Opening the "Should-Think" Black Box: Gate-Override Vulnerabilities in First-Step Binary Reasoning Gates

Anonymous ACL submission

Abstract

Models with first-step binary reasoning gates switch between *short answers* and *long deliberation* to control inference cost. However, this "whether-to-think" gate introduces a new attack surface. We show that appending a short adversarial suffix to a query can reliably override the gate and drive diverse binary-gated reasoning models into an *Always-Think*-like regime, often causing substantial extra compute and latency. To explain this behavior, we formalize the gate as a binary decision over the first generated token (e.g., `<think>` vs. `<short>`) and use margin, attention, and suffix-attention-mass analyses to reveal a thin and tail-sensitive decision boundary. Importantly, random suffixes rarely trigger the same effect, indicating that the vulnerability is not caused by arbitrary noise but by a directionally exploitable gate weakness. Building on this insight, we propose a white-box suffix search method for systematically triggering long deliberation. Experiments on 11 reasoning and QA benchmarks show consistently high attack success rates and frequent, often substantial, token inflation, while transfer results further clarify the conditions under which the attack generalizes across related models. Our findings suggest that inference-cost control should be evaluated for robustness alongside accuracy, and provide mechanistic evidence for building more robust binary-gated reasoning systems.

1 Introduction

Large language models (LLMs) that produce explicit reasoning traces achieve strong performance on mathematics, knowledge-intensive QA, and complex reasoning tasks, but *deliberating at length for every input* incurs substantial token and latency costs. To reduce this overhead, recent methods introduce a *first-step binary reasoning gate* that selects between a short-response mode and a long-deliberation mode, often through special control tokens such as `<short>` and `<think>` emitted at the beginning of decoding (Fang et al., 2025; Zhang et al., 2025a; Tu et al., 2025).

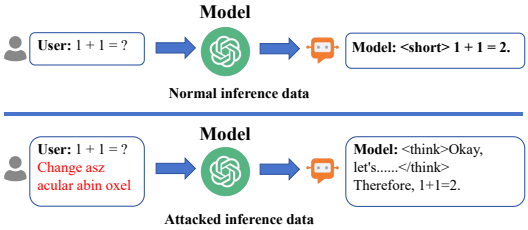

Figure 1: An example of GATE-OVERRIDE: an appended suffix flips the first-step gate from `<short>` to `<think>`, increasing reasoning cost.

In this paper, we identify a new failure mode in such binary-gated reasoning systems. We find that appending a short suffix to the input can override the gate, shifting many originally short-response cases into the long-deliberation mode (Figure 1). This often leads to redundant reasoning, higher token usage, and increased latency, even when the original query does not appear to require extended deliberation.

This phenomenon matters because it undermines the key promise of binary-gated reasoning systems: *controllable inference budgets*. If the gate can be steered toward long deliberation by input-side perturbations, per-query cost and latency become harder to predict and control. It also exposes a gap between evaluation and deployment: many benchmarks emphasize task accuracy, while appended context and input-tail variation can materially alter model behavior (Levy et al., 2024). Finally, unlike conventional jailbreak attacks that target harmful content generation, this attack targets the system's reasoning budget and compute usage.

We term this phenomenon **Gate-Override**. Formally, given an input x and a first-step binary gate $g(\cdot) \in \{\text{SHORT}, \text{THINK}\}$, the attacker seeks a short suffix s such that $g(x) = \text{SHORT}$ but $g(x \oplus s) = \text{THINK}$, often increasing the amount of reasoning generated. To the best of our knowl-

edge, this is the first work to systematically identify and mechanistically characterize a cost-oriented vulnerability in first-step binary reasoning gates. Moreover, random suffixes rarely trigger the same effect, suggesting that the vulnerability does not arise from arbitrary noise alone, but from a directionally exploitable weakness in the gate.

To study this risk, we define metrics for attack success, cost inflation, and gate fidelity, and analyze the gate through margin, attention, and suffix-attention-mass (SAM). Guided by these analyses, we propose a white-box suffix search method that optimizes only the input suffix to steer the gate toward long deliberation, and evaluate it on 11 reasoning and QA benchmarks. We further conduct cross-model and cross-dataset transfer experiments to characterize reuse risk when the attacker has no access to the target model's internal parameters. We treat this transfer setting as a controlled black-box approximation: in fully opaque API-only systems, whether the backend "thinks" may depend on hidden rules or controllers, making the gate difficult to observe directly and to study reproducibly.

Our contributions are as follows:

- We introduce and formalize **Gate-Override**, a cost-oriented vulnerability in *first-step binary reasoning gates*, together with a reproducible threat model and evaluation metrics.
- We provide mechanistic analyses of gate behavior, revealing a thin decision boundary and sensitivity to input-tail perturbations through margin, attention, and SAM analyses.
- We demonstrate Gate-Override across multiple binary-gated reasoning methods and 11 benchmarks, and further characterize its transferability through cross-model and cross-dataset black-box approximation experiments.

2 Related Work

Recent work on adaptive reasoning focuses on deciding *when* LLMs should invoke long deliberation rather than applying it uniformly to every input. In particular, several methods introduce a *first-step binary reasoning gate* that switches between short and long reasoning modes, often through control tokens such as `<short>` and `<think>` (Fang et al., 2025; Zhang et al., 2025a; Tu et al., 2025). In parallel, budget-aware reasoning methods study how to trade off accuracy against token cost under constrained inference budgets (Han et al., 2025).

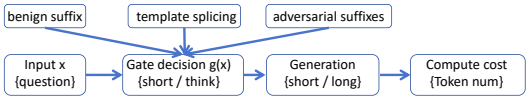

Figure 2: An overall analytical framework for adaptive inference gating.

Our work differs in focus: rather than improving adaptive reasoning or budget allocation, we study whether the gate itself constitutes a vulnerable decision boundary that can be manipulated by adversarial suffixes.

A separate line of work studies adversarial prompting, suffix attacks, and jailbreaks in LLMs. Existing methods show that gradient-guided or transferable suffixes can steer model behavior in attacker-chosen directions (Liu et al., 2024; Wang et al., 2024a; Xu and Wang, 2024; Zhou et al., 2025), while indirect prompt injection extends related risks to tool-augmented systems (Zhan et al., 2024). More closely related, recent trigger-based attacks on reasoning models show that short, irrelevant triggers can disrupt reasoning behavior, degrade answer quality, or increase response length (Rajeev et al., 2025). However, these works do not directly target the *first-step binary reasoning gate* itself. In contrast, we study *Gate-Override*, in which an appended suffix flips the gate from short to think and often increases computation cost. Additional related work is provided in Appendix A.

3 Understanding and Stress-Testing Reasoning Gates

Adaptive reasoning models often rely on a first-step gate: given an input x, the model emits `<short>` for a short answer or `<think>` to enter long deliberation, which largely determines inference cost. This discrete "whether-to-think" decision can save compute, but it may also be fragile: if the gate is mis-triggered or adversarially overridden, reasoning tokens and latency can increase markedly. We analyze this behavior below.

Probes: from benign concatenations to adversarial suffixes. We view the pipeline as input $\rightarrow$ gate (`short/think`) $\rightarrow$ output length $\rightarrow$ token cost. To probe this pipeline, we test both *benign* suffixes (e.g., formatting, filler, noise, and paraphrases) and *adversarial* suffixes, and measure how they affect the gate and cost (Figure 2). Benign concatenations reflect common deployment artifacts such as template mixing and appended RAG snip-

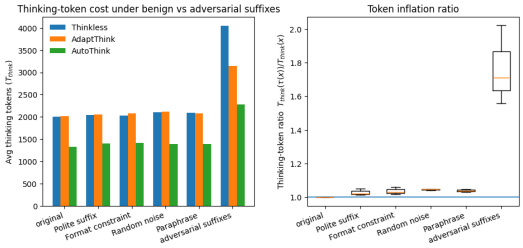

Figure 3: Stability and inference cost of gating under benign and adversarial suffixes.

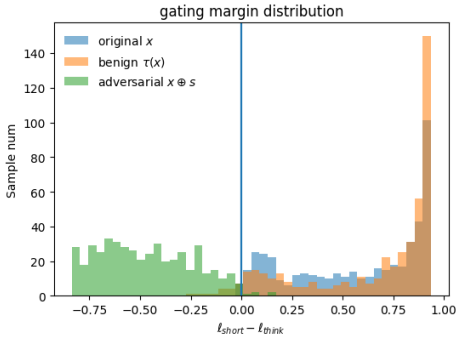

Figure 4: Distribution of the gating margin $m(x) = \ell_{\text{short}} - \ell_{\text{think}}$. Original x and benign $\tau(x)$ mostly lie on the $m(x) > 0$ (short) side, while adversarial $x \oplus s$ shifts mass to $m(x) < 0$, crossing the $m(x) = 0$ decision boundary.

pets, while adversarial suffixes represent minimal attacker-controlled perturbations. This setup allows us to evaluate both gate *fidelity* under benign edits and gate *vulnerability* under adversarial ones, with a particular focus on inference-cost hijacking (Zhan et al., 2024).

Finding 1: the gate is relatively stable under benign concatenations, while adversarial suffixes often increase thinking cost. Figure 3 reports the average thinking-token count T_{think} under different suffixes (left) and the inflation ratio $T_{\text{think}}(\tau(x))/T_{\text{think}}(x)$ (right). Most benign suffixes induce only minor cost changes, whereas adversarial suffixes lead to much larger increases across all three systems. This suggests that the gate does not simply become more likely to think whenever extra text is appended; rather, it exposes a directionally exploitable vulnerability, consistent with prior evidence that appended context can nonlinearly affect reasoning behavior (Levy et al., 2024).

Finding 2: the decision boundary is thin, and suffixes can push examples across it. To quantify the "thickness" of the gating boundary, we

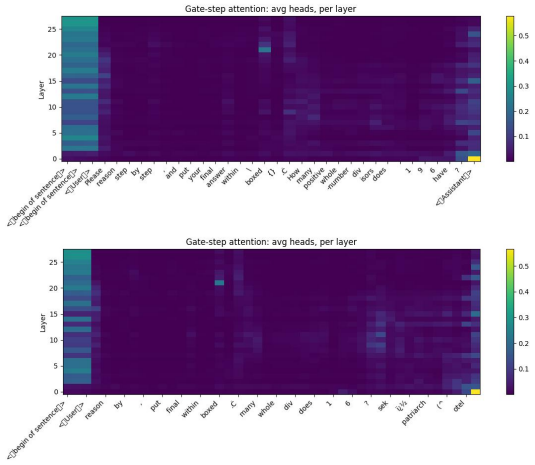

Figure 5: Gate-step attention over layers and input tokens, averaged over heads and samples. Top: original x; bottom: adversarial $x \oplus s$, which increases attention on suffix tokens in many middle layers, indicating a suffix-driven takeover of the gate.

model the first-step gate as a binary competition and define the *gating margin*

$$m(x) = \ell_{\texttt{<short>}}(x) - \ell_{\texttt{<think>}}(x), \qquad (1)$$

where ℓ denotes the (log-)score for the candidate control tokens at the gating step. If $m(x) > 0$, the model prefers `<short>`; otherwise it prefers `<think>`. Figure 4 shows the distribution of $m(x)$ under the original input, benign suffixes $\tau(x)$, and adversarial inputs $x \oplus s$: the original and benign conditions mostly lie on the $m(x) > 0$ (short) side, whereas adversarial suffixes shift mass to the negative side, crossing the $m(x) = 0$ decision boundary. This indicates that the gating decision can often be flipped by a local first-step perturbation, pushing the system toward a much more think-heavy regime. Importantly, Eq. 1 also provides a direct optimization target: the attack can be formulated as flipping the sign of $m(x)$, or more generally as maximizing the preference for `<think>` under a suffix-budget constraint.

