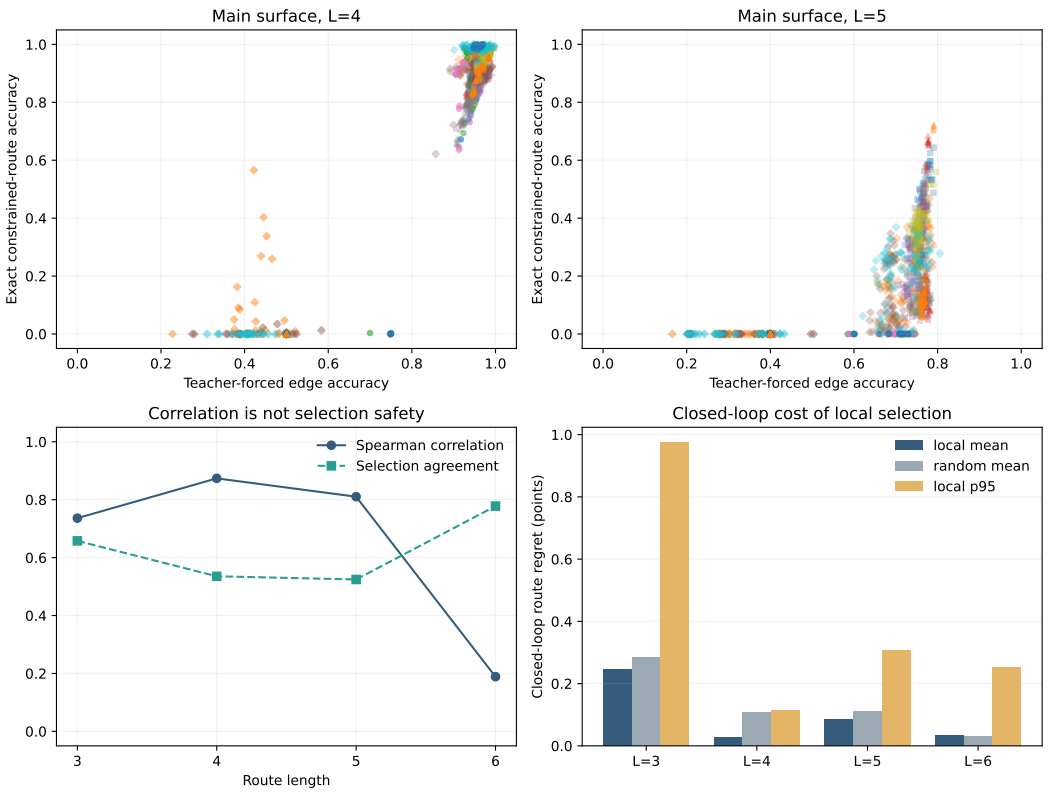


Figure 2: **Teacher-forced edge accuracy is a weak closed-loop selector.** Top: main-surface teacher-forced edge accuracy against exact constrained-route accuracy at $L{=}4$ and $L{=}5$. Bottom left: Spearman correlation and selection agreement by route length. Bottom right: local mean, uniform-random mean, and local 95th-percentile closed-loop regret of the teacher-forced scorer choice.

Independent SCAN confirmation. We next test whether the selection mismatch survives in natural-language action programs on data not used to choose a scorer. In the frozen transductive SCAN protocol, all four scorer families have near-floor autonomous exact match, so full-route normalized edit distance (NED) retains the needed resolution. We freeze the within-backbone teacher-edge winner on development optimization seeds 90–99, then evaluate that fixed choice on untouched seeds 100–109. The local metric selects a non-minimum-NED scorer in all 30 confirmation clusters. Its mean autonomous NED selection regret is 100.7 points (bootstrap 95% CI $[97.4, 104.1]$). NED here divides the edit distance by the length of the reference program, so it is unbounded above when a produced program is longer than its reference; the regret is reported on the same scale. The local-metric and NED-optimal winners never coincide, and all three backbone means are positive. This independent confirmation uses the unchanged SCAN protocol and establishes that the selection mismatch is not limited to the synthetic skill inventory.

The locked long-horizon study reverses the short-horizon ordering for some backbones, and the 20-seed replication likewise shows a backbone-dependent gap. Local edge accuracy remains similar while closed-loop behavior changes with scorer family and horizon. Held-out-program and row-level breakdowns are reported in Appendix A.

5.2 Longer horizons expose the limits of local accuracy

We extend held-out programs to lengths three through six in a separately locked protocol (Appendix A). Appendix Table 15 qualifies the short-horizon comparison in Table 1: the leading scorer can change with backbone and horizon, and every scorer reaches zero route accuracy by length five in this protocol.

The position diagnostics identify a boundary signal: the first unsupported position has negative gold-prefix margins for Scorers A–C in most backbone/length cells, while Scorer D is mixed (Appendix Figure 3). Because teacher and rollout margins share the same skill-only candidate set, this signal precedes any generated error, though it does not name the competing action, and some trajectories still recover later.

Beam search does not alter the measured failure. As a runtime control (Appendix L), we replace greedy decoding with beam search of widths 3, 5, and 10, while keeping the selector-predicted length and scorer unchanged. The beam results are exactly identical to greedy decoding for every tested backbone, input seed, optimization seed, scorer, and horizon; aggregated Scorer D accuracy remains $2.37 \pm 7.85\%$ at $L{=}4$. Under these fixed scores, wider search does not improve the measured routes. The control holds the learned scores fixed, which bounds the decoding explanation without explaining the scores themselves.

Baseline sensitivity. A parameter-matched scorer scan reaches 98.8–100.0% at $L{=}3$ (Appendix Table 3), so the short-horizon baseline is not a tuning artifact.

5.3 Gold-prefix controls localize some errors before feedback

The margin result predicts that correcting an earlier prefix need not restore the next conditional transition. We therefore force the previous skill to its gold value for k prefix actions during held-out decoding, then return to the model's own predictions, leaving the router, selector, hidden states, and training stream untouched. Across three backbones, three optimization seeds, two phase orders, and four scorers, forcing three gold prefix actions leaves full-route $L{=}4$ accuracy at the generated baseline (Appendix Table 16); the suffix from the first post-intervention position improves only modestly, to 10.7% on Gemma3 and at most 2.4% elsewhere. Correcting accumulated prefix state is therefore not enough to restore these conditional transitions. That localizes one failure pathway before generated feedback without fixing a unique cause, and it explains why full-route and conditional suffix accuracy have to be read separately.

5.4 Ruling out selector, termination, and uncertainty

Selector length accuracy is 98.8–100.0%, and the selector's expected attention positions are ordered and non-collapsed (Appendix A). Since all compared scorers use the same selector, selector differences cannot explain the between-scorer gap. This is supervised state access rather than unsupervised program discovery.

Termination is likewise separable from transition learning: the factorized presence controller reaches proper-stop accuracy of 100.0%, 100.0%, and 99.3% on Gemma, Qwen, and Llama, while the two length-control regimes (raw greedy and all-length global) both reach 0% on every backbone (Table 19, Appendix). The baseline adaptation protocol contains no positive three-skill continuation, so the positive result is composition under a constrained state length rather than unrestricted length generalization.

