SMART: SYSTEMATIC MEASUREMENT AND ADAPTIVE ROUTING OF THINKING FOR LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

How much should an LLM agent *think* before it acts? We present SMART (Systematic Measurement and Adaptive Routing of Thinking), a study that treats the per-step thinking budget—the maximum number of reasoning tokens allowed per step—as an economic quantity to be measured and optimally spent. Across 11,952 evaluation episodes on three agentic domains, budget sensitivity proves environment-dependent: success–budget curves saturate quickly in short-horizon dialogue but keep scaling in long-horizon code execution. We trace the pathology of mid-range budgets to a *truncation tax*: reasoning cut off mid-thought degrades into garbled steps that pollute the context. The pollution inflates later steps' token use by 13.2% on average (9–31% across domains), making intermediate-high budgets the least favorable trade on the grid. A 1.7B router trained on successful trajectories is Pareto-optimal in every domain—beating the pre-specified compute-optimal budget at 48.5% (airline) and 26.5% (retail) fewer reasoning tokens, and reaching the Pareto frontier in AppWorld, where success keeps scaling with budget—and the optimal allocation granularity matches the domain's interaction horizon: per-step in dialogue, per-task in long-horizon execution. Finally, we examine what paid-for thinking contains: budget vigorously reshapes what agents think about, yet no behavior predicts success positively—backtracking instead marks struggle, not failure. What budget buys is not a different mix of thoughts but intact ones.

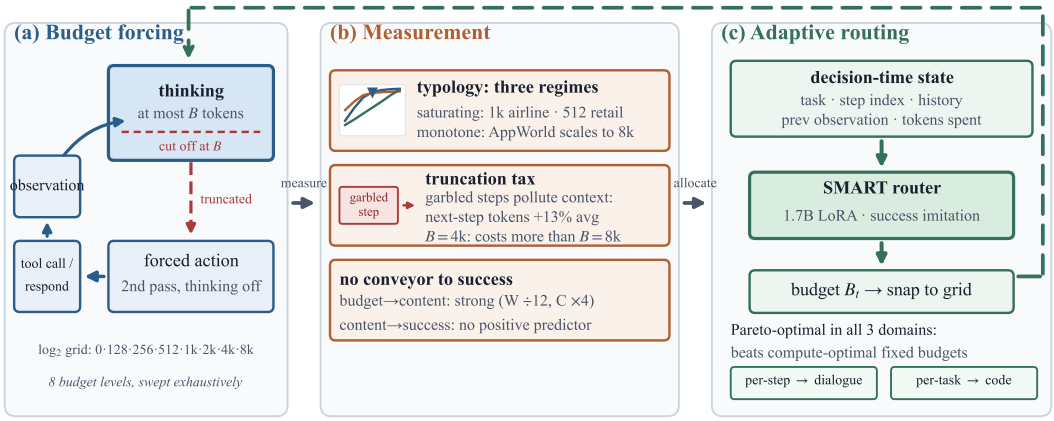

Figure 1: Overview of SMART. **(a) Budget forcing:** each agent step reasons with at most B tokens before being forced to act. **(b) Measurement:** success–budget curves form three regimes; truncation pollutes context; reasoning content responds to budget without predicting success. **(c) Adaptive routing:** a LoRA router sets budgets from decision-time state, is Pareto-optimal in all three domains, and matches granularity to interaction horizon.

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.

1 INTRODUCTION

Explicit reasoning has become a central lever for improving LLM problem-solving. Chain-of-thought prompting showed that eliciting intermediate steps substantially raises accuracy on complex tasks (Wei et al., 2022), and interleaving such reasoning with environment actions yields effective tool-use agents (Yao et al., 2023; Schick et al., 2023). A subsequent generation of reasoning models—OpenAI o1 (OpenAI, 2024), DeepSeek-R1 (Guo et al., 2025), and the Qwen3 family (Yang et al., 2025)—made long internal deliberation a default mode of operation, and test-time compute scaling studies confirmed that more reasoning tokens often buy more accuracy (Snell et al., 2025; Muennighoff et al., 2025).

Yet in agentic settings, reasoning is not free and not always beneficial. Each deliberation step costs tokens, wall-clock time, and—as we show—can actively damage the context when the reasoning itself degrades. Practitioners currently choose a thinking budget by folklore: enable thinking and hope, or cap it at an arbitrary value. This paper asks a sharper question: *what is the price of thought in LLM agents, and can a cheap controller learn to pay it optimally?* Figure 1 overviews our pipeline.

We answer in three parts. First, measurement. Test-time compute studies measure single-turn problem solving, where longer chains saturate and can even hurt through overthinking (Snell et al., 2025; Chen et al., 2025; Cuadron et al., 2025); whether these conclusions transfer to multi-step agents—where each reasoning step inherits the text of all previous ones—is unknown. Sweeping the per-step budget across a log grid in three agentic domains, we find that the success–budget relationship is not one curve but a *typology of regimes*—strongly saturating, weakly saturating, and monotone scaling—tracking the domain's interaction structure rather than the budget knob alone.

This typology raises a mechanistic question: why should more thinking ever hurt? We trace the answer to a *truncation tax*: reasoning cut off mid-thought degrades into garbled steps that pollute the context, making intermediate-high budgets the least favorable trade on the grid. A behavior-level analysis points the same way from the content side: budget massively reshapes what agents think about, yet what they think about does not predict whether they succeed. What budget buys is not a different mix of thoughts but intact ones. Unlike observational dissections of long chains (Marjanović et al., 2026; Wang et al., 2025b) and injected-distractor studies (Shi et al., 2023; Laban et al., 2026), the pollutant here is *endogenous*—the agent's own truncated thoughts—so its downstream cost can be measured against a placebo control.

These findings suggest treating the budget as an economic decision and learning to pay it. Existing routers choose *which model* answers a query (Ong et al., 2025; Ding et al., 2024) or executes a reasoning step (Kapoor et al., 2026), and adaptive-reasoning methods choose *whether* to think (Fang et al., 2025; Zhang et al., 2025); none prices *how much* a single model thinks at each step. We train a 1.7B-parameter LoRA router on successful trajectories; it is Pareto-optimal in all three domains, matching the strongest fixed budgets at up to 69% fewer reasoning tokens. The optimal granularity of allocation matches the domain's interaction horizon: per-step in short-horizon dialogue, per-task in long-horizon execution.

Contributions. (i) A budget-controlled measurement across three agentic domains, establishing a three-regime typology of budget sensitivity (§3); (ii) the *truncation tax*: mid-thought truncation, garbled steps, context pollution, and non-monotone token costs (§4); (iii) a lightweight LoRA router that is Pareto-optimal in every domain, with a granularity-matching law (§5); (iv) a two-link behavior analysis: budget reshapes reasoning content, but content incidence does not predict success (§6).

2 RELATED WORK

Test-time compute and thinking budgets. Scaling test-time compute improves accuracy on reasoning benchmarks (Snell et al., 2025; OpenAI, 2024; Guo et al., 2025), and budget forcing—hard caps on reasoning length—has been proposed as a cheap control knob (Muennighoff et al., 2025). Several works warn of overthinking: excessively long chains can hurt accuracy and efficiency (Chen et al., 2025; Cuadron et al., 2025). These studies concern single-turn problem solving. We study

multi-step agents, where reasoning is interleaved with environment feedback and degraded reasoning propagates through the context.[1]

Reasoning–acting trade-offs in agents. ReAct-style interleaving of thought and action is the dominant agent paradigm (Yao et al., 2023). Benchmarks such as τ-bench (Yao et al., 2025) and AppWorld (Trivedi et al., 2024) measure multi-turn tool use with database-backed state. Little is known about how per-step thinking budgets shape agent trajectories end-to-end; we provide what is, to our knowledge, the first budget-controlled measurement of per-step thinking budgets in multi-step tool-use agents, at a scale of 11,952 episodes over 8 budget levels in 3 domains.

LLM routing. A growing literature routes *queries* among models of different cost and capability (Ong et al., 2025; Ding et al., 2024; Hu et al., 2024; Feng et al., 2025). Two recent directions move the routing decision off the query level: TRIM routes individual reasoning *steps* between weak and strong models (Kapoor et al., 2026), and adaptive-reasoning methods train a single model to choose *whether* to think at all (Fang et al., 2025; Zhang et al., 2025). SMART routes a complementary control variable: *how much* to think, via the per-step token budget of a single model, decided at the granularity of individual agent steps. These axes compose rather than compete—a system may route a query to a model and then price each step of the resulting trajectory.