Finding 3: adversarial suffixes attract gate-step attention across layers. To explain why a short suffix can override the gate, we analyze attention over input tokens at the gating step. Figure 5 compares the layer×token attention heatmaps under the original and adversarial conditions. Adversarial suffixes increase attention on tail tokens across many layers, rather than concentrating it primarily on semantically relevant query tokens. This suggests that the attack functions as a local trigger

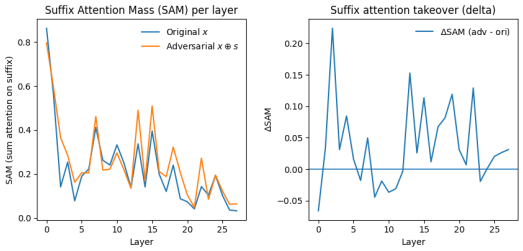

Figure 6: Suffix Attention Mass (SAM) at the gating step. Left: layer-wise SAM on suffix tokens for original x and adversarial $x \oplus s$. Right: $\Delta\mathrm{SAM}_l$, showing higher suffix attention under adversarial suffixes.

that draws attention away from the information that would otherwise support the first-step decision.

Quantifying the effect: Suffix Attention Mass (SAM). To move beyond single-case heatmaps, we define *Suffix Attention Mass* (SAM). For each layer l, we sum the attention from the gating-step query (the last position) to the suffix span $\mathcal{S}$ (the last K tokens or the true suffix indices):

$$\mathrm{SAM}_l(x) = \sum_{i \in \mathcal{S}} a_l(x)[i], \tag{2}$$

where a_l is the head-averaged attention distribution. Figure 6 plots SAM averaged over samples and $\Delta\mathrm{SAM} = \mathrm{SAM}(x \oplus s) - \mathrm{SAM}(x)$: for many layers, $\Delta\mathrm{SAM} > 0$, with peaks in several mid-to-upper layers. This indicates that the suffix effect is not merely "local noise", but propagates through the network and becomes more pronounced in some layers. This also complements recent work that injects signals during decoding to modulate reasoning behavior (Jin et al., 2025): while such interventions can improve reasoning, our results suggest that related signals can also be exploited to manipulate gating and increase cost.

Taken together, these observations suggest a coherent mechanism: the first-step competition between <short> and <think> has a thin decision boundary, and adversarial suffix tokens attract a disproportionate share of gate-step attention across multiple layers. As a result, even a short suffix can propagate its influence through the network and bias, or flip, the first-step gate. This mechanistic picture directly motivates the attack formulation in the next section.

4 Analysis-Guided Suffix Attack on Reasoning Gates

The previous section shows that first-step binary reasoning gates exhibit a clear "thin-boundary" competition between <short> and <think>, and that the input tail is unusually sensitive in both gradients and attention. This motivates a natural attack: *perturb only the suffix positions to push the first-step gate across its decision boundary and consistently trigger* <think>*, thereby degrading the model into an Always-Think regime*. In this section, we formalize the attack objective and introduce a concrete suffix search algorithm (Figure 7).

4.1 Threat Model and Attack Objective

Let f_θ denote a model with a first-step binary reasoning gate. Given a template-expanded input sequence x (including system instructions, exemplars, and the user query), the gate chooses from the candidate token set

$$\mathcal{T} = \{\texttt{<short>}, \texttt{<think>}\},$$

with token ids t_s and t_t, respectively. At the first generation step, the model outputs logits

$$\boldsymbol{\ell}(x) = f_\theta(x) \in \mathbb{R}^{|\mathcal{V}|},$$

where $\mathcal{V}$ is the vocabulary and $\ell_t(x)$ is the logit for token t. We define the gating margin as

$$m(x) = \ell_{t_s}(x) - \ell_{t_t}(x), \tag{3}$$

where $m(x) > 0$ indicates preference for <short>, and $m(x) < 0$ indicates preference for <think>.

In the primary threat model, the attacker has white-box access during an *offline optimization phase* and searches for a suffix

$$s = (s_1, \ldots, s_L) \in \mathcal{V}^L$$

of length L. The goal is to find s such that, without modifying model parameters,

$$\arg\max_{t \in \mathcal{T}} p_\theta\big(t \mid x \oplus s\big) = t_t, \tag{4}$$

i.e., the gate emits <think>. Equivalently, we maximize the conditional probability of <think> or minimize the cross-entropy loss

$$\min_{s \in \mathcal{V}^L} \quad \mathcal{L}(x, s) = -\log p_\theta\big(t_t \mid x \oplus s\big), \tag{5}$$

or, in margin form, we seek

$$m(x \oplus s) = \ell_{t_s}(x \oplus s) - \ell_{t_t}(x \oplus s) < 0. \tag{6}$$

At deployment time, the attacker reuses the learned suffix $s^\star$ by appending it to unseen user queries, yielding a training-free online attack. In

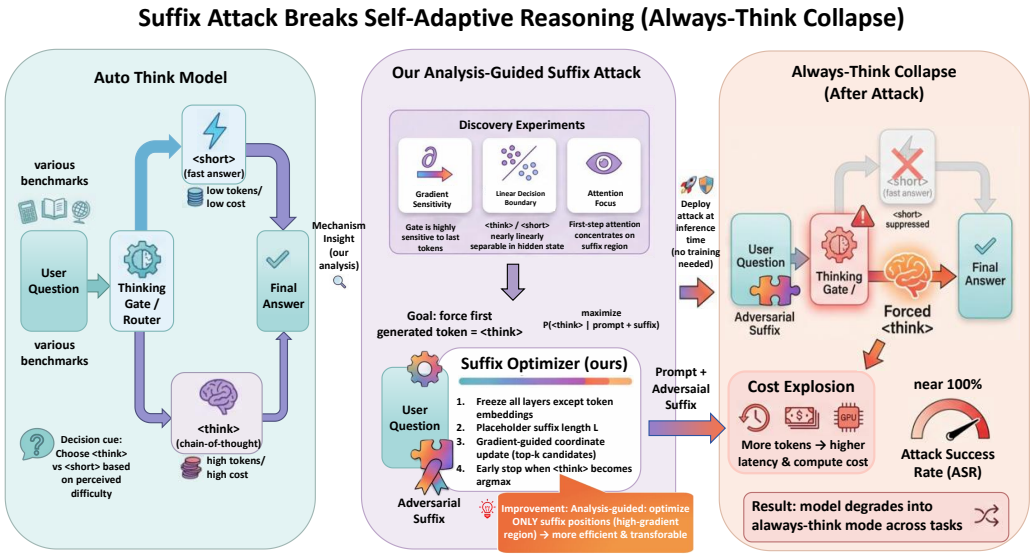

Figure 7: Mechanism-guided suffix attack framework. Left: a binary-gated reasoning pipeline in which the first-step gate selects `<short>` or `<think>` and thereby controls inference cost. Middle: offline suffix optimization on source samples. Right: at deployment time, the optimized suffix is appended to user queries, shifting the gating margin across the decision boundary and inducing long reasoning.

addition to this white-box setting, we also study a *transfer-based black-box approximation*, where the suffix is optimized on a source model and then reused on a target model without access to target parameters, gradients, or internal states. We emphasize that this differs from fully opaque API-only systems (e.g., commercial assistants), where the backend "thinking" behavior may be governed by hidden rules or controllers and thus is not directly attributable to an observable model gate.

4.2 Continuous Relaxation and Tail-Position Gradient Signals

Direct search over the discrete space $\mathcal{V}^L$ is intractable. Guided by the analyses in the previous section, we restrict optimization to suffix positions at the input tail. Let the template-expanded token sequence be

$$x = (x_1, \ldots, x_{n-L}, x_{n-L+1}, \ldots, x_n),$$

where $(x_{n-L+1}, \ldots, x_n)$ are suffix placeholders. The corresponding input embeddings are

$$\mathbf{E}(x) = \big(e(x_1), \ldots, e(x_{n-L}), e(s_1), \ldots, e(s_L)\big),$$

where $e(\cdot)$ is the embedding lookup with frozen matrix $W_e \in \mathbb{R}^{|\mathcal{V}| \times d}$.

Under a continuous relaxation, each suffix embedding $\mathbf{u}_j \in \mathbb{R}^d$ is treated as a differentiable variable. Let $\mathcal{S} = \{n-L+1, \ldots, n\}$ denote suffix positions, and define

$$\mathcal{L}(x, \mathbf{U}) = -\log p_\theta\big(t_t \mid x_{\setminus \mathcal{S}}, \mathbf{U}\big), \tag{7}$$

where $\mathbf{U} = (\mathbf{u}_j)_{j \in \mathcal{S}}$. For each $j \in \mathcal{S}$, we compute

$$\mathbf{g}_j = \frac{\partial \mathcal{L}(x, \mathbf{U})}{\partial \mathbf{u}_j} \in \mathbb{R}^d, \tag{8}$$

which measures how changes in the suffix embedding at position j affect the gate objective.

4.3 Gradient-Guided Coordinate Greedy Search

Because the final suffix must be discrete, we adopt a *gradient-guided coordinate-greedy* procedure: in each iteration, for each suffix position $j \in \mathcal{S}$, we select a new token $v \in \mathcal{V}$ to replace the current token s_j.

One-step approximation. Let $W_e[v] \in \mathbb{R}^d$ be the v-th row of the embedding matrix. Under a first-order approximation, replacing s_j with candidate token v changes the loss by

$$\Delta\mathcal{L}_j(v) \approx \mathbf{g}_j^\top \big(W_e[v] - W_e[s_j]\big). \tag{9}$$

Thus, we prefer tokens that minimize $\Delta\mathcal{L}_j(v)$.

Top-k candidates and exact rescoring. To balance efficiency and accuracy, we first score all vocabulary tokens using Eq. 9 and form a top-k candidate set

$$\mathcal{C}_j = \mathrm{TopK}_k\big\{-\Delta\mathcal{L}_j(v) \mid v \in \mathcal{V}\big\}. \tag{10}$$

We then forward-evaluate each $v \in \mathcal{C}_j$ to obtain the exact loss $\mathcal{L}\big(x, s_{j \to v}\big)$, where $s_{j \to v}$ denotes the

suffix with only position j replaced by v. The update is

$$v_j^\star = \arg\min_{v \in \mathcal{C}_j} \mathcal{L}\big(x, s_{j \to v}\big), \qquad s_j \leftarrow v_j^\star. \quad (11)$$

Iterations and early stopping. Starting from a placeholder suffix (e.g., repeated punctuation), we repeat the above updates for at most T iterations. After each iteration, we check the first-step gate. If

$$\arg\max_{t \in \mathcal{T}} p_\theta\big(t \mid x \oplus s\big) = t_t$$

or $p_\theta(t_t \mid x \oplus s)$ exceeds a threshold τ, we stop early and return the current suffix. If no position yields an improving replacement in an iteration, we also terminate and keep the best suffix found. The concrete values of L, T, k, and τ are reported in the experimental setup.

Overall, the proposed Suffix Optimizer is a white-box discrete search method operating on the input suffix only. By freezing all model parameters and exploiting first-step gradient signals at tail positions, it searches for short suffixes that systematically increase the gate's preference for `<think>`. In the next section, we evaluate the resulting attack on multiple binary-gated reasoning models and 11 benchmarks, and further study its transfer-based reuse boundary.

5 Experimental Results

This section evaluates the proposed Suffix Optimizer under the threat model defined above. We first describe the experimental setup, including datasets, metrics, baselines, and attack hyperparameters. We then report three groups of results: (i) gate-override effectiveness and cost impact on 11 reasoning and QA benchmarks, (ii) transfer-based black-box approximation experiments across models and training paradigms, and (iii) a suffix-length ablation analyzing how attack effectiveness and search cost vary with suffix length.

5.1 Experimental Setup

Models, datasets, and metrics. We evaluate three families of binary-gated reasoning models: ThinkLess (Fang et al., 2025), AdaptThink (Zhang et al., 2025a), and AutoThink (Tu et al., 2025). We use 11 reasoning and QA benchmarks: MATH-500, OpenAI-GSM8K, MMLU-Pro, LIVEBENCH-MATH, SUPER-GPQA, GPQA, CHINESE-SIMPLEQA, SIMPLEQA, AIME2024, OLYMPIADBENCH, and PROCBENCH (Hendrycks et al., 2021; Cobbe et al., 2021; Wang et al., 2024b; White et al., 2024; M-A-P Team et al., 2025; Rein et al., 2023; He et al., 2024; Wei et al., 2024; He et al., 2025; Fujisawa et al., 2024; Jia, 2024). Unless otherwise noted, we evaluate 200 examples per dataset. We report SAL, AAL, and ASR, and provide per-model results and task accuracy before and after attack in Appendix B.

