

# ParentingLLM: Pacing Reasoning Maturity for Bidirectional Test-Time Scaling under Token Budgets

Seunghwan Jeong<sup>1,2</sup>, Sugi Lee<sup>1</sup>, Wei-Jin Park<sup>1</sup>, Ikjun Yeom<sup>1,2</sup>, Honguk Woo<sup>2</sup>

<sup>1</sup>Acryl Inc.,

<sup>2</sup>Department of Computer Science and Engineering, Sungkyunkwan University  
{levi, daniel, ijeom, jin}@acryl.ai, hwoo@skku.edu

## Abstract

Reasoning LLMs under fixed token budgets often commit before verification finishes, or spend the budget on deliberation that never reaches a final answer. Existing inference-time controllers regulate reasoning *length*, which does not indicate whether the current prefix is ready to be finalized. We introduce PARENTINGLLM, an inference-time controller that paces *reasoning maturity*, defined as the prefix’s readiness to commit given the remaining budget, against a user-specified target schedule, without fine-tuning the underlying LLM. The maturity estimate combines consumed budget with a forked *forced-commit probe*; the controller injects WAIT when the estimate runs ahead of the schedule and HURRY when it lags. On MATH-500, GPQA-Diamond, AIME 24+25, and GSM-Hard with three Gemma-4 models and Qwen3.5-9B, PARENTINGLLM improves over prior inference-time baselines, with the largest gains on AIME 24+25 under a tight budget (−22.6 burst rate, +4.0 to +7.3 accuracy at  $B = 1024$ ).

## 1 Introduction

Reasoning models such as OpenAI’s o1/o3 series (OpenAI, 2024, 2025), DeepSeek-R1 (Guo et al., 2025), Qwen3 (Yang et al., 2025a), and Gemma 4 (Google, 2026) improve mathematical and scientific reasoning by generating longer chains of intermediate steps (Muennighoff et al., 2025). Allocating more compute at inference, i.e., *test-time scaling*, has become an alternative to scaling model size.

Longer reasoning, however, is not free. Each generated token extends autoregressive decoding, consumes additional GPU compute and KV-cache memory, and contributes directly to serving latency. Modern LLM serving systems therefore optimize scheduling and memory management under throughput and latency/SLO constraints (Yu et al., 2022; Kwon et al., 2023; Zhong et al., 2024),

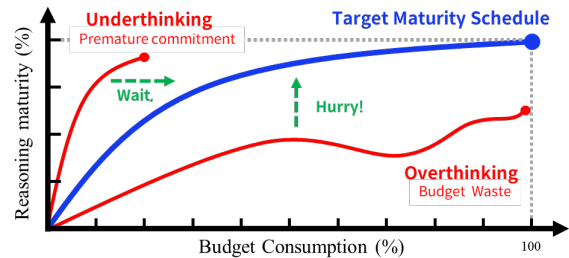


Figure 1: Two symmetric failures (red lines) of fixed-budget reasoning. Above: *underthinking*: premature commitment. Below: *overthinking*: budget wasted before finalization. Our method injects WAIT/HURRY cues to pace maturity along the target schedule (blue).

making uncontrolled generation length a practical systems concern. A response that exceeds its budget may be truncated mid-reasoning and provide little utility. The deployment-relevant question is therefore not *how much* a model can think in the abstract, but *how effectively* it uses a fixed reasoning budget.

A reasoning LLM’s decision to terminate involves two quantities: *what* the candidate answer is, and *whether the model is ready to commit to it*. Existing termination signals conflate these. Next-token entropy (Wang et al., 2025a) peaks mid-trace, since a model performing late-stage verification can hedge cautiously even while its candidate is stable. Answer confidence from a forced trial answer (Yang et al., 2025b) captures certainty in the candidate, but reasoning models routinely hold a high-confidence candidate while continuing to self-verify. Inter-boundary consistency (Fu et al., 2024) flags when the candidate stops changing, but a stable candidate under ongoing verification reflects cautious decoding rather than readiness to stop. None of these asks whether the model itself would terminate cleanly if forced to. A model that is ready to commit will assign high probability to ending its turn once a candidate answer is in place; a model still mid-verification will not, even when the same

candidate is on the table. This distinction is observable from the model’s own distribution and does not require a separate confidence estimate. We define a signal that asks exactly this:  $P_{\text{EOT}}$ , the end-of-turn probability after a forced-commit suffix.

This gap motivates the two failure modes we address (Fig. 1). *Underthinking* occurs when the model becomes willing to finalize too early, committing after only superficial verification and leaving useful budget unused. *Overthinking*, in our usage, is a trace-level budget failure: the model spends tokens on low-yield deliberation, redundant checks, or unproductive branches, so its commitment readiness fails to keep pace with the budget and the response risks truncation before finalization.<sup>1</sup> We argue that both failures are best understood as failures of *commitment readiness under budget*, not merely of length. We call this latent state *reasoning maturity*.

We propose PARENTINGLLM, an inference-time controller that paces reasoning maturity under a fixed token budget against a user-specified target schedule. At each reasoning boundary, PARENTINGLLM applies a *forced-commit probe* to read the end-of-turn probability after an elicited candidate answer, combines this signal with the current budget consumption to estimate reasoning maturity, and compares the estimate to the schedule. If the estimate exceeds the schedule, a WAIT cue is injected to delay commitment; if it lags, a HURRY cue is injected to apply finalization pressure. *Bidirectional test-time scaling* thus emerges from the sign of the maturity gap, without fine-tuning the underlying LLM.

Our contributions are as follows:

- We reformulate fixed-budget reasoning control as pacing *reasoning maturity*, and frame underthinking and overthinking as symmetric failures of maturity pacing.
- We propose PARENTINGLLM, a maturity-gap controller that realizes bidirectional test-time scaling via a forced-commit probe, implemented as a fork-and-discard path inside vLLM. The target maturity schedule exposes an operator-facing knob that shifts the controller between earlier commitment and longer verification, without fine-tuning.

<sup>1</sup>Prior work uses *overthinking* for post-solution surplus deliberation or within-step token redundancy; we use it in a trace-level, budget-relative sense.

## 2 Related Work

**Budget-aware reasoning and inference-time steering.** A growing line of work studies reasoning under explicit computational constraints (Alomrani et al., 2025). TALE-EP (Han et al., 2025) exposes the budget to the model and relies on its in-context self-regulation; Han et al. (2025) show that this works well when the LLM can produce accurate budget estimates (e.g., GPT-4o-mini), but the stated limit may still be treated as a soft preference and exceeded. Training-based methods such as BudgetThinker (Wen et al., 2025), L1 (Aggarwal and Welleck, 2025), and ThinkPrune (Hou et al., 2025) achieve tighter length control by updating model parameters, at the cost of portability across base models. Inference-time alternatives avoid retraining: s1-style budget forcing (Muennighoff et al., 2025) inserts WAIT tokens to extend reasoning, while Budget Guidance (Li et al., 2025) uses an auxiliary length predictor to align reasoning length with a budget. Related steering methods inject cues at reasoning transitions, including SmartSwitch’s deepening prompts (Zhang et al., 2025), certainty-guided reflection suppression (Huang et al., 2026), and difficulty-adaptive slow-thinking (Shen et al., 2025). These methods primarily control reasoning length along a single direction. PARENTINGLLM is complementary along a different axis: it treats commitment readiness as the control target and exposes the desired pacing as an operator-facing schedule rather than a fixed length policy. The controller chooses WAIT, HURRY, or no intervention from the sign of a maturity gap, addressing cases where length control or self-regulation alone does not produce timely finalization.

**Overthinking and underthinking.** Sui et al. (2025) survey overthinking, where reasoning models generate verbose or redundant chains without improving answer quality, and Wang et al. (2025b) study underthinking, where models prematurely abandon promising paths; SmartSwitch (Zhang et al., 2025) addresses the latter by injecting deepening prompts after backtracking. We build on this distinction in a budgeted form: underthinking as becoming willing to finalize before adequate verification, and overthinking as deliberation that fails to keep commitment readiness on pace with the budget. PARENTINGLLM unifies the two under maturity pacing, using a single controller to delay premature commitment or apply finalization pressure depending on the sign of the maturity gap.

**Probe-based signals and uncertainty.** Closer to our setting, training-free methods probe the model during generation to inform termination. DEER (Yang et al., 2025b) exits when an elicited trial answer is high-confidence; EAT (Wang et al., 2025a) exits when next-token entropy stabilizes; Dynasor-CoT (Fu et al., 2024) monitors trial-answer consistency across transitions. A broader literature on LLM uncertainty, semantic entropy (Kuhn et al., 2023; Farquhar et al., 2024), relevance-weighted aggregation (Duan et al., 2024), semantic entropy probes (Kossen et al., 2024), and momentum-based uncertainty (Yan et al., 2025), estimates uncertainty over answers or reasoning steps, typically for post-hoc analysis or compute allocation. PARENTINGLLM extends this line in two ways: our forced-commit probe captures *commitment willingness*, whether the model would terminate cleanly if forced to answer now, rather than answer-level uncertainty; and the signal is used bidirectionally, prolonging reasoning when premature commitment is detected as well as stopping it when ready. Trial-answer changes between boundaries serve as an auxiliary prolongation signal (Sec. 4.4), related to Dynasor-CoT’s consistency monitoring but used in the opposite direction.