Behavior analysis of reasoning traces. Process-level analyses of reasoning traces (Lightman et al., 2024; Huang et al., 2024) increasingly rely on LLM judges to label reasoning content. Our behavior audit (§6) adds a cautionary datapoint: the statistical object matters as much as the judge—count-based incidence models avoid the leverage artifacts that share-based regressions produce on rare behaviors. We therefore report behavior analyses under a blinded frontier-model judge with incidence modeling.

3 Measuring the Reasoning–Acting Trade-off

3.1 Setup

Budget control. The *thinking budget* B—hereafter simply the *budget*—is the maximum number of reasoning tokens allowed per agent step. We implement exact budget forcing with a two-stage protocol, adapting the budget-forcing recipe of Muennighoff et al. (2025) to agent steps: (1) generate reasoning with `max_tokens` $= B$ and stop token `</think>`; (2) if truncated, feed the truncated reasoning back with thinking disabled to force an action. $B{=}0$ disables thinking entirely. The agent decodes greedily (temperature 0) so that budget is the only manipulated variable.

Experimental grid. One backbone, Qwen3.6-35B-A3B (Qwen Team, 2026b), a thinking-enabled sparse MoE model (35B total, 3B active parameters) released April 2026 as part of the Qwen3 family (Yang et al., 2025). Three domains: τ-bench airline (50 tasks) and retail (115 tasks) (Yao et al., 2025), and AppWorld test-normal (168 tasks) (Trivedi et al., 2024). Budget grid $\{0, 128, 256, 512, 1024, 2048, 4096, 8192\}$; 6 seeds on τ-bench (7,920 episodes), 3 seeds on AppWorld (4,032 episodes). Stochasticity enters through the user simulator—a separate Qwen3.5-4B instance (Qwen Team, 2026a) in non-thinking mode at temperature 0.4, used on τ-bench (Appendix D)—and the environment. Episodes are capped at 30 agent steps on τ-bench and 40 on AppWorld.

3.2 A typology of budget sensitivity

Figure 2 shows the central measurement. Airline rises steeply from 36.0% ($B{=}0$) to a 50.3% plateau at $B{\in}\{1024, 2048\}$. It then dips to 45.3% at $B{=}4096$ and partially recovers to 48.0% at $B{=}8192$. Retail follows the same shape with much smaller amplitude (38.6% $\rightarrow$ 45.4% at $B{=}512$, flat thereafter). AppWorld never saturates within our grid: success climbs from 20.6% at $B{=}128$ to 39.3% at $B{=}8192$, its highest point. (6 of 504 AppWorld $B{=}8192$ episodes are conservatively imputed as failures after a serving incident; impact $\leq$1.2pp, Appendix H.) We define the compute-optimal

[1] Additional related work on concise, length-controlled, and cost-aware inference is reviewed in Appendix A.

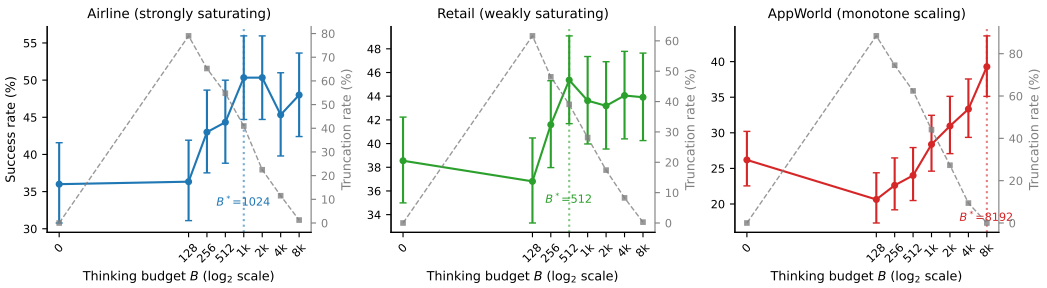

Figure 2: Success rate (solid, Wilson 95% CI) and reasoning-truncation rate (dashed gray) vs. the per-step thinking budget. The three domains instantiate three regimes: strongly saturating (airline), weakly saturating (retail), and monotone scaling (AppWorld). Truncation rate falls with budget and co-moves with the success plateau.

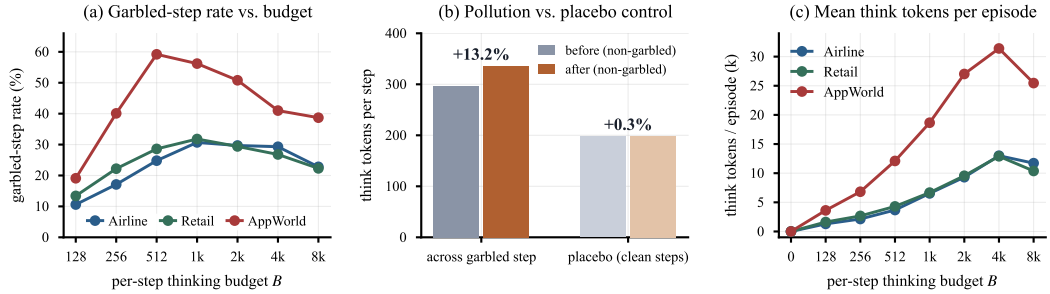

Figure 3: The truncation tax. (a) Garbled-step rate is non-monotone in budget, peaking in the mid-high range (all three domains). (b) Context pollution with placebo control (both conditions pool airline/retail/AppWorld): think tokens of non-garbled steps before vs. after a garbled step (real), and across three consecutive clean steps (placebo); the zero-centered placebo rules out a position artifact (statistics in the text). (c) Mean total reasoning tokens per episode vs. budget: at B=4096 agents pay more than at B=8192 in all three domains—truncated thinking is re-paid later.

budget B^* as the smallest budget whose *observed* success rate reaches 95% of the grid maximum: B^*=1024 (airline) and B^*=512 (retail), whereas AppWorld peaks at the grid edge, so its optimum lies beyond the grid. Parametric fits—a saturating family wins by BIC in all three domains—and the fitted curves for all three domains appear in Appendix C.

The airline dip (-5.0pp from the plateau) is directionally consistent with a localized overthinking regime: a paired task-level test gives p=0.109 (two-sided Wilcoxon) and p=0.035 (one-sided paired t). We therefore treat the dip as suggestive rather than definitive.

Budget sensitivity tracks the domain's interaction regime. Task-level analysis shows the optimal budget varies widely within a domain: on airline, the per-task success-maximizing budget ranges over the entire grid (10 of 50 tasks are best at B=0), and correlates only weakly with task difficulty (the task's success rate at B=0). Meanwhile the *shape* of the curve is stable across seeds. What differs across domains is the regime—saturation point, amplitude, and monotonicity—which tracks the interaction structure: short-horizon dialogue with dense feedback (airline/retail, 11–17 mean steps) saturates, while long-horizon code execution against a persistent state (AppWorld, 28–34 steps) keeps scaling.

4 The Truncation Tax

Why should mid-range budgets be wasteful? We trace the pathology to a single mechanism with three observable stages (Figure 3).

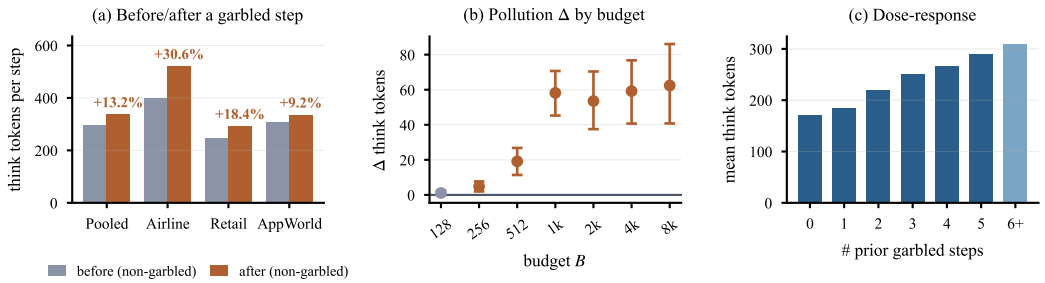

Figure 4: Context pollution, decomposed (all panels pool airline/retail/AppWorld). (a) Within-episode paired comparison of think tokens of the nearest non-garbled steps before vs. after a garbled step (means). (b) Pollution Δ by budget with 95% episode-cluster bootstrap CIs: insignificant at B=128 (gray), significantly positive from B=256 onward (orange). (c) Dose–response: mean think tokens of non-garbled steps grow with the number of prior garbled steps in the same episode.

Stage 1: budget pressure degrades reasoning. We flag garbled steps with three automatic criteria (character/token ratio, CJK fraction, repetition; thresholds and a short-thought guard in Appendix F). Garbled rates are non-monotone in B (Fig. 3a): lowest at B=128 (10.6–19.1% across domains; thoughts too short to derail), peaking at 24.8–59.2% for $B \in [512, 2048]$, where truncation cuts substantive reasoning mid-thought, and receding to 22.3–38.7% at B=8192, where step-level truncation is rare (0–1.7%).