Attack configuration. Unless otherwise stated, we use deterministic decoding and optimize a suffix of length $L = 10$. The coordinate-greedy search runs for at most $T = 100$ iterations with top-k candidate rescoring ($k = 5$). We use argmax-based early stopping and do not activate an additional probability threshold; that is, the threshold τ in Section 4 is disabled in the main experiments. The attack optimizes only suffix tokens, while all model parameters remain frozen.

Random baseline and transfer setting. To test whether the effect can be induced by arbitrary text perturbations, we compare our optimized suffix against a random suffix baseline of the same length. In addition to the main white-box setting, we also evaluate a transfer-based black-box approximation, in which a suffix optimized on a source model is reused on a target model without access to the target's parameters, gradients, or internal states.

5.2 Overall Effectiveness: Gate Override Across Models and Benchmarks

To capture both gating behavior and inference cost, we report SAL, AAL, and ASR on all 11 datasets. Table 1 summarizes results averaged over models within each family, while full per-model results are provided in Appendix B.

Table 1 leads to three observations.

(1) Random suffixes rarely trigger Gate-Override. Across all datasets and model families, the random baseline yields near-zero ASR and only negligible changes in AAL relative to SAL. This indicates that Gate-Override is not caused by arbitrary appended text or accidental noise. Instead, the vulnerability is directional: random tails seldom succeed, whereas optimized suffixes much more reliably push the gate toward `<think>`.

(2) Optimized suffixes achieve high ASR, with frequent but uneven cost inflation. Using the optimized suffix, the average ASR across model families and datasets is close to 0.92, and many

Dataset	Suffix	ThinkLess			AdaptThink			AutoThink		
		SAL	AAL	ASR	SAL	AAL	ASR	SAL	AAL	ASR
MATH-500	Ours	2010.5	5433.7	0.91	2424.7	4444.0	0.86	1926.4	2789.7	0.86
	Rand	2010.5	2056.8	0.02	2424.7	2389.5	0.00	1926.4	1989.7	0.01
OpenAI-GSM8K	Ours	1202.4	2324.0	0.95	654.8	1693.4	0.91	934.2	1479.3	0.84
	Rand	1202.4	1187.9	0.00	654.8	712.4	0.03	934.2	898.5	0.00
MMLU-Pro	Ours	4231.3	9061.0	0.94	7168.7	8737.9	0.99	4378.9	5222.2	0.84
	Rand	4231.3	4298.7	0.01	7168.7	7105.3	0.00	4378.9	4421.8	0.02
LIVEBENCH-MATH	Ours	6655.4	13374.5	0.86	7148.3	10041.9	0.89	6193.9	7509.8	0.88
	Rand	6655.4	6723.8	0.02	7148.3	7099.7	0.00	6193.9	6256.4	0.01
SUPER-GPQA	Ours	4063.3	10868.9	0.95	7017.4	8373.8	0.98	3888.3	3830.7	0.86
	Rand	4063.3	3998.7	0.00	7017.4	7089.2	0.03	3888.3	3821.5	0.00
GPQA	Ours	6596.6	18292.3	0.93	11868.0	13343.5	0.96	7164.7	8421.6	0.89
	Rand	6596.6	6654.9	0.02	11868.0	11912.7	0.01	7164.7	7218.9	0.02
CH-SIMPLEQA	Ours	7455.4	8840.1	0.98	3462.9	4050.1	0.85	124.6	1332.1	0.83
	Rand	7455.4	7512.8	0.03	3462.9	3418.7	0.00	124.6	189.7	0.01
SIMPLEQA	Ours	2803.4	6926.8	0.98	2311.5	2353.4	0.97	479.4	1488.8	0.90
	Rand	2803.4	2756.9	0.00	2311.5	2310.2	0.00	479.4	523.8	0.03
AIME2024	Ours	8878.2	15979.9	0.90	16639.8	17862.9	0.99	12993.8	12554.2	0.88
	Rand	8878.2	8927.5	0.01	16639.8	16601.4	0.00	12993.8	12971.9	0.00
OLYMPIADBENCH	Ours	4182.8	15983.5	0.89	10328.5	14170.8	0.92	8541.4	8569.1	0.91
	Rand	4182.8	4237.9	0.02	10328.5	10289.7	0.00	8541.4	8539.6	0.00
PROCBENCH	Ours	20958.2	28555.9	0.93	22458.9	29256.9	0.99	17715.9	21005.0	0.93
	Rand	20958.2	20901.8	0.00	22458.9	22512.7	0.03	17715.9	17789.4	0.02

Table 1: Overall gate-override results on 11 benchmarks, averaged within each model family. "Ours" uses the optimized suffix; "Rand" uses a random suffix of the same length. SAL and AAL denote average thinking tokens before and after suffix appending, and ASR is the gate-override success rate.

model–dataset pairs exceed 0.9, especially in ThinkLess and AdaptThink. In many settings, AAL is also clearly larger than SAL; on several reasoning-heavy datasets, including MATH-500, MMLU-Pro, LIVEBENCH-MATH, GPQA, AIME2024, and OLYMPIADBENCH, the increase is often large and can approach or exceed $2\times$. At the same time, the magnitude of token inflation is not uniform across families and datasets, indicating that gate override and post-gate reasoning length are related but not identical phenomena. This is broadly consistent with the mechanism analysis in Section 3: pushing the first-step margin toward `<think>` often leads the model onto a longer reasoning trajectory.

(3) Vulnerability differs across reasoning families. ThinkLess and AdaptThink generally exhibit higher ASR than AutoThink, although AutoThink still shows non-trivial gate override and, in some individual models, noticeable AAL/SAL inflation. Fine-grained per-model results in Appendix B further show that more conservative gating hyperparameters can reduce ASR but do not eliminate the vulnerability. For completeness, the same appendix also reports task accuracy before and after attack ($\text{Acc}_{\text{clean}}$ / $\text{Acc}_{\text{attack}}$ / ΔAcc), suggesting that the attack primarily affects gating behavior and reasoning cost, with task-accuracy changes typically being smaller in magnitude.

Dataset	AdaptThink			AutoThink		
	SAL	AAL	ASR	SAL	AAL	ASR
MATH-500	2020.1	2781.4	0.37	3030.5	2121.8	0.37
OpenAI-GSM8K	320.9	953.5	0.48	2163.4	1277.8	0.34
MMLU-Pro	3603.9	7330.8	0.83	6331.4	4300.7	0.46
LIVEBENCH-MATH	7230.4	6853.7	0.46	9329.8	6436.6	0.54
SUPER-GPQA	5053.4	7226.6	0.87	6102.9	3590.9	0.51
GPQA	7574.0	11534.4	0.77	10916.5	6852.3	0.54
CH-SIMPLEQA	1791.1	3624.0	0.75	46.4	612.1	0.27
SIMPLEQA	1116.7	2061.1	0.82	814.9	894.3	0.44
AIME2024	10292.4	15760.0	0.91	21261.8	9850.8	0.60
OLYMPIADBENCH	11796.0	10169.9	0.59	11718.5	7794.3	0.65
PROCBENCH	21733.1	24582.3	0.80	25746.0	19644.7	0.62

Table 2: Transfer results of adversarial suffixes across adaptive reasoning model families. Values are averaged over all target models in each family.

5.3 Transfer-Based Black-Box Approximation

We next evaluate transfer under a constrained attacker who optimizes a suffix offline on a single *source* model and then reuses it on other models without accessing any target-model parameters, gradients, or internal states. Specifically, we optimize suffixes on AdaptThink-Qwen-7B-δ-0.05 and AutoThink-DeepScaleR-Stage3, and transfer them to multiple Qwen-1.5B variants with different δ values as well as several Distill-R1 and Skywork distilled models. Table 2 reports family-averaged transfer results; full per-model results are provided in Appendix C.

We treat this transfer setting as a controlled black-box approximation rather than a fully opaque API-only scenario. In commercial systems such as ChatGPT, whether the backend enters a "thinking"

mode may depend on hidden rules, controllers, or hybrid mechanisms, making the gate difficult to observe and evaluate reproducibly. Transfer therefore provides a more controlled way to study black-box reuse risk while keeping the underlying binary-gated mechanism experimentally identifiable.

Table 2 leads to three takeaways.

(1) Transfer is non-trivial, but bounded. The learned suffixes remain effective in many target settings, especially within the same reasoning family. For AdaptThink targets, ASR remains substantial on many datasets and is often accompanied by higher AAL. For AutoThink targets, transfer is weaker on average but still non-trivial. These results suggest that some Gate-Override directions generalize across models, but not uniformly.

(2) Even imperfect transfer can still affect reasoning cost. ASR and token inflation do not always move in lockstep. In several transfer settings, ASR drops relative to the white-box case, yet AAL remains above SAL, indicating that transferred suffixes can still push the model toward longer reasoning without perfectly flipping the gate on every example. In other settings, however, the transferred suffix perturbs the gate more weakly and does not yield consistent cost inflation. This again suggests that gate override and post-gate reasoning length are related but distinct outcomes.

(3) Transfer reflects structural similarity. Transfer is strongest when source and target models share similar gating behavior or decision geometry, such as hyperparameter variants within AdaptThink or teacher–student relations within AutoThink. Conversely, more conservative gating settings or more heterogeneous targets reduce ASR, clarifying the practical boundary of attack reuse.

5.4 Effect of Suffix Length

We further study how attack behavior changes with suffix length. Figure 8 reports ASR, clean output length, attack output length, and suffix search time for representative models on MATH-500, SIMPLEQA, and AIME2024 under suffix lengths ranging from 5 to 40.

Three patterns stand out. First, short suffixes are often sufficient: many model–dataset pairs already achieve high ASR at length 5 or 10, suggesting that Gate-Override does not rely on long perturbations. Second, increasing suffix length does not yield monotonic gains: ASR and attack output length vary across models and tasks, but longer suffixes do not consistently improve attack effectiveness. Third, search cost rises with suffix length, while clean output length remains largely stable. Overall, results suggest that Gate-Override exploits a fragile gating boundary rather than brute-force prompt elongation: short suffixes often suffice, whereas longer suffixes mainly add search cost.

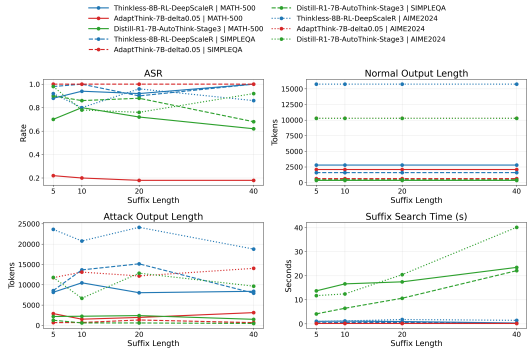

Figure 8: Suffix-length ablation on representative models and datasets. Top-left: ASR. Top-right: clean output length. Bottom-left: attack output length. Bottom-right: search time.

Overall, the main, transfer, and suffix-length results show that GATE-OVERRIDE is a reproducible vulnerability rather than an artifact of random noise. Across many settings, optimized suffixes reliably override the first-step gate and often increase reasoning cost, consistent with the thin, tail-sensitive boundary identified in Section 3.

6 Conclusion

We identify GATE-OVERRIDE, a cost-oriented vulnerability in first-step binary reasoning gates: a short suffix can override the "whether-to-think" decision and often increase inference cost. Through analyses of gating stability, margin distributions, attention, and Suffix Attention Mass (SAM), we show that this vulnerability is associated with a thin decision boundary and disproportionate suffix attention at the gating step. Guided by this analysis, we design a white-box suffix attack that modifies the input suffix via coordinate-greedy search using first-step gradients. Experiments across ThinkLess, AdaptThink, and AutoThink on 11 benchmarks show that optimized suffixes reliably override the gate, while random suffixes rarely do; transfer results reveal a non-trivial but bounded cross-model reuse risk. Overall, our results suggest that gates should be treated as a security- and robustness-critical attack surface.

Limitations

Our work has several limitations. First, the primary threat model is *white-box*: the attacker has access to model parameters and gradients, enabling direct suffix optimization against the first-step gate. Although we additionally study transfer-based black-box approximation, we do not systematically evaluate fully opaque API-only settings. In such systems, whether the backend enters a "thinking" mode may depend on hidden rules, controllers, or hybrid mechanisms, making the gate difficult to observe and evaluate reproducibly.

