EXPERTS OF EXIT: RISK-CONTROLLED HALTING IN LOOPED LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Looped language models allow recurrent depth to be adjusted at inference time, but adaptive depth requires choosing a stopping policy for each deployment. Candidate policies are heterogeneous: checkpoints may include trained gates, whereas confidence, convergence, and fixed-depth rules can be added post hoc. Our cross-model, cross-task evaluation of fixed-threshold stopping rules finds no reliable default among the tested signal families: their quality-cost rankings vary with maximum recurrent depth, model scale, and task. We formulate adaptive-depth inference as risk-controlled selection among exit experts. Given a user-specified quality-loss budget, the selector chooses the least expensive candidate that passes a simultaneous validation test. We implement this formulation in Experts of Exit (EOE) with EOE-SAFE, which evaluates each candidate on its own autoregressive generations and constructs paired bootstrap max-t lower bounds simultaneously calibrated over the candidate pool. Across 160 held-out selection trials on Ouro, point-estimate selection violates the quality budget in 55 trials, with 23 exceeding twice the budget. EOE-SAFE records 13 violations, none exceeding twice the budget, while retaining a mean 33.6% recurrent-step reduction. On all 20 splits of Ouro-2.6B/MATH500, where no candidate meets the validation criterion, it returns full depth. The simultaneous bounds have bootstrap-asymptotic validity. Deployment measurements distinguish analytical depth savings from hardware speedups. In our single-GPU prototype, deterministic fixed-depth skipping achieves $1.92\times$–$2.12\times$ decode throughput over full-step execution, while the native gate and a JSD rule achieve $1.31\times$ and $1.92\times$ relative to their corresponding full-compute baselines.

1 INTRODUCTION

To improve the quality of the next generated token, language models can either (1) extend the reasoning trace or (2) refine an internal representation. Looped language models expose the second axis by repeatedly applying the same Transformer block to an evolving hidden state, allowing the model to iteratively refine its internal representation without necessarily extending the visible reasoning trace (Geiping et al., 2025; Zhu et al., 2025; Bae et al., 2025). This flexibility, however, raises a new inference-time question: how many iterative steps should each token receive? Using a fixed maximum iteration depth for every generated token wastes computation when a shallow prediction already suffices, while exiting too early risks committing to a poor token that can propagate errors through the remainder of autoregressive generation.

Ouro models (1.4B and 2.6B) support up to four loops (full depth) and use a learned early-exit gate, referred to as the native gate, to adaptively choose the iteration depth. However, exiting early with this gate is not free: when its threshold is set to skip 57–60% of the loops, accuracy on multilingual grade-school math (MGSM; Shi et al., 2023) falls from 50.4% to 38.8% for Ouro-1.4B on Russian and from 81.2% to 70.8% for Ouro-2.6B on Spanish (Figure 1a).

The quality–cost trade-off of early exit also varies across models. At τ=0.2, which costs both models less than one point on GSM8K, the native gate lowers Ouro-1.4B's accuracy by 4.5 points on Compositional GSM (Hosseini et al., 2024), where each problem chains two GSM8K questions, whereas Ouro-2.6B shows no loss at a similar saving (Figure 1c). The rule selected under a one-point quality budget also changes across tasks: mostly max-softmax on math and code, and a committee

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

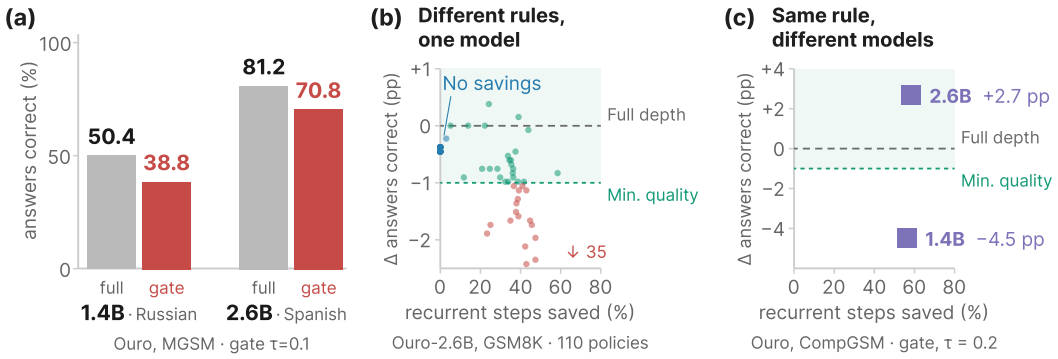


Figure 1: Why stopping policies require task-level evaluation (Ouro, autoregressive generation). **(a)** The native gate at τ=0.1 on MGSM, each model's worst language (250 problems each). **(b)** Accuracy change of 110 stopping policies on Ouro-2.6B GSM8K (all 1,319 problems) relative to full depth: some stay within one point, many fall below (35 lie off scale below -2.5 points). **(c)** On Compositional GSM (600 problems), the native gate at τ=0.2 costs Ouro-1.4B 4.5 points but not Ouro-2.6B. Dashed lines: full depth; dotted lines: one point below it.

rule on MMLU (Table 3, Appendix B). These results suggest that the choice of stopping policy should account for the model, the task, and the acceptable loss in answer quality (Figure 1b,c).

Existing approaches provide several ways to decide when to stop. Learned-halting methods train a gate together with the model (Graves, 2016; Dehghani et al., 2019; Zhu et al., 2025), while calibrated early-exit methods tune a threshold for a chosen confidence signal (Schuster et al., 2022). Convergence-based methods instead use changes in the model's internal state or output distribution as stopping criteria (Bai et al., 2019; Geiping et al., 2025). These approaches offer candidate stopping policies, but selecting a policy for a given model and task requires comparing their computational costs and effects on answer quality. We therefore study how to select the lowest-cost policy whose quality loss relative to full-depth decoding stays within a specified tolerance.

To address this problem, we introduce EoE, a framework for risk-controlled exit-expert selection. An exit expert provides a stopping rule, such as a learned gate or a confidence criterion, which is instantiated with specific parameters to form a candidate policy. EoE compares these policies with full-depth decoding on held-out validation data and selects the lowest-cost policy for which the data provide sufficient evidence that the quality loss is within the specified tolerance.

Selecting a stopping policy requires addressing three challenges. First, learned gates, confidence scores, and convergence criteria use different scales, so their raw scores cannot be compared directly. Policies must instead be compared by their effects on answer quality and computational cost. Second, an early-exit decision can change the generated token and thus the context for subsequent predictions. Each policy therefore needs to be evaluated on its own autoregressive generations. Third, selecting among many policies using finite validation data can favor a policy whose quality is overestimated by chance. A confidence bound computed for a single policy does not necessarily remain valid after this selection. The selection procedure must therefore account for all candidate policies when assessing whether the quality loss is within the specified tolerance.

EoE-Safe implements this selection procedure. It evaluates learned gates, confidence-based rules, convergence criteria, and fixed-depth policies by the same quality and cost measures. Each candidate generates its own answers on the validation set, and its quality is compared with full-depth decoding on the same examples. EoE-Safe then constructs simultaneous confidence bounds across the candidate set using a shared-resample max-t procedure, so that the uncertainty of the search across candidates is accounted for.