Stage 2: garbled steps pollute the context. Within an episode we pair each garbled step with its nearest non-garbled neighbours before and after: steps immediately after a garbled step spend 13.2% more tokens than their paired predecessors (n=56,011 transitions pooled over the three domains; 9–31% across domains; Figs. 3b, 4a). The same pairing across three consecutive clean steps finds no inflation (placebo: n=63,091 pooled; $\Delta = +0.6$ tokens, 95% CI $[-1.4, +2.5]$; per-domain Δ: Airline -2.1%, Retail $+0.2\%$, AppWorld $+1.9\%$), ruling out a position artifact. The per-budget Δ is significantly positive from B=256 upward and grows into the high range (Fig. 4b), and token use grows monotonically with the number of prior garbled steps ($+82\%$ from none to six or more; Fig. 4c): the agent re-derives, re-checks, and compensates for corrupted earlier reasoning.

Stage 3: the tax on the total bill. The compound effect is a non-monotone token cost (Fig. 3c): mean total reasoning tokens *fall* from B=4096 to B=8192 (airline: 13.0k→11.7k; retail: 12.9k→10.4k; AppWorld: 31.4k→25.5k), because at B=8192 thoughts complete naturally and nothing is re-paid. At the observed grid points B=8192 therefore offers a more favorable token–success trade-off than B=4096 in airline and AppWorld (higher cost, lower success at B=4096: 45.3% vs. 48.0%; 33.3% vs. 39.3%), and in retail is 20% cheaper at tied success. **Intermediate-high budgets are the least favorable observed trade on the grid.**

5 Adaptive Routing as Optimal Payment

§3–§4 establish that the right budget is domain-dependent, task-dependent, and expensive to get wrong. We therefore learn to allocate it.

5.1 Router

A Qwen3-1.7B LoRA adapter (rank 16) with a continuous regression head predicts the per-step budget, trained on steps of successful episodes with the label set to the step's own realized thinking length clipped up to the budget grid, $\ell_{e,s} = \min\{b \in \mathcal{B} : b \geq r_{e,s}\}$ (success imitation). The primary label rule is pre-specified, not selected from held-out router outcomes: the least restrictive nonzero seed-success threshold (sr $1/6$ for τ-bench, sr $1/3$ for AppWorld), which maximizes the available successful-trajectory pool. Inputs use only pre-decision information: task description, step index, history summary, previous observation, and accumulated thinking tokens. At inference the

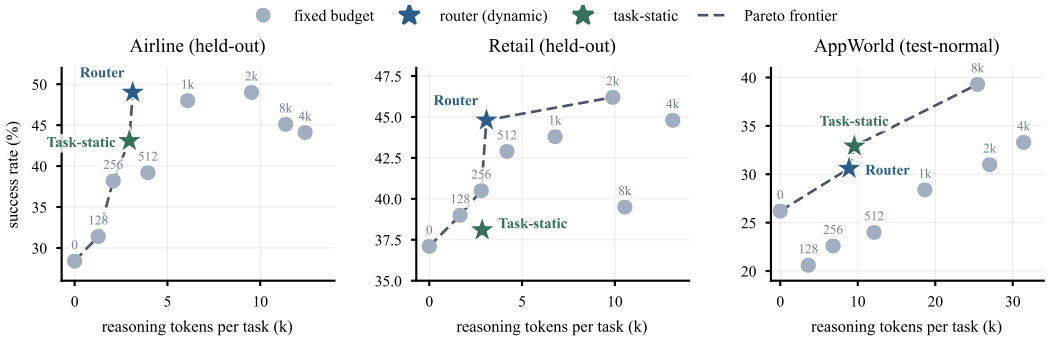

Figure 5: Success rate vs. reasoning tokens per task on held-out tasks. Gray: fixed budgets; stars: learned policies. Dashed line: Pareto frontier. In airline and retail the dynamic router sits above and to the left of the pre-specified compute-optimal B^*; in AppWorld both learned policies lie on the frontier and dominate interior fixed budgets, with the frontier running from $B{=}0$ through them to $B{=}8192$ (numbers in §5).

router scores each step in $<0.5\%$ of step wall-clock. Full training configuration and a worked input example appear in Appendix G.

5.2 The router is Pareto-optimal in every domain

We measure cost by T_{task}, the mean reasoning tokens per task, and take the Pareto frontier to be the policies undominated in the $(\text{SR}, T_{\text{task}})$ plane (dominance: weakly better on both coordinates, strictly on one). Figure 5 plots success vs. T_{task} for all policies on held-out tasks, with paired bootstrap comparisons. Primary fixed-budget comparisons use the pre-specified B^* of §3; the highest-SR fixed setting on the router held-out set is a descriptive secondary reference.

Airline. The dynamic router exceeds $B^*{=}1024$ (49.0% vs. 48.0%) at 48.5% fewer reasoning tokens (3.1k vs. 6.1k per task); descriptively, it also matches the held-out highest-SR fixed setting, $B{=}2048$ (49.0%), at 67% fewer tokens. The saving is realized in wall-clock, not only in token accounting: -43% per-task wall-clock at statistically indistinguishable step counts under matched serving, so savings come from cheaper steps rather than shorter episodes (Appendix E).

Retail. The router (44.8% SR at 3.1k tokens) exceeds $B^*{=}512$ (42.9% at 4.2k) at 26.5% fewer tokens, directly dominates $B{\in}\{512, 1024, 4096, 8192\}$, and is statistically indistinguishable from the held-out highest-SR fixed setting $B{=}2048$ (46.2%; paired $\Delta{-}1.4$pp, 95% CI $[-8.6, +6.2]$) at 69% fewer tokens.

AppWorld. Both the dynamic (30.6% at 8.9k) and task-static (32.9% at 9.6k) policies lie on the Pareto frontier: task-static directly dominates $B{=}1024$ (28.4% at 18.7k; paired $+4.6$pp, 95% CI $[+0.6, +8.7]$) and $B{=}2048$ (31.0% at 27.0k), whereas dynamic supplies a lower-token frontier trade-off. **In all three domains, the learned policy set contains an undominated, Pareto-optimal policy.**

5.3 Granularity matching: when does per-step routing pay?

The dynamic-vs-static contrast flips sign across domains. On airline (mean 12 steps), dynamic beats task-static by $+5.9$pp (49.0 vs. 43.1) and beats a shuffled control—same per-episode budget multiset, state matching destroyed—by $+3.9$pp (49.0 vs. 45.1): *where* budget is spent matters, not just how much. On AppWorld (mean 29 steps), task-static beats dynamic by $+2.4$pp (32.9 vs. 30.6) at only 7.6% more tokens. Figure 6 shows why: the airline router escalates budget over an episode (Spearman $\rho{=}{+}0.20$) with decisive, state-dependent switches; the AppWorld router switches budget on over 40% of steps while concentrating on small budgets, injecting depth incoherence into a long reasoning chain where each step builds on all previous ones. **The optimal granularity of budget**

Table 1: Routing results on held-out tasks (task×seed paired). T_{task}: mean reasoning tokens per task. Airline: 17 tasks × 6 seeds; retail: 35 × 6; AppWorld: 168 × 3. Reference policies as defined in §5; Airline additionally reports the stricter sr $4/6$ training-threshold router variant used in the training-threshold ablation (§5.4). Complete six-threshold sweep for both domains in Appendix B.

Domain	Policy	SR (%)	T_{task}	vs. primary reference
Airline	Router (dynamic)	**49.0**	**3,132**	-48.5% vs. B^*
	Router (sr $4/6$ training)	49.0	3,497	$+11.7\%$ tokens vs. sr $1/6$
	Task-static	43.1	2,952	
	Shuffled	45.1	3,201	-3.9pp vs. router
	B^*=1024 (primary ref.)	48.0	6,086	
	held-out high-SR B=2048	49.0	9,528	
Retail	Router (dynamic)	**44.8**	**3,079**	-26.5% vs. B^*
	Task-static	38.1	2,850	
	Shuffled	36.2	2,818	-8.6pp vs. router
	B^*=512 (primary ref.)	42.9	4,188	
	held-out high-SR B=2048	46.2	9,887	
AppWorld	Task-static	**32.9**	**9,557**	-62% vs. B=8192 frontier
	Router (dynamic)	30.6	8,879	
	B=8192 (best fixed)	39.3	25,458	
	B=1024	28.4	18,656	

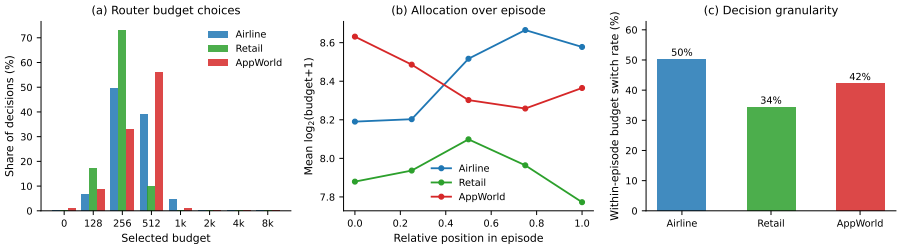

Figure 6: Router policy behavior. (a) Budget choices concentrate at 128–512 in all domains—many small clean thoughts rather than few large truncatable ones. (b) The airline router escalates budget within an episode; the AppWorld router does the opposite. (c) Within-episode switch rate: high everywhere, yet state matching pays only on short horizons (§5.3).

allocation matches the domain's interaction horizon: per-step for short-horizon dialogue, per-task for long-horizon execution.

