RECUR, SWITCH, RECUR, COMMIT: FUNCTION-ADAPTIVE HYBRID REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models can reason through explicit tokens or recurrent latent states, but hybrid systems still lack a principled rule for deciding which representation to use at each step. We show that this choice is governed more by *local reasoning function* than by position: latent recurrence better supports search and planning by retaining alternative candidates, whereas explicit computation better supports calculation and verification by preserving exact values and entities. Controlled interventions reverse these preferences as candidate multiplicity and exact-value demand change, and frozen hidden states predict them before router training. We therefore introduce *Function-Adaptive Hybrid Reasoning*, combining capability-preserving multi-path training, function-grounded switching supervision, and budgeted router-only reinforcement learning for bidirectional, state-conditioned routing. Across eight model configurations and twelve core tasks, the full method improves the twelve-task macro over explicit CoT-RL on every backbone, by 0.8–2.2 percentage points. On Qwen3-8B, it gains 1.9 points while using 0.72 normalized compute and a 0.75 estimated normalized latency. These results suggest that effective hybrid reasoning is less about when to compress thought than about matching each local computation to the representation that best supports it.

1 INTRODUCTION

Explicit chain-of-thought (CoT) helps large language models (LLMs) reason by generating intermediate text, but each additional step incurs autoregressive generation and context-processing costs. Latent reasoning instead computes with continuous hidden states (Hao et al., 2024; Shen et al., 2025), offering a way to reduce text generation without eliminating intermediate computation.

Recent hybrids use confidence, span reliability, or reinforcement learning to switch between representations (Xu et al., 2026; Xie et al., 2026; Zhang et al., 2026). Yet their accuracy–cost trade-offs vary: selective compression, for example, can remain less accurate than full explicit CoT (Xie et al., 2026). We ask a complementary question: *which local reasoning functions benefit from each representation, and why?*

We examine this question using a frozen model that supports both modes. Simple hybrids can outperform either pure mode, but their gains depend on placement. Comparisons across functions and positions reveal a more stable explanation: search and planning favor latent reasoning, whereas calculation and verification favor explicit reasoning. Candidate-path interventions and delayed-reuse analyses link these preferences to retaining alternatives and preserving precise information, respectively. The preferences are predictable from hidden states before routing training.

This evidence motivates *Function-Adaptive Hybrid Reasoning* (Figure 1). Multi-path training strengthens latent exploration while preserving explicit capability. Functional supervision then grounds switching in these measured preferences, and budgeted router-only reinforcement learning refines complete routes with the content model fixed. At inference, decisions depend on the current state and budget, without function annotations; final answers remain explicit.

On Qwen3-8B at the reference budget cap, Ours-Full improves the twelve-task core macro by 1.9 percentage points over explicit CoT-RL, with normalized compute/latency estimates of 0.72/0.75.

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.

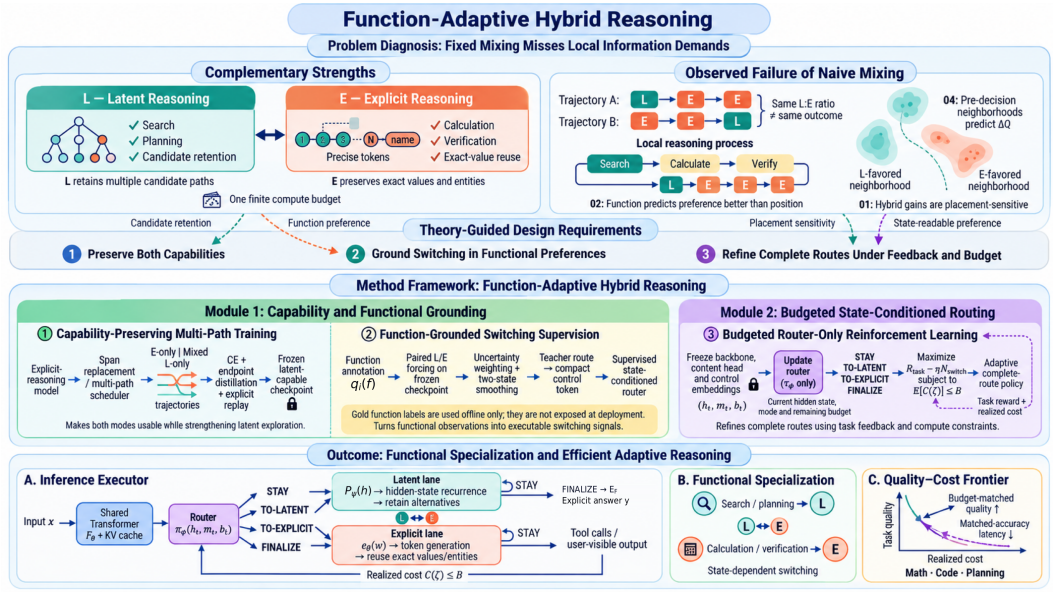


Figure 1: **Function-adaptive hybrid reasoning.** Complementary latent/explicit strengths and state-dependent preferences motivate capability-preserving multi-path training, function-grounded switching supervision, and budgeted router-only RL. During inference, a state-conditioned router selects STAY, TO-LATENT, TO-EXPLICIT, or FINALIZE; switching is bidirectional, outputs remain explicit, and function labels are used only during training.

These results connect local functional preferences to a practical policy for allocating reasoning computation.

2 Related Work

Continuous latent reasoning. Coconut (Hao et al., 2024) performs continuous reasoning through hidden-state recurrence, while CODI (Shen et al., 2025) transfers explicit CoT into latent states through self-distillation. Building on these capabilities, we identify the local reasoning functions favoring each representation and characterize how their preferences are organized in hidden-state space before routing supervision.

Adaptive and hybrid reasoning. System-1.5 (Wang et al., 2025) adaptively reduces computation through shortcuts across model depth and reasoning steps. ThinkRouter (Xu et al., 2026) routes using model confidence, while SLT (Xie et al., 2026) uses span reliability to selectively compress reasoning and preserve precision-critical computation. TARPO (Zhang et al., 2026) jointly optimizes a hidden-state router and the backbone through end-to-end reinforcement learning for token-wise mode selection. Our approach grounds mode selection in independently measured functional preferences, uses these preferences to supervise switching, and then refines the policy with router-focused reinforcement learning. This connects adaptive computation to an explicit account of when each representation is useful.

Multi-path reasoning mechanisms. Reasoning by Superposition (Zhu et al., 2025) provides theoretical and empirical evidence that continuous CoT can represent multiple search frontiers simultaneously in graph reachability. We connect this capability to the complementary strengths of latent and explicit reasoning through interventions on candidate-path information and analyses of precise intermediate-value reuse. The resulting analysis guides both multi-path training data for strengthening latent exploration and functional supervision for deciding where to use it.

3 Problem Setting and Discovery Protocol

Given an input x, a shared autoregressive backbone produces an explicit final answer while its intermediate computation may proceed through explicit token generation (E) or recurrent latent states (L). We evaluate a mixed trajectory under both task quality Q and realized computation cost C, keeping prompts, answer formats, decoding settings, and total budgets aligned across modes. The unit of analysis is a local reasoning segment defined by the operation it performs—for example, search, planning, exact calculation, or constraint verification—rather than by task domain or absolute token position.

To measure representation preference independently of a learned router, we first train a latent-capable model and then freeze its parameters. For each eligible segment, paired rollouts share the same preceding state and evaluation protocol, force either L or E for the target computation, and regenerate all downstream states. We report the paired quality effect $\Delta^Q = Q(L) - Q(E)$ separately from the corresponding cost effect $\Delta^C = C(L) - C(E)$. Function annotation is performed without access to the preferred mode, and observation, supervision, calibration, and final-test partitions are separated by source problem and transformation family.

4 Observations: When and Why to Mix Reasoning Modes

To ground adaptive hybrid reasoning, we investigate when latent and explicit reasoning complement each other and what should govern switching. Simple hybrid schedules first reveal budget-matched gains that depend on latent placement. Function-matched comparisons and representation analyses identify local operations as a more stable predictor of mode preference than position. Path decomposition, selective interventions, and delayed-reuse tests suggest complementary strengths: retaining unresolved candidates in latent states and preserving exact information through explicit reasoning. Manipulating these demands can reverse mode preference. Finally, functional rules and frozen-state probes establish the utility and predictability of these patterns, motivating capability preservation, state-dependent switching, and adaptive stopping.

Following Section 3, we freeze the backbone after latent-capability training and externally select latent recurrence (L) or explicit token generation (E). Free-form segment descriptions are mapped to predefined functions. Paired rollouts share prefixes, continuation policies, and sampling protocols while regenerating downstream states; their quality difference is $\Delta_i^Q = Q_i(L) - Q_i(E)$. General-purpose benchmarks, simulator-based interactive evaluations, and the anonymized real-data retail-business evaluation report quality and cost separately; annotation, splits, budget controls, and statistics are in Appendix D.

4.1 Simple Hybrids Reveal Gains and Placement Sensitivity

We compare pure modes with front-loaded, back-loaded, random, and alternating latent segments across latent proportions. Schedules use neither function labels nor answers; configurations are validation-selected. Front-40% reaches 56.4% Pass@1, exceeding explicit and latent reasoning by 1.6 and 3.5 percentage points, with gains persisting at Pass@8 (Figure 2a–b). Hybrids also solve examples missed by both pure modes under equal sampling budgets, partially replicating across independent seeds (Appendix D.5). This supports finite-budget trajectory complementarity.

Yet front- and back-loaded schedules differ by 2.6 points at the same 40% latent proportion. The placement effect persists under switch-count and compute matching and paired segment swaps. Latent usage alone is insufficient. Since planning often precedes calculation, does this preference reflect position or the operations performed there?

4.2 Local Functions Explain Preferences Reflected in Hidden States

Cross-position and cross-domain matching with hierarchical regression separates these explanations. Controlling for function reduces the positional effect from 2.4 to 0.3 points, with a 95% interval containing zero. Early calculation still favors explicit reasoning; late search favors latent reasoning (Figure 2c). Function explains preference more consistently than position alone.

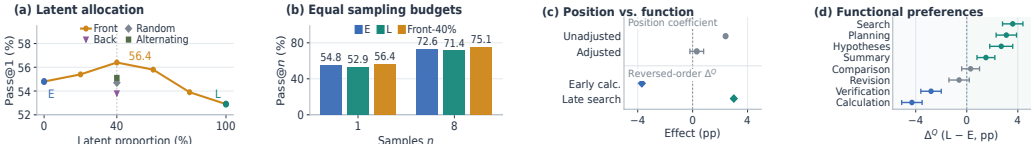


Figure 2: **From hybrid gains to functional preferences.** (a) Front sweep and 40%-ratio alternatives. (b) Pass@1/8. (c) Position coefficients and reversed-order Δ^Q. (d) Functional effects. Protocol: Appendix D.



Figure 3: **Mechanisms and preference reversals.** (a) Candidate retention and a qualitative ablation summary. (b) Recovery and delayed reuse. (c–d) Candidate and exact-value interventions (shared Δ^Q axis). Further interventions: Appendix I.

Retrieval/search, planning/decomposition, and hypothesis enumeration favor latent reasoning; exact calculation and constraint verification favor explicit reasoning; summarization/synthesis, comparison/selection, and self-correction/revision are instance-dependent (Figure 2d). Opposite preferences occur within one problem, while domain, difficulty, and model comparisons broadly preserve the ordering. Functions describe local operations, distinct from topics such as mathematics or coding; figures use shortened labels for readability.