We test selection reliability in 160 trials on Ouro, selecting a policy on a validation split and measuring its quality loss on a separate test split in each trial. A baseline that selects the lowest-cost policy meeting the quality requirement based on validation point estimates goes over the tolerance in 55 trials. In 23 of these trials, the loss is more than twice the tolerance. Accounting for uncertainty separately for each candidate reduces these counts to 22 and 4, respectively. EoE-Safe also

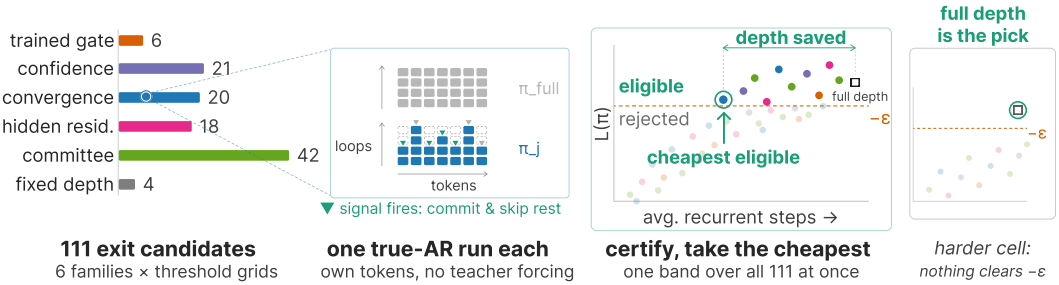


Figure 2: EOE-SAFE in one picture: six exit-expert families give 111 candidates (left), each measured on its own autoregressive generations (middle). The cheapest candidate whose simultaneous quality bound $L(\pi)$ clears $-\epsilon$ is deployed, or full-depth decoding when none does (right); because the bound covers all 111 candidates at once, certification remains valid after the search. Scatter positions are schematic; measured values are in Table 1a and Figure 4.

accounts for the search across candidates and reduces the counts to 13 and zero, while retaining a mean 33.6% reduction in recurrent depth.

Our contributions are fourfold. (i) We formulate adaptive-depth inference as *risk-controlled exit-expert selection* among candidates from different exit experts, with full depth as the fallback (Section 3). (ii) At thresholds shared across models, we show that no stopping rule is a reliable default: the same rule can be harmless on one model or task and cost points on another, and the maximum depth changes which rules are competitive (Sections 4 and 5). (iii) We propose EOE-SAFE, which certifies candidates on their own generations with paired, shared-resample max-t bounds; across 160 held-out trials, it cuts severe budget violations from 23 under point-estimate selection to zero while keeping a mean 33.6% depth reduction (Sections 6 and 7). (iv) We separate analytical step savings from measured speedups and identify cache-feasible skipped work as the quantity that determines runtime gains (Section 6.3 and Appendix D). Figure 2 summarizes the protocol.

2 BACKGROUND AND RELATED WORK

Adaptive computation and looped language models. Adaptive Computation Time (ACT) (Graves, 2016) introduced a differentiable halting mechanism that allows a recurrent neural network to adapt the number of computation steps used for each input. Universal Transformers (Dehghani et al., 2019) extended adaptive halting to repeated self-attention transformations. More recent language models use repeated computation to refine representations during generation. Ouro (Zhu et al., 2025) learns a native early-exit gate during pretraining, while Huginn (Geiping et al., 2025) increases effective depth by repeatedly applying a shared recurrent block. Mixture-of-Recursions (Bae et al., 2025) learns to assign different recursion depths to individual tokens, and PonderLM-2 (Zeng et al., 2026) feeds each position's last hidden state back as a latent thought, a recurrence computed with Jacobi iterations.

Early exit and halting signals. Confident Adaptive Language Modeling (CALM) calibrates a confidence threshold with Learn-then-Test on the early-exit model's own generations, falling back to full depth when no threshold passes (Schuster et al., 2022). FREE, LayerSkip, and EE-LLM improve training and decoding support for layer-wise exits (Bae et al., 2023; Elhoushi et al., 2024; Chen et al., 2024). ConsistentEE and COSEE align classifier training with early-exit behavior at inference (Zeng et al., 2024; He et al., 2025). EOE's convergence signals draw on fixed-point computation and recurrent-depth diagnostics (Bai et al., 2019; Geiping et al., 2025), and apply to existing checkpoints without modifying or retraining them.

Risk-controlled selection. Learn-then-Test uses multiple hypothesis testing to identify configurations satisfying a specified risk limit (Angelopoulos et al., 2025). Risk-controlling model selection searches for efficient configurations under such constraints (Laufer-Goldshtein et al., 2024). Pareto Testing configures early exit and pruning in Transformers by testing configurations in an order learned on a separate split (Laufer-Goldshtein et al., 2023). For early exit, EERO uses

classifiers with a reject option to select exits under a computational budget (Valade et al., 2025). Fast-yet-Safe controls quality loss by calibrating a confidence threshold shared across exit layers (Jazbec et al., 2024). EoE extends this perspective to candidates from different experts, which have no natural order; Bonferroni-type corrections remain valid but are conservative at our sample sizes (Section 7). EoE-Safe instead combines paired comparisons with full depth and shared-resample max-t bounds across candidates, selecting the cheapest qualifying policy with full depth as the fallback. Appendix E.2 discusses further related work.

3 Problem Setup

Consider a looped LM that performs at most K recurrent steps per generated token. An *exit expert* e defines a family of stopping rules, such as a trained gate, confidence, convergence, or fixed depth. Fixing its parameters, such as a threshold or an exit depth, defines an *exit candidate* π. At token position t, π selects an exit step $k_t \leq K$ and commits to the token predicted at that step. An *exit selector* uses validation data to choose one candidate from a pool Π for deployment. The pool includes the full-depth candidate π_{full}, which always executes all K steps and is the fallback.

For an evaluation set, let $C(\pi)$ denote the mean selected recurrent depth per generated token and $Q(\pi)$ the downstream task score. This cost counts recurrent steps; Section 6.3 examines their relationship to runtime. The quality budget ϵ specifies the maximum task-score decrease that the user allows relative to full-depth decoding. Our matched-quality objective selects the lowest-cost candidate within this budget:

$$\min_{\pi \in \Pi} C(\pi) \quad \text{s.t.} \quad Q(\pi) \geq Q(\pi_{\text{full}}) - \epsilon. \tag{1}$$

Validation scores estimate performance on unseen inputs from finite data. The selector must therefore account for estimation uncertainty when assessing whether a candidate meets this constraint.

Because recurrent steps reuse the same block, changes in output distributions and hidden states provide natural convergence signals, but an early exit can change the emitted token and so affect subsequent generation. We therefore evaluate each candidate on its own autoregressive trajectory and use teacher-forced diagnostics, which hold the token sequence fixed, only for analysis; Section 5 shows that they can be misleading.

4 Study Design

Evaluation design. We conduct three complementary evaluations. The policy analysis sweeps every expert family over a prespecified threshold grid on Ouro and measures, on the full test sets of four tasks, how answer quality and recurrent depth change relative to full-depth decoding. The fixed-threshold evaluation compares the signals at one prespecified threshold each, shared across models: it applies these thresholds to Huginn, whose maximum depth is eight times Ouro's, across 48 lm-evaluation-harness benchmarks, and reads the same thresholds from the policy analysis on Ouro. The selection evaluation chooses policies on validation data under the objective in Equation 1 and tests whether they remain within the quality budget on held-out test data.

Models. Ouro (1.4B and 2.6B; $K{=}4$) exposes a trained gate, per-step logits, and hidden states (Zhu et al., 2025). These outputs support comparisons between its native gate and stopping rules that require no additional training, so Ouro is the main model of the policy analysis and the selection evaluation. Huginn ($K{=}32$) provides much deeper recurrence and is used in the fixed-threshold evaluation (Geiping et al., 2025). Huginn and PonderLM-2 (410M and 1.4B; depth counted in Jacobi iterations, $K{=}4$) are also included in the selection evaluation (Section 7; Zeng et al., 2026).

Signals. We group the stopping rules into six expert families: the trained gate (Ouro only); output confidence (max-softmax, entropy, and margin); output convergence (top-1 agreement, Jensen–Shannon divergence (JSD) (Lin, 1991), and KL); hidden-state convergence, measured by the relative residual $\|h_k - h_{k-1}\|/\|h_k\|$ with a patience parameter; two-signal committees; and fixed-depth baselines. The policy analysis and the selection evaluation instantiate these families as 111 candidates

on the prespecified threshold grids in Appendix E. The fixed-threshold evaluation uses one configuration for each of five signals (max-softmax at 0.999, entropy at 1.0, top-1 agreement, JSD at 0.005, and the hidden residual at 0.05), plus the trained gate at 0.5 on Ouro, with thresholds fixed before evaluation (Appendix E.3); on Huginn it also evaluates all even fixed depths from 2 to 32.