5.4 Ablations

Two ablations carry the argument (Table 1): a quantity–quality trade-off in the training-threshold choice—the broader sr $1/6$ rule and the stricter sr $4/6$ rule both reach 49.0% airline SR, at 3,132 vs. 3,497 tokens—and a state-matching effect isolated by a shuffled control, which costs -3.9pp (45.1% vs. 49.0%).

6 What Does Thinking Buy? Behavior-level Analysis

The truncation tax (§4) explains how budget inflates the token bill mechanically. A prima facie competing hypothesis is that budget works through *content*: thinking longer helps because agents plan more, verify more, calculate more, or wander less. The content hypothesis has two links: budget must first change what agents think about, and those changes must then predict success. We test each link separately.

Setup. Let $\mathcal{L}_{e,s} \subseteq \{P, V, B, W, C\}$ be the label set of sentence s in episode e, and let N_e be the episode's sentence count (the exposure). Write $n_{e,L} = \sum_{s=1}^{N_e} \mathbf{1}[L \in \mathcal{L}_{e,s}]$ for the episode-level count of behavior L. The content hypothesis requires **both** a nonzero dose response in

$$\log \mathbb{E}[n_{e,L}] \;=\; \log N_e + \sum_b \delta_{L,b}\, \mathbf{1}[\text{budget}_e = b] \qquad \text{(link i: budget} \rightarrow \text{behavior)} \qquad (1)$$

Table 2: Link (i): behavior incidence by budget, pooled over the two τ-bench domains (airline and retail; rate ratios vs. B=128, task-cluster bootstrap 95% CI; $^{*}p<0.1$, $^{**}p<0.05$, $^{***}p<0.01$). Per-domain rate ratios (Table 7) and raw incidence rates (Table 8) are in Appendix I.

Label	B256	B512	B1024	B2048	B4096	B8192
Wander	1.00	0.58^{*}	0.29^{***}	0.07^{***}	0.11^{***}	0.08^{***}
Calc	2.24^{**}	3.80^{***}	3.04^{***}	2.64^{***}	2.79^{***}	4.00^{***}
Backtrack	1.53^{*}	1.92^{***}	1.83^{***}	1.82^{***}	1.93^{***}	2.07^{***}
Plan	1.16^{***}	1.14^{***}	1.20^{***}	1.11^{***}	1.15^{***}	1.07^{**}
Verify	0.93^{***}	0.90^{***}	0.86^{***}	0.88^{***}	0.85^{***}	0.89^{***}

and a nonzero outcome association in

$$\log \mathbb{E}[n_{e,L}] \;=\; \log N_e + \beta_L \cdot \mathbf{1}[\text{success}_e] + \sum_b \gamma_{L,b}\, \mathbf{1}[\text{budget}_e = b] \qquad \text{(link ii: behavior} \to \text{success)} \tag{2}$$

for at least one label L. Both are weighted Poisson incidence models with budget fixed effects (pooled fits add benchmark fixed effects): $\log N_e$ is an exposure offset, episode-level inverse-probability weights account for the sampling design, and task-cluster bootstrap yields confidence intervals. We report incidence rate ratios $\text{IRR} = \exp(\delta)$ and $\exp(\beta)$. Note the deliberate choice of statistical object: *incidence* (counts with exposure offset) rather than behavior *shares*—three of the five behaviors occur in only 0.1–2.3% of sentences (pooled rates), where share-based slopes are leverage-driven and unstable. A self-contained expansion of both models—the offset algebra, reference-cell coding, weighting, clustering, and worked examples—appears in Appendix J.

6.1 Blinded frontier-model labeling

We label every reasoning sentence with a five-category multi-label scheme (Plan/Verify/Backtrack/Wander/Calc) using Claude Opus 5 as the judge under a blinded protocol: a stratified 10% sample of complete episodes covering all 28 benchmark×budget×outcome cells (687 episodes, 79,527 sentences, 160 task clusters; sampling design, prompt, and protocol in Appendix I). The judge sees only anonymized sentence windows—never budget or outcomes. Against the same model at high reasoning effort, the low-effort judge returns the identical label set on 92% of sentences in a paired 100-sentence pilot (mean Jaccard 0.945). Sentences inside garbled steps (§4) are excluded by the labeling frame, so the labels below describe intact reasoning only.

6.2 Link (i): budget reshapes behavior composition

The first link is strong: budget massively reshapes what agents think about. Table 2 reports $\exp(\hat{\delta}_{L,b})$ from Eq. 1 (reference B=128, $\delta_{L,128} = 0$). Raising the budget from 128 to 8192 tokens collapses wandering twelve-fold (2.06→0.17 per 100 sentences), raises calculation 4.4-fold (0.52→2.29), and doubles backtracking (0.73→1.54); planning rises modestly and verification declines modestly. The response is graded across domains in the same ordering as §3's typology: in budget-sensitive airline, calculation incidence rises from near zero to 4.9 per 100 sentences and backtracking triples; in budget-insensitive retail, the calculation response is muted and the endpoint contrast is not significant (0.66→0.86, +30%; per-domain ratios in Table 7). *The behavior response is strongest exactly where budget matters most.*

6.3 Link (ii): behavior composition does not predict success

The second link is broken. Table 3 reports $\exp(\hat{\beta}_L)$ from Eq. 2. Planning and verification—by far the most common behaviors—are precisely null: successful and failed episodes plan and verify at identical per-sentence rates (pooled IRR 0.97 $[0.93, 1.02]$ and 1.01 $[0.97, 1.04]$), excluding even small content effects for the two behaviors that dominate reasoning. Wandering and calculation show no association. The only signal is backtracking: failed episodes backtrack at $1/0.76 \approx 1.3\times$ the rate of successful ones (p=0.004, driven by retail). Two probes show it is a *symptom of struggle, not a cause of failure*: (i) decomposing $\mathbb{E}[n_{e,L}] = P(n_{e,L} > 0) \cdot \mathbb{E}[n_{e,L} \mid n_{e,L} > 0]$, the association lives entirely in the first factor—among episodes that backtrack at all, intensity does not differ (IRR 0.96

Table 3: Link (ii): incidence rate ratio (success vs. failure) of each behavior, adjusted for budget fixed effects (the pooled column adds benchmark fixed effects); task-cluster bootstrap 95% CI. $^{**}p<0.01$.

Label	Pooled	Airline	Retail
Plan	0.97 [0.93, 1.02]	0.98 [0.89, 1.08]	0.97 [0.93, 1.02]
Verify	1.01 [0.97, 1.04]	1.05 [0.98, 1.12]	0.98 [0.95, 1.01]
Backtrack	**0.76**** [0.61, 0.92]	0.84 [0.62, 1.12]	**0.71**** [0.56, 0.89]
Wander	0.95 [0.56, 1.52]	1.06 [0.43, 2.29]	0.90 [0.49, 1.62]
Calc	0.82 [0.50, 1.36]	0.71 [0.26, 1.55]	1.05 [0.68, 1.68]

$[0.80, 1.16]$); (ii) restricted to the first three steps, before any difficulty has occurred, backtracking has no predictive power (IRR 0.71 $[0.46, 1.11]$, p=0.16), and its incidence is highest in stuck-type failures that exhaust the step limit (2.14% vs. 1.14% in successes; Table 9).

Taken together, the behavior channel is a *mirror of the budget, not a conveyor to success*: composition responds to budget vigorously and monotonically, yet none of it reaches the outcome. What remains is the integrity channel of §4.