Pre-decision states reflect this structure: 20-nearest-neighbor agreement on the sign of Δ^Q reaches 79.0%, versus 51.0% under label permutation. Frozen-state probes, local autocorrelation, and unsupervised clustering corroborate this finding. Categories precede representation analysis; projections serve visualization, and probe controls assess selectivity (Hewitt & Liang, 2019). Preference is thus predictable, but why do different operations favor different modes?

4.3 Candidate Retention and Precise Information Reuse

Evidence that continuous thoughts retain alternative paths (Hao et al., 2024) motivates a two-demand hypothesis: maintaining unresolved candidates (B) favors latent reasoning; reusing exact discrete values (D_{exact}) favors explicit reasoning. We test the empirical relationship $\Delta^Q \approx \alpha B - \beta D_{\text{exact}}$, $\alpha, \beta > 0$, through representation measurements, interventions, and controlled task variations.

Multi-label probes, path-prototype decomposition, and controlled decoding measure information about verifiable candidate paths. With four candidates, entropy-based effective path count, $\exp(H(p))$ for normalized path-support scores p, is 2.74 in latent states versus 1.31 in explicit states (Figure 3a). Removing a path-aligned component preferentially suppresses that path. Entropy-matched, random-direction, and compute-matched controls support candidate retention and downstream use (Appendix I), with intervention sensitivity checked following Zhang & Nanda (2024).

Conversely, explicit reasoning supports exact reuse: numerical-value reuse after four steps reaches 81.4%, versus 42.8% for latent reasoning; abstract intentions show no comparable gap (Figure 3b). Function-conditioned regressions and within-cluster analyses connect both findings: latent-favored neighborhoods retain more recoverable candidates, whereas explicit-favored neighborhoods better preserve discrete entities. Search and planning emphasize the former demand; calculation and verification emphasize the latter. Functional labels describe these operations, while hidden-state neighborhoods reflect their underlying information requirements.

We independently vary candidate count K and exact-value count D_{count} in controlled synthetic templates. Increasing K raises Δ^Q from +0.1 to +3.5 points; increasing D_{count} shifts it from +2.4 to -3.0 points (Figure 3c–d). Hybrids outperform both pure modes at high K and D_{count}. Exploration gains saturate with latent length, while prolonged latent computation impairs exact calculation. Information requirements thus affect mode preference. Can current states guide its selection?

4.4 From Mode Preferences to Learnable Routing Signals

An offline function rule reaches 56.9% Pass@1, recovering approximately 78% of the gain from a protocol-specific segment-counterfactual reference. A current-state probe achieves held-out-domain preference AUROC of 0.83, versus 0.65 for confidence alone. Functional regularities are useful and routing signals internally accessible, complementing confidence-based routing (Xu et al., 2026).

Stepwise decisions reduce premature and delayed stops relative to one-shot length prediction; boundary-tolerant evaluation and routing regret assess their consequences. These findings motivate capability-preserving supervision and bidirectional, state-dependent switching, with adaptive stopping evaluated against length prediction.

Section 5 develops supervised control grounding and routing-policy optimization from the frozen checkpoint; Section 7 evaluates end-to-end quality–cost trade-offs and the proposed mechanisms.

5 Function-Adaptive Hybrid Reasoning

The observations impose three requirements: placement effects call for state-conditioned switching; complementary information demands require both latent candidate retention and explicit precision; and function–state alignment can initialize routing but must be refined with task feedback under a compute budget. We therefore use *function-adaptive hybrid reasoning*, combining a shared interface, capability-preserving training, function-grounded switching, and budgeted policy refinement.

5.1 Hybrid Computation and State-Conditioned Control

Let x be an input and y its final answer. A single autoregressive Transformer F_θ uses one causal history and KV cache $\mathcal{K}$ for both modes. Given the latest hidden state h, one content step is

$$(h', \mathcal{K}') = F_\theta(v, \mathcal{K}), \qquad v = \begin{cases} e_\theta(w), & w \sim p_\theta(\cdot \mid h), \quad \text{explicit reasoning,} \\ P_\psi(h), & \text{latent reasoning,} \end{cases} \tag{1}$$

where e_θ is the token embedding and P_ψ projects a hidden state into the next continuous input; a latent step is therefore an executed Transformer recurrence rather than a decoded token (Hao et al., 2024; Shen et al., 2025).

We distinguish latent reasoning L, explicit reasoning E_R, and explicit answer generation E_F. Sections 3–4 use E as shorthand for explicit reasoning; from here onward, E_R denotes that reasoning state and E_F is reserved for final-answer generation. At state (h_t, m_t, b_t), a lightweight head selects from legal STAY, TO-LATENT, TO-EXPLICIT, and FINALIZE actions. STAY executes one content step; switching is bidirectional; FINALIZE irreversibly enters E_F; every transition emits a compact control token and at least one destination-mode step, while illegal actions are masked. This turns the placement effect in Section 4.1 into a state-conditioned decision process; state-machine and cache rules are in Appendix F, and serving details in Appendix C.

5.2 Capability-Preserving Three-Stage Training

The three stages follow the evidence chain in Section 4: make both modes usable, translate measured preferences into executable routes, then adapt those routes to complete-task outcomes.

Stage 1: learning both modes. Starting from an explicit-reasoning model, a scheduler replaces selected spans with the latent recurrence in Eq. (1), increasing latent coverage and depth while retaining pure-explicit, mixed, and pure-latent trajectories. Cross-entropy, frozen-teacher output-end distillation, and explicit replay preserve visible reasoning and the original path. We freeze this checkpoint before paired observations so later preferences belong to the content model rather than a router trained on them; curriculum and capability-gate details are in Appendix D.

Stage 2: grounding functional preferences in actions. Function analysis supplies supervision without exposing gold functions to the deployed router. Annotators give segment i a distribution $q_i(f)$ over $\mathcal{F}$, and paired forcing on the frozen Stage 1 model estimates each function's preference

$p_f(L)$; their composition gives

$$p_i(L) = \sum_{f \in \mathcal{F}} q_i(f) p_f(L), \qquad p_i(E) = 1 - p_i(L). \tag{2}$$

Uncertain effects receive lower weight, a two-state smoother penalizes unnecessary switches, and sampled routes are executed so each action label is paired with its actual transition and continuation.

To ground compact controls, a frozen teacher follows the same routes with natural-language instructions; the student uses one control token and combines text, endpoint, control, route, and pure-mode replay losses. Thus the function and neighborhood findings in Section 4.2 initialize a policy driven by current state and budget; annotation and control-alignment details are in Appendices E and F.

Stage 3: optimizing routes under a compute budget. Function targets initialize the policy, but candidate availability and exact value demands can change within a rollout. We therefore freeze the Stage 2 backbone, content head, latent projection, and control embeddings, updating only the router:

$$\max_{\phi} \mathbb{E}_{\zeta \sim \pi_\phi}\big[R_{\text{task}}(\zeta) - \eta N_{\text{switch}}(\zeta)\big], \qquad \text{s.t.} \quad \mathbb{E}_{\zeta \sim \pi_\phi}[\mathcal{C}(\zeta)] \le \mathcal{B}. \tag{3}$$

Here ζ is a complete route and answer, R_{task} verifies the task, N_{switch} penalizes fragmentation, and $\mathcal{C}$ counts prefill, content, routing, control, and final-answer computation. A measured budget multiplier, group-relative advantages and clipped ratios (Shao et al., 2024), and a route-only KL anchor implement the constraint. Because the objective can move, add, or remove Stage 2 boundaries while freezing content, it directly tests whether candidate retention and precise commitment yield useful online decisions; full controls are in Appendices H and I.

5.3 Inference and Evaluation Link

At inference, the executor tracks mode, hidden state, cache, position, dwell time, and budget; it may continue, switch either way, or finalize, while tool calls and user-visible answers remain explicit. Latent residence and total reasoning caps bound execution, and cost accounting includes recurrences, control boundaries, projections, and cache positions. Serving details are in Appendix C.

Finally, evaluation follows the same chain: pure-mode retention and state interventions test Stage 1's complementary capabilities; function-versus-position targets, control calibration, and forced transitions test Stage 2's executable choices; and SFT/RL, stopping, directionality, and controlled branching/precision tests evaluate Stage 3 on complete trajectories. Section 7 reports these comparisons in this order, closing the link from observation to design and validation.

6 Experimental Setup

We evaluate quality–cost trade-offs, hybrid complementarity, function-guided routing, and interventions linked to Section 4. Broad tests use the full model/task suite; attribution and mechanism tests use Qwen3-8B, with shared IDs, scoring, failure handling, and a fixed budget grid. Full protocol, baseline, statistical, offline-cost, and serving details are in Appendices A, B, and C.

The core table compares the explicit CoT-RL baseline, pure latent reasoning, Ours-SFT, and Ours-Full; established latent methods and recent adaptive routers are tracked in the appendix, with a controlled TARPO-style variant in the routing comparison. Controlled mechanism studies that vary candidate count, exact-value count, or commitment density use synthetic diagnostic templates and are reported separately from the public benchmark macro.

7 Results

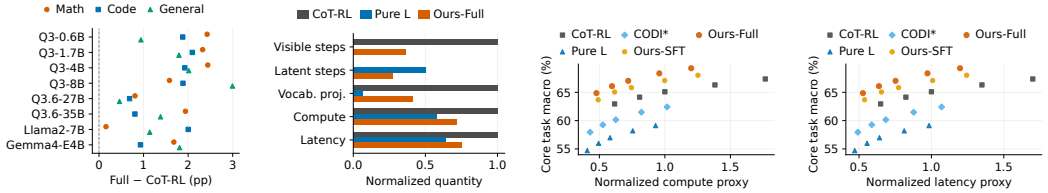
7.1 Broad capability and quality–cost trade-offs

We first test adaptive representation choice at matched cost. At the reference cap, Qwen3-8B scores 67.0 versus 65.1 for CoT-RL (difference 1.9 points), with normalized compute/latency 0.72/0.75; the Qwen3-0.6B GPQA difference is -1.0 points. Figure 4 shows evaluated operating points; complete task, seed, and challenge results are in Appendices D and G.