The calibration gain. Raw depth reductions are not comparable between models with $K{=}4$ and $K{=}32$. For a signal s at its shared threshold, let $\bar{d}(s)$ and $\bar{Q}(s)$ denote its mean exit depth and score on a task (or averaged over the model's tasks), and let $\bar{Q}_{\text{full}}$ be the corresponding full-depth score. We define the normalized depth saving E, the normalized quality loss R, and the calibration gain γ as

$$E(s) = 1 - \frac{\bar{d}(s)}{K}, \qquad R(s) = \max\left\{0,\ \frac{\bar{Q}_{\text{full}} - \bar{Q}(s)}{\bar{Q}_{\text{full}}}\right\}, \qquad \gamma(s) = E(s) - R(s).$$

Thus, $\gamma(s) > 0$ means that relative depth saving is larger than relative quality loss under equal weighting. The calibration gain puts depth saving and quality loss on one scale across models and tasks but weights them equally, so a rule can reach a large γ while violating the quality budget; Figure 3 therefore shows score change and depth saving, and Table 5 lists γ. Candidate selection follows the quality-constrained objective in Equation 1.

Tasks and implementation. Because early exit can behave differently across task types, our comparisons focus on four tasks with distinct characteristics: grade-school arithmetic reasoning (GSM8K (Cobbe et al., 2021)), competition mathematics (MATH500 (Hendrycks et al., 2021b; Lightman et al., 2024)), code generation (HumanEval+ (Chen et al., 2021; Liu et al., 2023)), and multiple-choice knowledge (MMLU (Hendrycks et al., 2021a)). We run them on our autoregressive harness with the reference checkpoints and greedy decoding: each candidate generates its own continuation, and the harness also records per-step signals along each full-depth output. Huginn's fixed-threshold evaluation uses a broad lm-evaluation-harness suite (Gao et al., 2024) run with our vLLM wrapper (Kwon et al., 2023) (Appendix C).

5 Findings

The Native Gate Trades Accuracy for Depth. The native gate saves recurrent steps at little cost only when it exits conservatively. At $\tau{=}0.5$ it skips 23–24% of steps on GSM8K with no loss in accuracy on either model. At $\tau{=}0.1$ it skips 62–65% of steps but lowers GSM8K accuracy by 4.1 points on Ouro-1.4B and 2.6 points on Ouro-2.6B, turning 124 and 82 correct answers into errors while fixing 70 and 48, and on MGSM it loses up to 11.6 points (Figure 1a). MATH500 shows the same pattern: at $\tau{=}0.1$, accuracy falls by 2.0 and 2.2 points while 64% of steps are skipped. The gate is therefore a tunable trade-off rather than a free saving, and its acceptable operating point must be established for each model and task.

One Early-Exited Token Can Derail a Solution. Token-level agreement with full depth does not predict answer quality. Along the full-depth outputs, max-softmax exit at $\tau{=}0.9$ reproduces the full-depth token at 99.8% (Ouro-1.4B) and 99.9% (Ouro-2.6B) of positions, but under autoregressive generation it lowers GSM8K accuracy by 6.0 and 5.0 points (95% CIs $[-7.4, -4.7]$ and $[-6.3, -3.8]$). In every problem whose output differs from full depth, the first divergence is an early-exited token whose prediction differs from the full-depth prediction, typically about halfway through the solution. Candidates must therefore be evaluated on their own generations; teacher-forced diagnostics serve only for analysis (Section 3).

No Rule Is a Reliable Default across Models and Tasks. The same rule can be harmless on one model and costly on another (Figure 3): at the same JSD threshold, Ouro-1.4B keeps its GSM8K accuracy while skipping 35% of steps, whereas Ouro-2.6B loses 1.5 points at a 38% saving; on Compositional GSM, the native gate at $\tau{=}0.2$ costs Ouro-1.4B 4.5 points and Ouro-2.6B none (Figure 1c). Different rules on one model also spread widely: of the 110 candidates on Ouro-2.6B GSM8K, 58 stay within one point of full depth and 52 fall below it (Figure 1b). Under calibrated selection, the chosen rule changes across tasks: max-softmax in most GSM8K, MATH500, and HumanEval+ cells, a margin–agreement committee on MMLU, and full depth on Ouro-2.6B/MATH500 (Table 3); wherever some family reduces depth, another family reduces it more than the native gate (Table 6).

Stopping rule	Ouro-1.4B (K=4) GSM8K	MATH500	HumanEval+	MMLU	Ouro-2.6B (K=4) GSM8K	MATH500	HumanEval+	MMLU	Huginn (K=32) GSM8K	MMLU	Avg.
Trained gate	**0.0** (23%)	**+1.6** (25%)	**+1.8** (36%)	**0.0** (1%)	**+0.4** (24%)	**−1.0** (27%)	**+2.4** (26%)	**0.0** (0%)	–	–	–
Max-softmax	**0.0** (20%)	**0.0** (37%)	**0.0** (50%)	**0.0** (1%)	**0.0** (22%)	**0.0** (36%)	**0.0** (45%)	**0.0** (0%)	**+0.9** (0%)	**0.0** (0%)	**0.0** (0%)
Entropy	**−15.3** (62%)	**−12.0** (67%)	**−10.4** (69%)	**−1.0** (17%)	**−17.7** (63%)	**−16.0** (69%)	**−3.0** (71%)	**−0.3** (10%)	**+0.5** (0%)	**0.0** (0%)	**0.0** (0%)
JSD	**+0.2** (35%)	**+1.2** (37%)	**+1.2** (45%)	**0.0** (0%)	**−1.5** (38%)	**−1.2** (38%)	**+2.4** (46%)	**0.0** (2%)	**−3.8** (68%)	**−6.7** (68%)	**−2.5** (68%)
Hidden residual	**−0.1** (0%)	**−0.4** (0%)	**0.0** (0%)	**0.0** (0%)	**−0.4** (0%)	**0.0** (0%)	**0.0** (0%)	**0.0** (0%)	**+0.9** (40%)	**−0.7** (40%)	**−0.3** (40%)
Top-1 agreement	**−2.1** (23%)	**−2.2** (23%)	**+1.2** (24%)	**−0.1** (11%)	**−1.9** (23%)	**−1.6** (23%)	**−1.8** (24%)	**−0.5** (13%)	**−21.1** (78%)	**−6.0** (78%)	**−7.0** (78%)

Figure 3: Score change from full depth (points, top) and depth saving E (bottom) at each rule's shared threshold; red deepens with the loss, green marks a gain (none significant). Ouro: our autoregressive harness. Huginn: lm-evaluation-harness suite (Avg.: suite mean), output-based rules read from a pre-coda proxy (Appendix E.3), one mean exit depth per rule. Tests and γ: Appendix C.

The Maximum Depth Changes Which Rules Are Competitive. The maximum recurrent depth changes both how much depth can be removed and which rules remove it at low cost. Halving the depth of every token lowers GSM8K accuracy by 13.7 and 8.3 points on Ouro-1.4B and Ouro-2.6B (K=4) in the policy analysis, but by only 1.7 points on Huginn (K=32) in the fixed-threshold evaluation (Table 4). On Huginn, fixed depth is therefore a strong candidate: stopping every token at step 18 skips 44% of the steps while changing GSM8K accuracy by 0.2 points and the mean score by 0.002. Output-convergence rules skip more steps, as expected when they stop at a step that does not depend on K (Proposition 2), but benchmark averages hide their losses on multi-step reasoning: JSD skips 68% of the steps and lowers the mean score by only 0.025 but costs 3.8 points on GSM8K, and top-1 agreement reduces GSM8K accuracy from 26.8% to 5.7%. EoE-Safe (Section 6) compares all of these candidates, including fixed depth, under one quality test.

6 EoE-Safe: Risk-Controlled Exit-Expert Selection

EoE-Safe certifies candidates with simultaneous lower bounds on their quality differences from full depth and selects the cheapest certified candidate, falling back to full-depth decoding if none qualifies.