### 3 Problem Definition

A reasoning LLM  $\mathcal{M}$  generates a response  $y$  to a prompt  $x$  under a hard token budget  $B$ : generation is terminated once  $|y| = B$ , regardless of whether  $\mathcal{M}$  has produced a final answer. Under this constraint, the two complementary failures of Fig. 1, *underthinking* and *overthinking*, are not reliably addressed by  $\mathcal{M}$  itself: even when  $B$  is stated in the prompt, the model does not enforce it as a hard external constraint.

We seek a controller  $\pi$  that intervenes during inference to maximize expected answer quality under the budget, without training the parameters of  $\mathcal{M}$ :

$$\max_{\pi} \mathbb{E}_{x \sim \mathcal{D}} [Q(y \mid x, \pi, \mathcal{M})] \quad \text{s.t.} \quad |y| \leq B, \quad (1)$$

where  $Q$  scores final answer quality (e.g., task accuracy) and  $\pi$  injects steering tokens (e.g., *Hurry up*, *Wait - let me think again*) at chosen reasoning boundaries. The controller may rely on lightweight auxiliary components fit offline, but  $\mathcal{M}$  itself is never updated, and all generated tokens, including intervention cues, count toward  $B$ .

Because  $Q$  depends on the final answer, it is unobservable at inference time and cannot be op-

timized directly by  $\pi$ . We instead introduce an intermediate control variable, *reasoning maturity*: the degree to which the current prefix is ready to support a reliable final answer given the remaining budget. Maturity is a control-time notion of commitment readiness, not a quality measurement: a mature prefix can still produce a wrong answer. The controller’s task is to keep maturity on pace with budget consumption at every boundary; the next section instantiates this via a forced-commit probe paced against a user-specified target schedule.

## 4 Proposed Method

### 4.1 Preliminaries

PARENTINGLLM controls generation at *reasoning boundaries* rather than at every token. A reasoning boundary is a natural segmentation point in the trace; in our experiments we use newline-delimited reasoning steps (Appendix A.2). At each boundary  $t_i$ , the controller observes the current prefix  $y_{1:t_i}$  and the fractional budget consumption

$$u_i = t_i/B \in [0, 1], \quad (2)$$

where  $t_i$  counts all output tokens generated so far.

### 4.2 Forced-Commit Probe

To estimate whether the current prefix is becoming ready to finalize, we run a side probe on a transient path: we append a commit suffix (e.g., “The final answer is”) to  $x \oplus y_{1:t_i}$ . Because the task prompt instructs the model to wrap its final answer in `\boxed{...}`, this elicits a boxed candidate answer  $\hat{a}_i$ . Let  $c_i = x \oplus y_{1:t_i} \oplus \text{commit suffix} \oplus \hat{a}_i$  denote the resulting probe context. We read the probability of the end-of-turn token immediately after the closing brace:

$$P_{\text{EOT},i} = \Pr_{\mathcal{M}}(\text{EOT} \mid c_i). \quad (3)$$

Intuitively,  $P_{\text{EOT},i}$  asks whether the model would terminate cleanly if forced to answer now. This is distinct from answer confidence: a model can be confident in a candidate answer while still continuing to verify or revise it. We therefore treat  $P_{\text{EOT},i}$  as a probe-based measurement of *commitment willingness*.

The probe is a side computation: the boxed candidate answer  $\hat{a}_i$  is used to compute  $P_{\text{EOT},i}$  and to detect inter-boundary answer changes; it is never appended to the returned trace. Sec. 4.5 describes

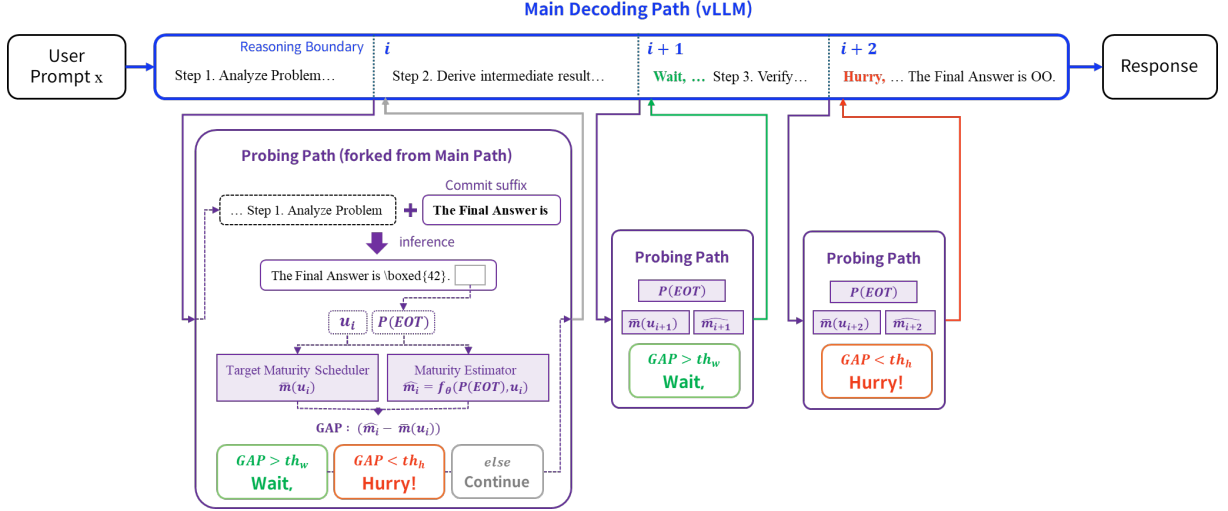


Figure 2: Inference-time pipeline of PARENTINGLLM. At each reasoning boundary, the runtime forks a transient probe path, appends a commit suffix, and reads  $P_{EOT,i}$ ; the maturity estimator  $f_\theta$  maps  $(u_i, P_{EOT,i})$  to  $\hat{m}_i$ , and the controller selects WAIT, HURRY, or CONTINUE from the sign of the maturity gap  $\hat{m}_i - \bar{m}(u_i)$ . The probe path is discarded after the read; only the selected cue (if any) is written to the main path.

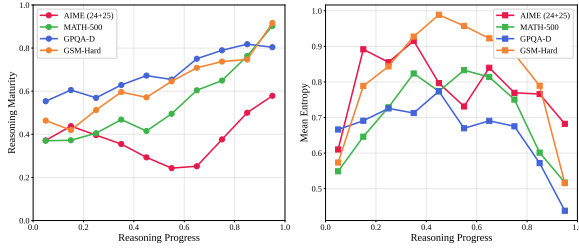


Figure 3: Per-benchmark trajectories as a function of reasoning progress ( $u$ , binned into deciles), averaged across traces. **Left:** estimated maturity  $\hat{m}$  from  $P_{EOT}$ . **Right:** mean answer-token entropy.

how this is implemented as a fork-and-discard path in vLLM.

Figure 3 contrasts  $P_{EOT}$ -based maturity with next-token entropy, a common uncertainty surrogate in prior early-exit methods (Wang et al., 2025a; Yan et al., 2025). Maturity rises monotonically with progress across all four benchmarks, while entropy is non-monotonic and peaks mid-trace. A signal that cannot distinguish a near-final prefix from a mid-deliberation one is unsuitable as a maturity proxy; we therefore use  $P_{EOT}$ , together with budget consumption, as the input to the maturity estimator.

### 4.3 Maturity Estimation and Target Schedule

The same forced-commit signal can mean different things early and late in the budget, so we combine it with  $u_i$  in a lightweight maturity estimator:

$$\hat{m}_i = f_\theta(u_i, P_{EOT,i}) \in [0, 1]. \quad (4)$$

We instantiate  $f_\theta$  as a gradient-boosted regressor (LightGBM (Ke et al., 2017)), fit once offline on boundary-level traces. Training labels are *judge-derived maturity scores*: an external LLM judge rates how ready a prefix is for reliable finalization under the remaining budget (prompt and rubric in Appendix B.2). These labels are used only to fit  $f_\theta$ ; at inference, the controller observes only  $u_i$  and  $P_{EOT,i}$ .  $f_\theta$  is fit *once per base model*, and the parameters of  $\mathcal{M}$  are never updated.

We compare  $\hat{m}_i$  against a user-specified *target maturity schedule*  $\bar{m} : [0, 1] \rightarrow [0, 1]$ , where  $\bar{m}(u)$  specifies the desired commitment readiness at budget fraction  $u$ . The schedule is a pacing preference, not a measurement of problem-solving progress: a more concave  $\bar{m}$  tolerates late maturity and biases the controller toward WAIT, while a flatter  $\bar{m}$  demands earlier maturity and biases toward HURRY. Unless otherwise stated, we use the default

$$\bar{m}(u) = 1 - (1 - u)^2, \quad (5)$$

a neutral concave shape that matches the empirically observed maturity trajectory of unconstrained Gemma-4 traces (Fig. 3, left). Operators can shift along this axis without retraining  $f_\theta$  or  $\mathcal{M}$ :  $\bar{m}$  is a fixed function with no parameters fit from data.