7 Applicability Boundary and Limitations

Applicability boundary. The router's success edge is largest where budget matters (airline) and degrades to a pure cost edge—no detectable success penalty—where it matters less (retail; §5). On AppWorld the learned policies trade 6.4–8.7pp of success against 62–65% token savings relative to B=8192—a Pareto-optimal trade, but not a free lunch. Task-level oracle analysis bounds the remaining headroom: defining $\mathrm{SR}_{\text{oracle}} = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \max_b \widehat{\mathrm{SR}}(t,b)$ (each task evaluated at its own best fixed budget), the oracle yields 63.3% (airline), 62.6% (retail), 55.0% (AppWorld)—14–16pp above any realized policy. No current router variant closes this gap; per-task difficulty (the task's success rate at B=0) predicts which fixed budget is best for a task only weakly on τ-bench and modestly on AppWorld (Spearman $\rho = -0.08$ airline, -0.18 retail, -0.29 AppWorld; easier tasks peak at smaller budgets, Appendix C), so the headroom is not cheaply accessible from difficulty alone.

Limitations: scope. The study uses one backbone and one router size, so the regime typology is a statement about domains *under this model*; separating domain effects from model×domain interactions requires cross-architecture validation.

Limitations: measurement and data. Garbled steps are flagged by automatic heuristics. Behavior analyses rely on a frontier-model silver reference (92% label-set agreement with a high-effort configuration of the same judge, Appendix I) over a 10% episode sample: the design is precisely powered for the common behaviors (Verify null within ±4%, Plan within ±7%) but excludes only large effects for the rare ones, and two audited episodes (0.29%) were excluded for unrecoverable API failures. AppWorld's B=8192 cell includes a small number of imputed failures after a serving incident (Appendix H). Wall-clock figures are environment-dependent; we report them only where serving loads are matched (Appendix E).

8 Conclusion

Thinking budgets in LLM agents obey economics, not monotonicity. Budget sensitivity tracks the interaction regime: short-horizon dialogue saturates, whereas long-horizon AppWorld continues to scale. Mid-thought truncation pollutes context, and the observed grid points place B=4096 at a less favorable token–success trade-off than B=8192 in airline and AppWorld. A 1.7B router exceeds the compute-optimal B^* with 48.5% fewer tokens on airline and 26.5% fewer on retail, also matching or statistically tying the respective held-out high-SR fixed references at 67% and 69% lower cost; the learned policy set traces AppWorld's Pareto frontier at task granularity. Behavior content tracks budget but does not predict success. SMART turns "how much should the agent think?" from intuition into a design decision.

AI USE STATEMENT

We used generative AI tools for code assistance, draft text editing, and data-analysis assistance. We have not used generative AI tools for generating experimental results or for the core research ideas, and human-subject research is not applicable to this work. All AI-assisted outputs were verified by the authors against the raw experimental data; every number in the paper traces to a reproducible analysis artifact. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only public benchmarks and does not involve human subjects or private data. Adaptive budget allocation reduces inference cost and energy consumption; we do not foresee harmful applications beyond those of general-purpose LLM agents.

REPRODUCIBILITY STATEMENT

All results derive from a layered pipeline (raw traces $\rightarrow$ episodes/steps $\rightarrow$ analysis) with fixed seeds and greedy agent decoding; analysis modules read only the standardized intermediate files. We will release the trace schemas, analysis code, router checkpoints, and the task manifests for held-out splits upon publication; the full episode-level aggregates needed to reproduce every table and figure are archived with the submission as supplementary material.

REFERENCES

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. In *Conference on Language Modeling (COLM)*, 2025.

Daman Arora and Andrea Zanette. Training language models to reason efficiently. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Simon A. Aytes, Jinheon Baek, and Sung Ju Hwang. Sketch-of-Thought: Efficient LLM reasoning with adaptive cognitive-inspired sketching. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 24296–24320, 2025. doi: 10.18653/v1/2025.emnlp-main.1236.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, et al. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, et al. Why do multi-agent LLM systems fail? In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do NOT think that much for 2+3=? on the overthinking of long reasoning models. In *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 9487–9499, 2025.

Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, et al. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *arXiv preprint arXiv:2502.08235*, 2025.

Mehul Damani, Idan Shenfeld, Andi Peng, Andreea Bobu, and Jacob Andreas. Learning how hard to think: Input-adaptive allocation of LM computation. In *International Conference on Learning Representations (ICLR)*, 2025.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid LLM: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations (ICLR)*, 2024.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: LLM learns when to think. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Tao Feng, Yanzhen Shen, and Jiaxuan You. GraphRouter: A graph-based router for LLM selections. In *International Conference on Learning Representations (ICLR)*, 2025.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-budget-aware LLM reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 24842–24855, 2025. doi: 10.18653/v1/2025.findings-acl.1274.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. RouterBench: A benchmark for multi-LLM routing system. *arXiv preprint arXiv:2403.12031*, 2024.

Jie Huang, Xinyun Chen, Swaroop Mishra, et al. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations (ICLR)*, 2024.

Vansh Kapoor, Aman Gupta, Hao Chen, Anurag Beniwal, Jing Huang, and Aviral Kumar. TRIM: Hybrid inference via targeted stepwise routing in multi-step reasoning tasks. In *International Conference on Learning Representations (ICLR)*, 2026.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. LLMs get lost in multi-turn conversation. In *International Conference on Learning Representations (ICLR)*, 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, et al. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics (TACL)*, 12:157–173, 2024.

Wenjie Ma, Jingxuan He, Charlie Snell, et al. Reasoning models can be effective without thinking. *arXiv preprint arXiv:2504.09858*, 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, et al. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Sara Vera Marjanović, Arkil Patel, Vaibhav Adlakha, Milad Aghajohari, Parishad BehnamGhader, Mehar Bhatia, et al. DeepSeek-R1 Thoughtology: Let's think about LLM reasoning. *Transactions on Machine Learning Research (TMLR)*, 2026.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 20275–20321, 2025. doi: 10.18653/v1/2025.emnlp-main.1025.

Sania Nayab, Giulio Rossolini, Marco Simoni, Andrea Saracino, Giorgio Buttazzo, Nicolamaria Manes, and Fabrizio Giacomelli. Concise thoughts: Impact of output length on LLM reasoning and cost. *Information Sciences*, 757:123939, 2026. doi: 10.1016/j.ins.2026.123939.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *International Conference on Learning Representations (ICLR)*, 2025.

OpenAI. Learning to reason with LLMs. https://openai.com/index/learning-to-reason-with-llms/, 2024.

Yuxiao Qu, Matthew Y. R. Yang, Amrith Setlur, Lewis Tunstall, Edward Emanuel Beeching, Ruslan Salakhutdinov, and Aviral Kumar. Optimizing test-time compute via meta reinforcement finetuning. In *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 50893–50925, 2025.

Qwen Team. Qwen3.5-4B. https://huggingface.co/Qwen/Qwen3.5-4B, 2026a. Open-source instruction-tuned model; used here as the τ-bench user simulator.

Qwen Team. Qwen3.6-35B-A3B. https://huggingface.co/Qwen/Qwen3.6-35B-A3B, April 2026b. Open-source sparse MoE reasoning model; 35B total, 3B active parameters.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31210–31227, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations (ICLR)*, 2025.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. Stop overthinking: A survey on efficient reasoning for large language models. *Transactions on Machine Learning Research (TMLR)*, 2025.

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 (ACL)*, pp. 16022–16076, 2024. doi: 10.18653/v1/2024.acl-long.850.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations (ICLR)*, 2025a.

Xuezhi Wang, Jason Wei, Dale Schuurmans, et al. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Yue Wang, Qiuzhi Liu, Jiahao Xu, et al. Thoughts are all over the place: On the underthinking of long reasoning models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models. *arXiv preprint arXiv:2408.00724*, 2024.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. TokenSkip: Controllable chain-of-thought compression in LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3351–3363, 2025. doi: 10.18653/v1/2025.emnlp-main.165.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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 (ICLR)*, 2023.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations (ICLR)*, 2025.

Murong Yue, Wenlin Yao, Haitao Mi, Dian Yu, Ziyu Yao, and Dong Yu. DOTS: Learning to reason dynamically in LLMs via optimal reasoning trajectories search. In *International Conference on Learning Representations (ICLR)*, 2025.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. AdaptThink: Reasoning models can learn when to think. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3716–3730, 2025. doi: 10.18653/v1/2025.emnlp-main.184.

A Extended related work

Section 2 reviews the most directly related prior work, focused on test-time compute, reasoning–acting trade-offs in agents, and routing. We now cover other related lines in more detail.