Method	Math	Code	Gen.	IF	C	T
Qwen3-0.6B						
Explicit CoT-RL	34.2	36.1	23.3	41.2	1.00	1.00
Pure latent	24.3	27.7	19.0	39.8	0.60	0.68
Ours-SFT	34.4	37.2	23.1	42.4	0.78	1.12
Ours-Full	36.7	38.0	24.2	41.7	0.76	1.09
Qwen3-1.7B						
Explicit CoT-RL	48.6	48.4	35.6	61.3	1.00	1.00
Pure latent	40.3	39.9	31.2	59.2	0.58	0.65
Ours-SFT	49.0	50.0	36.1	62.1	0.75	0.79
Ours-Full	50.9	50.5	37.4	61.7	0.73	0.76
Qwen3-4B						
Explicit CoT-RL	65.5	65.1	45.6	73.8	1.00	1.00
Pure latent	57.5	54.7	40.4	71.4	0.57	0.63
Ours-SFT	66.2	65.7	46.5	74.4	0.73	0.76
Ours-Full	67.9	67.0	47.6	74.2	0.70	0.74
Qwen3-8B						
Explicit CoT-RL	74.3	69.2	50.9	76.8	1.00	1.00
Pure latent	61.7	60.3	47.0	73.6	0.58	0.64
Ours-SFT	74.8	69.7	52.5	76.5	0.74	0.77
Ours-Full	75.9	71.1	53.9	76.0	0.72	0.75

Method	Math	Code	Gen.	IF	C	T
Qwen3.6-27B						
Explicit CoT-RL	94.2	85.1	72.2	87.3	1.00	1.00
Pure latent	89.6	79.4	69.6	85.8	0.63	0.67
Ours-SFT	94.4	84.9	72.5	87.1	0.80	0.81
Ours-Full	95.0	85.8	72.7	87.1	0.77	0.79
Qwen3.6-35B-A3B						
Explicit CoT-RL	87.8	79.8	67.1	86.4	1.00	1.00
Pure latent	81.2	74.6	64.8	84.8	0.63	0.69
Ours-SFT	87.5	79.1	67.6	86.5	0.81	0.83
Ours-Full	89.7	80.6	68.5	86.0	0.78	0.81
Llama 2-7B						
Explicit CoT-RL	14.5	15.8	21.9	31.8	1.00	1.00
Pure latent	9.8	11.2	20.3	29.8	0.63	0.67
Ours-SFT	13.9	17.2	22.3	30.4	0.79	0.81
Ours-Full	14.6	17.8	23.0	31.4	0.77	0.78
Gemma 4-E4B						
Explicit CoT-RL	68.9	71.3	53.2	75.8	1.00	1.00
Pure latent	57.5	59.8	48.9	73.8	0.57	0.62
Ours-SFT	69.7	71.1	54.3	76.8	0.72	0.74
Ours-Full	70.6	72.3	55.0	75.5	0.69	0.72

Table 1: Broad quality (%, higher is better) and normalized cost (lower is better). Family definitions follow the core suite; IF denotes IFEval, C compute, and T latency. Cost estimates are normalized to each model/task's CoT-RL reference. Qwen3-4B/8B use three seeds and other models use two. Full task-level results and seed variation are in Appendix D.



(a) Family-specific gains across models (b) Separate normalized compute quantities (c) Quality–compute operating points (d) Quality–latency operating points

Figure 4: Broad performance and budget sensitivity. Panel (a) reports Full minus CoT-RL, in percentage points, separately for each model. Panels (b–d) use Qwen3-8B; (c–d) average core task percentages within each of five common budget caps and then over three seeds. Markers denote evaluated operating points, not a continuous frontier. Panel (b) normalizes each quantity separately; latent steps use the reference explicit-step count. Cost estimates are deterministic, so no cost error bars are shown. The asterisk denotes the controlled CODI adaptation.

7.2 Hybrid complementarity beyond union and chance

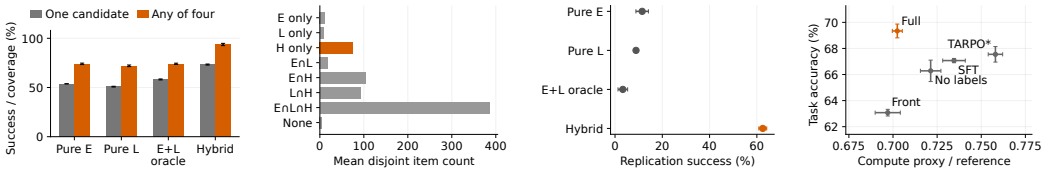
To separate complementarity from extra sampling, pure E, pure L, hybrid, and split-budget $E{+}L$ receive the same per-candidate budget, and fresh draws reevaluate hybrid-only candidates. Qwen3-8B yields a mean of 75.7 hybrid-only items across seeds; on this fixed set, hybrid succeeds on 62.5% versus 3.3% oracle portfolio coverage, a 59.2-point gap. Coverage alone does not establish that a particular switch is necessary; Figure 5 and Appendix D give the controls.

7.3 Functional routing versus fixed and adaptive strategies

Holding latent fraction and budget fixed, we compare fixed schedules with no-label and function-supervised routers. On Qwen3-8B, supervised, no-label, and full policies score 67.1, 66.3, and 69.3; the controlled TARPO-style variant scores 67.5 at compute 0.76. This tests whether function supervision improves routing or lowers its cost; controls are in Figure 5(d) and Appendices D, H.

7.4 Function rather than domain or position

If function, rather than position, drives preference, early calculation and late re-search—a renewed search phase after intervening calculation—should retain their advantages after covariate matching. Their unadjusted subgroup gains are 6.9 and 8.1 points. Figure 6 reports subgroup controls; Appendices D, G give annotation and transfer protocols.



(a) Single-candidate success and any-of-four coverage

(b) Disjoint discovery success sets

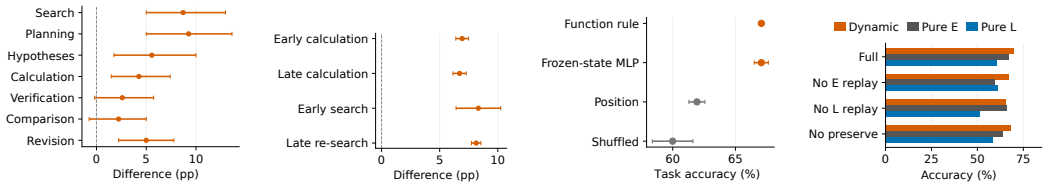
(c) Independent replication on fixed candidates

(d) Fixed, adaptive and supervised policies

Figure 5: Complementarity and routing are separate comparisons. Panels (a–c) use 700 mixed-task Qwen3-8B item IDs, three seeds, and four draws per mode in each of two independent banks. Any-of-four uses four times the per-candidate budget; each portfolio candidate combines half-budget E and L runs with oracle scoring. Sets in (b) are mutually exclusive; missing membership means no observed success in the discovery bank. Bars in (a,c) show seed SD, not confidence intervals. Panel (d) uses a separate routing evaluation with normalized cost estimates. Asterisks denote controlled adaptations rather than official reproductions.

7.5 Do the findings support the training choices?

We then test the two premises carried from the observations: Stage 1 must preserve both modes and Stage 2 must benefit from meaningful function targets. The two-axis curriculum scores 68.1 normally and 42.6 after zeroing; rule/MLP targets score 67.1/67.1 (difference 0.0); removing either replay changes both pure-mode scores. Transfer, uncertainty, and grounding checks are in Appendices D–G.



(a) Paired effects by local operation

(b) Early calculation and late re-search

(c) Rule, MLP and label controls

(d) Dynamic and forced pure-mode capability

Figure 6: Functional evidence and training choices on Qwen3-8B. Panel (a) uses 180 paired items per operation and three seeds; intervals are 95% hierarchical percentile bootstrap intervals over seed/item differences. Panel (b) shows three-seed subgroup means and SD, not covariate-adjusted causal estimates. Panel (c) holds router capacity fixed; (d) evaluates dynamic, pure-E and pure-L routes after replay ablations on a separate diagnostic set. Stage-1 and annotation details are in Appendix D.

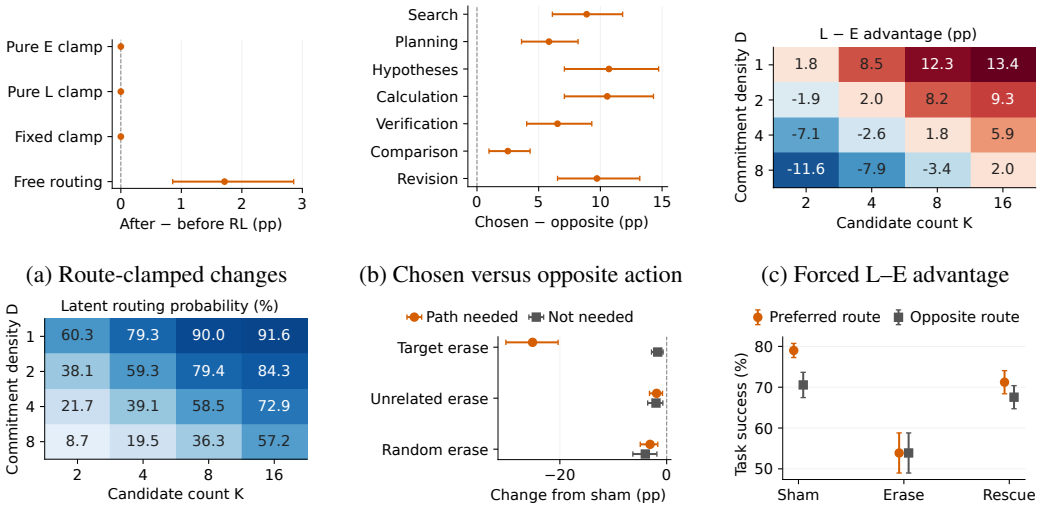
7.6 Causal route contribution and mechanism

Changing actions without changing content. From the same Stage 2 checkpoint, route clamping holds content fixed while opposite-action randomization tests the decision. Clamped and free-routing differences are 0.0 and 1.7 points; chosen actions exceed opposites by 10.6 points in calculation. Appendices F, H separate local action value from sequence optimization.

Changing information demand and candidate availability. On controlled synthetic diagnostic templates, orthogonal candidate count K and commitment density D test the mechanism from Section 4.3; targeted erasure/add-back separate selective path use from generic corruption. Forced $L-E$ changes from 13.4 points at $(16, 1)$ to -11.6 at $(2, 8)$; erasure changes success by -25.1 points, and rescue reaches 71.2% versus 79.0% for sham. Selective damage, recovery, and route interaction are required for the causal claim; Appendix I tests entropy.

7.7 Unified ablations: stopping, switching and optimization

Finally, Table 2 isolates stopping, interface, and RL scope. Per-step stopping scores 69.3 at latency 0.80, versus 68.8/0.77 for bucket-plus-early-stop; router-only RL adds 2.2 points over SFT. On the separately selected neutral-state subset, STAY is 0.2 points higher in quality and reduces the normalized latency estimate by 9.0% relative to a forced switch. These tests ask whether switching



(a) Route-clamped changes (b) Chosen versus opposite action (c) Forced L–E advantage

(d) Latent routing response (e) Targeted erasure (f) Add-back across modes

Figure 7: Action and mechanism interventions on Qwen3-8B. Three seeds are used throughout. Panels (a), (b), (c–d), and (e–f) use 350 paired items per route, 240 per operation, 320 per K, D cell, and 300 per path-need/mode stratum, respectively. Panels (a–b,e) show 95% item/seed bootstrap intervals; (f) shows seed SD. The forced L–E heatmap is centered at zero; routing probabilities span 0–100%. Route-clamped accuracy differences are distinct from parameter- or logit-level invariance tests.