### 4.4 Maturity-Gap Intervention

At each boundary, PARENTINGLLM computes the maturity gap

$$\Delta_i = \hat{m}_i - \bar{m}(u_i), \quad (6)$$

---

**Algorithm 1** Maturity-Gap Parenting Controller

---

```
1: Input: prompt  $x$ , model  $\mathcal{M}$ , estimator  $f_\theta$ ,  
   schedule  $\bar{m}$ , budget  $B$ , threshold  $(\tau_W, \tau_H)$   
2:  $y \leftarrow \emptyset$ ;  
3: while  $|y| < B$  and  $\mathcal{M}$  has not terminated do  
4:   Extend  $y$  by generating from  $\mathcal{M}(x \oplus y)$  until  
   the next reasoning boundary  $t_i$   
5:    $u_i \leftarrow t_i / B$   
6:    $(P_{\text{EOT},i}, \hat{a}_i) \leftarrow \text{PROBE}(\mathcal{M}, x, y_{1:t_i})$   
7:    $\hat{m}_i \leftarrow f_\theta(u_i, P_{\text{EOT},i})$   
8:    $\Delta_i \leftarrow \hat{m}_i - \bar{m}(u_i)$   
9:   if  $\Delta_i > \tau_W$  then  
10:    Append WAIT cue to  $y$ ;  
11:   else if  $\Delta_i < -\tau_H$  then  
12:    Append HURRY cue to  $y$ ;  
13:   else  
14:     if  $\hat{a}_i \neq \hat{a}_{i-1}$  then  
15:      Append WAIT cue to  $y$ ;  
16:     else  
17:      CONTINUE;  
18:     end if  
19:   end if  
20: end while  
21: return  $y$ 
```

---

and selects an action among  $\{\text{CONTINUE}, \text{WAIT}, \text{HURRY}\}$  from its sign. WAIT appends a short cue that encourages further verification; HURRY appends a cue that encourages concise commitment; CONTINUE is a no-op (cue strings in Appendix A.3).

A positive gap signals a risk of *underthinking*: the model appears more commit-ready than desired at the current budget position, and may finalize before adequate verification. When  $\Delta_i > \tau_W$ , the controller injects WAIT. A negative gap signals a risk of *overthinking* in our budget-level sense: deliberation is failing to translate into sufficient commitment readiness relative to the consumed budget. When  $\Delta_i < -\tau_H$ , the controller injects HURRY. Otherwise it takes CONTINUE.

The thresholds  $\tau_W, \tau_H \geq 0$  encode an asymmetric cost: a spurious WAIT mainly consumes extra tokens, while a spurious HURRY can force commitment on a still-unresolved problem. We therefore set  $\tau_H > \tau_W$ , using  $\tau_W = 0.1$  and  $\tau_H = 0.15$  held fixed for every model, benchmark and budget. We additionally cap the total number of cue injections per request at three. Beyond this, repeated cues begin to overwrite the model’s in-progress reasoning rather than redirect it, degrading answer quality in

preliminary runs.

As an auxiliary safeguard, the controller also caches the probe’s candidate answer  $\hat{a}_i$  (Sec. 4.2) across boundaries: when  $\hat{a}_i \neq \hat{a}_{i-1}$  and the maturity-gap rule would otherwise select CONTINUE, the controller substitutes a WAIT cue. A change in the trial answer between consecutive boundaries indicates that the prefix is still being revised, and prolonging reasoning prevents finalization on an unstable hypothesis. The safeguard only fills in boundaries where the maturity gap is inconclusive and never overrides WAIT or HURRY from the main rule.

## 4.5 Runtime Integration

PARENTINGLLM is implemented as an inference-time controller inside a modified vLLM (Kwon et al., 2023) runtime (Fig. 2). For each user request, the runtime maintains a single main decoding path whose tokens form the returned response. At each reasoning boundary, the runtime forks the main path into a transient probe path, appends the commit suffix only on the probe path, and generates a boxed candidate answer to read  $P_{\text{EOT},i}$ . The probe path is then immediately discarded. Thus, the forced-commit answer is never shown to the user, never enters the returned trace, and never modifies the main decoding state.

Only the controller’s selected action affects the returned response. If WAIT or HURRY is triggered, the corresponding cue is written to the main path and counts toward the response budget  $B$ . If CONTINUE is selected, the main path simply resumes decoding from its original pre-probe state. Probe tokens do not consume  $B$  because they are not part of the returned response, but their KV-cache footprint and forward-pass cost are real overheads, which we report in Sec. 5. The fork-and-discard design is compatible with vLLM-style continuous batching: transient probe paths can be scheduled alongside ordinary decoding paths and discarded after the probe is read.

## 5 Experiments

### 5.1 Experimental Setup

**Models.** We evaluate PARENTINGLLM on Gemma-4-31B-it and Gemma-4-E4B-it (Google, 2026) together with Qwen3.5-9B (Qwen Team, 2026) from a different model family; Gemma-4-26B-A4B-it is reported in Appendix C. This tests whether the controller generalizes both across

model scale and across model families.

**Datasets.** We use four test split of reasoning benchmarks spanning mathematics and science: **MATH-500** (Hendrycks et al., 2021), a 500-problem subset of MATH; **GPQA-Diamond** (Rein et al., 2023), graduate-level multiple-choice questions in physics, chemistry, and biology; **AIME 24+25** (60 problems combined), competition-level mathematics, and **GSM-Hard** (Gao et al., 2023), an adversarial variant of GSM8K with larger numeric operands. The first three benchmarks are split into calibration and evaluation portions and used to fit the maturity estimator  $f_\theta$  and tune the thresholds ( $\tau_W, \tau_H$ ); GSM-Hard is held out from calibration and used solely for evaluation, serving as a held-out generalization test. All tasks are evaluated with exact-match accuracy on the boxed final answer.

**Token budgets.** We evaluate at two budgets:  $B \in \{1024, 2048\}$  output tokens. The lower budget stresses the controller’s ability to force timely finalization; the higher budget tests whether the controller avoids spurious interventions when budget is less constraining.

**Baselines.** We compare against four inference-time baselines that, like PARENTINGLLM, do not modify the parameters of the base reasoning model: (i) **Vanilla**: the model decodes freely; if  $|y| = B$  before termination, the response is hard-truncated and the most recent boxed answer in  $y$  (if any) is taken as the prediction, otherwise the request is scored as incorrect. (ii) **s1** (Muennighoff et al., 2025): iterative budget forcing that suppresses end-of-turn with a WAIT cue until the budget is exhausted; no reduction direction. (iii) **TALE-EP** (Han et al., 2025): the budget  $B$  is stated explicitly in the prompt (e.g., “Please answer using less than  $B$  tokens for your reasoning.”), and no further intervention is applied. This tests whether the model self-regulates when made aware of the budget. (iv) **DEER** (Yang et al., 2025b): confidence-triggered early exit. At each paragraph break an answer is elicited on a trial path and generation is terminated as soon as that answer is high-confidence. Like ours it reads a discarded continuation, but it acts in one direction only and always terminates rather than pacing commitment (Appendix A.5).

**Metrics.** We report three answer-quality metrics per (model, dataset, budget, method) cell: (i) *Accuracy* (%), exact-match on the boxed answer; (ii)

*Token utilization* (%), defined as  $100 \cdot \overline{|y|} / B$  where  $\overline{|y|}$  is the mean response tokens per request (including all intervention cues for PARENTINGLLM); this measures how well a method exploits the available budget rather than the raw token count; (iii) *Burst rate* (%), the fraction of requests reaching  $B$  without a well-formed boxed answer.

## 5.2 Main Results

PARENTINGLLM improves the accuracy–budget trade-off across three models and four reasoning benchmarks (Table 1; per-model maturity trajectories and representative cases in Fig. 4). It raises example-weighted average accuracy over Vanilla by 0.4–10.5 points and reduces burst rate by up to 12.1 points, achieving the best accuracy in 17 of 24 and the best burst rate in 14 of 24 model–benchmark–budget settings.

The strongest gains concentrate on **AIME 24+25**, the hardest benchmark. At  $B = 1024$ , where the budget binds hardest, PARENTINGLLM reduces average AIME burst from 85.3% to 62.7% and improves accuracy by 4.0 to 7.3 points over the best baseline on each model. At  $B = 2048$  the burst reduction persists (57.6% to 37.1%) but the accuracy advantage largely disappears on the Gemma models: 23.7 against TALE-EP’s 24.7 on Gemma-4-E4B-it and 61.0 against 61.3 on Gemma-4-31B-it, though it holds on Qwen3.5-9B (28.7 against 26.7). Once the budget is loose enough for the model to finalize on its own, pacing commitment mainly buys reliable termination. HURRY interventions are most useful when maturity lags behind budget consumption (Fig. 4(d)).