Rollout-time confidence detectors do not fire at the measured held-out boundary; the target-aware margin is a retrospective diagnostic rather than a deployable alarm. Appendix J.2 reports the detector audit.

5.5 A full-matrix horizon-matched continuation intervention

We add length-four continuation supervision across Qwen3, Gemma3, and Llama3.1, three input seeds (36031, 39047, 41031), and five optimization seeds (70–74), with two phase orders and four scorer kinds. Appendix Figure 4 shows the complete $L{=}4$ slice of the intervention matrix; no scorer is selected. The confirmatory contrast is matched length-four continuation versus baseline at $L{=}4$. After averaging the repeated scorer and phase-order factors within each backbone/input/optimization-seed cluster ($n = 45$), the gain is 78.2 points (95% bootstrap CI $[74.9, 81.5]$, Monte Carlo sign-flip $p < 5 \times 10^{-6}$). The confirmatory condition adds the first N length-four programs in lexicographic order. Pre-specified controls add STOP-only, random length-four, or margin-guided length-four examples; the last condition takes the N programs with the smallest gold-prefix margins. Random

and margin-guided continuation perform similarly at this budget, so these results do not favor one continuation-selection rule. The intervention supports horizon-matched sequence supervision; longer-horizon cells are exploratory and appear in the appendix.

5.6 Exposure-bias controls: generated prefixes are not enough

Under the same budget, gold continuation, one-pass DAgger, and 0.5 scheduled sampling help at $L{=}4$ but not at $L{=}5$–6 (Appendix Table 17); exposure correction creates no support beyond the covered interface.

External scope checks. Five further surfaces carry frozen-prefix diagnostics: SCAN, an agent-like dependency chain (Appendix F), COGS, API-Bank (Appendix G), and a ToolBench control (Appendix I). As scope checks they stay out of the central conclusion, while the AppWorld result of Appendix H enters it.

End-to-end agent selection. The mismatch also appears on a real agent benchmark. Five backbones (Gemma-3-4B, Qwen3-8B, Llama-3.1-8B, Mistral-7B, Qwen3-30B) run the full 90-task pre-registered AppWorld pool under the official ReAct scaffold and are scored by the official evaluator. In the listed order, closed-loop success is 0%, 10.0%, 5.6%, 0%, and 13.3%. Gold-prefix next-API accuracy ranks Gemma first and Qwen3-30B second: the route-only metric selects Gemma on 43 of 90 task-level decisions. Its per-task selection regret, 0.156 (95% CI $[0.089, 0.233]$), is similar to the 0.176 of uniform-random selection. Requiring the call arguments as well as the route identifies the closed-loop winner on these labeled tasks. A second official scaffold preserves the ordering; Gemma then succeeds on 7.8% of tasks and Qwen3-30B on 23.3% (Appendix H).

6 Discussion and limitations

The AppWorld comparison shows the value of checking the complete call, including its arguments, on labeled validation trajectories. That offline metric selects the closed-loop winner on the studied tasks; we have not tested whether the ordering transfers to unseen tasks. On the controlled surface, tied top local scores carry about two thirds of the damage, but that warning does not survive the move to AppWorld (Appendix M).

The same checks bound what the paper can claim. The controlled surface is deterministic and supervised, so it is not an autonomous-agent result, and the gold-prefix margin uses the target and is an offline diagnostic rather than a deployable alarm. The length-four intervention restores performance at that horizon but does not show that we can choose continuation examples better than random.

Two pathways produce these failures. On the controlled surface the local score is well defined but can stop being a proxy once the state it conditions on is produced by the system rather than supplied as gold; matched continuation supervision addresses that composition problem. On AppWorld the metric is incomplete: it scores the route and ignores arguments, so a component that fails to emit the gold full call can still rank first; scoring the whole call changes the selection in the tested pool. The two evaluations probe different interfaces.

Search over intermediate states, including Tree of Thoughts (Yao et al., 2023a), may also depend on local scores. Our fixed-score beam control shows no gain from wider search on the controlled routes, but we have not evaluated Tree of Thoughts or the more unstructured WebArena benchmark (Zhou et al., 2024).

7 Conclusion

When local next-step accuracy is used to choose a component, that choice can be unreliable once the system runs. Prefix-conditioned and closed-loop measurements reveal failures the local score hides, matched continuation supervision recovers the horizon it covers, and on a real agent benchmark an offline metric that scores the whole call selects the most successful component in the tested pool. Component selection for a loop should be checked against closed-loop outcomes.

Reproducibility statement

The released archive contains the per-unit audit payloads extracted from the frozen decoders, the result artifacts, split definitions, training and aggregation code, optimization seeds, phase orders, locked protocol documents, an errata ledger, per-file provenance hashes, and an archive manifest. Appendix A specifies the evaluation semantics; the errata ledger records superseded analyses, and the paper cites none of them as evidence.

AI use statement

Generative AI tools were used for language editing and LaTeX presentation assistance only; they were not used to generate experimental measurements or to replace the author's analysis. The author verified all claims and compiled artifacts and takes responsibility for the content.

References

Lisa Alazraki, Lihu Chen, Ana Brassard, Joe Stacey, Hossein A. Rahmani, and Marek Rei. AgentCoMa: A compositional benchmark mixing commonsense and mathematical reasoning in real-world scenarios. *arXiv preprint arXiv:2508.19988*, 2025. URL https://arxiv.org/abs/2508.19988.

Zhiyu An and Wan Du. Representational homomorphism predicts and improves compositional generalization in transformer language models. *arXiv preprint arXiv:2601.18858*, 2026. URL https://arxiv.org/abs/2601.18858.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 39–48, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/Andreas_Neural_Module_Networks_CVPR_2016_paper.html.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/e995f98d56967d946471af29d7bf99f1-Abstract.html.

Lucas Caccia, Edoardo Maria Ponti, Zhan Su, Matheus Pereira, Nicolas Le Roux, and Alessandro Sordoni. Multi-head adapter routing for cross-task generalization. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2211.03831.

Róbert Csordás, Kazuki Irie, and Jürgen Schmidhuber. The devil is in the detail: Simple tricks improve systematic generalization of transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 619–634, 2021. URL https://aclanthology.org/2021.emnlp-main.49/.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena Hwang, Soumya Sanyal, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.neurips.cc/paper_files/paper/2023/hash/deb3c28192f979302c157cb653c15e90-Abstract-Conference.html.

MohammadReza Ebrahimi, Michaël Defferrard, Sunny Panchal, and Roland Memisevic. On the "induction bias" in sequence models. *arXiv preprint arXiv:2602.18333*, 2026. URL https://arxiv.org/abs/2602.18333.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. URL https://jmlr.org/papers/v23/21-0998.html.

Jingwen Fu and Nanning Zheng. On the statistical mechanisms of distributional compositional generalization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume

267, pp. 17909–17927, 2025. URL https://proceedings.mlr.press/v267/fu25j.html.

Yu Gai, Paras Jain, Wendi Zhang, Joseph Gonzalez, Dawn Song, and Ion Stoica. Grounded graph decoding improves compositional generalization in question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 1829–1838, 2021. URL https://aclanthology.org/2021.findings-emnlp.157/.