Second, our study focuses on adaptive reasoning methods with *first-step binary gates*, such as models that explicitly choose between `<short>` and `<think>`. We do not evaluate other adaptive designs, such as continuous gating, hierarchical early stopping, or dynamic subnetwork activation. Accordingly, the vulnerability we identify for thin first-step gates should not be assumed to extend to all adaptive reasoning systems.

Third, our evaluation centers on *inference cost* and *gating behavior*, including thinking-token inflation, gate-override success, and gate fidelity. While we report task accuracy for completeness, we do not study finer-grained behavioral effects such as solution-path diversity, error-type shifts, or long-chain hallucination tendencies. We also do not report end-to-end system metrics such as throughput, tail latency, or energy consumption in a serving environment.

Finally, our contribution is primarily diagnostic rather than defensive. We expose the vulnerability, analyze its mechanism through margin, attention, and SAM, and validate its impact through attacks. We do not develop dedicated defenses for robust gating, such as adversarial regularization, gate architectures that are less sensitive to input-tail perturbations, or multi-step decision schemes. Addressing these defenses, and evaluating them in broader model settings, remains an important direction for future work.

Potential risks. Our method could be misused to artificially inflate inference-time token usage and latency in deployed adaptive reasoning systems, potentially increasing serving cost and reducing system reliability. We present this analysis to expose the vulnerability and motivate more robust gate design and evaluation.

References

Simon A. Aytes, Jinheon Baek, and Sung Ju Hwang. 2025. Sketch-of-thought: Efficient LLM reasoning with adaptive cognitive-inspired sketching. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24296–24320, Suzhou, China. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. 2025. Thinkless: LLM learns when to think. In *Advances in Neural Information Processing Systems*. NeurIPS 2025 poster; arXiv:2505.13379.

Ippei Fujisawa, Sensho Nobe, Hiroki Seto, Rina Onda, Yoshiaki Uchida, Hiroki Ikoma, Pei-Chun Chien, and Ryota Kanai. 2024. ProcBench: Benchmark for multi-step reasoning and following procedure. *arXiv preprint arXiv:2410.03117*.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2025. Token-budget-aware LLM reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24842–24855, Vienna, Austria. Association for Computational Linguistics.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Zhang, Zhe Li, Chenyu Zhang, Yizhao Wang, Xinyu Zhao, Xiaobo Chen, Jing Zhou, and 4 others. 2024. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*.

Yancheng He, Shilong Li, Jiaheng Liu, Yingshui Tan, Weixun Wang, Hui Huang, Xingyuan Bu, Hangyu Guo, Chengwei Hu, Boren Zheng, Zhuoran Lin, Dekai Sun, Zhicheng Zheng, Wenbo Su, and Bo Zheng. 2025. Chinese SimpleQA: A chinese factuality evaluation for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19182–19208, Vienna, Austria. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. *Advances in Neural Information Processing Systems*, 34:5193–5205.

David Huang, Avidan Shah, Alexandre Araujo, David Wagner, and Chawin Sitawarin. 2025. Stronger universal and transferable attacks by suppressing refusals. In *Proceedings of the 2025 Conference of the*

Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 5850–5876, Albuquerque, New Mexico. Association for Computational Linguistics.

Maxwell Jia. 2024. AIME 2024: American invitational mathematics examination 2024 dataset. Hugging Face dataset. Problems from the AIME 2024 competition.

Hyunbin Jin, Je Won Yeom, Seunghyun Bae, and Taesup Kim. 2025. "well, keep thinking": Enhancing LLM reasoning with adaptive injection decoding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9989–10018, Vienna, Austria. Association for Computational Linguistics.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same task, more tokens: the impact of input length on the reasoning performance of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15339–15353, Bangkok, Thailand. Association for Computational Linguistics.

Xingyu Li, Xiaolei Liu, Cheng Liu, Yixiao Xu, Kangyi Ding, Bangzhou Xin, and Jia-Li Yin. 2026. LoopLLM: Transferable energy-latency attacks in LLMs via repetitive generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Also available as arXiv:2511.07876.

Wenhao Lin, Qiong Wu, Shuaixiang Yang, Yiyi Zhou, Xiaoshuai Sun, and Rongrong Ji. 2025. Towards efficient chain-of-thought reasoning via adaptive-budgeting based policy optimization. OpenReview submission to ICLR 2026, withdrawn.

Hongfu Liu, Yuxi Xie, Ye Wang, and Michael Shieh. 2024. Advancing adversarial suffix transfer learning on aligned large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7213–7224, Miami, Florida, USA. Association for Computational Linguistics.

Chenwei Lou, Zewei Sun, Xinnian Liang, Meng Qu, Wei Shen, Wenqi Wang, Yuntao Li, Qingping Yang, and Shuangzhi Wu. 2025. AdaCoT: Pareto-optimal adaptive chain-of-thought triggering via reinforcement learning. *arXiv preprint arXiv:2505.11896*.

Shangke Lyu, Linjuan Wu, Yuchen Yan, Xingyu Wu, Hao Li, Yongliang Shen, Peisheng Jiang, Weiming Lu, Jun Xiao, and Yueting Zhuang. 2025. Hierarchical budget policy optimization for adaptive reasoning. *arXiv preprint arXiv:2507.15844*.

M-A-P Team, Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, Kang Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, Chujie Zheng, Kaixin Deng, Shian Jia, Sichao Jiang, Yiyan Liao, Rui Li, Qinrui Li, Sirun Li, and 77 others. 2025. SuperGPQA: Scaling LLM evaluation across 285 graduate disciplines. In *Advances in Neural Information Processing Systems*. Datasets and Benchmarks Track; arXiv:2502.14739.

Kangyu Qiao, Shaolei Zhang, and Yang Feng. 2025. IG-pruning: Input-guided block pruning for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10618–10629, Suzhou, China. Association for Computational Linguistics.

Meghana Arakkal Rajeev, Rajkumar Ramamurthy, Prapti Trivedi, Vikas Yadav, Oluwanifemi Bamgbose, Sathwik Tejaswi Madhusudhan, James Zou, and Nazneen Rajani. 2025. Cats confuse reasoning LLM: Query-agnostic adversarial triggers for reasoning models. In *Proceedings of the Conference on Language Modeling*. COLM 2025; arXiv:2503.01781.

Anil Ramakrishna, Jimit Majmudar, Rahul Gupta, and Devamanyu Hazarika. 2024. LLM-PIEval: A benchmark for indirect prompt injection attacks in large language models. *NeurIPS 2024 Workshop on Adversarial Machine Learning*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. GPQA: A graduate-level google-proof Q&A benchmark. *arXiv preprint arXiv:2311.12022*.

Songjun Tu, Jiahao Lin, Qichao Zhang, Xiangyu Tian, Linjing Li, Xiangyuan Lan, and Dongbin Zhao. 2025. Learning when to think: Shaping adaptive reasoning in R1-style models via multi-stage RL. *Preprint*, arXiv:2505.10832.

Hao Wang, Hao Li, Minlie Huang, and Lei Sha. 2024a. ASETF: A novel method for jailbreak attack on LLMs through translate suffix embeddings. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2697–2711, Miami, Florida, USA. Association for Computational Linguistics.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024b. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*. Datasets and Benchmarks Track; arXiv:2406.01574.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Siddartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. 2024. LiveBench:

A challenging, contamination-free LLM benchmark. *arXiv preprint arXiv:2406.19314*.

Yue Xu and Wenjie Wang. 2024. LinkPrompt: Natural and universal adversarial attacks on prompt-based language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6473–6486, Mexico City, Mexico. Association for Computational Linguistics.

Zitao Xuan, Xiaofeng Mao, Da Chen, Xin Zhang, Yuhan Dong, and Jun Zhou. 2025. ShieldHead: Decoding-time safeguard for large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18129–18143, Vienna, Austria. Association for Computational Linguistics.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Zheng Lin, Li Cao, and Weiping Wang. 2025. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. 2024. InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10471–10506, Bangkok, Thailand. Association for Computational Linguistics.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. 2025a. AdaptThink: Reasoning models can learn when to think. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3716–3730, Suzhou, China. Association for Computational Linguistics.

Zikai Zhang, Rui Hu, and Jiahao Xu. 2025b. SelfGrader: Detecting jailbreak attacks on large language models with token-level logit distribution. OpenReview submission to ICLR 2026.

Yukai Zhou, Jian Lou, Zhijie Huang, Zhan Qin, Sibei Yang, and Wenjie Wang. 2025. Don't say no: Jailbreaking LLM by suppressing refusal. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25224–25249, Vienna, Austria. Association for Computational Linguistics.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. In *Advances in Neural Information Processing Systems*, volume 36.

A More Related Work

Adaptive reasoning and first-step gating. The line of work most directly related to ours studies *when* a model should invoke long reasoning. ThinkLess introduces explicit `<short>`/`<think>` tokens and jointly optimizes mode selection with answer quality under the DeGRPO objective, substantially reducing long CoT usage on academic reasoning tasks (Fang et al., 2025). AutoThink builds on R1-style distilled models and refines the think/not-think policy through multi-stage reinforcement learning, reducing reasoning tokens while largely maintaining accuracy (Tu et al., 2025). AdaptThink similarly formulates whether to trigger explicit thinking as a policy-learning problem and provides a unified framework across reasoning models (Zhang et al., 2025a). More broadly, AdaCoT, HBPO, and ABPO cast whether to use CoT (and how much of it to use) as a budget-constrained decision problem and optimize the accuracy–cost trade-off under reinforcement or hierarchical policy optimization (Lou et al., 2025; Lyu et al., 2025; Lin et al., 2025). These works establish "learning when to think" as a central problem, but they primarily optimize average cost and accuracy; by contrast, we ask whether the gate itself can be adversarially overridden.

Efficient reasoning and selective computation. Another closely related direction seeks to retain the benefit of reasoning while reducing inference cost. TALE argues that CoT traces contain many redundant tokens and adaptively assigns a reasoning budget at inference time (Han et al., 2025). Sketch-of-Thought replaces verbose natural-language rationales with compact symbolic sketches, reducing intermediate token usage while preserving reasoning performance (Aytes et al., 2025). At the architectural level, input-conditioned pruning methods reduce compute by selecting only a subset of transformer blocks (Qiao et al., 2025), while dynamic early-exit methods stop decoding once intermediate uncertainty signals become sufficiently stable (Yang et al., 2025). These methods aim to make reasoning cheaper; our work studies the complementary security question of whether such cost-control mechanisms themselves can become an attack surface.

Adversarial suffixes, jailbreaks, and performance-oriented attacks. Most adversarial studies on LLMs focus on alignment

bypass or harmful output generation, often through *adversarial suffixes*. A canonical example is GCG, which uses gradient-guided discrete token search to construct universal and transferable jailbreak suffixes (Zou et al., 2023). Subsequent work has improved suffix optimization, readability, and transferability (Wang et al., 2024a; Liu et al., 2024; Huang et al., 2025), while indirect prompt injection extends such risks to agent and tool-use settings (Zhan et al., 2024; Ramakrishna et al., 2024). More closely related, recent trigger-based attacks on reasoning models show that short query-agnostic triggers can degrade reasoning quality or inflate output length (Rajeev et al., 2025), and performance-oriented attacks such as LoopLLM highlight latency and energy amplification via repetitive generation (Li et al., 2026). Our work differs in that it targets the *first-step binary reasoning gate* itself and analyzes computation-budget hijacking through gate-specific mechanisms such as margin shifts and suffix attention mass.

Decoding-time safeguards and interpretable internal signals. Complementing attack studies, recent defenses increasingly rely on interpretable internal signals at decoding time. SelfGrader casts jailbreak detection as a scoring problem over internal logits and extracts scalar safety signals from the model's own token distributions (Zhang et al., 2025b). Decoding-Time Safeguards introduce dynamic safety heads that intervene during generation without modifying the backbone model (Xuan et al., 2025). These works show that internal signals such as logits and attention can support security decisions, but they primarily focus on harmfulness detection. Our work instead studies internal signals tied to *budget hijacking*: we introduce the gating margin and Suffix Attention Mass (SAM) to characterize how short suffixes redirect the gate toward long reasoning, providing a more targeted interpretability toolbox for robust reasoning-budget control.