6.1 Policy Evaluation and Selection

Paired Evaluation of Quality Changes. We evaluate quality through each candidate's score difference from full-depth decoding. EoE-Safe evaluates all candidates on the same locked validation split, using the autoregressive decoding and task scoring of the final test (Figure 2). Each candidate follows its own generation trajectory and is compared with full depth on the same examples, giving $\widehat{\Delta}(\pi) = Q_{\mathcal{V}}(\pi) - Q_{\mathcal{V}}(\pi_{\text{full}})$. Pairing exploits the correlation between the two policies' scores. In our experiments, it reduces estimation variance relative to unpaired comparisons (Section 7).

Simultaneous Bounds Account for Candidate Selection. A confidence bound calibrated for one candidate need not remain valid after searching and selecting among several (Angelopoulos et al., 2025; Laufer-Goldshtein et al., 2023). EoE-Safe therefore constructs simultaneous lower bounds across the candidate pool (Westfall & Young, 1993; Romano & Wolf, 2005). All candidates share the same bootstrap resamples. Within each resample, we standardize each candidate's downward deviation by its estimated standard error and take the maximum across candidates. The $(1-\alpha)$ quantile of these maxima gives a common critical value c and the max-t lower bound $L(\pi) = \widehat{\Delta}(\pi) - c\,\widehat{\text{se}}(\pi)$.

Cost Minimization among Eligible Candidates. Only candidates with $L(\pi) \geq -\epsilon$ and validation cost below full-depth decoding are eligible for selection. EoE-Safe returns the lowest-cost eligible candidate, or full-depth decoding if none qualifies (Algorithm 1).

Algorithm 1 EoE-Safe policy selection

Require: candidates Π, validation set $\mathcal{V}$, budget ϵ, level $1-\alpha$
1: Generate with π_{full} on $\mathcal{V}$ once; cache each example's score
2: **for** each policy $\pi \in \Pi$ **do**
3: Generate with π on $\mathcal{V}$ (true autoregressive decoding); cache the paired differences $\Delta_i(\pi)$
4: **end for**
5: Draw shared bootstrap resamples of $\mathcal{V}$; $c \leftarrow (1-\alpha)$ quantile of the maximum studentized downward deviation $\max_\pi \big(\widehat{\Delta}(\pi) - \widehat{\Delta}^*(\pi)\big)/\widehat{\mathrm{se}}(\pi)$
6: $L(\pi) \leftarrow \widehat{\Delta}(\pi) - c\,\widehat{\mathrm{se}}(\pi)$ for every $\pi \in \Pi$
7: $\Pi_{\text{eligible}} \leftarrow \{\pi \in \Pi \setminus \{\pi_{\text{full}}\} : L(\pi) \geq -\epsilon$ and $C_{\mathcal{V}}(\pi) < C_{\mathcal{V}}(\pi_{\text{full}})\}$
8: **if** $\Pi_{\text{eligible}} = \emptyset$ **then**
9: **return** π_{full} {conservative fallback}
10: **else**
11: **return** $\arg\min_{\pi \in \Pi_{\text{eligible}}} C_{\mathcal{V}}(\pi)$
12: **end if**

Table 1: Policy selection on Ouro (ϵ=0.01, α=0.05; 50/50 validation–test splits). **(a)** Four-task averages after max-t calibration over each selector's pool (per task: Table 6); Acc: test accuracy (%); #L: mean recurrent steps per token; bold/underline: best/second best; MATH500 uses a 512-token cap. **(b)** Eight (model, task) cells × 20 splits over 111 candidates; deficits vs. full depth.

(a) Quality and depth (three seeds)

	Ouro-1.4B		Ouro-2.6B	
Selector	Acc↑	#L↓	Acc↑	#L↓
Full depth	59.17	4.00	65.18	4.00
Fixed-depth	**60.80**	3.83	65.08	3.92
Trained gate	<u>60.63</u>	3.17	65.02	3.49
Max-softmax	58.93	2.53	65.00	2.96
Entropy	59.13	3.71	65.11	3.90
Margin	59.10	3.67	<u>65.14</u>	3.81
Convergence	59.62	3.32	**65.36**	3.70
Committee	59.06	3.35	65.10	3.29
EoE-Free	59.06	**2.48**	64.94	<u>2.84</u>
EoE-All	59.06	**2.48**	64.90	**2.83**

(b) Reliability (160 trials)

Rule	Viol.	Res.	Sev.	Δ (pp)	Cut
Point estimate	55/160	51/120	23	+0.40	49.7%
Marginal LCB	22/160	12/120	4	+0.37	36.6%
Bonferroni	16/160	2/120	3	−0.08	33.1%
Holm	13/160	1/120	0	−0.08	32.7%
LTT	0/160	0/120	0	+0.00	0.0%
EoE-Safe	13/160	1/120	0	+0.04	33.6%

Viol./Sev.: test deficit $> \epsilon$/$> 2\epsilon$. Res.: excludes HumanEval+ (Appendix E.5). Δ: mean test-score difference; Cut: mean depth saving. Paired marginal LCB; Bonferroni bootstrap; normal Holm (Holm, 1979); Hoeffding (Hoeffding, 1963) LTT with Bonferroni correction.

Here, *certification* means that a candidate passes the bootstrap-asymptotic quality test. Section 7 measures quality-budget violations on held-out test data.

The same selection rule applies to different candidate pools. Family-specific selectors search within one expert family; EoE-Free searches zero-training experts jointly, and EoE-All also includes the trained gate. Each selector computes the max-t bound over its own search pool. Cached candidate generations can be reused across quality budgets without further inference; caching and split protocols are detailed in Appendix A.

6.2 Quality–Depth Trade-offs

Signals Can Remain Useful after Task-Specific Calibration. On Ouro-2.6B, EoE-Free reduced mean recurrent depth from 4.00 to 2.84 steps, a 29.1% reduction. Mean accuracy across the four tasks decreased from 65.18% at full depth to 64.94%, a loss of 0.24 percentage points (Table 1a). Most of this reduction comes from max-softmax on GSM8K and HumanEval+ (Table 6), the same family that loses 5.0 points on Ouro-2.6B GSM8K at τ=0.9 (Section 5); a signal that fails at one threshold can still supply qualifying candidates at another.

Selected Policies Vary across Models and Tasks. Across the 24 (model, task, seed) combinations, EoE-All selected a zero-training signal in 20, a recalibrated gate in one, and full-depth decoding on all three Ouro-2.6B/MATH500 splits. On Ouro-2.6B/GSM8K, it selected max-softmax on two seeds and a recalibrated gate on one. The selected gate uses τ=0.5, which loses no GSM8K accuracy in the policy analysis; the aggressive τ=0.1 setting of Figure 1a is never selected. These

results support evaluating the gate as one candidate and retaining full-depth fallback when no candidate passes the quality check.

6.3 From Recurrent Depth to Runtime

Depth Savings Do Not Automatically Improve Throughput. The #L column in Table 1a reports the mean selected recurrent depth $\bar{k} = N^{-1} \sum_{t=1}^{N} k_t$ over N generated tokens. The evaluation harness records exits but executes all K steps, so the depth reductions in the table do not directly establish runtime savings.

Fixed Skipping Can Raise Throughput Proportionally. On an RTX A6000, a deterministic, token-exact Ouro-1.4B prototype that skipped steps 3–4 achieved $1.92\times$–$2.12\times$ the decode throughput of full-step execution at batch sizes 1, 8, and 32. This is close to $K/\bar{k}$=2, showing that fewer recurrent steps can directly reduce execution cost in this regime.

KV Cache Completion Limits Token-Adaptive Gains. Each recurrent level needs a contiguous prefix of KV entries, so a later token that returns to a deeper level must complete the missing cache, which can erase nominal depth savings. At batch size 1, against the same policy executed with all four steps and a complete cache, the native gate at mean depth 2.48 achieved only $1.31\times$ throughput because 130 depth increases forced 279 back-fill jobs, whereas JSD at mean depth 2.07 needed only 15 and achieved $1.92\times$. Runtime gains therefore depend on exit trajectories and cache overhead, not only on mean depth. A two-GPU pipeline eases back-fill but needs a second model replica, so its gains are a form of scale-out (Appendix D).