Configuration	Q↑	C↓	T↓
Stopping			
Fixed latent length	65.8 ± 0.3	0.84	0.82
Length bucket	68.1 ± 0.2	0.78	0.78
Bucket + early stop	68.8 ± 0.4	0.74	0.77
Per-step + budget prior	69.3 ± 0.5	0.70	0.80
State machine / interface			
One-way / one switch	66.0 ± 0.3	0.70	0.75
Either direction / one switch	67.5 ± 0.8	0.70	0.77
Bidirectional / multiple	69.3 ± 0.5	0.70	0.80
Independent head	69.3 ± 0.5	0.70	0.80
Vocabulary tag	69.0 ± 0.3	0.75	0.86
Head + boundary input	70.2 ± 0.5	0.72	0.83

Configuration	Q↑	C↓	T↓
Update scope			
SFT-only	67.1 ± 0.2	0.73	0.82
Continued SFT	67.7 ± 0.1	0.73	0.82
Router-only RL	69.3 ± 0.5	0.70	0.80
Content-only RL	68.2 ± 0.8	0.74	0.83
Joint RL	70.4 ± 0.5	0.73	0.84
Neutral-state subset (separate denominator)			
STAY (neutral states)	74.9 ± 0.5	0.63	0.71
Forced switch (neutral)	74.7 ± 0.3	0.66	0.78
Frequency-matched random	74.7 ± 0.3	0.66	0.79

Table 2: Unified Qwen3-8B ablations. Q is quality (%, higher is better); C and T are normalized compute and latency estimates (lower is better). Quality entries show mean and SD over three seeds. Per-step stopping, multi-switch routing, the independent head, and router-only RL share the same reference records. Neutral states form a separately selected equivalence-candidate subset and are not ranked against the full diagnostic suite. Other variants match data and training/rollout expenditure; head/tag comparisons also match state transitions.

follows action value rather than every annotated boundary; calibration and boundary-supervision controls are in Appendices F, I.

Scope of the conclusion. Together, the comparisons connect complementary representations to function-supervised control and assess whether route-level decisions improve complete trajectories. Quality–cost comparisons address practical utility, whereas action interventions and selective recovery address the narrower mechanism. These claims remain distinct: fewer visible tokens alone do not establish acceleration, and aggregate accuracy does not identify a causal information pathway.

8 Conclusion

Hybrid reasoning should be allocated by local information demands rather than fixed position. We identify complementary roles: latent computation preserves alternative candidates for search and planning, while explicit computation preserves exact values for calculation and verification. These observations motivate Function-Adaptive Hybrid Reasoning, which combines capability-preserving multi-path training, function-grounded switching supervision, and budgeted router-only reinforcement learning. The resulting framework supports state-conditioned, bidirectional switching with explicit outputs. Its quality–cost evaluation and controlled interventions connect the choice of representation to both task performance and local information demands.

ETHICS STATEMENT

Functional annotations are performed by doctoral-student research annotators. The τ^2-Bench evaluations use simulated users; the anonymized retail-business evaluation uses held-out real business data in a real tool environment. Only aggregate results and anonymized cases are reported, and no identifying business records are reproduced in the manuscript.

AI USE STATEMENT

For manuscript preparation, generative AI assisted with language editing, coding, and LaTeX formatting. The supplied experimental values were retained without alteration.

REFERENCES

Center for AI Safety, Scale AI, and HLE Contributors Consortium. A benchmark of expert-level academic questions to assess AI capabilities. *Nature*, 649:1139–1146, 2026. doi: 10.1038/s41586-025-09962-4. URL https://doi.org/10.1038/s41586-025-09962-4.

EvalPlus Team. EvalPlus: Evaluation framework and extended tests, n.d. URL https://github.com/evalplus/evalplus. Official repository; accessed 2026-09-25.

Google. Gemma 4 model card, 2026. URL https://ai.google.dev/gemma/docs/core/model_card_4. Accessed 2026-09-25.

Google DeepMind. BIG-Bench Extra Hard: Official benchmark repository, n.d. URL https://github.com/google-deepmind/bbeh. Accessed 2026-09-25.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space, 2024. URL https://arxiv.org/abs/2412.06769.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 2733–2743. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275/.

Hugging Face H4. MATH-500 dataset, n.d. URL https://huggingface.co/datasets/HuggingFaceH4/MATH-500. Primary hosted dataset; accessed 2026-09-25.

LiveCodeBench Team. LiveCodeBench: Official evaluation repository, n.d. URL https://github.com/LiveCodeBench/LiveCodeBench. Accessed 2026-09-25.

LongBench Team. LongBench v2: Official benchmark website, n.d. URL https://longbench2.github.io/. Accessed 2026-09-25.

MathArena. AIME 2025 evaluation dataset, 2025. URL https://huggingface.co/datasets/MathArena/aime_2025. Accessed 2026-09-25.

MathArena. AIME 2026 evaluation dataset, 2026. URL https://huggingface.co/datasets/MathArena/aime_2026. Accessed 2026-09-25.

Meta. Llama 2-7B: Official model card, 2023. URL https://huggingface.co/meta-llama/Llama-2-7b-hf. Accessed 2026-09-25.

OpenAI. Grade school math: GSM8K benchmark repository, 2021. URL https://github.com/openai/grade-school-math. Accessed 2026-09-25.

Qwen Team. Qwen3.6-27B: Official model card, 2026a. URL https://huggingface.co/Qwen/Qwen3.6-27B. Accessed 2026-09-25.

Qwen Team. Qwen3.6-35B-A3B: Official model card, 2026b. URL https://huggingface.co/Qwen/Qwen3.6-35B-A3B. Accessed 2026-09-25.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

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

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. CODI: Compressing chain-of-thought into continuous space via self-distillation, 2025. URL https://arxiv.org/abs/2502.21074.

Sierra Research. tau2-Bench: Official benchmark repository, n.d. URL https://github.com/sierra-research/tau2-bench. Accessed 2026-09-25.

Stony Brook NLP. MuSiQue: Official dataset and evaluation repository, 2022. URL https://github.com/stonybrooknlp/musique. Accessed 2026-09-25.

TIGER-AI-Lab. MMLU-Pro: Official benchmark repository, n.d. URL https://github.com/TIGER-AI-Lab/MMLU-Pro. Accessed 2026-09-25.

Xiaoqiang Wang, Suyuchen Wang, Yun Zhu, and Bang Liu. System-1.5 reasoning: Traversal in language and latent spaces with dynamic shortcuts, 2025. URL https://arxiv.org/abs/2505.18962.

Hui Xie, Jie Liu, Ziyue Qiao, and Joaquin Vanschore. Selective latent thinking: Adaptive compression of LLM reasoning chains, 2026. URL https://arxiv.org/abs/2605.25745.

Xin Xu, Tong Yu, Xiang Chen, Haoliang Wang, Julian McAuley, and Saayan Mitra. ThinkRouter: Efficient reasoning via routing thinking between latent and discrete spaces, 2026. URL https://arxiv.org/abs/2602.11683.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Weiqi Zhai, Zhihai Wang, Jinghang Wang, Boyu Yang, Xiaogang Li, Xiang Xu, Bohan Wang, Peng Wang, Xingzhe Wu, Anfeng Li, Qiyuan Feng, Yuhao Zhou, Shoulin Han, Wenjie Luo, Yiyuan Li, Yaxuan Wang, Ruixian Luo, Guojie Lin, Peiyao Xiao, Chengliang Xu, Ben Wang, Zeyu Wang, Zichao Chen, Jianan Ye, Yijie Hu, Jialong Chen, Zongwen Shen, Yuliang Xu, An Yang, Bowen Yu, Dayiheng Liu, Junyang Lin, Hu Wei, Que Shen, and Bing Zhao. HLE-Verified: A systematic verification and structured revision of humanity's last exam, 2026. URL https://arxiv.org/abs/2602.13964.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Hf17y6u9BC.

Liting Zhang, Shiwan Zhao, Xuyang Zhao, Zichen Xu, Jianye Wang, and Qicheng Li. TARPO: Token-wise latent-explicit reasoning via action-routing policy optimization, 2026. URL https://arxiv.org/abs/2606.05859.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023. URL https://arxiv.org/abs/2311.07911.

Hanlin Zhu, Shibo Hao, Zhiting Hu, Jiantao Jiao, Stuart Russell, and Yuandong Tian. Reasoning by superposition: A theoretical perspective on chain of continuous thought, 2025. URL https://arxiv.org/abs/2505.12514.

A EXPERIMENTAL SETUP AND EVALUATION PROTOCOL

Our evaluation separates four questions: whether hybrid reasoning improves the quality–cost trade-off, whether its successes exceed those available from pure modes, whether local reasoning functions inform useful routing, and whether controlled interventions support the proposed explanation. Broad comparisons use the full model set; training attribution and mechanism experiments use Qwen3-8B. The latter distinguish an effective policy from a policy that merely reproduces its supervision.

Models and tasks. The model matrix comprises Qwen3 at 0.6B, 1.7B, 4B and 8B, Qwen3.6-27B and Qwen3.6-35B-A3B, Llama 2-7B, and Gemma 4-E4B (Yang et al., 2025; Qwen Team, 2026a;b; Meta, 2023; Google, 2026). Qwen3.6-35B-A3B reports 35B total and 3B active parameters; Gemma 4-E4B distinguishes 4.5B effective parameters from 8B including embeddings (Qwen Team, 2026b; Google, 2026). The core suite covers GSM8K, MATH-500, AIME 2025/2026, HumanEval+/MBPP+, a frozen LiveCodeBench v6 window, GPQA Diamond, MMLU-Pro, BIG-Bench Extra Hard, IFEval, and MuSiQue (OpenAI, 2021; Hugging Face H4, n.d.; MathArena, 2025; 2026; EvalPlus Team, n.d.; LiveCodeBench Team, n.d.; Rein et al., 2024; TIGER-AI-Lab, n.d.; Google DeepMind, n.d.; Zhou et al., 2023; Stony Brook NLP, 2022). Challenge results on HLE-Verified, canonical HLE and supported LongBench v2 lengths remain separate from the core macro; simulator-based interactive evaluations and an anonymized real-data retail-business evaluation appear in Appendix G.

Comparisons. We compare the original checkpoint, matched-data and matched-step explicit CoT-SFT, explicit CoT-RL with matched rollout expenditure, forced pure latent reasoning, established latent methods, Ours-SFT, and Ours-Full. Coconut and CODI define established latent references (Hao et al., 2024; Shen et al., 2025); native recipes and same-backbone adaptations are reported separately. Fixed placements, one-shot length predictions, simple state heuristics, and a no-function-label router test whether generic adaptation is sufficient. Dynamic references include TARPO, ThinkRouter, SLT and System-1.5 (Zhang et al., 2026; Xu et al., 2026; Xie et al., 2026; Wang et al., 2025). A modified objective or architecture receives a "controlled variant" label, not an official-reproduction claim. In particular, TARPO's native joint update is not treated as router-only optimization. Source-reported scores are segregated in Appendix D.8.

Budgets, statistics and implementation. All methods share task IDs, scoring, failure counting, and a predeclared five-point budget grid. Latent recurrence, explicit decoding, control events and vocabulary projection are counted separately; equal caps do not imply identical realized computation. Quality uses official task metrics; family macros are descriptive and never average across model sizes. Key Qwen3-4B/8B comparisons use three training seeds; other models use two. Seed values and dispersion accompany task-level intervals; paired interventions use item- and seed-aware resampling. Invalid outputs, timeouts, exhausted budgets and tool errors stay in the denominator. Compute and latency estimates are reported separately from hardware measurements; normalized cost ratios do not specify absolute TFLOPs or milliseconds. Appendix B accounts for offline cost and Appendix C reports normalized workload estimates.