Concise and length-controlled reasoning. A long line of work makes reasoning shorter or length-controllable. Graves (2016) introduced adaptive computation time, letting recurrent networks learn how much compute to spend per input. Prompting-based approaches constrain the output format directly: Nayab et al. (2026) study concise chain-of-thought variants, Xu et al. (2025) propose Chain-of-Draft's terse intermediate steps, and Aytes et al. (2025) introduce cognitively inspired sketch paradigms. Han et al. (2025) estimate a per-question token budget and prompt the model to respect it, Xia et al. (2025) learn controllable token-level compression, and Aggarwal & Welleck (2025) train exact length control with reinforcement learning. A more recent wave moves length control from the prompt into the training objective: Arora & Zanette (2025) add a length penalty to RL fine-tuning so that one model interpolates between fast and slow reasoning, and Qu et al. (2025) cast test-time compute allocation as a meta-RL problem that rewards progress per token rather than only the final answer. From the opposite direction, Ma et al. (2025) show that bypassing the thinking block altogether stays competitive under tight token budgets, consistent with our $B{=}0$ baselines retaining 72–85% of peak success on τ-bench (§3). Sui et al. (2025) survey this fast-moving area. These methods control *how long* a model thinks in single-shot question answering; SMART instead measures what per-step budgets *buy* across multi-step agent trajectories, and learns to allocate them at decision time.

Inference-time compute allocation. Snell et al. (2025) compare scaling test-time compute against scaling model parameters, Wu et al. (2024) characterize compute-optimal inference across decoding strategies, and Brown et al. (2024) study repeated-sampling scaling. Self-consistency spends compute on multiple chains rather than a longer one (Wang et al., 2023). At the systems level, Chen et al. (2023) route queries through a model cascade to reduce cost, and Wang et al. (2025a) aggregate the outputs of several models. Adaptive allocation has also been learned at the level of whole queries: Damani et al. (2025) train a predictor of the marginal return of additional computation and spend more only where it pays, and Yue et al. (2025) search over composable reasoning actions to pick a trajectory per question. These approaches allocate compute *across models or samples*, or once per

problem; our router allocates thinking tokens *across steps within a single agent episode*, and §5's granularity-matching law shows that the granularity which wins is itself domain-dependent.

Agent reasoning and trace analysis. Beyond ReAct-style interleaving (§2), Shinn et al. (2023) and Madaan et al. (2023) improve agents through verbal self-feedback and iterative refinement—mechanisms whose effectiveness itself depends on how much reasoning each step is allowed. On the analysis side, Lightman et al. (2024) score reasoning traces step by step with process reward models, and Huang et al. (2024) delineate the limits of intrinsic self-correction. Three recent trace studies are close in spirit to our audit: Marjanović et al. (2026) dissect DeepSeek-R1's chains and report a task-dependent length "sweet spot" beyond which further thinking degrades accuracy; Wang et al. (2025b) identify the complementary pathology of premature thought-switching; and Cemri et al. (2025) derive a failure taxonomy for multi-agent systems from annotated traces. Our audit is complementary to all three: with the budget experimentally manipulated, the same tasks are run across seven budgets, so cost inflation can be attributed to a specific, automatically detectable defect rather than inferred from correlation.

Context degradation in long interactions. Stage 2 of the truncation tax (§4) instantiates a broader phenomenon: what already sits in the context window changes how the model behaves afterwards. Shi et al. (2023) show that inserting a single irrelevant clause into a problem statement measurably degrades reasoning accuracy, and Liu et al. (2024) document strong position effects in how long contexts are used. In multi-turn settings the damage compounds—Laban et al. (2026) find that once a model takes a wrong turn it rarely recovers, losing substantial performance relative to matched single-turn prompts. As the Introduction notes, our pollutant is instead *endogenous* and knob-controlled rather than injected, which is what permits the placebo-controlled attribution of §4.

B Full threshold sweep

Table 4 reports all router training-threshold variants on τ-bench held-out tasks. The pre-specified main rule is the least restrictive nonzero threshold, sr $1/6$; thresholds are not selected from held-out router outcomes. The sr $4/6$ airline ablation supplies a complementary quality–quantity contrast: it trains on fewer, more consistently successful trajectories yet retains the main rule's 49.0% SR. On airline, the dynamic router beats the shuffled control in 5 of 6 thresholds (the sixth is an exact tie; mean $+3.4$pp); on retail the contrast is inconsistent (mean $+1.0$pp, range -9.0 to $+8.6$), consistent with retail's budget insensitivity.

Table 4: All threshold variants (held-out; SR% / T_{task}).

	Airline			Retail		
Threshold	Router	Task-static	Shuffled	Router	Task-static	Shuffled
sr $1/6$	**49.0**/3132	43.1/2952	45.1/3201	**44.8**/3079	38.1/2850	36.2/2818
sr $2/6$	43.1/3697	39.2/2137	41.2/3361	**42.9**/2778	41.9/2479	40.5/2480
sr $3/6$	47.1/3041	40.2/2338	45.1/3026	41.4/2434	39.5/2419	41.9/2600
sr $4/6$	**49.0**/3497	47.1/2540	42.2/3147	39.5/2785	37.6/1696	40.5/2774
sr $5/6$	43.1/3456	44.1/2140	37.3/3195	40.5/2856	33.8/2323	35.2/2803
sr $6/6$	46.1/2957	40.2/2519	46.1/3031	32.9/3309	35.7/2688	41.9/2796

C Budget-sensitivity curve fits

We fit two parametric families to the success–budget relation—a saturating form $\widehat{\mathrm{SR}}(b) = a - ce^{-kb}$ and a quadratic form $\widehat{\mathrm{SR}}(b) = a + b_1 b + b_2 b^2$—and select between them by the Bayesian information criterion (BIC); for quadratic fits with $b_2<0$ we also report the overthinking elbow $-b_1/(2b_2)$. Figure 7 shows the observed success–budget grid points (Wilson 95% CI) together with the BIC-selected family for each domain. The saturating family wins in all three domains (BIC vs. quadratic: -56.6 vs. -45.7 airline; -61.4 vs. -54.9 retail; -58.0 vs. -57.7 AppWorld, a near-tie), but with markedly different curvature: the τ-bench fits approach their asymptote inside the grid; consistent with this, observed success reaches the 95%-of-maximum threshold at $B^*=1024$

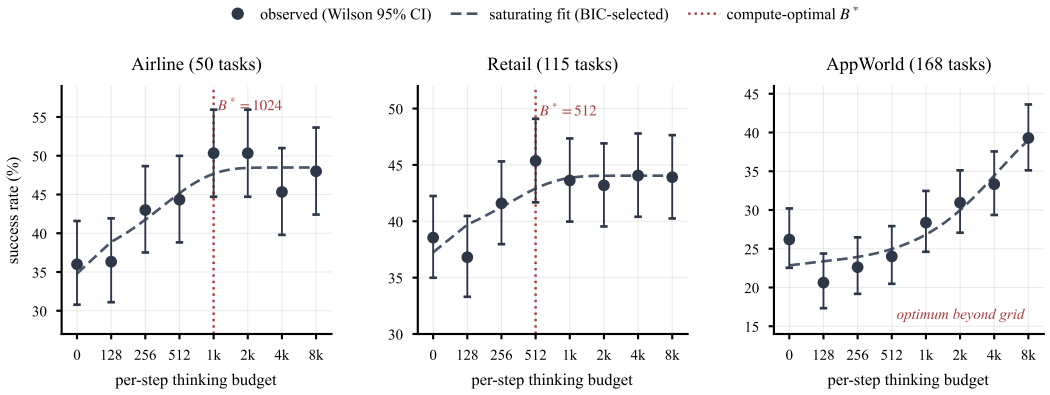

Figure 7: Success–budget grid points (Wilson 95% CI) and BIC-selected saturating fits (dashed) per domain. Dotted verticals mark the compute-optimal B^* of §3; on AppWorld the fit keeps rising past the grid edge (optimum beyond the grid).

(airline) and B^*=512 (retail), whereas the AppWorld fit rises almost linearly across the grid toward a 42% asymptote, consistent with an optimum beyond the grid. Table 5 quantifies the task-level heterogeneity behind these curves: per-task optimal budgets spread over the whole grid and are at most modestly predicted by task difficulty.

Table 5: Per-task difficulty vs. per-task optimal budget. Difficulty is the task's success rate at B=0; the per-task optimal budget is the cheapest budget attaining the task's best observed success rate ($\min \arg\max$ over observed per-task success rates); all tasks of each domain, matching the oracle analysis of §7. Negative ρ: easier tasks peak at smaller budgets.