Long Prefills Limit Pipeline Savings. For a prefill sequence of length N, Proposition 1 bounds idealized wavefront savings by $O(K/N)$, whereas the analytical depth saving $1 - \bar{k}/K$ is length-independent. On Huginn's fixed-threshold exit trajectories, the model maps JSD's 68.5% analytical depth saving to an estimated 5.0% pipeline saving (Appendix D). This analysis applies only to prefill; decode performance claims use measured execution results.

7 Reliability and Limits of Policy Selection

Pairing Reduces Uncertainty in Quality Differences. Candidate and full-depth scores are positively correlated across the same examples, and paired comparison uses this correlation to improve precision. With a correctness-flip rate of approximately 3% and 660 validation examples, the paired standard error is about 0.007, compared with 0.023 without pairing. At a one-sided 95% level, these errors correspond to margins of about 1.1 and 3.8 percentage points, so without pairing, a candidate with no true quality loss would typically fail a one- or two-point budget.

Simultaneous Calibration Reduces Violations after Selection. In each of 160 Ouro trials, a policy is selected from 111 candidates on a validation half and evaluated on a disjoint test half with a one-percentage-point quality budget (ϵ=0.01). A budget violation occurs when the selected policy's test score falls more than ϵ below full depth; in a severe violation it falls more than 2ϵ below. Paired bounds calibrated separately for each candidate lead to 22 budget violations, including 4 severe violations. Calibrating over the entire pool, EoE-Safe reduces these counts to 13 and zero while retaining a mean depth saving of 33.6% (Table 1b).

Violation Counts Must Be Read alongside Depth Savings. Holm also incurs 13 budget violations and no severe violations, with a slightly smaller mean depth saving of 32.7% than EoE-Safe's 33.6%. The tested Learn-then-Test baseline uses Hoeffding tests with Bonferroni correction. It accepts no early-exit candidate in any of the 160 trials, so it has neither violations nor savings.

The Generation Cap Affects Whether Candidates Qualify. On Ouro-2.6B/MATH500, raising the generation cap from 256 to 512 tokens improves full-depth accuracy by about 6.5 points. Under per-candidate calibration, a depth saving of about 32% qualifies at the 256-token cap but not at 512 tokens (Figure 4b), and at 512 tokens EoE-Safe returns full depth on all 20 splits. A shorter cap can therefore mask the quality loss of early exit. Some responses remain truncated at 512 tokens.

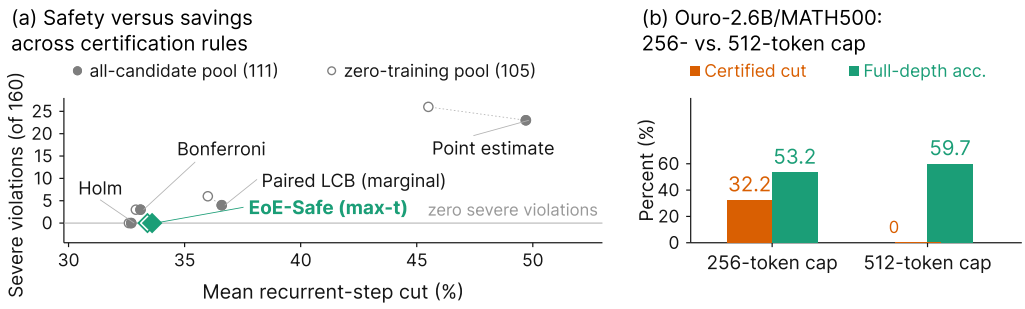


Figure 4: **(a)** Holm and EoE-Safe record no severe violations, and EoE-Safe retains a larger mean depth cut (33.6% versus 32.7%); point-estimate selection cuts more but incurs 23 severe violations (filled: all-candidate pool; hollow: zero-training pool). LTT, which accepts no candidate (0% cut), lies outside the plotted range. **(b)** On Ouro-2.6B/MATH500, under per-candidate (marginal) calibration with three seeds, a 32.2% cut qualifies under the 256-token cap; at 512 tokens, full-depth accuracy rises by 6.5 points and no early-exit candidate qualifies.

Other Architectures Show Both Savings and Fallback. On PonderLM-2's LAMBADA (Paperno et al., 2016) evaluation, both model sizes select a KL-convergence policy on all three seeds, and the 1.4B model cuts mean recurrent depth by 34.5% without losing task score (+0.45 points). On Huginn's MATH500 evaluation with a 256-token cap, EoE-Safe falls back to full depth on two of three splits and selects a fixed depth of 24 steps on the third (a 25% cut, 8.3% averaged over the splits), as the fixed-depth results of Section 5 suggest. KL and hidden-residual candidates pass the per-candidate test on some splits but not the simultaneous test. This evidence is limited to these two tasks.

8 Discussion and Conclusion

Learned Routers Face the Same Trajectory Shift. A learned router is trained on prefixes from one trajectory but acts on prefixes shaped by its own exits, so it inherits the shift that makes teacher-forced diagnostics misleading (Section 5). In our experiments (MATH500 capped at 256 tokens, marginal calibration), lightweight routers reached a median token-level AUC of 0.958 on teacher-forced trajectories, but a selector restricted to them failed the quality budget or fell back to full depth in 18 of 24 (model, task, seed) combinations under true autoregressive decoding. Calibrating any rule on fixed generations faces the same mismatch, so EoE-Safe, like CALM (Schuster et al., 2022), evaluates every candidate on its own generations.

Limitations. EoE-Safe requires labeled validation data for each model and task, and our selection experiments cover a limited set of tasks per model, so whether a selected policy transfers to new tasks remains open. The fixed-threshold evaluation uses one threshold per signal, approximate output signals on Huginn, and model-specific task sets and protocols, so it describes these rules rather than tuning them. Runtime results come from Ouro-1.4B on a single prompt; they show which execution regimes turn depth cuts into speedups, not the speedup to expect in serving.

Early exit in looped language models is better treated as a selection problem than as a choice of one default rule. The native gate costs accuracy when it exits aggressively, and no single rule is best across models, tasks, and maximum depths. EoE-Safe selects among such candidates with paired, simultaneous bounds: in 160 held-out trials on Ouro, it reduced budget violations from 55 under point-estimate selection to 13, none severe, with a mean 33.6% cut in recurrent steps. These cuts become speedups only when execution avoids the skipped work: fixed skipping reaches about 2× decode throughput, whereas token-adaptive gains shrink as later tokens force cache back-fill.

AI USE STATEMENT

Large language models were used to assist with manuscript writing and revision, including language polishing, grammar checking, and improving readability and clarity. They also helped with searching related work, coding, and running code. The authors reviewed all AI-assisted text and take full responsibility for the content of this submission.

REPRODUCIBILITY STATEMENT

The appendix documents the candidate policies, quality budgets, and supporting analyses. Upon publication, we will release the components produced by the authors' own measurement pipeline: the autoregressive harness, the record of every generated answer and the data splits for the policy analysis, the selector implementations (all six selection rules), and the selection results in both panels of Table 1. These records and the split seeds are already in the supplementary material. The Huginn part of the fixed-threshold evaluation was run with a separate vLLM-based implementation. We will release its threshold constants (Section 4), the score on each benchmark, mean exit depths, and early-exit fractions for every signal and fixed depth, and the pipeline summary behind Appendix D, from which every Huginn number in Tables 4 and 5 and the pipeline calibration can be recomputed. Outputs for individual examples of this part are not included, so it can be checked against the released tables but not re-run from raw outputs, whereas the policy analysis and selection evaluation can be re-run end to end from the released records.

REFERENCES

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1641–1662, 2025.

Sangmin Bae, Jongwoo Ko, Hwanjun Song, and Se-Young Yun. Fast and robust early-exiting framework for autoregressive language models with synchronized parallel decoding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5910–5924, 2023.

Sangmin Bae, Yujin Kim, Reza Bayat, Sungnyun Kim, Jiyoun Ha, Tal Schuster, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Aaron Courville, and Se-Young Yun. Mixture-of-Recursions: Learning dynamic recursive depths for adaptive token-level computation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 688–699, 2019.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Yanxi Chen, Xuchen Pan, Yaliang Li, Bolin Ding, and Jingren Zhou. EE-LLM: Large-scale training and inference of early-exit large language models with 3D parallelism. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pp. 7163–7189, 2024.