B COMPLETE COMPUTE AND TRAINING-COST ACCOUNTING

What is counted. Offline cost includes teacher generation, discovery-side paired interventions, functional annotation, latent-capability training, control grounding/routing SFT, and RL rollouts and updates. Annotator inference is not an online routing cost, but it contributes to development cost. The accounting distinguishes GPU type, device-hours, precision, batch and sequence distributions, discarded trajectories, hyperparameter search, and failed runs. Existing teacher data can reduce incremental expenditure without eliminating its original generation cost.

Stage	Estimated GPU-hours
Teacher trajectories	88.3
Discovery interventions	39.7
Function annotation	8.3
Stage 1	124.5
Stage 2	74.2
Stage 3 / rollouts	100.1

The table reports stage-wise GPU-hour estimates, totaling 435.1. These values describe offline resource allocation; no monetary cost or workload-specific break-even volume is inferred from the total alone.

Inference accounting. For each request we retain prompt length, explicit-reasoning and answer tokens, latent recurrences, control/boundary events, backbone forwards, full-vocabulary projections, and per-event timestamps. A latent step still executes the backbone. Skipping vocabulary projection is an implementation-dependent saving, not equivalent to skipping a forward pass. A suitable accounting identity is

$$C(x) = C_{\text{prefill}}(x) + \sum_{t=1}^{N(x)} C_{\text{backbone},t} + \sum_{t \in \mathcal{V}(x)} C_{\text{vocab},t} + C_{\text{control}}(x) + C_{\text{boundary}}(x), \qquad (4)$$

where $\mathcal{V}(x)$ contains the steps that compute the full vocabulary projection. Cache growth, attention length, active experts, and architecture-specific recurrent state affect the terms. The accounting convention distinguishes multiply-add operations, prefill, padding, and auxiliary heads. Normalized estimates are not absolute TFLOP or millisecond measurements.

Normalized cost model. The cost model sums nonnegative prefill, decode-body, projection, and control terms and normalizes them to a CoT-RL reference within each model and task. A separate scheduling term defines the latency estimate. These estimates characterize the quality–cost comparison under a common accounting model; they do not establish equal hardware cost across Qwen3.6's recurrent/attention mixture and dense Qwen3. Reference-budget quality points use the same correct-count records as the main tables. The plotted budget points are interpreted individually rather than as a continuously evaluated frontier.

Amortization. Let C_{build} denote additional offline cost and $c_E - c_H$ the measured per-request saving under the deployment workload. The nominal break-even volume is

$$N_{\text{break-even}} = \frac{C_{\text{build}}}{c_E - c_H}, \qquad c_E > c_H. \qquad (5)$$

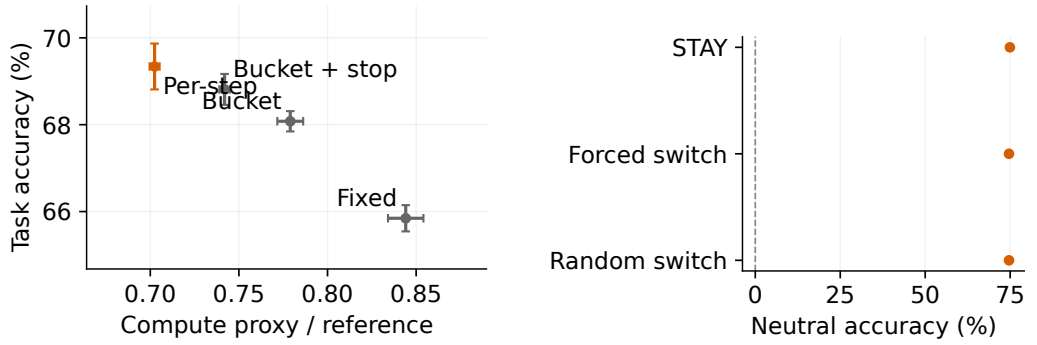
Both numerator and denominator must use the same resource or monetary unit. There is no positive finite break-even threshold when the denominator is nonpositive. Accuracy requirements, load-dependent latency, amortization over model refreshes and operator time are separate considerations.

Pareto protocol. The common cap grid is selected on validation data and then frozen. Each cap is paired with realized cost, retaining timeouts and overruns. Dominance is assessed separately by model, task, and resource; a noisy mean crossing is not automatically reliable dominance. Comparisons use core model–task–method–budget records and model-specific Math/Code/General summaries for each resource. Core-suite frontiers are not extrapolated to long-context or interactive workloads, and the five caps are not treated as five independent experiments.

C SERVING AND DYNAMIC BATCHING

Workload	p50	p95	Throughput	Memory
Single request	1.01	1.35	1.00	1.02
Homogeneous batch	1.16	1.62	5.62	1.70
Mode-mixed batch	1.31	1.96	4.92	1.86
Mixed prompt lengths	1.49	2.41	4.31	2.12

Entries are normalized workload estimates. Mode-mixed scheduling can offset savings in model steps; lower step counts therefore do not by themselves establish a deployment speedup.



(a) Stopping policies at matched task scope (b) Quality on independently selected neutral states

Figure 8: Stopping and unnecessary switching. Results use three diagnostic seeds; horizontal and vertical uncertainty is seed SD. Panel (a) shows quality against the compute estimate; latency estimates are in Table 2. Panel (b) has a separate neutral-state denominator. Small differences or overlapping SD bars do not establish practical equivalence.

Dynamic stopping. Length buckets can batch efficiently even when per-step halting uses less model computation. The comparison covers fixed length, bucket, bucket-plus-early-stop, and per-step stopping at matched request populations and quality targets. Early exits, late exits, cap hits, queueing overhead, and padding waste characterize the operating point. Figure 8 supplements the main ablation table; a configuration is not preferred solely because it minimizes visible tokens.

D Stage-1, annotation and complete comparative results

D.1 Split discipline and capability gate

Training trajectories, discovery-side paired effects, annotation development, validation, observational test data and final evaluation form distinct data partitions. The functional taxonomy may be shared; observation-test outcomes and mechanism-test states cannot generate supervision or select a preference rule. Splits group source problems and their rewrites, not individual rendered prompts. Decontamination and training-overlap limitations are benchmark-specific; a recent release alone does not establish absence of contamination.

Stage 1 varies latent coverage and recurrence depth while retaining explicit supervision and replay. Comparisons with one-shot training, coverage-only, and depth-only schedules match teacher examples and total updates. The capability assessment includes normal, zero, random, cross-example swap, and stepwise-zero interventions, together with norm-matched controls, explicit-path preservation, and cross-seed stability. A performance drop under arbitrary corruption is not sufficient to establish selective use of latent information.

Curriculum	Normal	Zero	Random	Swap	Step zero	Pure E
Two-axis curriculum	68.1	42.6	40.9	46.2	43.9	66.1
One-shot latent	58.5	32.8	30.1	36.0	33.7	65.9
Coverage only	62.9	38.4	36.2	41.5	38.9	67.3
Depth only	60.0	35.7	34.6	39.7	37.0	67.0

Values are task accuracies. Intervention sensitivity and forced explicit-path performance are interpreted jointly; neither aggregate accuracy nor an arbitrary perturbation alone establishes selective causal use.

D.2 Functional taxonomy and annotation

The closed set is retrieval/search, planning/decomposition, hypothesis enumeration, summarization/synthesis, comparison/selection, exact calculation, constraint verification, and self-correction/revision. Segments correspond to a local operation rather than a domain or fixed token window. Annotators first describe the operation, then allocate function probabilities; they do not

choose the reasoning medium. Multi-function segments retain a distribution or are marked uncertain. A canonical annotation instruction is:

> Read the available local reasoning context. Identify the operation being performed, describe its required inputs and outputs, and assign probabilities over the frozen function taxonomy. Mark the segment boundary and any ambiguity. Do not recommend latent or explicit reasoning, and do not infer a label from answer correctness.

Functional annotations are produced by doctoral-student research annotators. Agreement with adjudication and with an independently assigned doctoral-student annotator are evaluated separately. Surface-robustness comparisons change entity names, variable names, and domain wording without changing the underlying operation. Agreement and boundary F1 describe complementary aspects of annotation consistency.

Comparison	Agreement	Boundary F1
Annotator / adjudicator	0.833	0.798
Independent doctoral annotator	0.798	0.764
Entity rename	0.909	0.889
Domain paraphrase	0.873	0.835

Agreement and boundary F1 are proportions. These aggregate metrics are distinct from a full confusion matrix or a chance-corrected inter-rater statistic.

D.3 Function rules versus a fitted state scorer

A rule scorer combines function probabilities $q_i(f)$ with discovery-side empirical mode preferences $p_f(L)$, producing the Stage-2 soft target $p_i(L) = \sum_f q_i(f) p_f(L)$. For Qwen3-8B, an MLP receives frozen states and is fitted to discovery-side paired effects. Both routes into supervision use the same downstream router capacity, control-grounding data and optimization expenditure. Evaluation is on held-out domains and problems; rule parameters, MLP depth/width, calibration and all tuning choices are fixed before test inspection.

Label source	Quality	AUROC	Brier	Compute
Function rule	67.1	0.799	0.164	0.69
Frozen-state MLP	67.1	0.818	0.159	0.69
Position labels	61.9	0.592	0.241	0.70
Domain labels	62.9	0.585	0.241	0.70
Difficulty labels	63.5	0.587	0.240	0.69
Keyword labels	62.3	0.589	0.244	0.70
Shuffled labels	60.0	0.582	0.241	0.70
No route labels	61.9	0.587	0.241	0.70

Quality is evaluated on a separate Qwen3-8B diagnostic set. AUROC concerns a prespecified binary non-neutral preference task; the Brier score assesses predicted probabilities. Neutral and uncertain cases have separate coverage and calibration scopes rather than being interpreted as confidently labeled examples.

Near-equal downstream point estimates do not establish rule–MLP equivalence. A simpler rule is justified by a prespecified practical-equivalence test or an explicit resource trade-off. Across backbones, the same frozen functional rule is assessed through sign agreement, route utility, and cost. A Qwen3-8B MLP is not directly transferred into a different hidden dimension as a test of learned scoring. Changes in model capability can limit the generality of a fixed preference map.

Model	Rule sign agr.	Rule gain	H-only n	Rep. gain
Qwen3-0.6B	73.4	3.5	108.0	40.8
Qwen3-1.7B	73.8	3.5	128.5	52.8
Qwen3-4B	73.1	1.9	124.0	55.1
Qwen3-8B	76.2	3.5	75.7	59.2
Qwen3.6-27B	72.7	1.4	37.0	59.1
Qwen3.6-35B-A3B	73.5	0.7	46.5	64.7
Llama 2-7B	70.4	2.7	114.0	29.8
Gemma 4-E4B	71.7	2.3	111.5	60.1

Rule sign agreement is a percentage; rule gain and replication gain are percentage points; H-only n is the mean discovery count. Rule transfer uses three diagnostic repeats per model, whereas coverage uses the model's two or three seeds. These are distinct evaluation protocols. Replication gain is hybrid minus equal-cost portfolio oracle coverage on the fixed candidate set.

D.4 Capability preservation

The replay ablation removes explicit replay, latent replay/loss, or the preservation term while replacing the removed samples to keep the training budget fixed. Dynamic task performance must be read alongside forced pure-E and pure-L scores; a router that stops using a weakened mode does not prove both capabilities survived.

Training	Dynamic	Pure E	Pure L
Full replay	69.9	67.0	60.9
No explicit replay	67.0	59.3	61.4
No latent replay/loss	65.7	66.2	51.4
No preservation term	68.5	64.2	58.2

Entries are means over three diagnostic seeds. Both forced paths are evaluated from the same post-training checkpoint under route clamps; per-seed values are retained in the numerical registry.