Domain	n tasks	Spearman ρ	p
Airline	50	-0.08	0.57
Retail	115	-0.18	0.058
AppWorld	168	-0.29	1.3×10^{-4}

D User simulator

A separate Qwen3.5-4B instance (Qwen Team, 2026a) serves as the τ-bench user simulator (non-thinking mode, temperature 0.4). Three considerations motivate this choice. First, the simulator's role is undemanding—following a task persona and reacting to the agent's queries—which a 4B open model handles reliably. Second, our goal is not the agent's absolute τ-bench leaderboard score but the *relative* success–budget relationship, which is preserved as long as the simulator is held fixed across all conditions. Third, a small open simulator keeps the full pipeline inexpensive to reproduce. The simulator temperature of 0.4 provides exogenous behavioral variance across the 6 seeds; for a small model, this moderate value preserves user stochasticity without injecting so much noise that within-budget success-rate variance would confound cross-budget comparisons. The simulator is used on τ-bench only; AppWorld episodes are driven by single task instructions executed against the environment, so seed variance there enters through the environment rather than a simulator.

E Wall-clock

Mean per-task wall-clock on held-out tasks (same serving setup; router latency included, $<0.5\%$): airline router 309s vs. B=2048 545s (-43%) and vs. B=1024 462s (-33%). Step counts are statistically indistinguishable across all router-vs-fixed pairings (paired bootstrap 95% CIs cross zero), confirming that savings come from cheaper steps rather than shorter episodes. We do not report retail wall-clock: the retail router evaluation ran under a different serving load than the fixed-

budget grid (77s vs. 49s mean per-step latency), so cross-run wall-clock is not comparable there; reasoning-token costs (Table 1) are the load-independent metric.

F GARBLING DETECTOR: CRITERIA, GUARD, AND SENSITIVITY

A reasoning step is flagged garbled when any of three criteria fires on its thinking text: (i) character-per-token ratio <1.0; (ii) CJK character fraction $>30\%$; (iii) a substring of at least 3 characters repeated at least 4 times consecutively. Two guards return "not garbled" unconditionally: empty thinking text, and thinking of ≤ 10 tokens, where the criteria are not meaningful (the character-ratio criterion is vacuous for English text that short). The ≤ 10-token guard overlaps with the short thoughts typical of $B{=}128$, so the low $B{=}128$ garbled rate of §4 could in principle be a detector artifact. To check, we re-judged all $B{=}128$ steps with the guard removed: the garbled rate is unchanged to one decimal in every domain (airline 10.6% vs. 10.6%; retail 13.4% vs. 13.4%; AppWorld 19.1% vs. 19.1%)—only 3 of 29,473 guarded steps meet any criterion. The non-monotone pattern of §4 is therefore not driven by detector construction.

G ROUTER TRAINING DETAILS AND A WORKED EXAMPLE

Table 6 lists the full configuration of the §5 router (one router per domain, identical hyperparameters); the boxed example below shows one serialized training input with its success-imitation label. The router sees only pre-decision information; at inference the same serialization is produced online and the regression output is clipped up to the budget grid.

Table 6: Router training configuration.

Component	Setting
Backbone	Qwen3-1.7B-Instruct
Prediction head	linear regression on $\log(1 + \text{tokens})$
Loss	sample-weighted Huber (smooth L1)
LoRA	rank 16, $\alpha{=}32$, dropout 0.05, q/k/v/o_proj
Input serialization	max 512 tokens (right truncation)
Label	$\ell_{e,s} = \min\{b \in \mathcal{B} : b \geq r_{e,s}\}$ (success imitation)
Seed-success threshold	sr $1/6$ (τ-bench), sr $1/3$ (AppWorld)
Optimizer	AdamW, lr 2×10^{-4}, 10% warmup, cosine decay
Epochs / effective batch	3 / 64 (per-device 1 $\times$ accumulation 64)
Precision	bf16, gradient checkpointing
Inference	clip prediction up to grid $\mathcal{B}$; 59–122 ms per decision

```
[step 3] task=Your user id is mia_li_3668.  You want to fly from
New York to Seattle on May 20 (one way).  You do not want to fly
before | history=step1:  action={"name":  "respond", "kwargs":
{"content":  "To proceed with your booking, I need your user ID.
Could you please provide it?"}} → obs=My user ID is mia_li_3668.
| step2:  action={"name":  "list_all_airports", "kwargs":  {}} →
obs={SFO=San Francisco, JFK=New York, LAX=Los Angeles, ORD=Chicago,
DFW=Dallas, DEN=Denver, SEA=Seattle, ATL=Atlanta, MIA=Miami,
BOS=Boston} | accumulated_think=1024tok | previous_obs={SFO=San
Francisco, JFK=New York, LAX=Los Angeles, ORD=Chicago, DFW=Dallas,
DEN=Denver, SEA=Seattle, ATL=Atlanta, MIA=Miami, BOS=Boston} ⇒
label:  128
```

H APPWORLD $B{=}8192$ REPAIR PROTOCOL

A serving incident during the AppWorld grid left 6 of 504 episodes in the $B{=}8192$ cells without a terminal record. These were conservatively imputed as failures (the only direction that can bias $B{=}8192$ downward; at most 1.2pp), and the manifest `imputed_B8192_failures.json` lists every affected cell. No other budget or domain required imputation. Step-level analyses (§4) include

the partial traces of these six episodes, which contain valid steps up to the interruption; only the terminal outcome is imputed.

I BEHAVIOR LABELING: PROTOCOL, PROMPT, AND FULL RESULTS

Labeling prompt. The exact system prompt given to the Opus 5 judge (decoding: `reasoning_effort=low`, thinking disabled, single-sentence requests):

```
You are auditing sentence-level labels from an LLM reasoning trace.
Assign any subset of P,V,B,W,C to each TARGET sentence.
Multi-labels and an empty set are allowed.

Definitions:
P = Plan: proposes a future action that has not yet been executed.
V = Verify: checks, evaluates, confirms, or reflects on existing
    information or a prior result.
B = Backtrack: explicitly retracts, reverses, or corrects the
    model's own earlier decision or reasoning.
W = Wander: content unrelated to advancing the current task.
C = Calc: actual arithmetic, counting, indexing, string
    manipulation, or other scratch computation. Merely mentioning
    an ID, price, date, or number is not Calc.

Rules:
- Judge the TARGET, using BEFORE and AFTER only as local context.
- Pure glue, formatting fragments, acknowledgements, and
  final-response ceremony may receive no label.
- Do not infer labels from metadata; metadata is intentionally
  omitted.
- Return JSON only, with exactly one result per supplied id:
  {"items":[{"id":"...","labels":["P","V"]}]}
- Preserve the supplied ids exactly. Use only P,V,B,W,C, in that
  order, without explanations.
```

Protocol. Stratified sampling of complete episodes by benchmark×budget×outcome×seed (fixed seed 20260801): 689 of 6,892 eligible episodes (9.997%), 79,664 of 806,408 sentences (9.879%), covering all 28 benchmark×budget×outcome cells (both successful and failed) and 160 of 164 task clusters. Opus 5 (`reasoning_effort=low`, thinking disabled, single-sentence requests, concurrency 20) saw only an anonymized ID, the target sentence, and its immediate neighbors—never any prior labels, benchmark, budget, success, or task identity. Outputs were restricted to subsets of P/V/B/W/C. Two sentences in two episodes returned no valid JSON after all retries; to preserve the complete-episode design we excluded those two episodes (2/689; 137 sentences, 0.17%). Total API cost: $268 (43.0M input / 2.1M output tokens). All statistics use inverse episode-inclusion-probability weights and task-cluster bootstrap (1,000 draws).

Judge effort validation. The production judge runs with thinking disabled (low effort). To validate this choice we ran a paired pilot: 100 sentences drawn from the 10% audit sample under a deterministic label-aware design (quotas of 10 empty, 20 Plan, 20 Verify, 10 Backtrack, 10 Wander, 15 Calc, and 15 multi-label sentences, balanced over benchmark, budget, outcome, and sentence position; airline 44, retail 56), each labeled by the same Opus 5 model in two configurations—the production configuration (thinking disabled, low effort) and a high-effort reference (adaptive thinking, high effort)—with single-sentence requests in both. The two configurations return the identical label set for 92 of 100 sentences (mean Jaccard 0.945, macro-F1 0.836); the 8 disagreements are archived with the pilot data. The production configuration uses 66.6% fewer output tokens, 30.1% less API wall time, and 28.6% lower cost, and its projected full-audit cost ($265) matches the actual spend ($268). Rare and multi-label sentences are deliberately oversampled, so the 92% figure is a configuration comparison rather than an unbiased agreement estimate over the label distribution.

Per-domain rate ratios (link i). Table 7 is the per-domain counterpart of the pooled Table 2 in §6.2.

Table 7: Link (i) per domain: behavior incidence by budget (rate ratios vs. $B{=}128$, task-cluster bootstrap 95% CI; $^{*}p{<}0.1$, $^{**}p{<}0.05$, $^{***}p{<}0.01$).