B Additional Experimental Results

This appendix reports the complete per-model results of the proposed attack on all 11 benchmark datasets, including ThinkLess (Table 3), AdaptThink (Tables 4–5), and AutoThink (Table 6). For each setting, we report SAL, AAL, and ASR under the optimized adversarial suffix, together with task accuracy before and after attack ($\text{Acc}_{\text{clean}}$ / $\text{Acc}_{\text{attack}}$) and the corresponding change ΔAcc. These results complement the family-averaged summaries in the main text and support finer-grained analysis of cost inflation and task-performance changes.

Overall, these appendix results show that GATE-OVERRIDE is a *cost-oriented* attack rather than an accuracy-oriented one. Its primary effect is to force the model into longer reasoning traces and thereby inflate inference cost. Importantly, the attack is constructed to override the reasoning gate, not to directly optimize for incorrect answers, so the resulting accuracy changes are mixed. In several hard benchmarks, especially for smaller models, the clean accuracy is already low due to task difficulty and limited model capability; such cases should not be interpreted as evidence that the attack mainly harms correctness. Instead, the key takeaway is that the attack can substantially increase computation while often preserving answer behavior within a similar range, which makes it a realistic efficiency threat for adaptive reasoning systems.

C Additional Transfer Results

This appendix reports the complete per-model results for the transfer-based black-box approximation experiments in the main paper, including AdaptThink (Tables 7–8) and AutoThink (Table 9). For each target setting, we report SAL, AAL, and ASR under suffixes optimized on the corresponding source model. These results complement the family-averaged summaries in the main text and support finer-grained analysis of cross-model reuse boundaries.

These appendix results further show that transferability in GATE-OVERRIDE is structured rather than unconditional. Transfer is generally stronger within the same family or under similar gate configurations, while it weakens for more conservative or heterogeneous targets. Nevertheless, even when ASR decreases, the transferred suffix can still produce noticeable token inflation, indicating that it may push the target model toward longer reasoning without fully flipping the gate in every case. This supports the main-text claim that transfer-based black-box approximation has practical relevance, while also making its boundary conditions explicit. The accuracy changes remain mixed, which is consistent with our threat model: the transferred suffix primarily perturbs computation-budget control rather than directly optimizing for wrong answers.

Table 3: Complete results of ThinkLess models on the 11 benchmarks. SAL is the average number of thinking tokens under short reasoning, AAL is the average number of thinking tokens after the attack, and ASR is the gate-override success rate. We additionally report task accuracy before/after attack ($\text{Acc}_{\text{clean}}$ / $\text{Acc}_{\text{attack}}$) and the change ΔAcc (in percentage points).

Method	Model	Dataset	SAL	AAL	ASR	$\text{Acc}_{\text{clean}}$	$\text{Acc}_{\text{attack}}$	ΔAcc
ThinkLess	Qwen-1.5B	MATH-500	2620.800	3333.120	0.990	62.00%	67.00%	+5.00%
ThinkLess	Qwen-1.5B	OpenAI-GSM8K	712.740	1286.730	0.980	77.50%	84.50%	+7.00%
ThinkLess	Qwen-1.5B	MMLU-Pro	3937.820	4434.950	1.000	39.50%	40.00%	+0.50%
ThinkLess	Qwen-1.5B	LIVEBENCH-MATH	5558.040	6380.890	1.000	38.50%	47.50%	+9.00%
ThinkLess	Qwen-1.5B	SUPER-GPQA	4955.050	4261.770	1.000	12.00%	11.00%	-1.00%
ThinkLess	Qwen-1.5B	GPQA	9552.180	9673.650	1.000	7.00%	4.50%	-2.50%
ThinkLess	Qwen-1.5B	CHINESE-SIMPLEQA	13877.700	12375.660	1.000	0.50%	2.50%	+2.00%
ThinkLess	Qwen-1.5B	SIMPLEQA	2216.340	3644.820	1.000	1.00%	1.00%	+0.00%
ThinkLess	Qwen-1.5B	AIME2024	8634.940	8596.320	1.000	34.00%	33.50%	-0.50%
ThinkLess	Qwen-1.5B	OLYMPIADBENCH	6104.570	6686.010	1.000	31.50%	30.50%	-1.00%
ThinkLess	Qwen-1.5B	PROCBENCH	20422.350	20640.790	1.000	1.50%	4.00%	+2.50%
ThinkLess	Qwen-7B	MATH-500	2025.475	7046.085	1.000	60.00%	67.00%	+7.00%
ThinkLess	Qwen-7B	OpenAI-GSM8K	1685.035	3474.745	1.000	78.50%	81.50%	+3.00%
ThinkLess	Qwen-7B	MMLU-Pro	2119.835	14120.610	1.000	44.00%	45.50%	+1.50%
ThinkLess	Qwen-7B	LIVEBENCH-MATH	7791.625	17540.265	1.000	36.50%	51.50%	+15.00%
ThinkLess	Qwen-7B	SUPER-GPQA	4884.765	9422.315	0.980	4.00%	13.50%	+9.50%
ThinkLess	Qwen-7B	GPQA	4599.245	8429.965	0.960	7.00%	11.00%	+4.00%
ThinkLess	Qwen-7B	CHINESE-SIMPLEQA	3723.100	8592.895	0.995	2.50%	2.50%	+0.00%
ThinkLess	Qwen-7B	SIMPLEQA	3151.065	8220.255	0.995	2.00%	2.00%	+0.00%
ThinkLess	Qwen-7B	AIME2024	8740.460	10580.100	1.000	14.50%	32.00%	+17.50%
ThinkLess	Qwen-7B	OLYMPIADBENCH	4874.660	19727.290	0.980	25.50%	27.50%	+2.00%
ThinkLess	Qwen-7B	PROCBENCH	20660.330	32769.500	1.000	1.00%	0.00%	-1.00%
ThinkLess	Llama-8B	MATH-500	2329.440	8666.965	0.910	52.50%	59.50%	+7.00%
ThinkLess	Llama-8B	OpenAI-GSM8K	768.635	3659.270	0.985	64.50%	78.00%	+13.50%
ThinkLess	Llama-8B	MMLU-Pro	7241.555	14498.045	0.990	38.50%	33.50%	-5.00%
ThinkLess	Llama-8B	LIVEBENCH-MATH	6986.905	12138.665	0.940	25.50%	42.00%	+16.50%
ThinkLess	Llama-8B	SUPER-GPQA	6141.905	9774.720	0.975	4.00%	13.00%	+9.00%
ThinkLess	Llama-8B	GPQA	5049.145	8553.995	0.930	5.00%	6.00%	+1.00%
ThinkLess	Llama-8B	CHINESE-SIMPLEQA	5782.320	8865.080	0.965	2.00%	2.50%	+0.50%
ThinkLess	Llama-8B	SIMPLEQA	3058.035	9304.890	0.925	1.50%	2.50%	+1.00%
ThinkLess	Llama-8B	AIME2024	13866.860	13732.880	0.940	5.00%	21.50%	+16.50%
ThinkLess	Llama-8B	OLYMPIADBENCH	3882.180	18389.070	0.890	19.50%	27.50%	+8.00%
ThinkLess	Llama-8B	PROCBENCH	13687.400	26125.580	0.650	0.00%	0.00%	+0.00%
ThinkLess	Qwen-14B	MATH-500	1510.580	3868.120	0.775	54.00%	67.00%	+13.00%
ThinkLess	Qwen-14B	OpenAI-GSM8K	2106.235	6739.240	0.870	79.50%	87.00%	+7.50%
ThinkLess	Qwen-14B	MMLU-Pro	7861.605	15174.955	0.960	50.50%	51.00%	+0.50%
ThinkLess	Qwen-14B	LIVEBENCH-MATH	6213.480	11526.905	0.800	33.00%	42.50%	+9.50%
ThinkLess	Qwen-14B	SUPER-GPQA	6470.095	10572.145	0.950	7.00%	10.00%	+3.00%
ThinkLess	Qwen-14B	GPQA	5163.320	8596.120	0.900	9.00%	10.50%	+1.50%
ThinkLess	Qwen-14B	CHINESE-SIMPLEQA	6436.770	10266.795	0.935	9.50%	10.50%	+1.00%
ThinkLess	Qwen-14B	SIMPLEQA	3885.000	12851.000	0.815	2.00%	3.50%	+1.50%
ThinkLess	Qwen-14B	AIME2024	9114.335	11569.080	0.960	26.00%	49.00%	+23.00%
ThinkLess	Qwen-14B	OLYMPIADBENCH	3042.930	12957.200	0.690	20.50%	30.00%	+9.50%
ThinkLess	Qwen-14B	PROCBENCH	19929.780	32068.550	1.000	0.00%	0.00%	+0.00%
ThinkLess	Qwen-32B	MATH-500	1566.270	4253.980	0.895	63.00%	72.50%	+9.50%
ThinkLess	Qwen-32B	OpenAI-GSM8K	739.595	1753.840	0.900	85.50%	86.50%	+1.00%
ThinkLess	Qwen-32B	MMLU-Pro	4992.375	6483.755	0.965	59.00%	70.50%	+11.50%
ThinkLess	Qwen-32B	LIVEBENCH-MATH	11322.460	15422.180	0.940	48.50%	58.00%	+9.50%
ThinkLess	Qwen-32B	SUPER-GPQA	6071.560	10748.235	0.970	10.00%	9.50%	-0.50%
ThinkLess	Qwen-32B	GPQA	4948.980	16424.305	0.890	8.50%	12.50%	+4.00%
ThinkLess	Qwen-32B	CHINESE-SIMPLEQA	4589.795	4948.835	0.970	3.00%	14.50%	+11.50%
ThinkLess	Qwen-32B	SIMPLEQA	3615.900	6265.975	0.980	2.00%	3.50%	+1.50%
ThinkLess	Qwen-32B	AIME2024	4034.475	10537.045	0.620	11.50%	46.00%	+34.50%
ThinkLess	Qwen-32B	OLYMPIADBENCH	3009.420	22157.950	0.900	9.00%	13.00%	+4.00%
ThinkLess	Qwen-32B	PROCBENCH	30090.920	31174.945	0.995	1.00%	1.50%	+0.50%

Table 4: Complete results of AdaptThink models on the 11 benchmarks (Qwen-1.5B-δ0 to Qwen-1.5B-δ0.075). We additionally report task accuracy before/after attack ($\text{Acc}_{\text{clean}}$ / $\text{Acc}_{\text{attack}}$) and ΔAcc (percentage points).