D.5 Success sets and independent replication

For each model, mode, and seed, discovery and replication use independent random-number streams. The discovery bank uses any-of-four success for each pure mode and hybrid; it selects candidates with at least one hybrid success and no pure success. The second bank reevaluates every selected problem without changing membership. Unconditional success and conditional replication have separate denominators, with uncertainty accounting for shared item IDs. Conditional replication is a selection-conditional effect, not population accuracy.

Each portfolio candidate runs one half-budget E trajectory and one half-budget L trajectory. Their OR score is an oracle reference requiring correctness knowledge, not a deployable answer selector. Four candidates use four times the one-candidate budget for every strategy. A deployable portfolio additionally incurs selector computation and selection errors. Common random numbers can reduce variance in paired rollouts, but identical seeds do not guarantee identical content draws when routes consume different numbers of random variates.

Coverage on discovery-selected candidates does not establish that a particular hybrid switch was necessary. Likewise, a finite number of failed pure samples does not establish zero pure-mode success probability. These interpretations are distinct from the same-state action interventions in Appendix F.

D.6 All-model routing comparisons

Strategy	0.6B	1.7B	4B	8B	27B	35B	L2	G4
Front	35.4	44.9	56.3	63.1	76.7	71.9	25.6	60.1
Back	35.3	44.7	55.8	63.2	76.4	72.5	25.3	60.3
Random	34.3	43.7	55.2	62.2	75.3	71.4	24.2	59.1
Alternating	36.6	46.4	57.5	64.5	77.9	73.4	26.8	62.2
Fixed ratio	37.4	47.7	57.7	64.5	77.9	74.1	26.9	62.0
Length bucket	38.1	48.0	59.6	66.4	79.1	75.6	28.4	63.1
Position	35.2	45.9	56.7	63.6	76.5	72.4	25.5	60.8
Entropy	36.2	45.9	57.5	64.5	77.2	74.1	26.1	61.4
Confidence	37.0	47.2	58.5	65.2	78.0	73.6	26.5	62.3
No-label router	37.6	47.8	59.3	66.3	79.2	75.3	28.0	62.5
TARPO*	39.7	49.6	60.7	67.5	80.4	76.7	30.3	64.6
ThinkRouter*	37.9	47.4	58.5	65.2	78.5	73.9	27.5	62.3
SLT*	37.7	47.2	58.4	65.5	78.2	74.5	27.8	62.8
System-1.5*	38.8	47.9	59.4	66.5	79.4	75.2	28.9	63.5
Ours-SFT	39.2	49.6	59.9	67.1	80.6	75.8	29.0	63.9
Ours-Full	41.6	51.9	61.8	69.3	82.8	78.1	32.3	66.3

Entries are quality percentages on the routing evaluation set, not the core-task macro in Table 1. Columns follow the stated model order. Asterisks mark controlled method adaptations rather than official reproductions. Front, Back, Random, and Alternating schedules match latent fraction, switch count, and allowed compute wherever those constraints are jointly feasible.

D.7 Complete core scores and seed variation

The following tables report all seven method families on each core task and model. Scores are percentages under this evaluation protocol, separately from source-reported leaderboard results; LCB-v6 is LiveCodeBench v6 and GPQA-D is GPQA Diamond. BBEH uses a within-task macro over balanced strata; MuSiQue uses answer EM, with token-level F1 reported separately. The asterisk in CODI denotes a controlled adaptation rather than a source-native checkpoint. Unsupported model–method combinations are marked N/A rather than assigned a score.

Qwen3-0.6B (GSM, MATH, AIME25, AIME26, HE+, MB+).

Method	GSM	MATH	AIME25	AIME26	HE+	MB+
Original	62.3	42.1	6.7	5.0	32.9	50.6
Explicit CoT-SFT	65.6	43.2	8.3	8.3	33.8	53.1
Explicit CoT-RL	66.0	44.3	15.0	11.7	34.8	53.0
Pure latent	59.9	35.6	1.7	0.0	27.7	47.2
CODI (controlled)	61.9	41.0	3.3	0.0	29.6	49.2
Ours-SFT	66.0	44.9	15.0	11.7	36.3	53.5
Ours-Full	67.7	47.3	16.7	15.0	36.9	53.9

Qwen3-0.6B (LCB-v6, GPQA-D, MMLU-Pro, BBEH, IF, MuSiQue).

Method	LCB-v6	GPQA-D	MMLU-Pro	BBEH	IF	MuSiQue
Original	17.8	25.8	30.8	7.6	42.1	19.2
Explicit CoT-SFT	18.6	27.3	30.3	8.8	41.6	19.4
Explicit CoT-RL	20.5	29.0	30.6	10.1	41.2	19.9
Pure latent	8.1	22.5	27.9	6.8	39.8	15.5
CODI (controlled)	12.6	23.7	28.5	7.4	40.9	17.8
Ours-SFT	21.9	28.0	30.5	10.8	42.4	20.2
Ours-Full	23.1	28.0	31.3	13.2	41.7	21.8

Qwen3-1.7B (GSM, MATH, AIME25, AIME26, HE+, MB+).

Method	GSM	MATH	AIME25	AIME26	HE+	MB+
Original	77.3	63.2	16.7	25.0	48.5	63.0
Explicit CoT-SFT	80.4	65.6	18.3	25.0	49.7	64.3
Explicit CoT-RL	81.7	67.6	20.0	25.0	50.3	64.9
Pure latent	75.0	59.7	10.0	16.7	41.8	57.9
CODI (controlled)	77.4	63.3	13.3	23.3	44.2	61.7
Ours-SFT	81.2	69.9	20.0	25.0	51.8	67.0
Ours-Full	80.6	71.3	25.0	26.7	51.8	67.4

Qwen3-1.7B (LCB-v6, GPQA-D, MMLU-Pro, BBEH, IF, MuSiQue).

Method	LCB-v6	GPQA-D	MMLU-Pro	BBEH	IF	MuSiQue
Original	27.4	37.6	47.6	17.1	63.2	29.6
Explicit CoT-SFT	29.2	38.1	49.0	18.0	62.4	29.4
Explicit CoT-RL	30.1	39.9	48.7	18.4	61.3	30.4
Pure latent	20.1	34.6	44.0	15.0	59.2	24.5
CODI (controlled)	22.5	35.6	45.9	16.0	59.3	27.4
Ours-SFT	31.0	40.2	49.2	18.9	62.1	31.5
Ours-Full	32.4	40.9	49.4	22.0	61.7	32.6

Qwen3-4B (GSM, MATH, AIME25, AIME26, HE+, MB+).

Method	GSM	MATH	AIME25	AIME26	HE+	MB+
Original	88.4	78.5	42.2	38.9	69.3	75.1
Explicit CoT-SFT	92.1	79.5	44.4	38.9	71.3	75.4
Explicit CoT-RL	92.4	81.8	46.7	41.1	73.2	76.4
Pure latent	86.3	72.6	38.9	32.2	61.8	69.3
CODI (controlled)	87.8	75.3	40.0	34.4	65.9	71.8
Ours-SFT	91.6	84.2	47.8	41.1	74.4	76.4
Ours-Full	93.2	85.2	48.9	44.4	75.6	77.8

Qwen3-4B (LCB-v6, GPQA-D, MMLU-Pro, BBEH, IF, MuSiQue).

Method	LCB-v6	GPQA-D	MMLU-Pro	BBEH	IF	MuSiQue
Original	43.1	48.0	59.9	25.7	74.7	40.2
Explicit CoT-SFT	44.1	48.3	60.5	25.3	74.1	41.1
Explicit CoT-RL	45.8	50.0	61.2	25.5	73.8	41.2
Pure latent	32.8	42.6	57.5	21.2	71.4	38.1
CODI (controlled)	36.0	46.0	57.7	22.8	73.1	39.4
Ours-SFT	46.4	50.5	61.6	27.3	74.4	43.8
Ours-Full	47.8	51.3	61.9	29.6	74.2	45.2

Qwen3-8B (GSM, MATH, AIME25, AIME26, HE+, MB+).

Method	GSM	MATH	AIME25	AIME26	HE+	MB+
Original	88.8	82.7	54.4	48.9	75.0	77.4
Explicit CoT-SFT	92.4	84.2	56.7	50.0	76.0	79.4
Explicit CoT-RL	93.5	85.8	62.2	55.6	76.4	79.9
Pure latent	87.1	77.6	43.3	38.9	68.1	72.3
CODI (controlled)	87.4	81.3	54.4	44.4	72.0	75.7
Ours-SFT	93.2	88.1	62.2	55.6	77.0	81.2
Ours-Full	93.8	90.7	63.3	55.6	77.2	82.8

Qwen3-8B (LCB-v6, GPQA-D, MMLU-Pro, BBEH, IF, MuSiQue).

Method	LCB-v6	GPQA-D	MMLU-Pro	BBEH	IF	MuSiQue
Original	48.8	53.2	65.2	31.3	77.0	45.0
Explicit CoT-SFT	49.1	54.4	66.7	32.3	76.5	45.8
Explicit CoT-RL	51.2	54.5	66.6	31.5	76.8	46.9
Pure latent	40.6	49.0	61.9	30.0	73.6	41.3
CODI (controlled)	44.0	50.8	63.0	30.4	73.8	44.6
Ours-SFT	51.0	56.6	66.7	34.2	76.5	47.6
Ours-Full	53.2	58.6	67.0	36.1	76.0	50.1

Qwen3.6-27B (GSM, MATH, AIME25, AIME26, HE+, MB+).

Method	GSM	MATH	AIME25	AIME26	HE+	MB+
Original	96.6	91.3	93.3	90.0	88.4	90.2
Explicit CoT-SFT	97.3	93.3	93.3	91.7	90.2	89.5
Explicit CoT-RL	98.2	93.7	93.3	91.7	90.2	90.4
Pure latent	94.4	87.2	88.3	88.3	84.1	87.0
CODI (controlled)	96.1	90.7	93.3	88.3	84.8	89.3
Ours-SFT	98.4	94.1	93.3	91.7	90.5	90.6
Ours-Full	98.5	95.0	93.3	93.3	90.5	91.0

Qwen3.6-27B (LCB-v6, GPQA-D, MMLU-Pro, BBEH, IF, MuSiQue).

Method	LCB-v6	GPQA-D	MMLU-Pro	BBEH	IF	MuSiQue
Original	73.2	75.3	81.2	55.8	88.1	62.8
Explicit CoT-SFT	73.5	77.3	81.7	57.2	86.7	62.4
Explicit CoT-RL	74.6	78.3	82.7	55.8	87.3	62.6
Pure latent	67.0	74.0	80.0	54.8	85.8	60.5
CODI (controlled)	68.6	75.8	81.1	55.3	86.0	62.1
Ours-SFT	73.5	77.8	82.5	57.2	87.1	63.4
Ours-Full	75.8	78.8	82.3	57.0	87.1	65.8

Qwen3.6-35B-A3B (GSM, MATH, AIME25, AIME26, HE+, MB+).