Kellen Gillespie and Robyn Perry. Scaling enterprise agent routing: Degradation, diagnosis, and recovery. *arXiv preprint arXiv:2606.17519*, 2026. URL https://arxiv.org/abs/2606.17519.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Dongxin Guo, Jikun Wu, and Siu Ming Yiu. The deterministic horizon: When extended reasoning fails and tool delegation becomes necessary. *arXiv preprint arXiv:2606.00376*, 2026. URL https://arxiv.org/abs/2606.00376.

Tianxing He, Jingzhou Zhang, Zhaopeng Zhou, and James Glass. Exposure bias versus self-recovery: Are distortions really incremental for autoregressive text generation? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5683–5695, 2021. URL https://aclanthology.org/2021.emnlp-main.415/.

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://arxiv.org/abs/2106.09685.

Yichen Jiang and Mohit Bansal. Neural module networks for reasoning over text. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1912.04971.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1912.09713.

Najoung Kim and Tal Linzen. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 9087–9105, 2020. URL https://aclanthology.org/2020.emnlp-main.731/.

Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pp. 2873–2882, 2018. URL https://proceedings.mlr.press/v80/lake18a.html.

Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. doi: 10.1126/science.aab3050. URL https://doi.org/10.1126/science.aab3050.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. Api-bank: A comprehensive benchmark for tool-augmented llms. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://arxiv.org/abs/2304.08244.

Qing Li, Yixin Zhu, Yitao Liang, Ying Nian Wu, Song-Chun Zhu, and Siyuan Huang. Neural-symbolic recursive machine for systematic generalization. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2210.01603.

Yaokun Liu, Yifan Liu, Daniel Yue Zhang, Ruichen Yao, Zelin Li, and Dong Wang. PropUQ-MAS: Propagation-aware uncertainty quantification for LLM multi-agent systems. *arXiv preprint arXiv:2608.22130*, 2026. URL https://arxiv.org/abs/2608.22130.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, et al. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*, 2024. URL https://arxiv.org/abs/2404.19756.

Jelena Luketina, Jack Lanchantin, Sainbayar Sukhbaatar, and Arthur Szlam. Compositional interfaces for compositional generalization. In *Proceedings of the 3rd Conference on Lifelong Learning Agents*, volume 274, pp. 692–709, 2025. URL https://proceedings.mlr.press/v274/luketina25a.html.

Edoardo Maria Ponti, Alessandro Sordoni, Yoshua Bengio, and Siva Reddy. Combining parameter-efficient modules for task-level generalisation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 687–702, 2023. URL https://aclanthology.org/2023.eacl-main.49/.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.16789.

Stéphane Ross, Geoffrey Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15, pp. 627–635, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Laura Ruis, Alessandro Roncone, and David Wingate. A benchmark for systematic generalization in grounded language understanding. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/e5a90182cc81e12ab5e72d66e0b46fe3-Abstract.html.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1701.06538.

Saurav Singla, Aarav Singla, Advik Gupta, and Parnika Gupta. AgentWeave: Routing before reasoning for efficient function calling in tool-rich language models. *arXiv preprint arXiv:2608.23078*, 2026. URL https://arxiv.org/abs/2608.23078.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. Appworld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 16022–16076, 2024. URL https://aclanthology.org/2024.acl-long.850/.

Sondre Wold, Étienne Simon, Lucas Charpentier, Egor Kostylev, Erik Velldal, and Lilja Øvrelid. Compositional generalization with grounded language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3447–3460, 2024. URL https://aclanthology.org/2024.findings-acl.205/.

Sondre Wold, Lucas Georges Gabriel Charpentier, and Étienne Simon. Systematic generalization in language models scales with information entropy. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 1807–1819, 2025. URL https://aclanthology.org/2025.findings-acl.90/.

Zhengxuan Wu, Christopher D. Manning, and Christopher Potts. Recogs: How incidental details of a logical form overshadow an evaluation of semantic interpretation. *Transactions of the Association for Computational Linguistics*, 11:1719–1733, 2023. URL https://aclanthology.org/2023.tacl-1.96/.

Xu Xu, Xin Li, Xingwei Qu, Jie Fu, and Binhang Yuan. Local success does not compose: Benchmarking large language models for compositional formal verification. *arXiv preprint arXiv:2509.23061*, 2025. URL https://arxiv.org/abs/2509.23061.

Wanli Yang, Fei Sun, Jiajun Tan, Xinyu Ma, Qi Cao, Dawei Yin, Huawei Shen, and Xueqi Cheng. The mirage of model editing: Revisiting evaluation in the wild. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025. URL https://arxiv.org/abs/2502.11177.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, 2023a. URL https://arxiv.org/abs/2305.10601.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023b. URL https://arxiv.org/abs/2210.03629.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.13854.

A PROTOCOL AND REPRODUCIBILITY DETAILS

The route scorer is trained on atoms and directed pairs only. A held-out triple is evaluated in closed loop, feeding the predicted state from one transition into the next. Constrained evaluation uses only the selector-predicted presence prefix to determine the number of transitions; it does not reveal the correct skill identities. Gaussian uncertainty is added to selector logits before softmax and is never used during router training. Pointer-residual conditions preserve the ordered pointers and apply the matched relation function to the residual transition score. All caches, aggregators, protocols, and the archive manifest are included with the code.

Scorer mapping. To keep the main comparison architecture-neutral, all scorer labels in the main text and current summary tables use Scorer A–D: Scorer A is the matched global MLP, Scorer B the local B-spline scorer (the local KAN implementation), Scorer C the RBF scorer, and Scorer D the global Chebyshev scorer. Historical tables that used "MLP" and "KAN" refer to Scorers A and B, respectively; no additional scorer definition is implied.

Task transformations. Input values are lowercase strings of length five to nine over the letters `a`–`z`. The five skills are deterministic string transformations: `rotate_left` moves the first character to the end; `mask_vowels` replaces every vowel with the hash character; `duplicate` repeats the complete value twice; `sort_chars` arranges the characters in alphabetical order; and `trim_edges` removes the first and last character. Program texts are rendered through four natural-language templates, e.g. "Begin with {`value`}. Perform these operations in order: {`operations`}. Return only the final string." All strings are disjoint across decoder fitting, selector fitting, adaptation, and evaluation.

Our larger transductive SCAN study calibrates state access on a disjoint subset of test-regime commands. Because its access differs from the standard inductive protocol, it is reported here for completeness and never as a SOTA claim.

B MLP TUNING AND LAYER SENSITIVITY AUDITS

Table 3: Focused parameter-matched MLP scan on the locked held-out-triple protocol. Each entry is the best constrained route accuracy at $L{=}3$ over the recorded learning-rate, weight-decay, phase-epoch, and ten optimization-seed grid; the MLP width remains eight (857 trainable parameters). This scan addresses short-horizon tuning and is not used to select a scorer for the long-horizon results. Here MLP denotes Scorer A; see the scorer mapping in Appendix A.

Backbone	Best MLP $L{=}3$ route accuracy (%)
Qwen3	100.0
Gemma3	100.0
Llama3.1	98.8

The reported parameter counts are independently checkable from the implementation: the shared relational feature vector has dimension $14 + \binom{14}{2} = 105$, so the width-eight MLP has $(105 + 1) \times 8 + 9 = 857$ parameters, while the eight-basis spline scorer has $105 \times 8 + 1 = 841$.