TALE-EP confirms that exposing the budget in the prompt can help, but does not fully solve budgeted commitment control: on AIME 24+25 it reduces burst over Vanilla, yet PARENTINGLLM reduces it further, especially at  $B = 2048$ . The relative strength of the two methods varies with base-model capability: TALE-EP is competitive on Gemma-4-31B-it for MATH-500 and GPQA-Diamond, where sufficiently capable models self-regulate from the prompt alone, while PARENTINGLLM’s gains concentrate on the smaller model (E4B-it) and the hardest benchmark (AIME 24+25), where prompt-based self-regulation leaves substantial residual burst. The two are complementary rather than competing. s1-style budget forcing mostly increases token utilization with only marginal accuracy gains, consistent with its one-sided design: it can extend reasoning but cannot ap-

Table 1: Main results across three models, four benchmarks, and two budgets ( $B \in \{1024, 2048\}$ ), each cell averaged over 5 seeds (temperature 0.7). Acc denotes Accuracy (%), Util denotes Token Utilization (%; mean response tokens divided by  $B$ ), and Burst denotes Burst Rate (%; fraction of requests reaching  $B$  without a well-formed boxed answer). Avg is the example-weighted average of each metric across the four benchmarks, with  $\Delta$  relative to Vanilla shown in parentheses (+ improvement for Acc, – improvement for Burst).

| $B$            | Method  | MATH-500    |       |             | GPQA-Dia.   |       |             | AIME 24+25  |       |             | GSM-Hard    |       |             | Avg                                    |                |                                        |
|----------------|---------|-------------|-------|-------------|-------------|-------|-------------|-------------|-------|-------------|-------------|-------|-------------|----------------------------------------|----------------|----------------------------------------|
|                |         | Acc.        | Util. | Burst       | Acc.        | Util. | Burst       | Acc.        | Util. | Burst       | Acc.        | Util. | Burst       | Acc. ( $\Delta$ )                      | Util.          | Burst ( $\Delta$ )                     |
| Gemma-4-E4B-it |         |             |       |             |             |       |             |             |       |             |             |       |             |                                        |                |                                        |
| 1024           | Vanilla | 60.2        | 68.8  | 32.3        | 13.6        | 95.6  | 80.5        | 4.7         | 99.3  | 95.3        | 36.4        | 51.8  | 12.5        | 39.0 $\pm$ 0.3                         | 61.4 $\pm$ 0.1 | 26.1 $\pm$ 0.3                         |
|                | s1      | 59.4        | 74.9  | 32.3        | 14.3        | 97.4  | 79.8        | 4.3         | 99.5  | 95.7        | 42.3        | 65.7  | 12.1        | 42.7 $\pm$ 0.3 (+3.7)                  | 71.9 $\pm$ 0.3 | 25.8 $\pm$ 0.2 (−0.3)                  |
|                | TALE-EP | 63.2        | 67.6  | 29.2        | 15.5        | 94.9  | 77.5        | 5.0         | 99.0  | 94.7        | 36.1        | 50.8  | 11.5        | 39.8 $\pm$ 0.5 (+0.8)                  | 60.4 $\pm$ 0.2 | 24.4 $\pm$ 0.3 (−1.7)                  |
|                | DEER    | 53.0        | 58.0  | 21.6        | 14.4        | 93.3  | 74.7        | 5.7         | 96.8  | 89.0        | 39.0        | 41.4  | <b>4.6</b>  | 39.0 $\pm$ 0.5 (+0.0)                  | 51.9 $\pm$ 0.3 | 17.8 $\pm$ 0.4 (−8.3)                  |
|                | OURS    | <b>68.4</b> | 61.3  | <b>14.7</b> | <b>24.8</b> | 84.2  | <b>52.4</b> | <b>12.0</b> | 87.7  | <b>57.0</b> | <b>47.8</b> | 56.8  | 6.1         | <b>49.5<math>\pm</math>0.5</b> (+10.5) | 61.4 $\pm$ 0.2 | <b>14.1<math>\pm</math>0.2</b> (−12.1) |
| 2048           | Vanilla | 76.5        | 44.1  | 12.3        | 40.0        | 77.5  | 40.0        | 24.0        | 90.6  | 70.3        | 37.9        | 28.5  | 1.2         | 47.0 $\pm$ 0.6                         | 38.7 $\pm$ 0.3 | 9.6 $\pm$ 0.4                          |
|                | s1      | 75.4        | 48.0  | 11.0        | 38.1        | 82.1  | 41.7        | 24.0        | 92.4  | 70.3        | 43.3        | 36.8  | 1.1         | 50.0 $\pm$ 0.6 (+3.0)                  | 45.4 $\pm$ 0.2 | 9.4 $\pm$ 0.4 (−0.2)                   |
|                | TALE-EP | <b>76.9</b> | 44.0  | 11.1        | 38.3        | 77.7  | 42.5        | <b>24.7</b> | 89.4  | 67.7        | 37.7        | 29.1  | 1.3         | 46.8 $\pm$ 0.6 (−0.1)                  | 39.1 $\pm$ 0.3 | 9.5 $\pm$ 0.4 (−0.1)                   |
|                | DEER    | 60.3        | 35.7  | 8.3         | 43.0        | 71.1  | 29.3        | 19.0        | 87.1  | 67.0        | 40.0        | 21.7  | <b>0.3</b>  | 44.6 $\pm$ 0.8 (−2.4)                  | 31.6 $\pm$ 0.1 | 7.0 $\pm$ 0.4 (−2.6)                   |
|                | OURS    | 74.8        | 42.1  | <b>5.0</b>  | <b>46.3</b> | 65.8  | <b>22.9</b> | 23.7        | 72.4  | <b>34.3</b> | <b>47.6</b> | 37.4  | 0.7         | <b>53.3<math>\pm</math>0.7</b> (+6.4)  | 42.2 $\pm$ 0.1 | <b>4.8<math>\pm</math>0.2</b> (−4.7)   |
| Gemma-4-31B-it |         |             |       |             |             |       |             |             |       |             |             |       |             |                                        |                |                                        |
| 1024           | Vanilla | 74.4        | 51.9  | 9.4         | 66.6        | 72.4  | 20.5        | 29.3        | 92.3  | 68.3        | 60.9        | 35.3  | 5.1         | 63.8 $\pm$ 0.5                         | 44.5 $\pm$ 0.2 | 9.4 $\pm$ 0.5                          |
|                | s1      | 73.5        | 58.6  | 9.6         | 65.9        | 78.4  | 19.9        | 29.3        | 95.3  | 68.0        | <b>62.5</b> | 44.4  | 5.1         | 64.5 $\pm$ 0.1 (+0.8)                  | 52.5 $\pm$ 0.1 | 9.4 $\pm$ 0.3 (−0.0)                   |
|                | TALE-EP | 75.5        | 50.1  | 8.2         | 68.4        | 68.6  | 15.7        | 33.7        | 91.0  | 63.3        | 61.3        | 34.2  | 4.2         | 64.6 $\pm$ 0.4 (+0.9)                  | 43.0 $\pm$ 0.1 | 8.0 $\pm$ 0.2 (−1.4)                   |
|                | DEER    | 67.2        | 45.2  | <b>4.2</b>  | 63.8        | 60.1  | <b>9.7</b>  | 25.7        | 88.7  | 59.0        | 58.5        | 30.8  | <b>0.9</b>  | 60.1 $\pm$ 0.3 (−3.6)                  | 38.7 $\pm$ 0.2 | <b>4.2<math>\pm</math>0.1</b> (−5.2)   |
|                | OURS    | <b>77.4</b> | 58.4  | 4.8         | <b>71.0</b> | 71.3  | 10.4        | <b>41.0</b> | 88.2  | <b>47.7</b> | 62.3        | 52.7  | 2.3         | <b>66.2<math>\pm</math>0.4</b> (+2.4)  | 56.8 $\pm$ 0.2 | 5.0 $\pm$ 0.3 (−4.4)                   |
| 2048           | Vanilla | 79.7        | 28.1  | 2.0         | <b>77.2</b> | 40.1  | 3.9         | 57.3        | 70.2  | 32.3        | 61.0        | 19.3  | 2.3         | 66.9 $\pm$ 0.3                         | 24.9 $\pm$ 0.2 | 3.2 $\pm$ 0.3                          |
|                | s1      | 78.9        | 32.1  | 1.6         | 76.7        | 44.8  | 4.1         | <b>61.3</b> | 75.0  | 31.3        | 62.7        | 24.3  | 2.3         | 67.9 $\pm$ 0.3 (+0.9)                  | 29.6 $\pm$ 0.1 | 3.2 $\pm$ 0.1 (−0.1)                   |
|                | TALE-EP | <b>79.9</b> | 28.0  | 1.6         | 76.4        | 40.8  | 5.2         | <b>61.3</b> | 68.5  | 30.3        | 61.1        | 19.7  | 2.2         | 67.1 $\pm$ 0.3 (+0.2)                  | 25.1 $\pm$ 0.1 | 3.1 $\pm$ 0.1 (−0.1)                   |
|                | DEER    | 69.2        | 23.4  | <b>0.4</b>  | 68.5        | 31.6  | <b>1.5</b>  | 45.3        | 63.0  | 26.7        | 58.4        | 15.7  | <b>0.3</b>  | 61.6 $\pm$ 0.7 (−5.3)                  | 20.4 $\pm$ 0.1 | <b>1.2<math>\pm</math>0.2</b> (−2.0)   |
|                | OURS    | 79.2        | 39.2  | 0.7         | <b>77.2</b> | 50.6  | 3.2         | 61.0        | 67.0  | <b>13.3</b> | <b>63.3</b> | 36.1  | 0.8         | <b>68.4<math>\pm</math>0.5</b> (+1.5)  | 39.1 $\pm$ 0.1 | 1.4 $\pm$ 0.2 (−1.9)                   |
| Qwen3.5-9B     |         |             |       |             |             |       |             |             |       |             |             |       |             |                                        |                |                                        |
| 1024           | Vanilla | 63.8        | 63.9  | 26.8        | 24.0        | 93.2  | 72.6        | 6.7         | 99.2  | 92.3        | <b>58.4</b> | 43.6  | 14.1        | 54.9 $\pm$ 0.5                         | 54.8 $\pm$ 0.1 | 25.0 $\pm$ 0.6                         |
|                | s1      | 55.7        | 80.1  | 26.5        | 24.7        | 98.1  | 71.4        | 5.0         | 99.7  | 91.3        | 56.0        | 72.1  | 13.6        | 51.5 $\pm$ 0.5 (−3.5)                  | 77.3 $\pm$ 0.3 | 24.5 $\pm$ 0.4 (−0.5)                  |
|                | TALE-EP | 63.2        | 66.3  | 26.9        | 27.9        | 92.8  | 68.7        | 8.0         | 98.4  | 90.0        | 56.7        | 48.1  | 15.9        | 54.1 $\pm$ 0.6 (−0.8)                  | 58.2 $\pm$ 0.2 | 25.7 $\pm$ 0.4 (+0.8)                  |
|                | DEER    | 61.6        | 57.2  | 20.7        | 27.2        | 91.3  | 69.2        | 8.7         | 98.1  | 87.7        | 58.0        | 40.5  | <b>10.0</b> | 54.5 $\pm$ 0.8 (−0.4)                  | 51.0 $\pm$ 0.5 | <b>20.5<math>\pm</math>0.4</b> (−4.5)  |
|                | OURS    | <b>67.5</b> | 64.8  | <b>19.2</b> | <b>37.4</b> | 87.2  | <b>55.9</b> | <b>12.7</b> | 96.4  | <b>83.3</b> | 56.4        | 60.4  | 15.8        | <b>56.0<math>\pm</math>0.6</b> (+1.0)  | 65.0 $\pm$ 0.2 | 22.4 $\pm$ 0.6 (−2.6)                  |
| 2048           | Vanilla | 74.9        | 40.8  | 12.8        | 46.3        | 74.9  | 47.7        | 26.3        | 88.9  | 70.3        | <b>59.8</b> | 27.5  | 8.9         | 61.2 $\pm$ 0.3                         | 37.0 $\pm$ 0.2 | 15.3 $\pm$ 0.4                         |
|                | s1      | 72.1        | 54.9  | 12.1        | 46.3        | 86.7  | 47.9        | 22.3        | 94.6  | 68.7        | 58.5        | 49.6  | 8.6         | 59.6 $\pm$ 0.5 (−1.6)                  | 55.7 $\pm$ 0.3 | 14.9 $\pm$ 0.2 (−0.4)                  |
|                | TALE-EP | 73.0        | 45.4  | 14.5        | 41.5        | 79.6  | 54.1        | 26.7        | 89.7  | 70.3        | 58.2        | 34.2  | 12.2        | 59.3 $\pm$ 0.2 (−1.9)                  | 42.9 $\pm$ 0.3 | 18.4 $\pm$ 0.4 (+3.1)                  |
|                | DEER    | 69.3        | 35.0  | 9.0         | 49.8        | 71.7  | 42.5        | 25.0        | 86.3  | 66.0        | 59.0        | 23.8  | <b>4.9</b>  | 59.6 $\pm$ 0.7 (−1.6)                  | 32.9 $\pm$ 0.6 | <b>11.2<math>\pm</math>0.5</b> (−4.1)  |
|                | OURS    | <b>75.1</b> | 42.0  | <b>8.8</b>  | <b>56.2</b> | 70.9  | <b>34.6</b> | <b>28.7</b> | 86.2  | <b>63.7</b> | 58.8        | 40.2  | 8.5         | <b>61.6<math>\pm</math>0.4</b> (+0.4)  | 44.9 $\pm$ 0.1 | 12.7 $\pm$ 0.3 (−2.6)                  |