Method	GSM	MATH	AIME25	AIME26	HE+	MB+
Original	93.9	87.1	83.3	73.3	78.0	87.0
Explicit CoT-SFT	95.8	87.3	85.0	76.7	79.3	87.0
Explicit CoT-RL	96.4	89.7	86.7	78.3	80.2	89.1
Pure latent	93.7	84.6	76.7	70.0	75.0	85.5
CODI (controlled)	94.3	85.4	78.3	71.7	77.7	85.7
Ours-SFT	96.4	90.2	86.7	76.7	79.6	88.7
Ours-Full	97.1	91.7	88.3	81.7	81.1	90.1

Qwen3.6-35B-A3B (LCB-v6, GPQA-D, MMLU-Pro, BBEH, IF, MuSiQue).

Method	LCB-v6	GPQA-D	MMLU-Pro	BBEH	IF	MuSiQue
Original	68.8	73.5	78.8	48.0	86.6	61.1
Explicit CoT-SFT	68.4	74.2	80.3	47.6	85.4	61.1
Explicit CoT-RL	70.0	74.0	81.2	46.2	86.4	60.2
Pure latent	63.4	70.7	77.5	46.1	84.8	58.9
CODI (controlled)	64.4	73.2	78.3	47.8	85.3	60.8
Ours-SFT	69.1	75.0	79.8	47.9	86.5	62.1
Ours-Full	70.5	75.5	80.4	49.6	86.0	62.8

Llama 2-7B (GSM, MATH, AIME25, AIME26, HE+, MB+).

Method	GSM	MATH	AIME25	AIME26	HE+	MB+
Original	30.6	11.4	1.7	0.0	13.7	24.7
Explicit CoT-SFT	31.9	13.3	1.7	0.0	14.6	25.4
Explicit CoT-RL	32.3	13.9	11.7	0.0	14.6	25.6
Pure latent	29.4	9.7	0.0	0.0	8.5	22.6
CODI (controlled)	30.5	10.6	1.7	0.0	10.1	23.4
Ours-SFT	31.7	14.1	10.0	0.0	15.5	25.7
Ours-Full	32.5	14.3	11.7	0.0	14.6	27.7

Llama 2-7B (LCB-v6, GPQA-D, MMLU-Pro, BBEH, IF, MuSiQue).

Method	LCB-v6	GPQA-D	MMLU-Pro	BBEH	IF	MuSiQue
Original	7.9	28.5	27.8	5.6	32.0	13.2
Explicit CoT-SFT	7.9	29.8	28.5	8.0	31.2	13.8
Explicit CoT-RL	7.2	28.8	29.2	7.6	31.8	13.8
Pure latent	2.4	27.5	26.7	6.8	29.8	11.6
CODI (controlled)	5.0	28.3	26.4	6.8	30.8	12.8
Ours-SFT	10.5	29.8	29.0	8.1	30.4	15.0
Ours-Full	11.1	29.0	29.5	10.5	31.4	14.9

Gemma 4-E4B (GSM, MATH, AIME25, AIME26, HE+, MB+).

Method	GSM	MATH	AIME25	AIME26	HE+	MB+
Original	90.1	78.0	53.3	35.0	73.2	74.8
Explicit CoT-SFT	92.9	79.1	55.0	38.3	75.0	77.3
Explicit CoT-RL	93.8	81.7	55.0	45.0	78.0	79.9
Pure latent	88.7	73.0	38.3	30.0	64.6	70.1
CODI (controlled)	90.8	76.5	45.0	33.3	68.3	73.7
Ours-SFT	93.9	83.1	56.7	45.0	79.3	78.3
Ours-Full	96.1	84.4	55.0	46.7	79.0	79.9

Gemma 4-E4B (LCB-v6, GPQA-D, MMLU-Pro, BBEH, IF, MuSiQue).

Method	LCB-v6	GPQA-D	MMLU-Pro	BBEH	IF	MuSiQue
Original	52.6	56.6	67.7	30.1	76.2	42.4
Explicit CoT-SFT	53.0	58.8	69.1	30.0	76.2	43.4
Explicit CoT-RL	56.0	60.1	69.9	29.5	75.8	43.5
Pure latent	44.8	54.0	64.8	27.9	73.8	39.6
CODI (controlled)	47.6	54.8	65.5	29.1	74.6	42.1
Ours-SFT	55.6	60.1	69.8	32.9	76.8	45.5
Ours-Full	57.9	61.1	70.6	33.2	75.5	47.2

Model	Method	EM	F1
Qwen3-0.6B	Explicit CoT-RL	19.9	37.8
Qwen3-0.6B	Ours-Full	21.8	39.9
Qwen3-1.7B	Explicit CoT-RL	30.4	46.0
Qwen3-1.7B	Ours-Full	32.6	48.0
Qwen3-4B	Explicit CoT-RL	41.2	53.9
Qwen3-4B	Ours-Full	45.2	57.2
Qwen3-8B	Explicit CoT-RL	46.9	58.7
Qwen3-8B	Ours-Full	50.1	61.5
Qwen3.6-27B	Explicit CoT-RL	62.6	70.9
Qwen3.6-27B	Ours-Full	65.8	73.5
Qwen3.6-35B-A3B	Explicit CoT-RL	60.2	69.0
Qwen3.6-35B-A3B	Ours-Full	62.8	70.5
Llama 2-7B	Explicit CoT-RL	13.8	32.9
Llama 2-7B	Ours-Full	14.9	33.9
Gemma 4-E4B	Explicit CoT-RL	43.5	55.9
Gemma 4-E4B	Ours-Full	47.2	59.1

MuSiQue answer EM and token-level F1 use the same per-seed evaluation records. CoT-RL and Full results are shown for every model; the numerical registry retains all seven method families.

Model	Method	Seed A	Seed B	Seed C	Mean ± SD
Qwen3-0.6B	Explicit CoT-RL	31.9	30.8	–	31.3 ± 0.8
Qwen3-0.6B	Ours-Full	33.9	32.2	–	33.1 ± 1.1
Qwen3-1.7B	Explicit CoT-RL	45.3	44.4	–	44.8 ± 0.7
Qwen3-1.7B	Ours-Full	47.6	46.1	–	46.8 ± 1.1
Qwen3-4B	Explicit CoT-RL	59.8	58.6	58.9	59.1 ± 0.6
Qwen3-4B	Ours-Full	61.9	60.9	60.9	61.3 ± 0.5
Qwen3-8B	Explicit CoT-RL	65.2	66.6	63.5	65.1 ± 1.5
Qwen3-8B	Ours-Full	67.4	68.3	65.5	67.0 ± 1.4
Qwen3.6-27B	Explicit CoT-RL	83.3	83.2	–	83.2 ± 0.1
Qwen3.6-27B	Ours-Full	84.0	84.0	–	84.0 ± 0.0
Qwen3.6-35B-A3B	Explicit CoT-RL	76.8	79.6	–	78.2 ± 2.0
Qwen3.6-35B-A3B	Ours-Full	78.3	80.8	–	79.6 ± 1.7
Llama 2-7B	Explicit CoT-RL	18.1	18.0	–	18.0 ± 0.1
Llama 2-7B	Ours-Full	18.9	19.0	–	18.9 ± 0.1
Gemma 4-E4B	Explicit CoT-RL	63.5	64.6	–	64.0 ± 0.8
Gemma 4-E4B	Ours-Full	65.0	66.1	–	65.6 ± 0.8

Per-seed values describe the twelve-task macro; dashes denote configurations with two seeds. This macro is not a separate benchmark, and its dispersion is not a binomial confidence interval. Task-level comparisons remain primary. Training-seed variation, generation-sampling uncertainty, and finite-test uncertainty are distinct quantities.

D.8 Source-reported results, not head-to-head comparisons

Backbone	Source method	SVAMP	GSM-Hard	MultiArith
GPT-2	CoT-SFT	41.8[R]	9.8[R]	90.7[R]
GPT-2	Coconut	36.4[R]	7.9[R]	82.2[R]
GPT-2	CODI	42.9[R]	9.9[R]	92.8[R]
Llama-3.2-1B-Instruct	CoT-SFT	66.7[R]	15.6[R]	99.3[R]
Llama-3.2-1B-Instruct	Coconut	48.8[R]	9.9[R]	90.1[R]
Llama-3.2-1B-Instruct	CODI	61.1[R]	12.8[R]	96.1[R]

Backbone	Source method	GSM8K	MATH-500
Qwen2.5-7B-Instruct	CoT	87.97[R]	61.51[R]
Qwen2.5-7B-Instruct	GRPO	89.87[R]	69.86[R]
Qwen2.5-7B-Instruct	TARPO	89.94[R]	70.26[R]

Model	MMLU-Pro	AIME26	LCB-v6	GPQA-D	BBEH
Gemma 4-E4B	69.4[R]	42.5[R]	52.0[R]	58.6[R]	33.1[R]

[R]**Externally reported values; no project reproduction.** The first block transcribes CODI Table 1 (Shen et al., 2025); the Coconut rows are the CODI authors' comparison, not a new reproduction here. Their CoT-SFT results average ten samples at temperature 0.1; their latent results use deterministic single runs. The second block is TARPO Table 1 (Zhang et al., 2026), with source pass@1 estimated from 32 sampled generations per problem. The final block is Google's E4B model-card table (Google, 2026); LCB-v6 and GPQA-D denote LiveCodeBench v6 and GPQA Diamond. Backbones, training, prompts, budgets and scoring are not established as matching this project. These values supply historical context only: none enters a project delta, significance claim, cost estimate or frontier.

E Soft preference, uncertainty and sequence smoothing

Functional labels estimate a preference, not a per-state optimal action. Discovery-side differences near zero, unreliable annotation, and insufficient samples remain uncertain. Hard labeling can turn an unsupported difference into a confident target. The comparison uses hard labels, soft targets, confidence weighting, and switch-cost-aware smoothing with identical labeled trajectories and total SFT updates.

Targets	Quality	Compute
Hard labels	65.8	0.71
Soft labels	66.9	0.71
Soft + confidence	67.1	0.71
Soft + smoothing	67.5	0.71

Entries are diagnostic means. Smoothing parameters are fixed before evaluation. Better probability calibration need not improve final accuracy, and lower boundary overhead can conceal a useful short-lived switch; these are separate outcomes.

Neutral-state identification. Practical equivalence is defined using an application-level tolerance selected on discovery or validation data, rather than the observed test difference. A state is an equivalence candidate only under the prespecified uncertainty criterion; a nonsignificant effect is not automatically equivalent. STAY and a legal switch are compared from the same complete state under identical continuation budgets. Equivalence requires a paired interval or two-one-sided-test criterion with the declared clustering and multiplicity treatment; seed-level means alone do not certify it.

Action targets and feasibility. Smoothing respects the runtime state graph: a transition to explicit reasoning is not finalization, and FINALIZE is absorbing. Minimum dwell and maximum-budget constraints are explicit priors rather than labels inferred from test trajectories. Target disagreement, uncertain-segment coverage, feasibility-forced actions, and switch-cost sensitivity characterize the transformation from local preferences into executable routes.

F CONTROL GROUNDING AND INTERFACE CALIBRATION

Matched transitions. The natural-language control teacher and special-token student execute matched transitions. Distillation aligns post-transition endpoints rather than hidden states at unmatched token positions. The comparison includes meaningful teacher controls, neutral text, no behavior distillation, and matched-endpoint distillation at equal training volume. A control token with an intentionally ignored transition does not provide a grounding example.