C LAYER SENSITIVITY AUDIT

The primary protocol reads layer -8. As a small audit, the same Qwen3 input seed and intervention runner were evaluated with caches corresponding to layers 4, 8, and 12, using the baseline pairwise support and three optimization seeds. Table 4 averages the two phase orders, so it speaks to representation sensitivity rather than adding a backbone comparison.

Table 4: Qwen3 layer-offset sensitivity (mean constrained route accuracy, percent). B-spline and Chebyshev denote Scorers B and D, respectively; see the scorer mapping in Appendix A.

Layer	Scorer	$L = 3$	$L = 4$	$L = 5$	$L = 6$
4	B-spline	1.3	0.0	0.0	0.0
8	B-spline	51.1	0.0	0.0	0.0
12	B-spline	0.5	0.0	0.0	0.0
4	Chebyshev	53.5	0.0	0.0	0.0
8	Chebyshev	51.9	0.0	0.0	0.0
12	Chebyshev	50.4	0.0	0.0	0.0

Interface information probe. To test whether the critical interface is represented at all, we train five-fold linear probes on the position-3 interface (the first three selector slots plus the source skill), separately for optimization seeds 70–72. The next-skill probe reaches $82.9 \pm 1.7\%$ (chance 20%), whereas a held-out-boundary membership probe reaches only $59.6 \pm 6.1\%$ (chance 50%). Thus next-action information is present, while the protocol split label is only modestly separable; the trained relation scorer nevertheless has a negative conditional margin there. None of these probes informs scorer selection, and none identifies a mechanism.

Transition-mask control. The held-out route failures might be an artifact of the no-self-transition mask rather than of the conditional transition. Table 5 removes the mask: the first-boundary margin reversal and the zero length-four route accuracy both persist.

Table 5: No-self-transition mask control on Qwen3 (input seed 36031; three optimization seeds and two phase orders). Removing the mask leaves the first held-out-boundary margin reversal and does not restore length-four routes. MLP, B-spline, and Chebyshev denote Scorers A, B, and D, respectively; see Appendix A.

Scorer	Position-3 gold margin	$L = 4$ route	$L = 5$ route
MLP	-3.614	0.0	0.0
B-spline	-3.025	0.0	0.0
RBF	-2.971	0.0	0.0
Chebyshev	-4.640	0.0	0.0

D SCAN SCOPE CHECK

Table 6 reports the complete scorer-level pilot summary. Each entry is Exact/NED/STOP, averaged across held-out lengths 24–30 for one optimization seed; no scorer is selected within a seed in this appendix table. After retaining the baseline action edges in the corrected implementation, positive-STOP support produces small nonzero STOP rates but still only 0–0.1% exact route accuracy.

Table 6: Detailed SCAN intervention results. Each cell reports Exact/NED/STOP (percent), averaged over held-out lengths 24–30. MLP, B-spline, and Chebyshev denote Scorers A, B, and D, respectively; see Appendix A.

Intervention	Scorer	Seed 90	Seed 91	Seed 92
Continuation	MLP	17.8/30.1/23.8	12.4/28.8/21.0	10.5/32.4/18.2
Continuation	B-spline	9.0/21.0/15.5	2.5/38.9/5.4	5.0/44.7/11.3
Continuation	RBF	10.7/41.5/14.5	2.2/72.3/4.4	3.6/78.8/7.2
Continuation	Chebyshev	15.2/9.5/24.3	9.0/13.0/16.7	9.9/12.0/19.1
Positive STOP	MLP	0.0/100.0/0.3	0.0/140.8/0.1	0.0/100.0/0.9
Positive STOP	B-spline	0.1/55.1/5.1	0.0/52.2/7.6	0.1/55.8/6.8
Positive STOP	RBF	0.0/83.4/2.1	0.0/67.5/3.7	0.1/87.2/3.9
Positive STOP	Chebyshev	0.1/30.9/7.0	0.0/38.3/4.5	0.1/32.0/8.0

Table 7: Representation-adaptation pilot (percent). A matched Qwen3 LoRA adapts the final eight decoder blocks on the atomic/pair stream, then reuses the frozen-state selector and relation protocol. Values are means over three optimization seeds, two phase orders, and three representation-adaptation seeds. LoRA adaptation leaves $L = 4$–6 at zero, indicating that the observed bottleneck is not removed by a modest representation update. Kind-level agent-like results are reported below. MLP and Chebyshev denote Scorers A and D, respectively; see Appendix A.

Condition	L=3	L=4	L=5	L=6
LoRA + MLP	0.0	0.0	0.0	0.0
LoRA + Chebyshev	88.5	0.0	0.0	0.0

E CROSS-BENCHMARK SCOPE-CHECK SUMMARY

The COGS extension reproduces the pre-rollout signature but not stable route recovery: teacher-forced and closed-loop position accuracies are nearly identical for MLP, while Chebyshev changes by at most 3.7 points, and the first-boundary margins are negative for both scorers. On the complete train-only SCAN test, action content improves but selector sequence exactness and termination do not. Table 8 collects the two scope checks.

Table 8: Compact scope-check summary. COGS values are held-out position accuracies at positions 1–4; SCAN values use all 3,920 train-only test commands and the fixed Chebyshev scorer (Scorer D; see Appendix A). MLP denotes Scorer A. TF abbreviates teacher-forced and CL closed-loop.

Dataset	Condition	Diagnostic result	Result boundary
COGS	MLP	TF=CL: 27.6/30.2/32.9/31.7	margins -0.72 to -0.41
COGS	Chebyshev	TF/CL gap ≤ 3.7pt	margins -4.25 to -2.28
SCAN	Qwen3 / Gemma3 / Llama3.1	**Cheb. NED 16.49 / 15.98 / 28.18**	route exact $< 0.4\%$; STOP 6.6–16.4%

F AGENT-LIKE SCOPE CHECK

The agent-like surface reuses the same frozen decoders and the same pairing protocol, but changes the stream construction. A *program* is a chain of tool-like operations applied to a structured input record: a record has up to six slots, each slot holds one of six candidate skills, and each operation copies, duplicates, rotates, or reorders slots under a skill-consistency constraint (task variant `agent_like_structured_dependency_pilot`, built by `build_agent_like_cache_v1.py` with `max_slots=6`). The hidden state that feeds the selectors is taken from decoder layer -8 (counting from the final layer), matching the lockout protocol of the main caches. Training and evaluation lengths, seeds, and the in-support-L=4 and STOP-only additions mirror the main protocol: Gemma3/Llama3.1 use five optimization seeds (70–74), Qwen3 uses the earlier three-seed pilot, all crossed with two phase orders and two scorer kinds (MLP and Chebyshev) in the 160-epoch locked setting. Table 9 reports the scorer-kind-level cells used in the scope-check summary.

Table 9: Agent-like dependency-chain surface, scorer-kind-level constrained route accuracy (percent). Gemma3 and Llama3.1 cells average five optimization seeds and two phase orders; Qwen3 cells use the earlier three-seed pilot. MLP and Chebyshev denote Scorers A and D, respectively; see Appendix A.