ply finalization pressure when maturity lags. DEER attains the lowest burst rate in 10 of 24 settings, but the lowest accuracy among the baselines in four of the six model–budget blocks, and falls below Vanilla in five of six. Because it terminates as soon as a trial answer is confident, it cannot recover a problem that needed more reasoning: on AIME 24+25 at  $B = 1024$  it scores 5.7, 25.7 and 8.7 where we score 12.0, 41.0 and 12.7. Fig. 4(a–c) shows this directly: OURS pulls the maturity trajectory toward the target schedule across all model scales, while Vanilla drifts above or below depending on the benchmark.

**GSM-Hard as a held-out generalization test.** GSM-Hard (Gao et al., 2023) also serves as a distribution shift in our setting, since no GSM-family data is used to fit  $f_\theta$  or tune  $(\tau_W, \tau_H)$ . PARENT-INGLLM improves Vanilla accuracy in four of six model–budget settings, with the largest gains on Gemma-4-E4B-it (+11.4 and +9.7 points), but loses 2.0 and 1.0 points on Qwen3.5-9B. Burst-rate reductions are also mixed and DEER attains a lower burst rate on every model (Fig. 4(e) shows a representative underthink case). Accuracy therefore transfers out of domain within the calibrated family but not across it, which suggests the estimator’s calibration is tied to a family’s chain-of-thought

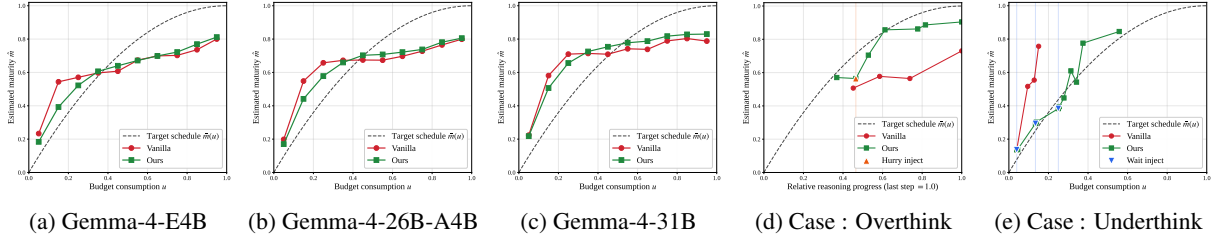


Figure 4: Schedule tracking and representative cases at  $B = 2048$ . **(a–c)** Per-model average maturity trajectories: estimated maturity  $\hat{m}$  averaged across all evaluation traces. Vanilla (red) is the model’s natural trajectory; OURS (green) adds the maturity-gap controller. The black dashed line is the target schedule  $\bar{m}(u) = 1 - (1 - u)^2$ ; OURS tracks  $\bar{m}$  more closely than Vanilla across all model scales. **(d)** A representative best-case sample where Vanilla overthinks: maturity lags the schedule and the budget is exhausted before finalization; the controller injects HURRY to apply finalization pressure. **(e)** A representative best-case sample where Vanilla underthinks: maturity exceeds the schedule and the model commits prematurely; the controller injects WAIT to delay commitment.

Table 2: Signal ablation on Gemma-4-E4B-it: alternative inputs to the maturity estimator  $f_\theta$ .

| $B = 1024$             |                        |                        |
|------------------------|------------------------|------------------------|
| $f_\theta$ inputs      | Acc.                   | Burst                  |
| Vanilla                | $39.0 \pm 0.3$         | $26.1 \pm 0.3$         |
| $u_i$ only             | $48.6 \pm 0.7 (+9.6)$  | $13.7 \pm 0.3 (-12.5)$ |
| $u_i + \text{entropy}$ | $49.1 \pm 0.6 (+10.0)$ | $13.0 \pm 0.4 (-13.1)$ |
| $u_i + P_{\text{EOT}}$ | $49.5 \pm 0.5 (+10.5)$ | $14.1 \pm 0.2 (-12.1)$ |
| $B = 2048$             |                        |                        |
| $f_\theta$ inputs      | Acc.                   | Burst                  |
| Vanilla                | $47.0 \pm 0.6$         | $9.6 \pm 0.4$          |
| $u_i$ only             | $52.4 \pm 0.6 (+5.4)$  | $4.4 \pm 0.3 (-5.1)$   |
| $u_i + \text{entropy}$ | $52.8 \pm 0.8 (+5.9)$  | $4.7 \pm 0.4 (-4.9)$   |
| $u_i + P_{\text{EOT}}$ | $53.3 \pm 0.7 (+6.4)$  | $4.8 \pm 0.2 (-4.7)$   |

Table 3: Direction ablation on Gemma-4-E4B-it: bidirectional control vs. WAIT and HURRY-only variants.

| $B = 1024$ |                        |                        |
|------------|------------------------|------------------------|
|            | Acc.                   | Burst                  |
| Vanilla    | $39.0 \pm 0.3$         | $26.1 \pm 0.3$         |
| WAIT only  | $44.1 \pm 0.4 (+5.0)$  | $28.8 \pm 0.4 (+2.7)$  |
| HURRY only | $48.1 \pm 0.3 (+9.1)$  | $13.0 \pm 0.4 (-13.1)$ |
| OURS       | $49.5 \pm 0.5 (+10.5)$ | $14.1 \pm 0.2 (-12.1)$ |
| $B = 2048$ |                        |                        |
|            | Acc.                   | Burst                  |
| Vanilla    | $47.0 \pm 0.6$         | $9.6 \pm 0.4$          |
| WAIT only  | $52.3 \pm 0.4 (+5.3)$  | $10.1 \pm 0.2 (+0.6)$  |
| HURRY only | $51.8 \pm 0.6 (+4.9)$  | $4.2 \pm 0.4 (-5.4)$   |
| OURS       | $53.3 \pm 0.7 (+6.4)$  | $4.8 \pm 0.2 (-4.7)$   |

distribution.

### 5.3 Ablation Studies

We isolate the contribution of each design choice through two ablations on Gemma-4-E4B-it across all four benchmarks.