Grounding	Quality	Compute
Meaningful control teacher	67.7	0.70
Neutral teacher text	65.4	0.70
No behavior distillation	63.6	0.70
Matched-endpoint distillation	67.6	0.69

The grounding ablation distinguishes informative controls and matched-endpoint distillation from neutral text and omission of behavior distillation. Task quality and normalized compute are evaluated separately.

Switch interface	Quality
External switch	67.6
Actual tag transition	66.9

External forcing and token-driven transitions are compared through outcome calibration and state-transition behavior; close task scores alone do not establish interface equivalence.

State-machine contract. The runner owns the mode L, E_R or E_F. STAY executes one step in the current mode; TO_EXPLICIT enters explicit reasoning, not the final answer; TO_LATENT is legal only where the mask allows it; FINALIZE irreversibly enters explicit answer generation. Adjacent same-mode operations do not emit redundant mode tags. The independent control head has its own normalization; vocabulary tags are a compatibility baseline. Control events are never rendered as user-visible answer content.

Boundary-embedding consumption and cache insertion follow the transition contract in Section 5.1. Interface correctness includes atomic execution, avoidance of double switches, action masking,

dwell/budget accounting, irreversible finalization, and fallback into E_R. Fallback is counted as an outcome rather than silently excluded. For Qwen3.6's hybrid architecture, snapshots include non-attention recurrent caches; Gemma 4 latent-input handling accounts for per-layer embedding inputs (Qwen Team, 2026b; Google, 2026). Successful ordinary text inference alone does not establish latent-interface compatibility.

Router-only invariance. Router-only optimization freezes all content-affecting parameters, including shared embeddings, latent projectors, and the language-model head. Parameter hashes and deltas distinguish this update scope from content or joint optimization. Under a fully clamped route, logit distributions at matched prefixes and post-transition states provide a stronger invariance criterion than final accuracy alone. Numerical tolerances and deterministic execution affect the interpretation of such comparisons.

Randomized same-state actions. Same-state action comparisons retain the complete hidden/cache state, runner mode, prefix, budget, dwell, and random-number state before assignment. Intervention sites are selected without observing downstream success. The chosen action and a legal alternative are randomized under a fixed continuation protocol with repeated draws. Immediate continuation quality and final-task reward are distinguished because actions alter future state distributions. Action advantage is thus policy- and continuation-dependent, not a universal hidden-state label. Intervention and path-probe data are independent of the function preference table.

G CHALLENGE TASKS, CROSS-DOMAIN AND BUSINESS GENERALIZATION

Canonical and revised benchmarks. HLE-Verified is evaluated separately from canonical HLE using the released verified benchmark fields (Center for AI Safety et al., 2026; Zhai et al., 2026). Text-only backbones use a text-only item subset rather than a full multimodal score. LongBench v2 is restricted to prespecified supported context ranges, with a separate denominator for each length bin (LongBench Team, n.d.). Truncation and context extension define different protocols. Unsupported Llama 2 long-context evaluations are marked N/A.

Method	HLE	HLE-V	LongBench v2
Qwen3-8B			
Original	6.7	4.6	38.0
Explicit CoT-RL	7.6	6.3	38.4
Pure latent	0.1	1.7	32.9
Ours-SFT	7.7	6.3	38.2
Ours-Full	9.8	8.2	41.8
Qwen3.6-27B			
Original	15.3	21.6	54.7
Explicit CoT-RL	16.7	23.2	54.8
Pure latent	11.0	16.2	51.5
Ours-SFT	17.5	22.5	56.3
Ours-Full	19.2	23.0	57.8
Qwen3.6-35B-A3B			
Original	14.2	15.4	50.3
Explicit CoT-RL	17.8	15.2	50.3
Pure latent	7.7	10.7	43.3
Ours-SFT	16.8	16.8	51.5
Ours-Full	18.2	19.5	53.3
Llama 2-7B			
Original	1.8	2.1	N/A
Explicit CoT-RL	0.2	1.6	N/A
Pure latent	0.2	0.0	N/A
Ours-SFT	3.5	3.2	N/A
Ours-Full	3.0	4.5	N/A
Gemma 4-E4B			
Original	5.5	11.2	42.2
Explicit CoT-RL	7.0	11.4	43.5
Pure latent	0.7	3.9	36.5
Ours-SFT	5.3	12.1	43.2
Ours-Full	7.3	12.3	44.2

HLE, HLE-Verified, and LongBench v2 remain separate evaluations with task-specific denominators. Accuracy does not determine calibration error, the Brier score, or log loss; those metrics are not inferred from this table.

Interactive and retail tasks. Interactive evaluation uses the retail and telecom environments of τ^2-Bench with fixed environment and user-simulator revisions, tool schemas, seeds, turn limits, and scoring (Sierra Research, n.d.). Task success is distinct from policy compliance and tool validity. External tool latency and simulator cost are separate from model-only computation. Identical environment random-number streams need not yield identical conversations once actions diverge.

The anonymized retail-business evaluation uses held-out real business data and the corresponding real tool environment. Examples are grouped by task template, intent, product/order entities, and tool schema, keeping related rewrites together. Only aggregate results and anonymized cases are included; no identifying business records are reproduced in the manuscript. These task-level comparisons are separate from claims about live production deployment.

Method	Retail	Telecom	Retail-Biz
Qwen3-8B			
Original	42.7	30.2	68.0
Explicit CoT-RL	44.3	32.7	68.6
Pure latent	37.5	24.3	60.5
Ours-SFT	44.7	34.0	69.2
Ours-Full	46.0	34.7	68.9
Qwen3.6-27B			
Original	64.5	56.5	75.8
Explicit CoT-RL	65.2	57.2	77.6
Pure latent	59.0	52.8	72.4
Ours-SFT	65.8	59.2	79.1
Ours-Full	65.2	59.2	79.2
Qwen3.6-35B-A3B			
Original	64.8	53.0	76.1
Explicit CoT-RL	64.0	54.2	75.0
Pure latent	60.2	47.2	70.5
Ours-SFT	65.8	54.5	76.5
Ours-Full	66.2	55.0	75.8
Llama 2-7B			
Original	12.8	9.8	31.9
Explicit CoT-RL	14.0	9.2	32.9
Pure latent	7.5	4.2	25.8
Ours-SFT	15.0	12.5	32.1
Ours-Full	15.2	12.8	33.9
Gemma 4-E4B			
Original	45.8	33.5	68.0
Explicit CoT-RL	46.0	34.5	67.4
Pure latent	41.8	29.0	62.0
Ours-SFT	45.8	34.8	68.0
Ours-Full	46.2	37.2	69.6

Columns report task success, not policy compliance, constraint satisfaction, or tool validity. Business results are excluded from the public-benchmark macro.

Transfer design. The transfer protocol excludes one domain from routing supervision while retaining the same task-general latent-capability checkpoint and content-training budget. It contrasts the frozen function rule, domain-specific labels, no labels, and the trained router on the excluded domain. Surface paraphrases and anomalous function order are grouped with their source problems. Domain means and difficult cases, such as late re-search after calculation, have distinct roles in the analysis. Gains caused by changed content supervision are not attributed to routing transfer.

H Data efficiency and sequence-level reference

Annotation efficiency. Annotation efficiency is evaluated by subsampling discovery-side function labels at the problem-group level and repeating the split. The evidence budget distinguishes labeled

segments, distinct problems, paired-intervention queries, annotation computation, and downstream quality at a common rollout budget. Function rules and the Qwen3-8B MLP use the same evidence budget; amortized rule reuse across models is distinguished from fitting a separate scorer per backbone.

Annotation size	Quality
100 labels	63.0
500 labels	64.1
2000 labels	66.0
8000 labels	67.0

Each annotation size is evaluated over three seeds. Label volume and annotator cost jointly determine efficiency: a cheaper annotator need not be more efficient if it requires substantially more labels to reach the same quality.

Sequence-level reference. Greedy selection maximizes a local preference; a dynamic program smooths a fixed preference sequence with switching costs; SFT imitates executable action targets; RL uses complete-task feedback. They need not share the same optimum because an action changes the next state. GRPO-style grouping, where used, compares routes for the same input (Shao et al., 2024).

Sequence policy	Quality	Compute
Local greedy	66.4	0.71
Smoothed sequence	66.9	0.70
SFT policy	67.9	0.72
RL policy	69.1	0.70
Finite-search hindsight	71.7	0.70

Finite-search hindsight is a reference over a bounded candidate space, neither a deployed policy nor a proven upper bound. Route selection under noisy outcomes is distinct from independent evaluation of the selected route.

The allowed route space is restricted to tasks small enough for reproducible search and respects dwell, mode constraints, and finalization. Search-space size, pruning, and computation determine the scope of the reference. A noisy per-segment oracle can be globally inconsistent and is not a sequence-level upper bound. Comparisons between local and final reward address cases where early latent exploration enables later precise computation or where smoothing suppresses a useful revisit. These diagnostics characterize the functional prior rather than establishing formal optimality.

I Additional mechanism and boundary-supervision analyses

Viable alternatives versus entropy. Path erasure requires an operationally identified candidate component learned on a disjoint probe/discovery split. The target, intervention layer, strength and norm matching are fixed before evaluation. Within a matched problem pair, later evidence either requires that candidate or makes it irrelevant. Selective target damage must exceed damage from unrelated and norm-matched random components; sham executes the same intervention code without removing information. Add-back restores the same component at the same site, not an extra answer-bearing hint.

The auxiliary entropy comparison distinguishes viable candidates, already-excluded candidates, norm-matched noise and a temperature control. Interpretation requires matching total entropy and perturbation magnitude where feasible, together with independent checks of candidate validity. High entropy alone does not imply useful multi-path retention.

Alternative set	Quality
Viable alternatives	74.7
Excluded alternatives	67.0
Matched-entropy noise	66.0
Temperature control	65.4

Task-quality contrasts distinguish viable alternatives from excluded candidates, matched-entropy noise, and temperature controls. Entropy matching and target specificity are separate conditions for interpreting selective candidate retention.

Route–intervention interaction. The route–intervention analysis crosses route choice with sham, targeted erasure and add-back on the same pre-action task state, comparing route-specific effects and their paired difference-in-differences; an ordinary intervention main effect cannot explain the hybrid advantage. The target-erasure interaction is -8.4 points, with interval [-11.2, -5.7]; the rescue-versus-sham interaction is -4.8. Randomization of K and D changes task demands, but excessive difficulty can produce floor effects. Interpreting the $K \times D$ grid therefore requires absolute pure-mode accuracy and uncertainty in addition to the L–E contrast and routing response. Dwell-time changes are useful descriptive evidence, not proof of a candidate-count code in the representation.

Commitment-density and delayed exact-reuse results are consistent with stable explicit commitment; they do not, by themselves, locate a particular value representation or demonstrate its causal use. Likewise, quality–cost gains alone do not establish selective path use. Mechanism conclusions are restricted to effects supported by the corresponding intervention controls.

Boundary-free RL. The boundary-free comparison removes boundary supervision while retaining comparable initial content capability and rollout expenditure, and evaluates boundary agreement on independent annotated states. Agreement is meaningful only alongside action-value tests; rediscovering annotation boundaries is not itself better reasoning.

Boundary supervision	Quality	Boundary agreement
Boundary-supervised RL	69.4	72.1
No-boundary RL	67.8	63.3

Entries are task-quality and boundary-agreement percentages. Higher agreement without improved task quality or cost does not establish useful rediscovery of switching boundaries.