Backbone	Condition	Scorer	$L = 3$	$L = 4$	$L = 5$	$L = 6$
Qwen3	Baseline	MLP	0.0	0.0	0.0	0.0
		Chebyshev	100.0	0.0	0.0	0.0
Qwen3	In-support L=4	MLP	33.3	33.3	0.0	0.0
		Chebyshev	100.0	99.7	60.2	27.9
Gemma3	Baseline	MLP	0.1	0.0	0.0	0.0
		Chebyshev	86.5	0.2	0.0	0.0
Gemma3	In-support L=4	MLP	56.8	33.4	0.1	0.0
		Chebyshev	95.4	80.0	17.9	0.8
Gemma3	STOP-only	MLP	0.1	0.0	0.0	0.0
		Chebyshev	86.5	0.2	0.0	0.0
Llama3.1	Baseline	MLP	8.2	0.0	0.0	0.0
		Chebyshev	36.0	0.0	0.0	0.0
Llama3.1	In-support L=4	MLP	37.1	19.8	0.0	0.0
		Chebyshev	57.8	37.8	4.8	0.0
Llama3.1	STOP-only	MLP	8.2	0.0	0.0	0.0
		Chebyshev	36.0	0.0	0.0	0.0

Table 10: **Historical pilot detail** (percent). These rows belong to the superseded dose-response/early-replication protocol (three optimization seeds and selected input seeds), not the current 9-cell, 1,800-unit intervention matrix. They are retained only as historical controls; the current aggregate is Appendix Figure 4. The p column reports paired exact sign-flip tests on the condition differences (one-sided, p_{greater}).

		L=4				L=5				
Backbone (input seed)	N	Base	STOP	Rand	Sorted	Margin	Rand	Sorted	Margin	p (L4 / L5)
Qwen3 (36031)	3	0.0	0.0	60.6	58.7	59.9	9.9	8.9	7.8	0.60 / 0.88
Qwen3 (36031)	5	0.0	0.0	70.2	70.1	70.3	15.0	13.2	9.9	0.48 / 0.99
Qwen3 (36031)	10	0.0	0.0	78.9	78.7	78.9	19.7	17.9	18.7	0.51 / 0.74
Qwen3 (36031)	15	0.0	0.0	78.9	78.3	79.3	27.0	25.1	26.5	0.040 / 0.64
Qwen3 (39047)	10	0.0	0.0	79.9	79.8	79.4	21.3	23.7	19.4	0.72 / 0.78
Gemma3 (39047)	10	0.4	0.4	75.9	76.0	76.0	28.5	26.5	28.2	0.31 / 0.73
Llama3.1 (39047)	10	0.1	0.1	72.5	72.5	72.6	17.3	14.2	17.2	0.42 / 0.51
Gemma3 (36031)	10	2.7	2.7	75.7	75.5	75.9	14.8	10.9	15.1	0.11 / 0.39
Llama3.1 (36031)	10	0.1	0.1	73.1	71.7	72.0	13.1	16.9	15.7	0.97 / 0.013
Qwen3 agent-like	10	0.0	0.0	66.5	66.5	66.6	32.0	30.1	21.5	0.13 / 0.89
Gemma3 agent-like	10	0.1	0.1	51.0	50.7	50.4	11.2	10.4	8.6	0.78 / 0.94
Llama3.1 agent-like	10	0.0	0.0	32.2	32.0	31.9	2.1	3.9	0.4	0.78 / 0.56

G API-BANK SCOPE CHECK

To test whether the diagnostic signature survives outside the controlled surfaces, we use a small, reproducible subset of the released API-Bank benchmark (Li et al., 2023). It contains 108 real two-call dialogues for pairwise adaptation, 10 disjoint multi-call dialogues for continuation, and 59 held-out dialogues or official scenario scripts with three to six API calls. We retain the natural user/assistant context but mask API names, arguments, and returned values before extracting the frozen Qwen3 layer--8 state; hence the route label is not copied into the representation. The tool vocabulary has 37 APIs. The check is diagnostic and stops short of an end-to-end agent benchmark.

Using separate hidden states for each call context and retaining only 20 rows whose first two-call interface is already supported, Chebyshev accuracy drops at the first unsupported third call and its boundary margin is negative (Table 11). Ten disjoint real continuation dialogues make that margin less negative and improve three-call suffix exact match, although the small continuation pool does not restore longer routes reliably. The result supports transfer of the pre-rollout diagnostic signal and

gives a small short-route continuation effect, while showing that recovery requires coverage of the actual tool-state interface.

Table 11: API-Bank matched-support tool-use diagnostic (percent except margin). Values average five optimization seeds on 20 held-out real dialogues or scenario scripts whose first two-call interface is supported by pairwise adaptation. Hidden states are extracted separately before each API call; the first call is gold-controlled. MLP and Chebyshev denote Scorers A and D, respectively; see Appendix A.

Condition	Scorer	TF call 2	TF call 3	CL call 3	Margin call 3	3-call suffix
Pairwise	MLP	5.0	5.0	2.0	−302.2	0.0
Pairwise	Chebyshev	27.0	33.0	34.0	−0.42	15.0
+ continuation	MLP	1.0	10.0	7.0	−289.3	0.0
+ continuation	Chebyshev	36.0	41.0	42.0	−0.29	23.3

H APPWORLD SCOPE CHECK

We next test whether the conditional-error signature survives in a larger natural API environment. AppWorld is a programmatically evaluated world of interactive applications and people with hundreds of executable APIs and hundreds of tasks (Trivedi et al., 2024). Our frozen-state extraction uses all available tasks in the released train and development portions of the provided split: 90 train tasks (4,663 API-call contexts) and 57 held-out development tasks (3,192 contexts). The cache contains all 17 application-level interface labels; exact API routes are retained for the route-level control. We train the transition scorer on the first two calls of each train task and evaluate both teacher-forced and closed-loop prefixes. Results use five optimization seeds (70–74), with no development result used for model or hyperparameter selection.

The audited scale check reproduces staged interface recovery. On the fixed 57-task anchor, two-call adaptation gives 0% exact 3/4/5-call prefixes. Adding the third call raises Chebyshev exact 3-call prefixes to $69.1 \pm 2.0\%$; four-call adaptation yields $68.4 \pm 1.2\%$ exact 4-call prefixes and $55.4 \pm 1.0\%$ exact 5-call prefixes, while five-call adaptation yields $66.3 \pm 4.7\%$ and $58.6 \pm 4.2\%$, respectively. The corresponding pooled scorer values are 40.2%, 34.2%, and 29.3%. Full-task exact remains 0% because these development trajectories can be much longer; it is therefore reported as a separate strict metric rather than as the short-route recovery measure. Table 12 reports the dose response. This result demonstrates that coverage can restore exact performance at a short real interface and that later route success requires recursive coverage.

Table 12: **AppWorld short-route dose response.** Percentages are means ± standard deviation over five optimization seeds on the same 57-task development anchor. Each entry is exact closed-loop prefix accuracy for the indicated route length; full-task exact accuracy is reported separately.