**A1. Signal ablation.** We replace the input to  $f_\theta$  with alternative signals (Table 2):  $u_i$  alone (budget consumption only),  $u_i$  paired with next-token entropy, and the full  $(u_i, P_{\text{EOT}})$ . The full configuration is the most accurate at both budgets but has the highest burst rate of the three:  $u_i$  alone recovers +9.6 of its +10.5 gain at  $B = 1024$  with lower burst (13.7 against 14.1). Most of the gain therefore comes from the intervention rather than the signal, and entropy matches  $P_{\text{EOT}}$  end-to-end despite being non-monotonic in reasoning progress (Fig. 3).

**A2. Direction ablation.** We isolate the value of bidirectional control by comparing single-direction

variants (Table 3): *Wait-only* (no HURRY), *Hurry-only* (no WAIT), and the full bidirectional controller. The bidirectional version is the most accurate at both budgets. Wait-only fails under tight budgets ( $B = 1024$ ): without finalization pressure, additional verification cues push more requests past the budget, raising burst rate. Hurry-only attains the lowest burst rate at both budgets (13.0 and 4.2) but is less accurate than the bidirectional controller (48.1 and 51.8 against 49.5 and 53.3), because premature commitment is not corrected on problems that would benefit from extended reasoning. The bidirectional rule resolves both failure modes by selecting the intervention direction from the maturity-gap sign.

### 5.4 Probe and Serving Overhead

Table 4 reports probe cost on Gemma-4-E4B-it at  $B = 2048$ . Each probe is short because the commit suffix elicits only a boxed answer. On AIME 24+25 the HURRY cues remove more main-path tokens

Table 4: Probe overhead on Gemma-4-E4B-it at  $B = 2048$ , per-request means.  $N_p$ : probes per request.  $T_p$ : total probe tokens.  $T_m$ : main-path tokens for OURS and Vanilla. Net  $\Delta$ : total forward-pass tokens vs. Vanilla.

| Dataset    | $N_p$ | $T_p$ | $T_m^{\text{OURS}}$ | $T_m^{\text{Van.}}$ | Net $\Delta$ |
|------------|-------|-------|---------------------|---------------------|--------------|
| AIME 24+25 | 13.2  | 242   | 1483                | 1856                | -7.1%        |
| MATH-500   | 10.8  | 169   | 863                 | 904                 | +14.2%       |
| GPQA-D     | 12.4  | 275   | 1347                | 1587                | +2.2%        |
| GSM-Hard   | 11.1  | 215   | 765                 | 584                 | +68.0%       |

Table 5: Serving cost of the probe loop on Gemma-4-E4B-it (MATH-500,  $B=1024$ , H200) as a function of request concurrency  $C$ . Decode rate is returned output tokens per second per request. Parenthesised values are the ratio of OURS to Vanilla.

| $C$ | Latency p95 (s) ↓ |                         | Decode rate (tok/s) ↑ |                         |
|-----|-------------------|-------------------------|-----------------------|-------------------------|
|     | Van.              | OURS                    | Van.                  | OURS                    |
| 1   | 7.14              | 9.98 ( $\times 1.40$ )  | 142.8                 | 110.7 ( $\times 0.78$ ) |
| 4   | 5.92              | 11.27 ( $\times 1.90$ ) | 158.1                 | 89.5 ( $\times 0.57$ )  |
| 16  | 6.67              | 14.40 ( $\times 2.16$ ) | 142.7                 | 64.4 ( $\times 0.45$ )  |
| 32  | 7.52              | 17.71 ( $\times 2.36$ ) | 117.0                 | 50.7 ( $\times 0.43$ )  |

than the probes add, so the net forward-pass cost is -7.1%; on MATH-500 and GPQA-D the offset is partial, leaving +14.2% and +2.2%. On GSM-Hard, Vanilla traces are short (584 tokens) and the controller injects WAIT more often than HURRY, prolonging the main path; the +68.0% net cost reflects intervention against premature commitment rather than probe overhead.

Forward-pass tokens understate the cost of a synchronous probe. Table 5 measures the same controller under load: per-request decode rate falls to  $0.78\times$  Vanilla at concurrency 1 and to  $0.43\times$  at concurrency 32, while p95 latency rises from  $1.40\times$  to  $2.36\times$ . The degradation tracks concurrency rather than response length: each probe stalls the main path for its own decode, and the transient probe branches occupy batch slots and KV blocks that would otherwise serve other requests. Peak memory can transiently reach  $2\times$  Vanilla while the main and probe KV states coexist.

## 6 Conclusion

We controlled reasoning under a fixed token budget by pacing how ready a prefix is to be finalized rather than how long it is. PARENTINGLLM reads an end-of-turn probability from a forced-commit probe at reasoning boundaries and injects WAIT or HURRY depending on whether that estimate leads or lags a schedule set by the operator. Across four

benchmarks and four base models this raises accuracy and lowers burst rate, most clearly at the tighter budget. Budget consumption alone recovers most of the accuracy gain, and the probe accounts for a large share of the serving cost. A cheaper maturity signal is the most useful thing to look for next.

## Limitations

**Model coverage.** Our experiments span three Gemma-4 scales and one model from another family (Qwen3.5-9B). Other reasoning families such as DeepSeek-R1 remain untested.

**Serving cost of the probe.** The probe forks a path at every reasoning boundary and is synchronous in our implementation, so the main path stalls while it decodes. Under load this reduces per-request decode rate to  $0.78\times$  Vanilla at concurrency 1 and  $0.43\times$  at concurrency 32 (Table 5), and peak memory can transiently reach  $2\times$  Vanilla. The controller is therefore currently suited to settings where per-request budget adherence matters more than aggregate throughput.

Speculative decoding suggests a way to hide most of this cost. Because the probe is short and its output is discarded, it behaves like a speculative draft that is always rejected, and a runtime that already verifies drafted tokens in one batched pass could verify it there instead of stalling the main path. The probe would still need its own KV branch, since the commit suffix changes its prefix.

**Maturity label dependency.** Training labels for  $f_\theta$  come from an LLM judge whose monotone output is enforced by clamping (Appendix C.2); the estimator therefore inherits the judge’s biases, and the monotonicity of  $\hat{m}$  along  $u$  is partly an artefact of the label-shaping procedure. Replacing judge supervision with a task-grounded target (e.g., final-answer correctness on held-out samples) is a natural next step.

## Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation(IITP) under the Leading Generative AI Human Resources Development(IITP-2026-25543726) grant funded by the Korea government(MSIT).

## References

- Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*.
- Mohammad Ali Alomrani, Yingxue Zhang, Derek Li, Qianyi Sun, Soumyasundar Pal, Zhanguang Zhang, Yaochen Hu, Rohan Deepak Ajwani, Antonios Valkanias, Raika Karimi, and 1 others. 2025. Reasoning on a budget: A survey of adaptive and controllable test-time compute in llms. *arXiv preprint arXiv:2507.02076*.
- Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5050–5063.
- Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.
- Yichao Fu, Junda Chen, Siqi Zhu, Zheyu Fu, Zhongdongming Dai, Yonghao Zhuang, Yian Ma, Aurick Qiao, Tajana Rosing, Ion Stoica, and 1 others. 2024. Efficiently scaling llm reasoning with certaintindex. *arXiv preprint arXiv:2412.20993*.
- Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International conference on machine learning*, pages 10764–10799. PMLR.
- Google. 2026. Gemma 4. <https://huggingface.co/collections/google/gemma-4>. Hugging Face collection.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.
- 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.
- 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. *arXiv preprint arXiv:2103.03874*.
- Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. 2025. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*.
- Jiameng Huang, Baijiong Lin, Guhao Feng, Jierun Chen, Di He, and Lu Hou. 2026. Efficient reasoning for large reasoning language models via certainty-guided reflection suppression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 31176–31184.
- Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.
- Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. 2024. Semantic entropy probes: Robust and cheap hallucination detection in llms. *arXiv preprint arXiv:2406.15927*.
- Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.
- Junyan Li, Wenshuo Zhao, Yang Zhang, and Chuang Gan. 2025. Steering llm thinking with budget guidance. *arXiv preprint arXiv:2506.13752*.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. 2025. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20275–20321.
- OpenAI. 2024. Learning to reason with llms. <https://openai.com/index/learning-to-reason-with-llms/>.
- OpenAI. 2025. Introducing openai o3 and o4-mini. <https://openai.com/index/introducing-o3-and-o4-mini/>.
- Qwen Team. 2026. Qwen3.5: Towards native multi-modal agents.
- 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*.

- Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, and Shiguo Lian. 2025. Dast: Difficulty-adaptive slow-thinking for large reasoning models. *arXiv* <https://arxiv.org/abs/2503.04472>.
- Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, and 1 others. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*.
- Xi Wang, James McInerney, Lequn Wang, and Nathan Kallus. 2025a. Eat: Entropy after </think> for reasoning model early exiting. *arXiv preprint arXiv:2509.26522*.
- Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, and 1 others. 2025b. Thoughts are all over the place: On the underthinking of o1-like llms. *arXiv preprint arXiv:2501.18585*.
- Hao Wen, Xinrui Wu, Yi Sun, Feifei Zhang, Liye Chen, Jie Wang, Yunxin Liu, Yunhao Liu, Ya-Qin Zhang, and Yuanchun Li. 2025. Budgetthinker: Empowering budget-aware llm reasoning with control tokens. *arXiv preprint arXiv:2508.17196*.
- Hang Yan, Fangzhi Xu, Rongman Xu, Yifei Li, Jian Zhang, Haoran Luo, Xiaobao Wu, Luu Anh Tuan, Haiteng Zhao, Qika Lin, and 1 others. 2025. Mur: Momentum uncertainty guided reasoning for large language models. *arXiv preprint arXiv:2507.14958*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. 2025b. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*.
- Gyeong-In Yu, Joo Seong Jeong, Geon-Woo Kim, Soojeong Kim, and Byung-Gon Chun. 2022. Orca: A distributed serving system for {Transformer-Based} generative models. In *16th USENIX symposium on operating systems design and implementation (OSDI 22)*, pages 521–538.
- Xichen Zhang, Sitong Wu, Haoru Tan, Shaozuo Yu, Yinghao Zhu, Ziyi He, and Jiaya Jia. 2025. Smartswitch: Advancing llm reasoning by overcoming underthinking via promoting deeper thought exploration. *arXiv preprint arXiv:2510.19767*.
- Yinmin Zhong, Shengyu Liu, Junda Chen, Jianbo Hu, Yibo Zhu, Xuanzhe Liu, Xin Jin, and Hao Zhang. 2024. {DistServe}: Disaggregating prefill and decoding for goodput-optimized large language model serving. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pages 193–210.

## A Implementation Details

### A.1 Experimental Environment

All experiments are run on a single node with 8×NVIDIA H200 GPUs (141 GB HBM3e each), CUDA 12.8, and PyTorch 2.10. Inference uses our modified vLLM runtime, built on the public vLLM v0.19.0 codebase, with continuous batching enabled. The maturity estimator  $f_\theta$  is trained with LightGBM 4.6.0 on a separate CPU node. All methods and baselines use temperature=0.7, top\_p=1.0. Each reported cell is the mean over five seeds (0–4);  $\pm$  values are the standard deviation across seeds.

### A.2 Reasoning Boundary Definition

A reasoning boundary is a paragraph-level segmentation point in the generated trace at which the controller is invoked. Concretely, the sampler fires a boundary when (i) the last emitted token terminates the current paragraph (either a single token whose decoding ends in `\n\n`, or two consecutive single-newline tokens) and (ii) the next token’s argmax falls within the set of Step-prefixed tokens (e.g., `Step`, `Step_1`, `Step:`). This pattern matches the explicit `Step_1: \dots Step_2: \dots` chain-of-thought format that the instruction-tuned Gemma-4 models follow by default under our task prompt template. The set of `\n\n`-ending tokens and Step-prefix tokens is derived per model at engine startup by scanning the tokenizer vocabulary, so the same definition transfers to other instruction-tuned models without changes; this vocabulary scan is performed once per model load and adds no runtime overhead during inference.

Empirically, at least one boundary fires on  $\geq 99.8\%$  of reasoning traces across all four evaluation datasets (AIME 24+25, MATH-500, GPQA-Diamond, GSM-Hard) and all Gemma-4 model sizes.

Models that do not emit step markers, including Qwen3.5-9B, use a variant that fires on paragraph seams alone, without inspecting the following token. All Qwen3.5-9B results in this paper use this variant; the Gemma-4 results use the step-header rule above.

The probe is read at every boundary, giving the  $N_p \approx 10\text{--}15$  calls per request in Table 4. Cues are written only when the maturity gap exceeds a threshold, and at most three times per request (Sec. 4.4).

### A.3 Wait and Hurry Cue Strings

Both cues are short natural-language insertions appended directly to the main decoding path. The exact strings used in all experiments are framed as out-of-band user interjections (mimicking a user breaking into the assistant’s mid-response), since instruction-tuned models react more strongly to a perceived user message than to a self-uttered hedge:

- WAIT cue: “(user: wait, check your answer)”
- HURRY cue: “(user: hurry up and finalize)”

The user-interjection framing was selected after preliminary experiments showed that politer hedges in the assistant’s own voice (e.g., “Based on the analysis so far, the answer is”) produced weaker termination behavior on AIME 24+25, where the model otherwise tends to over-elaborate. The two strings above are used verbatim in every experiment reported here and in the released implementation. Both strings are tokenized by the model’s native tokenizer and count toward the response budget  $B$ .

### A.4 Maturity Estimator Training

The maturity estimator  $f_\theta$  is a LightGBM (Ke et al., 2017) gradient-boosted regressor mapping  $(u_i, P_{\text{EOT},i}) \in [0, 1]^2$  to a scalar maturity score in  $[0, 1]$ . Hyperparameters (Table 6) are shared across all base models; we did not tune them per model.

**Training data.** For each base model, we generate *probe-only* boundary traces (the controller’s PROGRESS\_MEASURE mode, which inserts the probe at every boundary but performs no WAIT/HURRY injection) on the calibration splits of MATH-500, GPQA-Diamond, and AIME 24+25 at  $B = 2048$ . Each boundary yields one  $(u_i, P_{\text{EOT},i})$  feature pair and a judge-derived maturity label (Appendix B.2). GSM-Hard is held out and used only as an out-of-distribution evaluation set.

**Judge model.** Labels come from an open-weight judge, openai/gpt-oss-120b, served locally on the same vLLM cluster. The judge outputs a monotone non-decreasing cumulative progress array (Appendix B.2); scores are clipped to  $[0, 1]$  and used directly as the regression target. The judge is invoked only at training time and never at inference.

Table 6: LightGBM hyperparameters for  $f_\theta$ , shared across all base models.

| Parameter        | Value               |
|------------------|---------------------|
| num_boost_round  | 500 (early stop 30) |
| learning_rate    | 0.05                |
| num_leaves       | 31                  |
| min_data_in_leaf | 50                  |
| feature_fraction | 0.9                 |
| bagging_fraction | 0.9                 |
| bagging_freq     | 5                   |
| Objective        | regression (RMSE)   |

### A.5 Baseline Implementations

All baselines use the same sampling parameters as PARENTINGLLM and run inside the same vLLM runtime to isolate the contribution of the controller.

**Vanilla.** The model decodes freely until either it emits the end-of-turn token or  $|y| = B$ . If hard-truncated, we attempt to extract the most recent well-formed `\boxed{...}` answer from  $y$ ; if none exists, the request is scored as incorrect.

**s1.** A client-side re-implementation of budget forcing (Muennighoff et al., 2025). We iteratively suppress termination: whenever the model emits the end-of-turn token (`finish_reason="stop"`) with budget remaining, we append the literal cue “\_Wait” and re-issue the request, conditioning on the accumulated assistant text. This repeats until either the model fails to emit a boxed answer before exhausting  $B$ , or no budget remains. If the final continuation overruns  $B$  mid-generation, we recover the last brace-balanced `\boxed{...}` from the prior natural-stop turn rather than the truncated tail. Cues are injected only on natural stop, never after length truncation.

**TALE-EP.** Following Han et al. (2025), the prompt is augmented with the explicit instruction “Please answer using less than  $B$  tokens for your reasoning.”, where  $B$  is the actual budget. No further inference-time intervention is applied. This isolates whether the model self-regulates when made aware of the budget.

**DEER.** We re-implement confidence-triggered early exit (Yang et al., 2025b) as a client-side controller over stock vLLM, matching our prompts, budget and main-path sampling. At each paragraph break past 256 tokens it forks a trial path, appends DEER’s answer inducer, and decodes it greedily for at most 24 tokens; the geometric mean of the per-token maximum probability is the confidence,

and generation stops once it exceeds  $\lambda$ . Otherwise the inducer is revoked, so DEER never writes to the main path and can only terminate, never prolong. We use its reference setting ( $\lambda=0.95$ , paragraph markers).

## B Prompts

### B.1 Task Prompt Templates

The user message concatenates a domain-specific instruction, the problem, and the META\_DUAL\_EXEMPLAR control-signal block. This block is shared across all methods (including vanilla, s1, and TALE-EP) for prompt parity; signals are injected only by PARENTINGLLM.

#### Mathematics (MATH-500, AIME 24+25, GSM-Hard).

Solve the math problem step by step. Format each reasoning step on its own with a 'Step N:' header. Separate steps with a blank line so each new step starts with "\n\nStep N:". After all reasoning steps, on a new paragraph write exactly 'The Final Answer is \boxed{<answer>}.'

Problem: {problem}

Begin your solution with 'Step 1:'.

{META\_DUAL\_EXEMPLAR}

**GPQA-Diamond.** Four options are deterministically permuted using SHA-256 of the record ID and pre-concatenated as "A) , B) , C) , D) ".

You are an expert answering hard graduate-level science multiple choice questions. Solve the question step by step. Format each reasoning step on its own with a 'Step N:' header. ... After all reasoning steps, write exactly 'The Final Answer is \boxed{<single letter>}.' where <single letter> is A, B, C, or D.

Question (choose one letter):

{problem with embedded "A) ... \n B) ..."}

Begin your solution with 'Step 1:'.

{META\_DUAL\_EXEMPLAR}

{META\_DUAL\_EXEMPLAR}

**\*\*Important control signals\*\*** (the SYSTEM may insert one of these phrases between your reasoning steps as an interruption from the user side):

- '(user: hurry up and finalize)' - the user is impatient. Briefly acknowledge, then continue with much shorter, more

clipped phrasing. Drop filler and self-explanation.

- '(user: wait, check your answer)' - the user wants you to double-check the previous step. Briefly acknowledge ("Let me reconsider"), then revisit the most recent step with a careful re-derivation.

Example (both signals on the same problem):

Problem: Find the roots of  $x^2 - 5x + 6 = 0$ .