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. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Amit Damri and Bracha Laufer-Goldshtein. Confident and adaptive generative speech recognition via risk control. In *International Conference on Learning Representations (ICLR)*, 2026.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal Transformers. In *International Conference on Learning Representations (ICLR)*, 2019.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, Ahmed Aly, Beidi Chen, and Carole-Jean Wu. LayerSkip: Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 12622–12642, 2024.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness. Software, 2024. URL `https://github.com/EleutherAI/lm-evaluation-harness`.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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

Jianing He, Qi Zhang, Hongyun Zhang, Xuanjing Huang, Usman Naseem, and Duoqian Miao. COSEE: Consistency-oriented signal-based early exiting via calibrated sample weighting mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 24023–24031, 2025.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021a.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021b.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Arian Hosseini, Alessandro Sordoni, Daniel Toyama, Aaron Courville, and Rishabh Agarwal. Not all LLM reasoners are created equal. *arXiv preprint arXiv:2410.01748*, 2024.

Metod Jazbec, Alexander Timans, Tin Hadži Veljković, Kaspar Sakmann, Dan Zhang, Christian A. Naesseth, and Eric Nalisnick. Fast yet safe: Early-exiting with risk control. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Ahmadreza Jeddi, Marco Ciccone, and Babak Taati. LoopFormer: Elastic-depth looped transformers for latent reasoning via shortcut modulation. In *International Conference on Learning Representations (ICLR)*, 2026.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, pp. 611–626, 2023.

Filipe Laitenberger, Dawid Kopiczko, Cees G. M. Snoek, and Yuki M. Asano. What layers when: Learning to skip compute in LLMs with residual gates. In *International Conference on Learning Representations (ICLR)*, 2026.

Bracha Laufer-Goldshtein, Adam Fisch, Regina Barzilay, and Tommi Jaakkola. Efficiently controlling multiple risks with Pareto testing. In *International Conference on Learning Representations (ICLR)*, 2023.

Bracha Laufer-Goldshtein, Adam Fisch, Regina Barzilay, and Tommi Jaakkola. Risk-controlling model selection via guided Bayesian optimization. *Transactions on Machine Learning Research*, 2024.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024.

Jianhua Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

nostalgebraist. Interpreting GPT: The logit lens. LessWrong post, 2020. URL `https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru`.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1525–1534, 2016.

Joseph P. Romano and Michael Wolf. Stepwise multiple testing as formalized data snooping. *Econometrica*, 73(4):1237–1282, 2005.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. Language models are multilingual chain-of-thought reasoners. In *International Conference on Learning Representations (ICLR)*, 2023.

Florian Valade, Mohamed Hebiri, and Paul Gay. EERO: Early exit with reject option for efficient classification with limited budget. In *Proceedings of the 41st Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 286 of *Proceedings of Machine Learning Research*, pp. 4290–4308, 2025.

Peter H. Westfall and S. Stanley Young. *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment*. Wiley, New York, 1993.

Boyi Zeng, He Li, Shixiang Song, Yixuan Wang, Zitong Wang, Ziwei He, Xinbing Wang, and Zhouhan Lin. PonderLM-2: Pretraining LLM with latent thoughts in continuous space. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Ziqian Zeng, Yihuai Hong, Hongliang Dai, Huiping Zhuang, and Cen Chen. ConsistentEE: A consistent and hardness-guided early exiting method for accelerating language models inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 19506–19514, 2024.

Rui-Jie Zhu, Zixuan Wang, Kai Hua, Tianyu Zhang, Ziniu Li, Haoran Que, Boyi Wei, Zixin Wen, Fan Yin, He Xing, Lu Li, Jiajun Shi, Kaijing Ma, Shanda Li, Taylor Kergan, Andrew Smith, Xingwei Qu, Mude Hui, Bohong Wu, Qiyang Min, Hongzhi Huang, Xun Zhou, Wei Ye, Jiaheng Liu, Jian Yang, Yunfeng Shi, Chenghua Lin, Enduo Zhao, Tianle Cai, Ge Zhang, Wenhao Huang, Yoshua Bengio, and Jason Eshraghian. Scaling latent reasoning via looped language models. *arXiv preprint arXiv:2510.25741*, 2025.

A THE BUDGET DIAL ON OURO

Selection protocol. For the results in Table 1a, we cache one greedy-decoded generation per candidate and example, then repeat selection on three seeded 50/50 validation–test splits. Changing the quality budget ϵ reuses these cached generations without further inference.

Table 2 re-runs max-t selection on the frozen candidate pool at five budgets (same cached generations, seeds, and α as Table 1a; the ϵ=0.01 row reproduces it exactly). Mean depth falls monotonically as ϵ grows: tightening ϵ from 0.01 to 0.002 buys back a fraction of a point of accuracy for about four points of depth cut, and loosening it to 0.05 reaches a 50% average cut while test accuracy changes by -2.1 to $+0.5$ points relative to full depth. Budgets of 0.002 and 0.005 are smaller than the approximate uncertainty margin in Equation 4 (about 0.011 in the illustrated setting); at these budgets, the selector is conservative and prefers deeper candidates. The fallback branch is exercised throughout: Ouro-2.6B/MATH500 returns full-depth decoding on all three seeds for every $\epsilon \leq 0.02$ and is first certified only at ϵ=0.05.

Table 2: The budget ϵ as a deployment dial under the max-t selector (test accuracy and steps averaged over four tasks and three seeds; full-depth references 59.17 and 65.18 at 4.00 loops; MATH500 at a 512-token cap). Ouro-2.6B/MATH500 falls back to full-depth decoding on every seed for $\epsilon \leq 0.02$.

	EOE-FREE				EOE-ALL			
	1.4B		2.6B		1.4B		2.6B	
ϵ	Acc	#L	Acc	#L	Acc	#L	Acc	#L
0.002	59.12	2.64	65.07	3.01	59.12	2.64	65.07	3.01
0.005	59.08	2.57	65.02	2.92	59.08	2.57	65.02	2.92
0.01	59.06	2.48	64.94	2.84	59.06	2.48	64.90	2.83
0.02	58.75	2.24	64.69	2.67	58.84	2.21	64.67	2.66
0.05	57.90	2.03	63.11	2.07	59.70	1.95	63.40	1.93

B WHICH CANDIDATE IS SELECTED WHERE

Table 3 lists the candidate selected by the unrestricted EOE-SAFE (max-t, ϵ=0.01) for each seed of every (model, task) cell. Max-softmax dominates the generative math and code tasks, a margin-plus-agreement committee takes both MMLU cells on all seeds, and Ouro-2.6B/MATH500 returns full-depth decoding on every seed at the 512-token cap. No expert family is selected everywhere, and even where the family repeats, the threshold changes across cells.

C FIXED-THRESHOLD BEHAVIOR

Table 4 reports the fixed-threshold evaluation on Huginn for every signal and for representative fixed depths (all depths are in the released CSV): the mean exit depth, the fraction of tokens exiting before step K, the unweighted mean score over 48 benchmarks, and GSM8K accuracy, each with its difference from full depth. Two full-depth runs differ by 0.9 GSM8K points, and max-softmax and entropy, which almost never exit, differ from full depth by 0.5 to 0.9 points; GSM8K differences within about one point therefore reflect run-to-run variation. In the policy analysis on Ouro (K=4), which uses our harness rather than the lm-evaluation-harness protocol of Table 4, stopping every token at step 2 lowers GSM8K accuracy by 13.7 points on Ouro-1.4B and 8.3 points on Ouro-2.6B, and stopping at step 3 lowers it by 1.2 and 1.7 points.

Table 3: Selected candidates per seed under the max-t selector ($\wedge$ marks a committee; `@` gives the instantiated threshold).