Labels	Adaptation	Scorer	3-call exact	4-call exact	5-call exact
App	2 calls	MLP	0.0	0.0	0.0
App	2 calls	Chebyshev	0.0	0.0	0.0
App	3 calls	MLP	11.2 ± 22.3	0.0	0.0
App	3 calls	Chebyshev	69.1 ± 2.0	0.0	0.0
App	4 calls	MLP	0.0	0.0	0.0
App	4 calls	Chebyshev	68.4 ± 1.2	68.4 ± 1.2	55.4 ± 1.0
App	5 calls	MLP	0.0	0.0	0.0
App	5 calls	Chebyshev	66.3 ± 4.7	66.3 ± 4.7	58.6 ± 4.2

This frozen-prefix analysis is separate from the end-to-end agent evaluation below. The App-level analysis coarsens the 457-API environment to 17 application labels. The short-route results show that additional continuation examples can produce nonzero exact closed-loop routes at the corresponding horizon, while later calls remain unresolved; full-task exact is zero on these long trajectories. The result is therefore a bounded stress test that says nothing about general tool-use recovery or full-task success.

Table 13: **Whether local selection is safe depends on how much of the call the metric measures, and the failure replicates across scaffolds.** One 90-task set and five backbones; only the underlying model differs between rows. The two task-success columns are official AppWorld evaluations under two officially released scaffolds (percent); the two metric columns are scaffold-independent gold-prefix accuracies. Metric A ranks Gemma first under both scaffolds and Metric B ranks the closed-loop winner first.

Component	ReAct	Full code	Metric A (route)	Metric B (full call)
Qwen3-30B-A3B	13.3	**23.3**	0.3230	**0.0897**
Qwen3-8B	10.0	17.8	0.2222	0.0585
Llama-3.1-8B-Instruct	5.6	12.2	0.2124	0.0703
Gemma-3-4B-it	0.0	7.8	**0.3554**	0.0000
Mistral-7B-Instruct-v0.3	0.0	4.4	0.1210	0.0000

End-to-end component selection. The frozen-state results above use AppWorld as a labelled interface source. We also use it as an end-to-end agent benchmark, which is the setting the selection claim is actually about. Five backbones share one scaffold: the official `appworld-agents` simplified ReAct code agent with AppWorld's own prompt file, 50 interaction steps (the value AppWorld ships), greedy decoding, and official scoring by `evaluate_task(...).success`. The components differ only in backbone: Gemma-3-4B-it, Qwen3-8B, Llama-3.1-8B-Instruct, Mistral-7B-Instruct-v0.3, and Qwen3-30B-A3B. All five run the full 90-task pre-registered pool (60 core, 15 development, 15 confirmation) in the same order.

We score each component with two local metrics on the same tasks. Metric A is the conventional one: with the gold API-call prefix supplied, choose the next route from the top-12 retrieved public APIs, with no API executed. Metric B uses the same prefix and retrieval but also requires the arguments, so a step counts only when route and arguments both match the gold call; 773 of 1,871 gold calls carry arguments, so the two metrics differ on 41% of steps. Table 13 gives both measurements.

The closed-loop order is Qwen3-30B-A3B, then Qwen3-8B, then Llama-3.1-8B, with Gemma-3-4B and Mistral-7B tied at zero. Metric A ranks Gemma first, so its argmax is the worst component under ReAct and fourth of five under the full-code agent; it spends 43 of 90 selection decisions on Gemma, and its per-task selection regret of 0.156 (95% bootstrap CI $[0.089, 0.233]$) is close to the 0.176 of uniform-random selection. Metric B ranks Qwen3-30B first, selects it on 88 of 90 decisions, and reaches regret 0.100 (CI $[0.044, 0.167]$), below the uniform-random mean of 0.176. Metric B is a strict per-step match against one gold trajectory, so it understates what an agent can do: agents reach the goal through alternative valid calls and recover from mistakes using environment feedback, which is why closed-loop success (13.3% under ReAct) exceeds its exact-step rate (9.0%). We use it as a ranking signal, not as a capability estimate. It recovers the winner rather than the exact ordering: it puts Llama-3.1-8B ahead of Qwen3-8B, whereas both closed-loop scaffolds order them the other way, giving Spearman 0.900 against the closed-loop order. Qwen3-30B succeeds on $12/90$ tasks against Gemma's $0/90$; all 12 discordant tasks favor Qwen3-30B (two-sided exact McNemar $p = 0.00049$). In all, 21 of 90 tasks are solved by at least one component.

The failure modes explain why a route-level metric cannot see the difference. Gemma-3-4B rarely exhausts the interaction budget: it ends most tasks by itself and, after an unrepairable authentication error, declares failure outright rather than continuing. Qwen3-8B and Llama-3.1-8B instead run into the step cap on roughly a third of tasks, and Qwen3-8B was observed failing the final file write of an otherwise correct trajectory after losing an access token. Mistral-7B emits a loop that mass-removes every Spotify song and album on two tasks, which deadlocks the simulator; both are scored as failures, so its denominator stays at 90 and all 90 runs fail. None of these behaviours is visible to a one-step gold-prefix metric.

The mismatch also occurs under a second scaffold. Replacing the ReAct agent with the officially released `simplified_full_code_agent` increases success for every component (30B: $13.3\% \to 23.3\%$; Gemma: $0.0\% \to 7.8\%$) but reproduces the identical component ordering, and Metric A still selects Gemma, still spending the same 43 of 90 decisions on it. Under this scaffold, Metric A's regret is 0.189 (CI $[0.111, 0.278]$) against 0.191 for uniform random, again similar. Against Gemma's $7/90$ successes, Qwen3-30B solves $21/90$ tasks; the discordant counts are 17

versus 3 (two-sided exact McNemar $p = 0.00258$). The two official scaffolds have different success rates but produce the same backbone ordering and route-only selection failure.

This is a demonstration rather than a powered comparison. Five components and 90 tasks resolve the contrast between the closed-loop winner and the route-only metric's choice ($p = 0.00049$ under ReAct), but they do not resolve adjacent backbones such as Qwen3-30B (13.3%) and Qwen3-8B (10.0%). Both local metrics are low in absolute terms, so they are used as ranking signals rather than capability estimates, and Gemma's zero on Metric B partly reflects a refusal to the argument prompt. The supported claim is narrow: on this labeled task pool, the route-only local metric does not select the best closed-loop component, while the full-call metric does. This same-task comparison does not test selection on unseen tasks.

I TOOLBENCH SCOPE CHECK

To separate interface composition from large-vocabulary tool retrieval, we construct a train-internal control from ToolBench (Qin et al., 2024). We retain one repeated tool family (`free_nba`), seven exact APIs, 33 pairwise adaptation trajectories, and 17 disjoint held-out trajectories whose first two-call interface is supported but whose third-call interface is not. The first call is gold-controlled; only subsequent calls enter the metric. The current target API specification is removed from each context, other tool descriptions are retained, and we use the full token-level hidden-state sequence rather than pooled features. The intervention adds eight disjoint continuation trajectories covering six of the 16 missing three-call interfaces.

Across five optimization seeds, Chebyshev gold-first exact 3-call route accuracy increases from $3.1 \pm 6.9\%$ to $24.6 \pm 6.4\%$ after adding continuation trajectories, while the position-3 margin moves from -1.2 ± 0.2 to -0.4 ± 0.2. MLP remains substantially weaker and does not show reliable recovery. Table 14 reports the cells. This small train-internal result is a narrow positive control; it neither validates a selection rule nor establishes broader recovery beyond this repeated-interface subset.