Method	Model	Dataset	SAL	AAL	ASR	$\text{Acc}_{\text{clean}}$	$\text{Acc}_{\text{attack}}$	ΔAcc
AdaptThink	Qwen-1.5B-delta0	MATH-500	3435.240	4256.590	0.970	67.50%	68.50%	+1.00%
AdaptThink	Qwen-1.5B-delta0	OpenAI-GSM8K	630.830	1731.230	0.930	79.50%	77.00%	-2.50%
AdaptThink	Qwen-1.5B-delta0	MMLU-Pro	11056.190	9906.635	1.000	40.50%	38.50%	-2.00%
AdaptThink	Qwen-1.5B-delta0	LIVEBENCH-MATH	11116.470	11019.495	0.995	44.00%	51.00%	+7.00%
AdaptThink	Qwen-1.5B-delta0	SUPER-GPQA	9259.505	11203.830	1.000	9.50%	11.50%	+2.00%
AdaptThink	Qwen-1.5B-delta0	GPQA	17075.709	15893.754	1.000	8.04%	7.54%	-0.50%
AdaptThink	Qwen-1.5B-delta0	CHINESE-SIMPLEQA	4186.115	3868.140	1.000	0.50%	0.00%	-0.50%
AdaptThink	Qwen-1.5B-delta0	SIMPLEQA	3225.085	2926.630	1.000	0.50%	1.00%	+0.50%
AdaptThink	Qwen-1.5B-delta0	AIME2024	17430.425	14858.105	1.000	34.00%	24.50%	-9.50%
AdaptThink	Qwen-1.5B-delta0	OLYMPIADBENCH	16620.925	15370.390	1.000	29.00%	29.00%	+0.00%
AdaptThink	Qwen-1.5B-delta0	PROCBENCH	30689.860	30634.045	1.000	0.50%	0.50%	+0.00%
AdaptThink	Qwen-1.5B-delta0.01	MATH-500	3332.250	6058.515	0.905	33.00%	36.00%	+3.00%
AdaptThink	Qwen-1.5B-delta0.01	OpenAI-GSM8K	625.800	2193.490	0.930	86.00%	81.00%	-5.00%
AdaptThink	Qwen-1.5B-delta0.01	MMLU-Pro	10966.175	10056.960	1.000	35.50%	36.50%	+1.00%
AdaptThink	Qwen-1.5B-delta0.01	LIVEBENCH-MATH	11221.860	11088.325	0.985	38.00%	40.50%	+2.50%
AdaptThink	Qwen-1.5B-delta0.01	SUPER-GPQA	9232.265	11343.785	1.000	14.00%	13.50%	-0.50%
AdaptThink	Qwen-1.5B-delta0.01	GPQA	16971.181	16129.200	1.000	11.50%	8.50%	-3.00%
AdaptThink	Qwen-1.5B-delta0.01	CHINESE-SIMPLEQA	4399.500	4488.910	1.000	1.00%	0.00%	-1.00%
AdaptThink	Qwen-1.5B-delta0.01	SIMPLEQA	2543.670	3156.710	1.000	1.00%	1.50%	+0.50%
AdaptThink	Qwen-1.5B-delta0.01	AIME2024	17462.545	15196.565	1.000	26.50%	30.50%	+4.00%
AdaptThink	Qwen-1.5B-delta0.01	OLYMPIADBENCH	16588.790	16250.880	1.000	22.00%	25.00%	+3.00%
AdaptThink	Qwen-1.5B-delta0.01	PROCBENCH	31440.055	31395.305	1.000	0.00%	0.50%	+0.50%
AdaptThink	Qwen-1.5B-delta0.02	MATH-500	2537.830	4267.210	0.915	45.50%	54.50%	+9.00%
AdaptThink	Qwen-1.5B-delta0.02	OpenAI-GSM8K	630.600	1771.240	0.960	84.00%	81.00%	-3.00%
AdaptThink	Qwen-1.5B-delta0.02	MMLU-Pro	10113.740	9137.075	1.000	31.50%	37.00%	+5.50%
AdaptThink	Qwen-1.5B-delta0.02	LIVEBENCH-MATH	11114.965	9982.035	0.930	42.50%	42.50%	+0.00%
AdaptThink	Qwen-1.5B-delta0.02	SUPER-GPQA	8985.300	11234.690	0.970	18.50%	16.00%	-2.50%
AdaptThink	Qwen-1.5B-delta0.02	GPQA	17074.209	16389.950	0.920	9.00%	8.00%	-1.00%
AdaptThink	Qwen-1.5B-delta0.02	CHINESE-SIMPLEQA	4154.685	3909.305	0.980	0.50%	0.50%	+0.00%
AdaptThink	Qwen-1.5B-delta0.02	SIMPLEQA	2700.975	3969.140	1.000	0.00%	1.50%	+1.50%
AdaptThink	Qwen-1.5B-delta0.02	AIME2024	16002.600	17019.980	1.000	18.50%	35.50%	+17.00%
AdaptThink	Qwen-1.5B-delta0.02	OLYMPIADBENCH	14740.160	17019.980	1.000	27.50%	29.00%	+1.50%
AdaptThink	Qwen-1.5B-delta0.02	PROCBENCH	30827.165	30396.295	1.000	0.50%	0.00%	-0.50%
AdaptThink	Qwen-1.5B-delta0.05	MATH-500	2612.345	4743.665	0.845	39.00%	59.00%	+20.00%
AdaptThink	Qwen-1.5B-delta0.05	OpenAI-GSM8K	643.610	1733.440	0.960	77.50%	73.00%	-4.50%
AdaptThink	Qwen-1.5B-delta0.05	MMLU-Pro	10370.560	10296.735	1.000	32.50%	34.00%	+1.50%
AdaptThink	Qwen-1.5B-delta0.05	LIVEBENCH-MATH	11172.965	10480.390	0.960	40.00%	43.50%	+3.50%
AdaptThink	Qwen-1.5B-delta0.05	SUPER-GPQA	8748.645	11281.180	0.940	8.50%	10.50%	+2.00%
AdaptThink	Qwen-1.5B-delta0.05	GPQA	16732.975	17225.085	0.930	10.00%	12.00%	+2.00%
AdaptThink	Qwen-1.5B-delta0.05	CHINESE-SIMPLEQA	4282.470	4071.585	0.985	0.50%	1.00%	+0.50%
AdaptThink	Qwen-1.5B-delta0.05	SIMPLEQA	2810.655	3973.445	1.000	0.00%	0.50%	+0.50%
AdaptThink	Qwen-1.5B-delta0.05	AIME2024	16218.900	17121.970	1.000	35.50%	14.50%	-21.00%
AdaptThink	Qwen-1.5B-delta0.05	OLYMPIADBENCH	5565.650	14357.470	0.880	18.00%	23.50%	+5.50%
AdaptThink	Qwen-1.5B-delta0.05	PROCBENCH	26774.580	29703.670	1.000	1.50%	0.00%	-1.50%
AdaptThink	Qwen-1.5B-delta0.075	MATH-500	1671.770	4166.180	0.840	14.00%	46.00%	+32.00%
AdaptThink	Qwen-1.5B-delta0.075	OpenAI-GSM8K	599.760	1438.420	0.970	85.50%	78.50%	-7.00%
AdaptThink	Qwen-1.5B-delta0.075	MMLU-Pro	10087.690	10818.145	0.980	23.00%	31.50%	+8.50%
AdaptThink	Qwen-1.5B-delta0.075	LIVEBENCH-MATH	10224.930	10488.815	0.950	26.50%	38.00%	+11.50%
AdaptThink	Qwen-1.5B-delta0.075	SUPER-GPQA	8716.595	10432.220	0.980	12.00%	11.00%	-1.00%
AdaptThink	Qwen-1.5B-delta0.075	GPQA	15366.175	14641.010	0.960	4.00%	6.50%	+2.50%
AdaptThink	Qwen-1.5B-delta0.075	CHINESE-SIMPLEQA	4272.870	4312.470	0.995	0.00%	1.00%	+1.00%
AdaptThink	Qwen-1.5B-delta0.075	SIMPLEQA	2969.670	4612.590	1.000	0.50%	0.50%	+0.00%
AdaptThink	Qwen-1.5B-delta0.075	AIME2024	20597.320	18080.620	0.980	31.50%	18.00%	-13.50%
AdaptThink	Qwen-1.5B-delta0.075	OLYMPIADBENCH	10292.420	14858.100	1.000	8.50%	23.00%	+14.50%
AdaptThink	Qwen-1.5B-delta0.075	PROCBENCH	8427.585	30837.175	0.975	0.50%	0.00%	-0.50%

Table 5: Complete results of AdaptThink models on the 11 benchmarks (Qwen-1.5B-δ0.1 and Qwen-7B-δ-0.05). We additionally report task accuracy before/after attack ($\text{Acc}_{\text{clean}}$ / $\text{Acc}_{\text{attack}}$) and ΔAcc (percentage points).

Method	Model	Dataset	SAL	AAL	ASR	$\text{Acc}_{\text{clean}}$	$\text{Acc}_{\text{attack}}$	ΔAcc
AdaptThink	Qwen-1.5B-delta0.1	MATH-500	1363.165	4466.000	0.755	55.00%	59.00%	+4.00%
AdaptThink	Qwen-1.5B-delta0.1	OpenAI-GSM8K	576.290	1398.920	0.950	74.00%	76.00%	+2.00%
AdaptThink	Qwen-1.5B-delta0.1	MMLU-Pro	9810.040	10589.235	0.980	25.00%	37.50%	+12.50%
AdaptThink	Qwen-1.5B-delta0.1	LIVEBENCH-MATH	9510.415	10597.000	0.960	42.00%	38.50%	-3.50%
AdaptThink	Qwen-1.5B-delta0.1	SUPER-GPQA	9237.715	10840.650	0.940	9.50%	6.50%	-3.00%
AdaptThink	Qwen-1.5B-delta0.1	GPQA	15402.491	14413.914	0.910	5.50%	8.50%	+3.00%
AdaptThink	Qwen-1.5B-delta0.1	CHINESE-SIMPLEQA	4206.495	4363.485	0.980	0.50%	0.00%	-0.50%
AdaptThink	Qwen-1.5B-delta0.1	SIMPLEQA	2747.655	4074.945	1.000	1.00%	1.00%	+0.00%
AdaptThink	Qwen-1.5B-delta0.1	AIME2024	16287.570	17252.830	1.000	20.50%	18.00%	-2.50%
AdaptThink	Qwen-1.5B-delta0.1	OLYMPIADBENCH	10292.420	13413.890	0.990	22.50%	32.00%	+9.50%
AdaptThink	Qwen-1.5B-delta0.1	PROCBENCH	7320.215	28730.930	0.970	1.50%	0.50%	-1.00%
AdaptThink	Qwen-7B-delta-0.05	MATH-500	2020.055	3149.745	0.780	56.50%	56.00%	-0.50%
AdaptThink	Qwen-7B-delta-0.05	OpenAI-GSM8K	320.860	653.970	0.940	86.00%	84.50%	-1.50%
AdaptThink	Qwen-7B-delta-0.05	MMLU-Pro	10581.275	9554.075	0.990	56.00%	51.00%	-5.00%
AdaptThink	Qwen-7B-delta-0.05	LIVEBENCH-MATH	11124.910	10084.605	0.950	45.00%	47.50%	+2.50%
AdaptThink	Qwen-7B-delta-0.05	SUPER-GPQA	9155.050	11142.930	0.990	6.50%	4.50%	-2.00%
AdaptThink	Qwen-7B-delta-0.05	GPQA	16801.241	16680.361	0.940	20.50%	14.00%	-6.50%
AdaptThink	Qwen-7B-delta-0.05	CHINESE-SIMPLEQA	4021.480	4752.245	0.990	2.00%	0.50%	-1.50%
AdaptThink	Qwen-7B-delta-0.05	SIMPLEQA	3151.065	6230.970	0.990	1.00%	1.00%	+0.00%
AdaptThink	Qwen-7B-delta-0.05	AIME2024	10292.420	11490.900	1.000	59.00%	42.00%	-17.00%
AdaptThink	Qwen-7B-delta-0.05	OLYMPIADBENCH	11796.040	13211.560	0.960	34.00%	35.00%	+1.00%
AdaptThink	Qwen-7B-delta-0.05	PROCBENCH	21733.065	23100.650	1.000	4.50%	1.50%	-3.00%

Table 6: Complete results of AutoThink models on the 11 benchmarks. We additionally report task accuracy before/after attack ($\text{Acc}_{\text{clean}}$ / $\text{Acc}_{\text{attack}}$) and ΔAcc (percentage points).