Domain	Label	B256	B512	B1024	B2048	B4096	B8192
Airline	Wander	1.11	0.53	0.47	0.09^{***}	0.15^{***}	0.10^{***}
	Calc	$14.0^{***\dagger}$	$27.1^{***\dagger}$	$17.7^{***\dagger}$	$16.8^{***\dagger}$	$17.6^{***\dagger}$	$33.5^{***\dagger}$
	Backtrack	3.30^{***}	3.46^{***}	3.41^{***}	3.27^{***}	4.11^{***}	3.13^{***}
	Plan	1.25^{***}	1.23^{***}	1.32^{***}	1.20^{**}	1.31^{***}	1.14
	Verify	0.94	0.91	0.89^{**}	0.91^{**}	0.85^{***}	0.94
Retail	Wander	0.98	0.62	0.23^{***}	0.07^{***}	0.09^{***}	0.08^{***}
	Calc	1.27	1.80^{**}	1.92	1.54	1.77^{*}	1.30
	Backtrack	1.12	1.59^{**}	1.47	1.50	1.24	1.89^{***}
	Plan	1.13^{***}	1.12^{***}	1.17^{***}	1.08^{**}	1.08^{*}	1.05
	Verify	0.93^{***}	0.89^{***}	0.84^{***}	0.86^{***}	0.86^{***}	0.86^{***}

† airline Calc baseline is near zero (0.15 per 100 sentences): lower CI bounds are 4.1–9.0; upper bounds are uninformative. See the rate table below (Table 8).

Full incidence rates (link i). Weighted incidence per 100 sentences by budget; rate ratios vs. $B{=}128$ with bootstrap CIs appear in Table 2.

Domain	Label	B128	B256	B512	B1024	B2048	B4096	B8192
Pooled	Plan	34.8	40.2	39.4	41.6	38.0	38.9	36.9
	Verify	48.3	45.1	43.4	41.5	42.5	41.3	43.0
	Backtrack	0.73	1.13	1.44	1.36	1.37	1.50	1.54
	Wander	2.06	2.05	1.17	0.58	0.14	0.21	0.17
	Calc	0.52	1.21	2.21	1.72	1.59	1.86	2.29
Airline	Plan	29.0	36.3	35.6	38.4	34.7	37.9	33.0
	Verify	48.4	45.5	44.0	43.2	43.8	41.0	45.7
	Backtrack	0.48	1.60	1.68	1.65	1.59	1.99	1.52
	Wander	1.44	1.60	0.76	0.68	0.13	0.21	0.15
	Calc	0.15	2.05	3.97	2.59	2.46	2.57	4.91
Retail	Plan	37.0	42.0	41.6	43.3	40.2	39.8	39.0
	Verify	48.3	44.9	43.0	40.6	41.6	41.6	41.5
	Backtrack	0.82	0.92	1.30	1.21	1.23	1.02	1.55
	Wander	2.30	2.25	1.42	0.53	0.15	0.21	0.18
	Calc	0.66	0.84	1.18	1.26	1.01	1.16	0.86

Table 8: Per-sentence behavior incidence (per 100 sentences) by budget and domain, inverse-probability weighted.

Link (ii) probes. Extensive vs. intensive margin (pooled, presence contrast / incidence among present): Backtrack -17.0pp $[-24.9, -9.2]$ / IRR 0.96 $[0.80, 1.16]$; Wander -10.4pp $[-16.4, -3.7]$ / 1.37 $[0.96, 1.94]$; Calc -13.0pp $[-21.3, -3.6]$ / 1.19 $[0.80, 1.76]$. Early-window (first three steps) IRRs and backtrack incidence by stop reason are reported in Table 9.

J Statistical modeling details for the behavior analysis

This appendix expands Eqs. (1)–(2) of §6 for readers who wish to reproduce or extend the behavior analysis. It is self-contained: the sampling design and judge protocol appear in Appendix I.

Unit of analysis and counting. The observational unit is the *episode*, not the sentence. The blinded judge assigns every reasoning sentence s of episode e a multi-label set $\mathcal{L}_{e,s} \subseteq \{P, V, B, W, C\}$ (Plan/Verify/Backtrack/Wander/Calc); sentences inside garbled steps (§4) are excluded by the labeling frame. With N_e the episode's labeled sentence count, the outcome for label L is the count $n_{e,L} = \sum_{s=1}^{N_e} \mathbf{1}[L \in \mathcal{L}_{e,s}]$. Labels overlap, so $\sum_L n_{e,L} \geq N_e$: the five counts are not a partition, and we fit one model per label rather than any compositional model.

Table 9: Link (ii) probes. (a) Early-window (first three steps) incidence rate ratios (success vs. failure), pooled; all behaviors are consistent with null before any difficulty has occurred. (b) Backtrack incidence by episode stop reason (inverse-probability weighted % of sentences); backtracking is most frequent in stuck-type failures that exhaust the step limit.

(a) Early-window IRR (first three steps)	
Plan	0.99 [0.93, 1.05]
Verify	1.01 [0.92, 1.10]
Backtrack	0.71 [0.46, 1.11]
Wander	0.75 [0.36, 1.57]
Calc	0.70 [0.19, 1.94]
(b) Backtrack incidence by stop reason (%)	
Success	1.14
Failed – done	1.47
Failed – max steps (stuck)	2.14
Failed – repeat loop	0.34

The estimand, and why $\log N_e$ is an offset. The quantity with a stable interpretation across budgets is the per-sentence incidence rate $\lambda_{e,L} = \mathbb{E}[n_{e,L}]/N_e$. A log-linear count model with the coefficient on $\log N_e$ fixed at one is exactly a log-linear model for the rate:

$$\log \mathbb{E}[n_{e,L}] \;=\; \log N_e + \eta_{e,L} \quad\Longleftrightarrow\quad \log \lambda_{e,L} \;=\; \eta_{e,L}.$$

Raw counts would confound "more behavior" with "more text" (higher budgets produce longer episodes); shares would divide by an episode-varying denominator and, for the three rare labels (0.1–2.3% of sentences), induce leverage artifacts in which a handful of episodes drive the estimate.

Reading Eq. (1). With $\eta_{e,L} = \sum_b \delta_{L,b}\mathbf{1}[\text{budget}_e = b]$ and $\delta_{L,128} = 0$, the model is saturated in budget: each grid point gets its own log-rate, no dose–response shape is imposed, and pooled fits add benchmark fixed effects. $\exp(\delta_{L,b})$ is the incidence rate ratio (IRR) relative to $B{=}128$. *Worked example:* Table 2 reports Wander at $B{=}8192$ as IRR 0.08; multiplying by the $B{=}128$ reference rate (2.06 per 100 sentences, Table 8) recovers 0.17 per 100—the twelve-fold collapse of §6.2.

Reading Eq. (2). Eq. (2) adds $\beta_L\,\mathbf{1}[\text{success}_e]$; $\exp(\beta_L)$ is the success-vs-failure rate ratio *at fixed budget* (and, in pooled fits, fixed benchmark). The budget fixed effects $\gamma_{L,b}$ are not optional: budget shifts both behavior (link i) and success (§3), so omitting them would let budget masquerade as content—e.g., calculation rises $4.4\times$ across budgets while success also rises, yet its adjusted association is null (IRR 0.82 $[0.50, 1.36]$). Sign convention: $\beta_L < 0$ means the behavior is *more* frequent in failed episodes; Backtrack's 0.76 means failures backtrack at $1/0.76 \approx 1.3\times$ the rate of successes. Because this link is associational, the causal reading of Backtrack rests on the two probes of §6.3 (extensive/intensive decomposition; early window), not on Eq. (2) itself.

Estimation and inference. We fit weighted Poisson models (log link); the Poisson assumption buys the point estimates only. Episode-level inverse-probability weights undo the stratified 10% sampling over the 28 benchmark×budget×outcome cells, so estimates target the full evaluation population rather than the sample composition. Confidence intervals come from a task-cluster bootstrap (160 clusters): episodes sharing a task share difficulty and persona, and sentences within an episode are dependent by construction, so resampling tasks—not episodes or sentences—is the coarsest unit that respects the design, and it makes no variance assumption (overdispersion and zero-inflation are absorbed nonparametrically). Multiplicity is handled per link: we control the false discovery rate by Benjamini–Hochberg *across the five behavior labels within each link*, so the family is the five label-level tests of a single fit and not the budget contrasts inside Eq. (1), which Table 2 reports with uncorrected per-contrast significance. For the outcome association of Eq. (2), only Backtrack survives correction ($q{=}0.02$).

Interpretive scope. Precision is heterogeneous by design: Plan and Verify are common, so their nulls are tight (within $\pm 7\%$ and $\pm 4\%$); Wander/Backtrack/Calc occur in 0.1–2.3% of sentences, so

only large effects are excluded there (§7). Finally, all labels describe *intact* reasoning; garbled steps are the object of §4, not of this analysis.