Table 14: Strict ToolBench intra-family control (percent except margin). Values are means $\pm$ standard deviation over five optimization seeds; the first API call is gold-controlled, and exact-route accuracy requires all remaining calls to be correct. MLP and Chebyshev denote Scorers A and D, respectively; see Appendix A. The Scorer D rows are the narrow continuation control cited in the main text.

Condition	Scorer	Pos. 1	Pos. 2	Gold-first exact route	Pos.-3 margin
Pairwise	MLP	23.5±11.0	14.1±3.2	4.6±4.2	-193.1 ± 33.0
Continuation	MLP	16.5±8.7	15.3±10.7	3.1±6.9	-126.0 ± 32.3
Pairwise	Chebyshev	17.6±5.9	17.6±9.3	3.1±6.9	-1.2 ± 0.2
Continuation	Chebyshev	44.7±3.2	31.8±9.8	24.6±6.4	-0.4 ± 0.2

J ADDITIONAL CONTROLLED ANALYSES

Tables 19, 20, and 21 support the decomposition, uncertainty, and sensitivity analyses referenced in the Results section. Backbone order is Gemma 3, Qwen3, Llama 3.1 throughout.

J.1 DEFERRED MAIN-TEXT CONTROLS

This subsection collects the controls the Results section cites but does not print: the horizon table, the second-seed gold-prefix rescue, the exposure-bias controls, the route decomposition, the selector-noise and secondary-replication tables, and the historical pilot detail. Each table's caption states the protocol it belongs to; the historical pilot detail is Table 10.

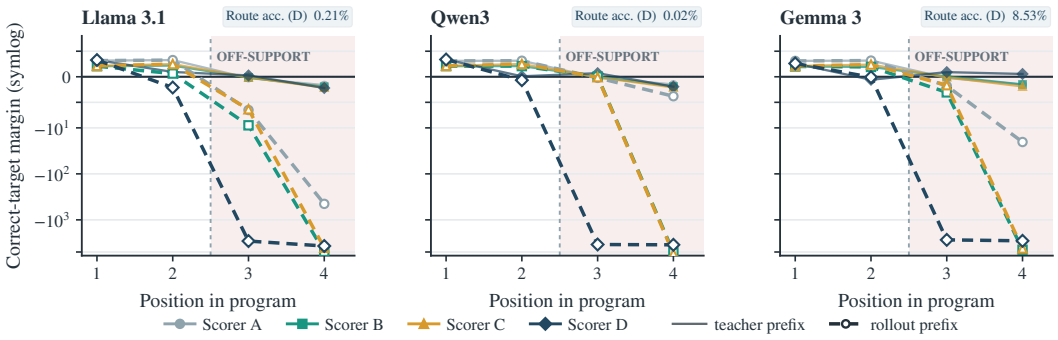


Figure 3: **Correct-target margin on the held-out L=4 cohort.** Solid lines score from the gold prefix (teacher forcing); dashed lines from the model's own predictions (rollout). The first unsupported position is position 3: teacher margins are negative for most scorer/backbone cells, while Scorer D is mixed. Catastrophic negative rollout margins reflect error propagation and the no-self-transition mask. We plot the locked L=4 cohort; the independently sampled L=3 cohort is reported in Appendix Table 15.

Table 15: **Exact constrained route accuracy across horizons** (percent) in the locked long-horizon protocol: bare basis scorers and twenty position-balanced held-out programs per length. Each cell averages input seeds 36031/39047, phase orders 5301/8701, and optimization seeds 70–79.

Backbone	Scorer	$L=3$	$L=4$	$L=5$	$L=6$
Gemma 3	Scorer A	0.06	0	0	0
	Scorer B	15.50	0	0	0
	Scorer C	8.03	0	0	0
	Scorer D	**89.96**	**8.53**	0	0
Qwen3	Scorer A	0	0	0	0
	Scorer B	1.32	0	0	0
	Scorer C	0.20	0	0	0
	Scorer D	**86.53**	0.02	0	0
Llama 3.1	Scorer A	0.13	0	0	0
	Scorer B	4.05	0	0	0
	Scorer C	0.53	0	0	0
	Scorer D	**53.29**	0.21	0	0

Table 16: **Gold-prefix rescue control** on the second input seed (39047). Values are means over 24 route units per backbone (three optimization seeds, two phase orders, four scorers). "Suffix after 3 gold" evaluates only the continuation beginning at the first position after the forced prefix.

Backbone	L=3 generated	L=4 generated	L=4 full after 3 gold	L=4 suffix after 3 gold
Qwen3	24.4	0.0	0.0	0.9
Gemma3	25.9	0.3	0.3	**10.7**
Llama3.1	21.6	0.1	0.1	2.4

Table 17: Exposure-bias controls on Qwen3 (constrained route accuracy, percent; means ± standard deviation over three seeds, two phase orders, and two scorers). All continuation conditions use the same ten length-four programs.

Condition	$L=3$	$L=4$	$L=5$	$L=6$	$p_{L=4}$
Baseline	42.0 ± 50.7	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	–
Gold continuation	99.7 ± 0.5	**82.8 ± 14.0**	0.0 ± 0.0	0.0 ± 0.0	$< .001$
DAgger	83.0 ± 38.8	66.1 ± 34.8	0.0 ± 0.0	0.0 ± 0.0	$< .001$
Scheduled sampling	99.7 ± 0.5	81.7 ± 14.7	0.0 ± 0.0	0.0 ± 0.0	$< .001$

J.2 CONFIDENCE-DETECTOR NEGATIVE CONTROL

The confidence-only diagnostic calibrates four rollout-time detectors (max-softmax probability, entropy, top-two gap, STOP mass) on adaptation-covered positions. Table 18 shows that teacher-forced accuracy is at ceiling on positions 1–2 but collapses at position 3, while calibrated confidence alarms fire on at most 0.1% of boundary rows. These confidence signals therefore do not detect the measured conditional error; the target-aware margin is retrospective rather than a deployable detector.

Table 18: **Confidence-only detectors cannot predict the boundary.** Values summarize both scorers; alarms are calibrated on on-support positions.

Backbone	pos1	pos2	pos3	Boundary failure	Alarm
Qwen3	100.0	100.0	0.0	100.0	0.0
Gemma3	100.0	100.0	0.0	100.0	0.0
Llama3.1	100.0	99.8	0.1	100.0	0.1

Table 19: Route decomposition (percent). Teacher-edge values use the independent-task evaluator and include the terminal STOP edge; the separate long-horizon skill-only metric is reported in Appendix Table 15.

Backbone/model	Teacher edge	Constrained	Route with learned STOP	Raw	Global
Gemma / Scorer A	53.1	9.8	9.8	0.0	0.0
Gemma / Scorer B	51.3	97.5	97.5	0.0	0.0
Qwen / Scorer A	52.5	9.7	9.7	0.0	0.0
Qwen / Scorer B	50.0	72.6	72.6	0.0	0.0
Llama / Scorer A	52.6	9.3	9.3	0.0	0.0
Llama / Scorer B	50.8	45.7	45.7	0.0	0.0

Table 20: Held-out constrained route accuracy under selector-logit noise $\sigma = 2.0$ (percent).