Model	Task	Selected candidates (3 seeds)
1.4B	GSM8K	`maxsoft@.999` $\times 3$
1.4B	HumanEval+	`maxsoft@.9`, `maxsoft@.95`, `entropy@.5`
1.4B	MATH500	`maxsoft@.99` $\times 2$, `maxsoft@.999`
1.4B	MMLU	`margin@.4`$\wedge$`agree` $\times 3$
2.6B	GSM8K	`maxsoft@.99`, `maxsoft@.999`, `gate@.5`
2.6B	HumanEval+	`maxsoft@.95` $\times 2$, `maxsoft@.9`
2.6B	MATH500	full-depth decoding $\times 3$
2.6B	MMLU	`margin@.4`$\wedge$`agree` $\times 3$

Table 4: Fixed-threshold evaluation on Huginn (K=32) over 48 lm-evaluation-harness benchmarks (`quac` excluded because it is reported on a 0–100 scale). $\bar{k}$: mean exit depth. Cut: $1 - \bar{k}/K$. EE%: fraction of tokens exiting before step K. $\bar{Q}$: unweighted mean score. GSM8K: flexible-extract exact match over 1,319 problems. Fixed-depth rows run every token for the given number of steps; fixed depth 32 is an independent repeat of full depth. All even depths from 2 to 32 are in the released CSV.

				48 benchmarks		GSM8K (%)	
Signal	$\bar{k}$	Cut	EE%	$\bar{Q}$	$\Delta\bar{Q}$	Acc	Δ
Full depth	32.00	0.0%	0.0	0.357	0.000	26.8	0.0
Max-softmax	32.00	0.0%	0.0	0.357	−0.000	27.7	+0.9
Entropy	31.99	0.0%	0.0	0.357	+0.000	27.2	+0.5
Hidden residual	19.05	40.5%	99.6	0.355	−0.003	27.7	+0.9
JSD	10.15	68.3%	100.0	0.332	−0.025	23.0	−3.8
Top-1 agreement	7.08	77.9%	100.0	0.288	−0.070	5.7	−21.1
Fixed depth 32	32	0.0%	–	0.357	−0.001	27.7	+0.9
Fixed depth 24	24	25.0%	–	0.356	−0.002	28.8	+2.0
Fixed depth 20	20	37.5%	–	0.356	−0.001	27.1	+0.4
Fixed depth 18	18	43.8%	–	0.356	−0.002	26.6	−0.2
Fixed depth 16	16	50.0%	–	0.352	−0.006	25.0	−1.7
Fixed depth 12	12	62.5%	–	0.337	−0.020	21.8	−4.9
Fixed depth 10	10	68.8%	–	0.322	−0.035	17.1	−9.6
Fixed depth 8	8	75.0%	–	0.295	−0.062	8.2	−18.6

Table 5 applies the calibration gain of Section 4 at the thresholds shared across models, with the task-score change relative to full depth in parentheses. The rule with the largest gain changes across models and tasks, but a large gain can hide a large loss: on Ouro, entropy has the largest gain in six of eight cells while losing up to 17.7 points.

No score gain in Figure 3 is statistically significant. On Ouro, an exact paired (McNemar) test against full depth gives $p \geq 0.125$ for every gain. Answers change even without early exit, through run-to-run variation of the harness: the hidden residual never exits early on Ouro, but on Ouro-1.4B it turns 23 correct GSM8K answers into errors and fixes 22. On Huginn, whose outputs for individual examples are unavailable, all gains are on GSM8K and stay within the 0.9-point difference between its two full-depth runs.

D Wall-Clock Derivations and Measurement Details

Formally, with τ per loop step, the sequential schedule costs $T_{\text{seq}} = \tau N \bar{k}$ (the quantity our analytical tables report), while an idealized cross-token wavefront (a best-case latency model; assumptions below) gives

$$T_{\text{pipe}}(k) \;=\; \tau \cdot \max_{1 \leq t \leq N} \big(t + k_t - 1\big), \tag{2}$$

with full-depth references $T_{\text{seq}}^{\text{full}} = \tau N K$ and $T_{\text{pipe}}^{\text{full}} = \tau(N + K - 1)$.

Table 5: Calibration gain γ at the thresholds shared across models (Section 4), with the task-score change relative to full depth in points in parentheses. Cut: depth saving E averaged over tasks. Ouro: autoregressive harness on the full test sets. Huginn: fixed-threshold evaluation, whose exit depth is available only as a mean over all benchmarks, so E is shared across its columns; its Avg. covers 48 benchmarks, and MATH500 and HumanEval+ are not in its suite. Huginn has no trained gate.

Signal	Cut	GSM8K	MATH500	HumanEval+	MMLU	Avg.
Ouro-1.4B (K=4)						
Trained gate	21.2%	+0.23 (0.0)	+0.25 (+1.6)	+0.36 (+1.8)	+0.01 (0.0)	+0.21 (+0.9)
Max-softmax	26.9%	+0.20 (0.0)	+0.37 (0.0)	+0.50 (0.0)	+0.01 (0.0)	+0.27 (0.0)
Entropy	53.5%	+0.42 (−15.3)	+0.44 (−12.0)	+0.45 (−10.4)	+0.15 (−1.0)	+0.37 (−9.7)
JSD	29.3%	+0.35 (+0.2)	+0.37 (+1.2)	+0.45 (+1.2)	0.00 (0.0)	+0.29 (+0.7)
Hidden residual	0.0%	0.00 (−0.1)	−0.01 (−0.4)	0.00 (0.0)	0.00 (0.0)	0.00 (−0.1)
Top-1 agreement	20.2%	+0.20 (−2.1)	+0.19 (−2.2)	+0.24 (+1.2)	+0.11 (−0.1)	+0.19 (−0.8)
Ouro-2.6B (K=4)						
Trained gate	19.4%	+0.24 (+0.4)	+0.26 (−1.0)	+0.26 (+2.4)	0.00 (0.0)	+0.19 (+0.5)
Max-softmax	26.0%	+0.22 (0.0)	+0.36 (0.0)	+0.45 (0.0)	0.00 (0.0)	+0.26 (0.0)
Entropy	53.5%	+0.43 (−17.7)	+0.42 (−16.0)	+0.64 (−3.0)	+0.10 (−0.3)	+0.39 (−9.2)
JSD	30.9%	+0.36 (−1.5)	+0.36 (−1.2)	+0.46 (+2.4)	+0.02 (0.0)	+0.31 (−0.1)
Hidden residual	0.0%	0.00 (−0.4)	0.00 (0.0)	0.00 (0.0)	0.00 (0.0)	0.00 (−0.1)
Top-1 agreement	20.8%	+0.21 (−1.9)	+0.20 (−1.6)	+0.19 (−1.8)	+0.12 (−0.5)	+0.19 (−1.4)
Huginn (K=32)						
Max-softmax	0.0%	0.00 (+0.9)	–	–	0.00 (0.0)	0.00 (0.0)
Entropy	0.0%	0.00 (+0.5)	–	–	0.00 (0.0)	0.00 (0.0)
JSD	68.3%	+0.54 (−3.8)	–	–	+0.51 (−6.7)	+0.61 (−2.5)
Hidden residual	40.5%	+0.40 (+0.9)	–	–	+0.39 (−0.7)	+0.40 (−0.3)
Top-1 agreement	77.9%	−0.01 (−21.1)	–	–	+0.62 (−6.0)	+0.58 (−7.0)

Proposition 1 *For a prefill request whose per-token exit steps are $k_1, \ldots, k_N$, the pipelined wall-clock time defined by Equation 2 satisfies* (i) $T_{\text{pipe}}(k) \;\geq\; \tau\,(N + k_N - 1)$*;* (ii) *the pipelined wall-clock saving is bounded by*

$$\text{Save}_{\text{pipe}} \;\leq\; \frac{K - k_N}{N + K - 1}; \tag{3}$$

(iii) *if $N \gg K$, then* $\text{Save}_{\text{pipe}} = O(K/N)$*, whereas the analytical saving* $\text{Save}_{\text{seq}} = 1 - \bar{k}/K$ *is independent of N.*

Assumptions of the pipelined dispatch model. The wavefront model applies to *prefill* (teacher-forced) requests, where every input token is known in advance and only the within-token order of recurrent steps (step k reads the hidden state of step $k-1$) constrains the schedule; it does not model autoregressive decode, whose cross-token commit dependency serializes execution and whose behavior we characterize only through the measured dispatch regimes. The idealized wavefront of Equation 2 assumes every in-flight token advances one loop step per tick and that there is enough parallel capacity to run up to K loop steps of *different* tokens concurrently (e.g., one device per pipeline stage); it is a best-case latency model, not a single-device execution model, and the main text uses it strictly as a lower bound on pipelined prefill wall-clock.