Step 1: Discriminant  $25 - 24 = 1$ , perfect square; roots are integers.

Step 2: Factors of 6 summing to -5: (-2, -3).

(user: wait, check your answer)

Step 2 (re-check):  $(x-2)(x-3) = x^2 - 5x + 6$ . Matches.

Step 3: From  $(x-2)(x-3) = 0$ , one factor is zero.

(user: hurry up and finalize)

Step 4:  $x = 2$  or  $x = 3$ .

The Final Answer is \boxed{2, 3}.

CRITICAL: Do NOT generate these (user: ...) phrases yourself.

The two phrases are the exact strings injected by PARENTINGLLM's controller at "\n\n" boundaries.

### B.2 Judge Prompt for Maturity Labels

Training labels for  $f_\theta$  come from an external judge (openai/gpt-oss-120b) that scores all reasoning steps of a single trace jointly as a monotone non-decreasing cumulative progress curve. The trace is split on blank-line boundaries (regex "\n\s\*\n") into  $N$  steps; each step is truncated to 800 characters with a "...[trunc]" marker when needed. The judge sees the ground-truth final answer (target) to ground its progress estimates; this is acceptable because labels are used only for training of  $f_\theta$ .

You are evaluating a reasoning trace.

CORRECT FINAL ANSWER: {target}

The trace below is split into {N} numbered steps separated by blank lines. For EACH step, output the CUMULATIVE progress fraction toward the final answer at the end of that step, as a decimal in [0.00,1.00].

Guidelines:

0.00-0.15: just started; 0.15-0.40: approach chosen, derivation begun; 0.40-0.70: main reasoning; 0.70-0.95: approaching answer; 0.95-1.00: final answer committed (boxed).

Table 7: Results for Gemma-4-26B-A4B-it across four benchmarks and two budgets ( $B \in \{1024, 2048\}$ ), each cell averaged over 5 seeds (temperature 0.7). Acc denotes Accuracy (%), Util denotes Token Utilization (%), mean response tokens divided by  $B$ ), and Burst denotes Burst Rate (%), fraction of requests reaching  $B$  without a well-formed boxed answer). Avg is the example-weighted average of each metric across the four benchmarks, with  $\Delta$  relative to Vanilla shown in parentheses (+ improvement for Acc,  $-$  improvement for Burst).

| $B$                | Method  | MATH-500    |       |            | GPQA-Dia.   |       |             | AIME 24+25  |       |             | GSM-Hard    |       |            | Avg                                   |                |                                        |
|--------------------|---------|-------------|-------|------------|-------------|-------|-------------|-------------|-------|-------------|-------------|-------|------------|---------------------------------------|----------------|----------------------------------------|
|                    |         | Acc.        | Util. | Burst      | Acc.        | Util. | Burst       | Acc.        | Util. | Burst       | Acc.        | Util. | Burst      | Acc. ( $\Delta$ )                     | Util.          | Burst ( $\Delta$ )                     |
| Gemma-4-26B-A4B-it |         |             |       |            |             |       |             |             |       |             |             |       |            |                                       |                |                                        |
| 1024               | Vanilla | 66.2        | 60.6  | 18.1       | 43.0        | 86.6  | 50.3        | 21.3        | 96.1  | 76.7        | 59.6        | 48.6  | 17.3       | 58.5 $\pm$ 0.3                        | 56.5 $\pm$ 0.2 | 22.4 $\pm$ 0.5                         |
|                    | s1      | 66.2        | 68.8  | 17.4       | 43.4        | 92.6  | 49.7        | 22.7        | 99.3  | 75.7        | 61.7        | 62.2  | 17.7       | 59.9 $\pm$ 0.5 (+1.4)                 | 67.8 $\pm$ 0.3 | 22.4 $\pm$ 0.6 (+0.0)                  |
|                    | TALE-EP | 68.0        | 58.9  | 15.6       | 49.3        | 83.7  | 41.4        | 24.3        | 94.7  | 74.0        | 60.2        | 47.5  | 15.0       | 60.0 $\pm$ 0.4 (+1.5)                 | 55.1 $\pm$ 0.1 | 19.4 $\pm$ 0.2 (−3.0)                  |
|                    | DEER    | 61.5        | 53.1  | 11.3       | 45.3        | 79.2  | 39.7        | 19.3        | 92.2  | 70.0        | 58.0        | 39.4  | <b>4.7</b> | 56.5 $\pm$ 0.3 (−2.0)                 | 48.0 $\pm$ 0.2 | <b>11.5<math>\pm</math>0.4</b> (−10.9) |
|                    | OURS    | <b>71.6</b> | 62.0  | <b>9.9</b> | <b>53.7</b> | 81.1  | <b>31.5</b> | <b>32.0</b> | 92.4  | <b>63.0</b> | <b>62.7</b> | 59.1  | 8.6        | <b>63.1<math>\pm</math>0.6</b> (+4.6) | 62.8 $\pm$ 0.3 | 12.6 $\pm$ 0.4 (−9.7)                  |
| 2048               | Vanilla | 75.4        | 35.1  | 5.5        | 64.0        | 58.8  | 20.7        | 45.3        | 79.2  | 52.0        | 60.8        | 31.3  | 11.5       | 64.2 $\pm$ 0.4                        | 36.2 $\pm$ 0.2 | 12.1 $\pm$ 0.1                         |
|                    | s1      | 74.2        | 41.7  | 5.5        | 64.5        | 65.0  | 21.4        | 42.3        | 86.1  | 54.3        | 63.9        | 39.7  | 11.8       | 65.8 $\pm$ 0.3 (+1.6)                 | 44.0 $\pm$ 0.3 | 12.5 $\pm$ 0.4 (+0.3)                  |
|                    | TALE-EP | 75.0        | 36.0  | 6.0        | 66.6        | 59.8  | 19.5        | 47.3        | 79.4  | 51.0        | 61.3        | 31.7  | 11.1       | 64.7 $\pm$ 0.3 (+0.5)                 | 36.8 $\pm$ 0.1 | 11.8 $\pm$ 0.3 (−0.3)                  |
|                    | DEER    | 66.1        | 29.4  | <b>2.6</b> | 63.4        | 50.4  | <b>14.6</b> | 38.3        | 75.2  | 45.3        | 57.8        | 21.4  | <b>2.3</b> | 59.8 $\pm$ 0.7 (−4.4)                 | 27.7 $\pm$ 0.3 | <b>4.8<math>\pm</math>0.5</b> (−7.3)   |
|                    | OURS    | <b>76.3</b> | 44.8  | 3.6        | <b>67.8</b> | 62.6  | 15.6        | <b>50.3</b> | 78.9  | <b>42.3</b> | <b>64.4</b> | 44.4  | 4.7        | <b>67.2<math>\pm</math>0.1</b> (+3.0) | 47.3 $\pm$ 0.2 | 6.5 $\pm$ 0.5 (−5.6)                   |

The values must be MONOTONE NON-DECREASING. Output STRICTLY a JSON array of EXACTLY  $\{N\}$  floats in order. NOTHING ELSE.

STEPS: [1] {step\_1\_text} [2]  
{step\_2\_text} ... [N] {step\_N\_text}

Non-monotone outputs are clamped post-hoc by running maximum. Step  $k$ 's label is paired with the budget consumption  $u_k$  at the matching  $\backslash n \backslash n$  boundary.

### B.3 Forced-Commit Probe Suffix

At every  $\backslash n \backslash n$  boundary, the probe path forks the main KV cache and appends the literal suffix

“ The Final Answer is”

(leading space; re-tokenised per tokenizer at engine startup for cross-model portability). The probe freely generates up to 64 tokens with logprobs = 20; a brace-balanced parser locates the proper closing  $\}$  of  $\backslash boxed\{ \dots \}$ . The probe terminates—and its KV blocks are freed—once five positions are emitted beyond the closing brace;  $P_{\text{EOT},i}$  is the maximum end-of-turn probability over those five trailing positions after the closing brace.

## C Additional Results

### C.1 Gemma-4-26B-A4B-it

Table 7 reports the third Gemma-4 scale, which we moved out of the main table for space. The pattern matches the two scales in Table 1: PARENTINGLLM raises example-weighted average accuracy by 4.6 and 3.0 points and lowers burst rate

Table 8: Sensitivity of the maturity estimator to its label pipeline, on Gemma-4-E4B-it at  $B = 1024$ .

| Label pipeline                   | Acc.           |
|----------------------------------|----------------|
| Ours (clamped, gpt-oss-120b)     | 49.5 $\pm$ 0.5 |
| No monotonicity clamp            | 49.8 $\pm$ 0.2 |
| Judge $\rightarrow$ Qwen3.5-397B | 49.2 $\pm$ 0.3 |

by 9.7 and 5.6 points, and DEER again reaches a lower burst rate at the cost of accuracy.

### C.2 Sensitivity to Judge Supervision

The maturity estimator is trained on labels from an external judge whose output is clamped to be monotone (Appendix B.2). Two properties of that pipeline could drive our results rather than the controller itself: the clamp, and the choice of judge. We varied each in turn on Gemma-4-E4B-it at  $B = 1024$  (Table 8). Removing the running-maximum constraint, so that the judge may lower a progress score after a backtrack or an error, changes accuracy by +0.3 points. Replacing gpt-oss-120b with Qwen3.5-397B, a judge roughly three times larger and from a different model family, changes it by  $-0.3$ .