Backbone	Pointer	Learned MLP	Learned KAN	Pointer+MLP	Pointer+KAN
Gemma 3	99.5	59.9	27.9	99.6	99.6
Qwen3	99.9	59.9	1.1	99.9	99.9
Llama 3.1	99.9	60.0	10.8	99.9	99.9

Table 21: Secondary replication results (constrained route, percent). These runs are not pooled as independent task constructions.

Analysis / backbone	MLP	KAN	Δ
20-seed / Gemma	40.00	89.05	+49.05
20-seed / Qwen	40.00	38.91	-1.09
20-seed / Llama	40.00	50.00	+10.00
New split / Gemma	20.00	75.37	+55.37
New split / Qwen	20.00	50.11	+30.11
New split / Llama	20.00	37.50	+17.50

K ADDITIONAL CONTROLS

K.1 SCALE SANITY CHECK

The existing modular-LoRA baseline on Qwen3-30B-A3B was run on four GPUs, four skills, forty optimization steps per skill, and seeds 0–3. Two-skill exact-match composition was $15/25/15/25\%$ (mean 20.0%, sample standard deviation 5.8); the atomic mean was 46.3%. Table 7 reports the frozen-state counterpart. The run is a loading and evaluation sanity check that leaves the full 30B selector/scorer protocol aside.

L BEAM-SEARCH NEGATIVE CONTROL

The diagnostic account suggests a search-based runtime control, but triggering it with the gold-target margin would leak the evaluation label. We therefore test a strictly label-free alternative: width $B{=}1$ is greedy, while $B \in \{3, 5, 10\}$ is always-on beam search using only scorer logits and the selector-predicted length. Gold programs are consulted only after decoding. Across three backbones, two input caches, three optimization seeds, two phase orders, and both the parameter-matched MLP and Chebyshev scorers, every beam width exactly matches greedy (Table 22). In particular, beam search does not restore the $L{=}4$–6 horizon. This is a negative remedy result: at the held-out boundary, the correct continuation is not a competitive candidate under the learned scorer, so search does not change the measured horizon failure. It does not identify why the candidate is low-ranked, and the continuation experiment establishes only a horizon-matched positive control. Search remains a plausible fallback if a future intervention changes the candidate ranking.

Table 22: Label-free beam-search control. Route accuracy is percent, averaged over backbones, two input caches, three optimization seeds, and two phase orders; values are mean ± sample SD. Widths 3, 5, and 10 are always-on and use no gold target or gold margin.

Scorer	Beam width	$L{=}3$	$L{=}4$	$L{=}5$	$L{=}6$
MLP	1 (greedy)	0.21±0.61	0.00±0.00	0.00±0.00	0.00±0.00
MLP	3/5/10	0.21±0.61	0.00±0.00	0.00±0.00	0.00±0.00
Chebyshev	1 (greedy)	81.56±30.51	2.37±7.85	0.00±0.00	0.00±0.00
Chebyshev	3/5/10	81.56±30.51	2.37±7.85	0.00±0.00	0.00±0.00

M WHERE THE SELECTION REGRET CONCENTRATES

We analyze where regret occurs across the same locked 1,800 contexts. The uncertainty analysis resamples whole input–optimization clusters ($5{,}000$ bootstrap resamples); Table 23 reports the tie audit.

Table 23: **Exact local ties carry most of the selection regret.** Locked 1,800 scorer-choice contexts. Tied means at least two candidates share the top teacher-forced edge accuracy, so the local metric provides no basis for choosing among them.

Subset	Contexts	Mean regret (pts)	Share of total regret
Tied top local scores	732	15.55	64.2%
Untied top local scores	1,068	5.93	35.8%

The difference is 9.62 points (cluster 95% CI $[6.82, 12.32]$). On this surface, roughly two thirds of the selection damage is locatable with a check that needs no rollout at all: if the top local scores are equal, the local metric has not made a decision.

The rule does not transfer to AppWorld. On the 90-task pool, top local scores tie on 25 of 90 route-level decisions, and those contexts carry mean regret 0.160 against 0.154 for untied ones; under the full-call metric the ordering reverses (0.088 tied against 0.200 untied, 80 of 90 tied). These AppWorld scores are based on per-task call counts rather than the fixed route-unit set of the controlled surface. Exact ties are therefore useful for auditing the controlled selection rule, but are not a general warning signal.

We also tested the more elaborate version of this idea. The calibration bound of Section 3 bounds regret by twice the local score's deviation from closed-loop utility, but that deviation cannot be measured without running the loop. Each cell is nevertheless measured under ten replicates (five optimization seeds, two phase orders), so the spread of a *local* score across replicates is available without any rollout. Flagging every context whose top-two local gap falls below the combined replicate standard deviation separates nothing useful: the flagged contexts average 0.101 regret against 0.095 for the rest, a difference whose bootstrap interval comfortably covers zero, and the

flagged set is not monotone in the gap. Replicate noise is therefore not a usable calibration proxy here, and the exact-tie rule above is the diagnostic that survives.

N INTERVENTION MATRIX BY CELL

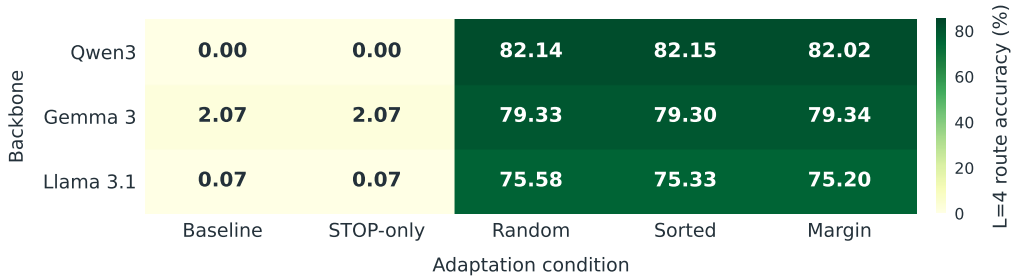


Figure 4: **Full-matrix horizon-matched continuation intervention at $L{=}4$.** Cells show constrained route accuracy (percent), averaged over four scorer kinds, two phase orders, five optimization seeds, and three input seeds; each backbone/input-seed payload contains 200 route units. The confirmatory comparison is length-four continuation versus baseline; the other columns are pre-specified exploratory controls.

O SELECTION-REGRET DETAIL BY ROUTE LENGTH

Table 24: **Correlation is not selection safety.** Main-surface scorer choice by evaluated horizon. Spearman is between teacher-forced edge accuracy and exact constrained-route accuracy; agreement and regret compare the teacher-forced winner with the closed-loop oracle. Random regret is the expected loss of uniform candidate selection; intervals are cluster bootstrap 95% CIs. Regret is in route-accuracy points.

Length	Spearman	Agreement	Local regret	Random regret	Cluster 95% CI	Tie rate
$L{=}3$	0.736	65.8	24.73	28.43	20.64–28.64	82.0
$L{=}4$	**0.874**	53.6	2.75	10.83	1.88–3.72	35.8
$L{=}5$	0.810	52.4	8.54	11.17	7.14–9.90	30.2
$L{=}6$	0.189	77.8	3.36	3.05	2.02–4.83	14.7