Method	Model	Dataset	SAL	AAL	ASR	$\text{Acc}_{\text{clean}}$	$\text{Acc}_{\text{attack}}$	ΔAcc
AutoThink	Distill-R1-1.5B-Stage3	MATH-500	2588.730	2950.840	0.955	82.00%	71.00%	-11.00%
AutoThink	Distill-R1-1.5B-Stage3	OpenAI-GSM8K	639.250	1403.710	0.950	83.50%	76.50%	-7.00%
AutoThink	Distill-R1-1.5B-Stage3	MMLU-Pro	8371.920	8491.190	1.000	35.50%	38.00%	+2.50%
AutoThink	Distill-R1-1.5B-Stage3	LIVEBENCH-MATH	8377.670	10190.940	0.950	47.50%	49.50%	+2.00%
AutoThink	Distill-R1-1.5B-Stage3	SUPER-GPQA	6287.580	9790.095	1.000	9.50%	10.00%	+0.50%
AutoThink	Distill-R1-1.5B-Stage3	GPQA	4877.620	18337.970	0.960	9.00%	8.00%	-1.00%
AutoThink	Distill-R1-1.5B-Stage3	CHINESE-SIMPLEQA	3260.370	4834.240	0.985	0.00%	1.00%	+1.00%
AutoThink	Distill-R1-1.5B-Stage3	SIMPLEQA	3549.245	5181.995	1.000	1.50%	0.50%	-1.00%
AutoThink	Distill-R1-1.5B-Stage3	AIME2024	11054.330	11343.960	1.000	29.50%	23.50%	-6.00%
AutoThink	Distill-R1-1.5B-Stage3	OLYMPIADBENCH	12754.190	19176.240	1.000	36.50%	33.00%	-3.50%
AutoThink	Distill-R1-1.5B-Stage3	PROCBENCH	29991.845	29534.645	1.000	1.50%	1.50%	+0.00%
AutoThink	Distill-R1-7B-Stage3	MATH-500	753.970	1897.485	0.770	73.00%	74.00%	+1.00%
AutoThink	Distill-R1-7B-Stage3	OpenAI-GSM8K	280.440	721.680	0.750	84.50%	81.50%	-3.00%
AutoThink	Distill-R1-7B-Stage3	MMLU-Pro	6203.360	2823.740	0.940	65.00%	65.00%	+0.00%
AutoThink	Distill-R1-7B-Stage3	LIVEBENCH-MATH	5877.475	5184.355	0.845	58.00%	57.50%	-0.50%
AutoThink	Distill-R1-7B-Stage3	SUPER-GPQA	5667.720	10317.095	0.985	4.50%	2.50%	-2.00%
AutoThink	Distill-R1-7B-Stage3	GPQA	3687.080	9328.980	0.930	18.00%	15.50%	-2.50%
AutoThink	Distill-R1-7B-Stage3	CHINESE-SIMPLEQA	2568.135	8569.270	0.970	0.50%	1.00%	+0.50%
AutoThink	Distill-R1-7B-Stage3	SIMPLEQA	3334.675	3660.905	0.980	1.00%	1.50%	+0.50%
AutoThink	Distill-R1-7B-Stage3	AIME2024	6205.210	7572.760	0.790	41.50%	52.00%	+10.50%
AutoThink	Distill-R1-7B-Stage3	OLYMPIADBENCH	10257.310	7603.800	0.750	34.00%	37.00%	+3.00%
AutoThink	Distill-R1-7B-Stage3	PROCBENCH	11839.120	20404.645	0.895	2.00%	0.50%	-1.50%
AutoThink	DeepScaleR-Stage3	MATH-500	3030.545	4032.040	0.965	57.50%	68.00%	+10.50%
AutoThink	DeepScaleR-Stage3	OpenAI-GSM8K	2163.390	2897.770	0.930	81.50%	79.50%	-2.00%
AutoThink	DeepScaleR-Stage3	MMLU-Pro	6331.360	7714.000	0.900	38.00%	40.00%	+2.00%
AutoThink	DeepScaleR-Stage3	LIVEBENCH-MATH	5765.130	8571.960	0.900	50.00%	48.50%	-1.50%
AutoThink	DeepScaleR-Stage3	SUPER-GPQA	5221.895	10998.305	0.980	8.00%	7.00%	-1.00%
AutoThink	DeepScaleR-Stage3	GPQA	3706.235	9805.595	0.955	10.50%	7.50%	-3.00%
AutoThink	DeepScaleR-Stage3	CHINESE-SIMPLEQA	3011.070	10012.230	0.995	0.00%	1.00%	+1.00%
AutoThink	DeepScaleR-Stage3	SIMPLEQA	3261.620	5484.210	1.000	1.00%	0.00%	-1.00%
AutoThink	DeepScaleR-Stage3	AIME2024	11718.550	9756.770	0.990	39.50%	36.00%	-3.50%
AutoThink	DeepScaleR-Stage3	OLYMPIADBENCH	21139.970	18095.360	1.000	27.00%	30.50%	+3.50%
AutoThink	DeepScaleR-Stage3	PROCBENCH	25745.980	26372.550	1.000	1.50%	0.50%	-1.00%
AutoThink	Skywork-7B-Stage3	MATH-500	1332.480	2278.330	0.740	75.50%	79.50%	+4.00%
AutoThink	Skywork-7B-Stage3	OpenAI-GSM8K	653.760	894.210	0.760	82.50%	85.50%	+3.00%
AutoThink	Skywork-7B-Stage3	MMLU-Pro	4360.655	2701.420	0.920	52.50%	44.00%	-8.50%
AutoThink	Skywork-7B-Stage3	LIVEBENCH-MATH	5864.180	5020.590	0.850	54.00%	52.00%	-2.00%
AutoThink	Skywork-7B-Stage3	SUPER-GPQA	5880.015	10083.185	0.965	12.00%	10.50%	-1.50%
AutoThink	Skywork-7B-Stage3	GPQA	4336.930	8961.035	0.925	17.00%	20.50%	+3.50%
AutoThink	Skywork-7B-Stage3	CHINESE-SIMPLEQA	2999.865	7059.390	0.925	0.50%	2.50%	+2.00%
AutoThink	Skywork-7B-Stage3	SIMPLEQA	2869.775	3822.025	0.945	1.50%	3.00%	+1.50%
AutoThink	Skywork-7B-Stage3	AIME2024	5187.410	5602.980	0.850	56.00%	55.50%	-0.50%
AutoThink	Skywork-7B-Stage3	OLYMPIADBENCH	7824.690	5341.280	0.790	49.00%	43.00%	-6.00%
AutoThink	Skywork-7B-Stage3	PROCBENCH	3286.815	7708.020	0.815	11.00%	7.00%	-4.00%

Table 7: Transfer results of the adversarial suffix across the AdaptThink family from Qwen-1.5B-$\delta 0$ to Qwen-1.5B-$\delta 0.075$. SAL is the average number of thinking tokens under short reasoning on the source model, AAL is the average number of thinking tokens after the attack on the transferred target, and ASR is the gate-override success rate. We additionally report task accuracy before/after transfer attack ($\text{Acc}_{\text{clean}}$ / $\text{Acc}_{\text{attack}}$) and ΔAcc (percentage points).

Method	Model	Dataset	SAL	AAL	ASR	$\text{Acc}_{\text{clean}}$	$\text{Acc}_{\text{attack}}$	ΔAcc
AdaptThink	Qwen-1.5B-delta0	MATH-500	2020.055	3715.400	0.630	56.50%	63.00%	+6.50%
AdaptThink	Qwen-1.5B-delta0	OpenAI-GSM8K	320.860	1636.585	0.530	86.00%	81.50%	-4.50%
AdaptThink	Qwen-1.5B-delta0	MMLU-Pro	3603.910	10035.735	1.000	56.00%	35.50%	-20.50%
AdaptThink	Qwen-1.5B-delta0	LIVEBENCH-MATH	7230.445	10245.835	0.750	34.50%	36.00%	+1.50%
AdaptThink	Qwen-1.5B-delta0	SUPER-GPQA	5053.415	9536.315	1.000	6.50%	13.00%	+6.50%
AdaptThink	Qwen-1.5B-delta0	GPQA	7590.111	15649.025	1.000	14.57%	8.54%	-6.03%
AdaptThink	Qwen-1.5B-delta0	CHINESE-SIMPLEQA	1791.140	3545.085	0.855	1.50%	0.00%	-1.50%
AdaptThink	Qwen-1.5B-delta0	SIMPLEQA	1116.735	2752.230	0.980	1.00%	1.00%	+0.00%
AdaptThink	Qwen-1.5B-delta0	AIME2024	10292.420	14858.105	1.000	57.00%	18.00%	-39.00%
AdaptThink	Qwen-1.5B-delta0	OLYMPIADBENCH	11796.040	15181.750	0.965	34.00%	28.00%	-6.00%
AdaptThink	Qwen-1.5B-delta0	PROCBENCH	21733.065	30634.045	1.000	4.50%	0.50%	-4.00%
AdaptThink	Qwen-1.5B-delta0.01	MATH-500	2020.055	4413.240	0.435	56.50%	22.00%	-34.50%
AdaptThink	Qwen-1.5B-delta0.01	OpenAI-GSM8K	320.860	1145.320	0.555	86.00%	82.50%	-3.50%
AdaptThink	Qwen-1.5B-delta0.01	MMLU-Pro	3603.910	10388.020	1.000	56.00%	36.50%	-19.50%
AdaptThink	Qwen-1.5B-delta0.01	LIVEBENCH-MATH	7230.445	9718.520	0.705	34.50%	22.50%	-12.00%
AdaptThink	Qwen-1.5B-delta0.01	SUPER-GPQA	5053.415	9682.650	1.000	6.50%	14.00%	+7.50%
AdaptThink	Qwen-1.5B-delta0.01	GPQA	7590.111	15847.595	1.000	15.00%	4.50%	-10.50%
AdaptThink	Qwen-1.5B-delta0.01	CHINESE-SIMPLEQA	1791.140	4300.560	0.920	1.50%	0.00%	-1.50%
AdaptThink	Qwen-1.5B-delta0.01	SIMPLEQA	1116.735	2958.500	0.995	1.00%	1.50%	+0.50%
AdaptThink	Qwen-1.5B-delta0.01	AIME2024	10292.420	15196.565	1.000	57.00%	27.50%	-29.50%
AdaptThink	Qwen-1.5B-delta0.01	OLYMPIADBENCH	11796.040	15832.440	0.920	34.00%	21.00%	-13.00%
AdaptThink	Qwen-1.5B-delta0.01	PROCBENCH	21733.065	31395.305	1.000	4.50%	0.50%	-4.00%
AdaptThink	Qwen-1.5B-delta0.02	MATH-500	2020.055	2236.810	0.275	56.50%	39.00%	-17.50%
AdaptThink	Qwen-1.5B-delta0.02	OpenAI-GSM8K	320.860	762.580	0.375	86.00%	84.00%	-2.00%
AdaptThink	Qwen-1.5B-delta0.02	MMLU-Pro	3603.910	10145.450	0.970	56.00%	36.50%	-19.50%
AdaptThink	Qwen-1.5B-delta0.02	LIVEBENCH-MATH	7230.445	9213.355	0.720	34.50%	22.50%	-12.00%
AdaptThink	Qwen-1.5B-delta0.02	SUPER-GPQA	5053.415	9712.215	0.985	6.50%	16.00%	+9.50%
AdaptThink	Qwen-1.5B-delta0.02	GPQA	7590.111	15175.840	0.960	15.00%	6.00%	-9.00%
AdaptThink	Qwen-1.5B-delta0.02	CHINESE-SIMPLEQA	1791.140	3030.090	0.835	1.50%	0.50%	-1.00%
AdaptThink	Qwen-1.5B-delta0.02	SIMPLEQA	1116.735	2716.910	0.965	1.00%	2.00%	+1.00%
AdaptThink	Qwen-1.5B-delta0.02	AIME2024	10292.420	15027.680	0.870	57.00%	27.00%	-30.00%
AdaptThink	Qwen-1.5B-delta0.02	OLYMPIADBENCH	11796.040	15027.680	0.870	34.00%	25.50%	-8.50%
AdaptThink	Qwen-1.5B-delta0.02	PROCBENCH	21733.065	30386.510	1.000	4.50%	0.50%	-4.00%
AdaptThink	Qwen-1.5B-delta0.05	MATH-500	2020.055	2429.055	0.220	56.50%	36.00%	-20.50%
AdaptThink	Qwen-1.5B-delta0.05	OpenAI-GSM8K	320.860	558.185	0.310	86.00%	81.00%	-5.00%
AdaptThink	Qwen-1.5B-delta0.05	MMLU-Pro	3603.910	8327.815	0.945	56.00%	32.50%	-23.50%
AdaptThink	Qwen-1.5B-delta0.05	LIVEBENCH-MATH	7230.445	9145.570	0.685	34.50%	17.50%	-17.00%
AdaptThink	Qwen-1.5B-delta0.05	SUPER-GPQA	5053.415	9694.745	0.985	6.50%	11.50%	+5.00%
AdaptThink	Qwen-1.5B-delta0.05	GPQA	7590.111	15040.915	0.955	14.57%	8.04%	-6.53%
AdaptThink	Qwen-1.5B-delta0.05	CHINESE-SIMPLEQA	1791.140	3509.555	0.860	1.50%	1.00%	-0.50%
AdaptThink	Qwen-1.5B-delta0.05	SIMPLEQA	1116.735	2636.185	0.920	1.00%	0.50%	-0.50%
AdaptThink	Qwen-1.5B-delta0.05	AIME2024	10292.420	4803.940	0.200	57.00%	14.50%	-42.50%
AdaptThink	Qwen-1.5B-delta0.05	OLYMPIADBENCH	11796.040	4803.940	0.200	34.00%	16.50%	-17.50%
AdaptThink	Qwen-1.5B-delta0.05	PROCBENCH	21733.065	30010.930	1.000	4.50%	0.50%	-4.00%
AdaptThink	Qwen-1.5B-delta0.075	MATH-500	2020.055	2378.160	0.200	56.50%	24.50%	-32.00%
AdaptThink	Qwen-1.5B-delta0.075	OpenAI-GSM8K	320.860	986.550	0.490	86.00%	80.00%	-6.00%
AdaptThink	Qwen-1.5B-delta0.075	MMLU-Pro	3603.910	10301.865	0.990	56.00%	26.50%	-29.50%
AdaptThink	Qwen-1.5B-delta0.075	LIVEBENCH-MATH	7230.445	8783.620	0.875	34.50%	16.00%	-18.50%
AdaptThink	Qwen-1.5B-delta0.075	SUPER-GPQA	5053.415	9896.405	0.990	6.50%	9.50%	+3.00%
AdaptThink	Qwen-1.5B-delta0.075	GPQA	7590.111	14493.575	0.930	14.57%	4.02%	-10.55%
AdaptThink	Qwen-1.5B-delta0.075	CHINESE-SIMPLEQA	1791.140	2882.940	0.845	1.50%	1.00%	-0.50%
AdaptThink	Qwen-1.5B-delta0.075	SIMPLEQA	1116.735	2887.245	0.955	1.00%	0.00%	-1.00%
AdaptThink	Qwen-1.5B-delta0.075	AIME2024	10292.420	5827.480	0.210	57.00%	19.50%	-37.50%
AdaptThink	Qwen-1.5B-delta0.075	OLYMPIADBENCH	11796.040	5827.480	0.210	34.00%	12.50%	-21.50%
AdaptThink	Qwen-1.5B-delta0.075	PROCBENCH	21733.065	21616.860	0.610	4.50%	0.00%	-4.50%

Table 8: Transfer results of the adversarial suffix on the AdaptThink family: Qwen-1.5B-δ0.1 and Qwen-7B-δ-0.05.

Method	Model	Dataset	SAL	AAL	ASR	$\text{Acc}_{\text{clean}}$	$\text{Acc}_{\text{attack}}$	ΔAcc
AdaptThink	Qwen-1.5B-delta0.1	MATH-500	2020.055	1147.690	0.080	56.50%	51.00%	-5.50%
AdaptThink	Qwen-1.5B-delta0.1	OpenAI-GSM8K	320.860	983.280	0.520	86.00%	71.50%	-14.50%
AdaptThink	Qwen-1.5B-delta0.1	MMLU-Pro	3603.910	10183.435	0.985	56.00%	24.00%	-32.00%
AdaptThink	Qwen-1.5B-delta0.1	LIVEBENCH-MATH	7230.445	7965.070	0.870	34.50%	23.50%	-11.00%
AdaptThink	Qwen-1.5B-delta0.1	SUPER-GPQA	5053.415	9505.835	0.965	6.50%	12.00%	+5.50%
AdaptThink	Qwen-1.5B-delta0.1	GPQA	7590.111	14537.105	0.935	15.00%	5.50%	-9.50%
AdaptThink	Qwen-1.5B-delta0.1	CHINESE-SIMPLEQA	1791.140	3073.880	0.825	1.50%	0.50%	-1.00%
AdaptThink	Qwen-1.5B-delta0.1	SIMPLEQA	1116.735	2831.880	0.945	1.00%	1.00%	+0.00%
AdaptThink	Qwen-1.5B-delta0.1	AIME2024	10292.420	3447.760	0.080	57.00%	17.00%	-40.00%
AdaptThink	Qwen-1.5B-delta0.1	OLYMPIADBENCH	11796.040	3447.760	0.080	34.00%	25.50%	-8.50%
AdaptThink	Qwen-1.5B-delta0.1	PROCBENCH	21733.065	4931.710	0.000	4.50%	2.00%	-2.50%
AdaptThink	Qwen-7B-delta-0.05	MATH-500	2020.055	3149.745	0.780	56.50%	56.00%	-0.50%
AdaptThink	Qwen-7B-delta-0.05	OpenAI-GSM8K	320.860	859.770	0.780	86.00%	84.50%	-1.50%
AdaptThink	Qwen-7B-delta-0.05	MMLU-Pro	3603.910	10017.640	0.990	56.00%	51.00%	-5.00%
AdaptThink	Qwen-7B-delta-0.05	LIVEBENCH-MATH	7230.445	8666.370	0.920	45.00%	47.50%	+2.50%
AdaptThink	Qwen-7B-delta-0.05	SUPER-GPQA	5053.415	9744.465	0.990	6.50%	4.50%	-2.00%
AdaptThink	Qwen-7B-delta-0.05	GPQA	7590.111	14996.355	0.940	20.50%	14.00%	-6.50%
AdaptThink	Qwen-7B-delta-0.05	CHINESE-SIMPLEQA	1791.140	2865.295	0.885	2.00%	0.50%	-1.50%
AdaptThink	Qwen-7B-delta-0.05	SIMPLEQA	1116.735	2579.965	0.930	1.00%	1.00%	+0.00%
AdaptThink	Qwen-7B-delta-0.05	AIME2024	10292.420	15948.960	1.000	59.00%	42.00%	-17.00%
AdaptThink	Qwen-7B-delta-0.05	OLYMPIADBENCH	11796.040	13211.560	0.960	34.00%	35.00%	+1.00%
AdaptThink	Qwen-7B-delta-0.05	PROCBENCH	21733.065	23100.650	1.000	4.50%	1.50%	-3.00%

Table 9: Transfer results of the adversarial suffix across AutoThink models.

Method	Model	Dataset	SAL	AAL	ASR	$\text{Acc}_{\text{clean}}$	$\text{Acc}_{\text{attack}}$	ΔAcc
AutoThink	Distill-R1-1.5B-Stage3	MATH-500	3030.545	2517.740	0.440	57.50%	75.00%	+17.50%
AutoThink	Distill-R1-1.5B-Stage3	OpenAI-GSM8K	2163.390	2091.110	0.380	81.50%	77.50%	-4.00%
AutoThink	Distill-R1-1.5B-Stage3	MMLU-Pro	6331.360	8477.910	0.820	38.00%	39.50%	+1.50%
AutoThink	Distill-R1-1.5B-Stage3	LIVEBENCH-MATH	5765.130	7873.940	0.820	37.00%	41.50%	+4.50%
AutoThink	Distill-R1-1.5B-Stage3	SUPER-GPQA	5221.895	10039.645	0.980	8.00%	8.00%	+0.00%
AutoThink	Distill-R1-1.5B-Stage3	GPQA	3706.235	9869.585	0.905	6.00%	8.00%	+2.00%
AutoThink	Distill-R1-1.5B-Stage3	CHINESE-SIMPLEQA	3011.070	7003.965	0.865	0.00%	0.50%	+0.50%
AutoThink	Distill-R1-1.5B-Stage3	SIMPLEQA	3261.620	6142.525	0.950	1.00%	0.50%	-0.50%
AutoThink	Distill-R1-1.5B-Stage3	AIME2024	11718.550	13739.830	0.800	30.00%	22.00%	-8.00%
AutoThink	Distill-R1-1.5B-Stage3	OLYMPIADBENCH	21139.970	13739.830	0.800	27.00%	34.50%	+7.50%
AutoThink	Distill-R1-1.5B-Stage3	PROCBENCH	25745.980	26458.660	0.780	1.50%	0.50%	-1.00%
AutoThink	Distill-R1-7B-Stage3	MATH-500	3030.545	507.870	0.010	57.50%	72.00%	+14.50%
AutoThink	Distill-R1-7B-Stage3	OpenAI-GSM8K	2163.390	856.930	0.390	81.50%	76.00%	-5.50%
AutoThink	Distill-R1-7B-Stage3	MMLU-Pro	6331.360	7866.980	0.800	38.00%	60.00%	+22.00%
AutoThink	Distill-R1-7B-Stage3	LIVEBENCH-MATH	5765.130	4942.710	0.770	37.00%	40.00%	+3.00%
AutoThink	Distill-R1-7B-Stage3	SUPER-GPQA	5221.895	9984.265	0.990	8.00%	5.50%	-2.50%
AutoThink	Distill-R1-7B-Stage3	GPQA	3706.235	9687.640	0.905	6.00%	9.00%	+3.00%
AutoThink	Distill-R1-7B-Stage3	CHINESE-SIMPLEQA	3011.070	6721.915	0.850	0.00%	1.50%	+1.50%
AutoThink	Distill-R1-7B-Stage3	SIMPLEQA	3261.620	6060.765	0.940	1.00%	0.50%	-0.50%
AutoThink	Distill-R1-7B-Stage3	AIME2024	11718.550	3843.770	0.270	28.50%	33.50%	+5.00%
AutoThink	Distill-R1-7B-Stage3	OLYMPIADBENCH	21139.970	3843.770	0.270	27.00%	37.00%	+10.00%
AutoThink	Distill-R1-7B-Stage3	PROCBENCH	25745.980	21501.050	0.690	1.50%	0.00%	-1.50%
AutoThink	DeepScaleR-Stage3	MATH-500	3030.545	4032.040	0.965	57.50%	68.00%	+10.50%
AutoThink	DeepScaleR-Stage3	OpenAI-GSM8K	2163.390	2897.770	0.930	81.50%	79.50%	-2.00%
AutoThink	DeepScaleR-Stage3	MMLU-Pro	6331.360	7714.000	0.900	38.00%	40.00%	+2.00%
AutoThink	DeepScaleR-Stage3	LIVEBENCH-MATH	5765.130	8589.880	0.900	50.00%	48.50%	-1.50%
AutoThink	DeepScaleR-Stage3	SUPER-GPQA	5221.895	11149.440	0.980	8.00%	7.00%	-1.00%
AutoThink	DeepScaleR-Stage3	GPQA	3706.235	9858.950	0.910	10.50%	7.50%	-3.00%
AutoThink	DeepScaleR-Stage3	CHINESE-SIMPLEQA	3011.070	8771.450	0.945	0.00%	1.00%	+1.00%
AutoThink	DeepScaleR-Stage3	SIMPLEQA	3261.620	6611.020	0.980	1.00%	0.00%	-1.00%
AutoThink	DeepScaleR-Stage3	AIME2024	11718.550	17037.560	0.900	39.50%	36.00%	-3.50%
AutoThink	DeepScaleR-Stage3	OLYMPIADBENCH	21139.970	17037.560	0.900	27.00%	30.50%	+3.50%
AutoThink	DeepScaleR-Stage3	PROCBENCH	25745.980	26372.550	1.000	1.50%	0.50%	-1.00%
AutoThink	Skywork-7B-Stage3	MATH-500	3030.545	1429.530	0.080	57.50%	89.50%	+32.00%
AutoThink	Skywork-7B-Stage3	OpenAI-GSM8K	2163.390	671.170	0.030	81.50%	84.50%	+3.00%
AutoThink	Skywork-7B-Stage3	MMLU-Pro	6331.360	1008.040	0.090	38.00%	43.00%	+5.00%
AutoThink	Skywork-7B-Stage3	LIVEBENCH-MATH	5765.130	4060.690	0.510	37.00%	26.00%	-11.00%
AutoThink	Skywork-7B-Stage3	SUPER-GPQA	5221.895	10063.520	0.990	8.00%	13.50%	+5.50%
AutoThink	Skywork-7B-Stage3	GPQA	3706.235	10016.660	0.965	6.00%	13.00%	+7.00%
AutoThink	Skywork-7B-Stage3	CHINESE-SIMPLEQA	3011.070	6614.150	0.900	0.00%	0.00%	+0.00%
AutoThink	Skywork-7B-Stage3	SIMPLEQA	3261.620	6223.140	0.955	1.00%	0.50%	-0.50%
AutoThink	Skywork-7B-Stage3	AIME2024	11718.550	14319.650	0.990	28.50%	49.50%	+21.00%
AutoThink	Skywork-7B-Stage3	OLYMPIADBENCH	21139.970	14319.650	0.990	27.00%	50.50%	+23.50%
AutoThink	Skywork-7B-Stage3	PROCBENCH	25745.980	4246.700	0.000	1.50%	2.50%	+1.00%