Proof of Proposition 1. For (i), the maximum in Equation 2 is at least its value at $t = N$, namely $N + k_N - 1$; interior tokens can only keep this maximum or raise it, so the lower bound holds. (ii) substitutes (i) into $\text{Save}_{\text{pipe}} = 1 - T_{\text{pipe}}/T^{\text{full}}_{\text{pipe}}$ with $T^{\text{full}}_{\text{pipe}} = \tau(N + K - 1)$. (iii) is the $N \to \infty$ limit of (ii). □

The tail corollary. If the terminal answer token is treated as high-uncertainty and cannot exit early, then $k_N = K$ and the bound of Equation 3 is zero: no wall-clock saving survives, however aggressive the average. To convert an analytical saving into a real single-stream saving, the selected policy must exit early on the tail tokens; the governing quantity is $\max_t(t + k_t - 1)$, not $\bar{k}$. The

proposition does not exclude an interior $t+k_t$ larger than $N+k_N$, but that only worsens the realized saving.

Measurement setup and the two-device pipeline. Batching bridges the dispatch regimes: B concurrent requests fill the skipped slots, and aggregate wall-clock approaches the sequential cost over B. All measurements use RTX A6000 GPUs with Ouro-1.4B (BF16) under greedy decoding from a fixed prompt, in a PyTorch eager prototype that is deterministic across repeats and, with eager attention, token-exact against the reference generation; the default SDPA backend selects different kernels in the per-step executor and in the reference forward pass, and the resulting BF16 differences flip generated tokens. Over 512 generated tokens, true skipping (commit at step 2 of 4, never execute the rest) measures $2.00\times$ / $1.92\times$ / $2.12\times$ at batch sizes 1/8/32. Same-device overlap, which commits at step 2 but still executes steps 3–4 on a second CUDA stream, reaches $1.30\times$ / $1.24\times$ / $1.11\times$ drained throughput without removing any computation. A two-GPU pipeline that commits shallow tokens on one device while the second keeps steps 3–4 cache-complete reaches $2.00\times$ foreground and $1.97\times$–$1.98\times$ drained throughput when every token commits at step 2 (64 generated tokens), and $1.41\times$–$1.84\times$ drained throughput when max-softmax (0.9) or JSD (0.02) decides per token whether to wait for step 4. Its outputs and exit depths equal the single-GPU reference, and it holds a second model replica: a latency/throughput scale-out, not a compute saving.

Token-adaptive exit with cache back-fill. At batch size 1, a lazy executor runs only the steps each token selects and back-fills a skipped step of an earlier token only when a later token requests that depth; its outputs equal those of the same policy executed with all four steps. Against that full-compute execution, the native gate at τ=0.5 reaches $1.31\times$ (mean selected depth 2.48; 130 depth increases trigger 279 back-fill jobs, so it executes 76% of the full-depth jobs), JSD at 0.01 reaches $1.92\times$ (depth 2.07; 15 back-fill jobs; 52% of the jobs), and max-softmax at 0.9 reaches $3.62\times$ (depth 1.07; 28% of the jobs). The executed job counts match the cache-prefix formula, so the runtime cost follows cache-feasible jobs rather than the mean selected depth.

Huginn prefill estimates. Huginn prefill requests average about 385 tokens. JSD's analytical saving of 68.5% becomes a pipelined saving of 5.0% (ratio 0.072), and the hidden residual drops from 40.5% to 2.3% (ratio 0.057). These ratios match the $O(K/N)$ bound: $32/385 \approx 0.083$.

E Additional Details and Results

Prespecified threshold grids. The 111 candidates instantiate each expert family on a grid fixed in the harness source before the policy analysis (`run_fixed_point_halting.py`): the hidden residual (with patience 1 and 2), JSD, and KL on $\{0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5\}$, entropy on $\{0.5, 1, 2, 3, 4, 5, 6\}$, margin on $\{0.05, 0.1, 0.2, 0.4, 0.6, 0.8, 0.9\}$, max-softmax on $\{0.6, 0.8, 0.9, 0.95, 0.99, 0.999, 0.9999\}$, the trained gate on $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.7\}$, top-1 agreement with patience 1 and 2, and fixed depths of 1, 2, and 3 steps, together with full depth. Committees use dedicated reduced grids (`run_fixed_point_halting.py`): max-softmax $\{0.6, 0.8, 0.9\}\times$ JSD $\{0.01, 0.05, 0.1\}$, margin $\{0.1, 0.4, 0.8\}\times$ JSD $\{0.01, 0.05, 0.1\}$, entropy $\{2, 4, 6\}\times$ top-agreement (patience 1), margin $\{0.1, 0.4, 0.8\}\times$ top-agreement (patience 1), and confidence $\times$ hidden-residual pairs on max-softmax $\{0.6, 0.8, 0.9\}$ / entropy $\{2, 4, 6\}$ $\times$ hidden $\{0.02, 0.05, 0.1\}$. None of the grids is tuned on validation data; only the selection among the resulting candidates uses the validation split.

E.1 Data for Figure 1(a) and (c)

Both panels use our autoregressive harness with greedy decoding and the same 17-policy batch as the Ouro GSM8K runs (full depth, fixed depths of 1–3 steps, six native-gate thresholds, and seven max-softmax thresholds). MGSM translates the first 250 GSM8K test problems into each language (Shi et al., 2023); we evaluate English, Spanish, French, German, Russian, Chinese, and Japanese with English five-shot exemplars, the question in the target language, and a 256-token output cap. The English outputs are identical token for token to our GSM8K run on the same problems. Panel (a) shows, for each model, the language with the largest loss at τ=0.1: Russian for Ouro-1.4B (-11.6 points, 95% CI $[-16.4, -6.4]$; 38 correct answers become wrong and 9 wrong answers become correct) and Spanish for Ouro-2.6B (-10.4, $[-15.6, -5.6]$; 36 and 10). Over all

seven languages the losses are 5.4 and 5.7 points. A policy that never exits early, run in a separate batch, changes a language's accuracy by at most 2.8 points; 29% of Ouro-1.4B's full-depth Russian outputs reach the output cap. Compositional GSM chains two GSM8K problems so that the second uses the answer to the first (Hosseini et al., 2024); we use the first 600 of its 1,200 problems with the GSM8K five-shot prompt and a 512-token cap. At τ=0.2 the gate changes accuracy by -4.5 points on Ouro-1.4B ($[-8.5, -0.5]$) and $+2.7$ on Ouro-2.6B ($[-1.7, +7.0]$), a difference of 7.2 points ($[1.2, 13.2]$, paired by problem); at τ=0.1 the changes are -4.8 and -3.5 points.

E.2 Extended Related Work

Reading intermediate computation. Our per-step signals inherit a methodological lineage of reading intermediate representations in feed-forward models: the logit lens and tuned lens project intermediate layers onto the vocabulary space (nostalgebraist, 2020; Belrose et al., 2023). Looped models swap the depth axis of this lineage for a time axis: we read the outputs of one weight block at successive steps. Unlike the lens literature, which reads these outputs for interpretation, we use them to decide when to stop, as layer-wise early exit does in feed-forward models.

Recent related work. LoopFormer trains looped Transformers to run at a user-specified depth budget that applies to the whole sequence (Jeddi et al., 2026), whereas EoE selects token-level stopping rules for existing checkpoints without training. GateSkip fine-tunes pretrained models with residual-stream gates that skip layers token by token (Laitenberger et al., 2026). It reports that prior adaptive-compute methods often collapse on generative reasoning, and that FREE keeps strong log-likelihood scores while remaining flat on generation; this gap between scoring fixed text and generating it resembles the one we find between teacher-forced diagnostics and autoregressive evaluation (Section 5). Damri & Laufer-Goldshtein (2026) apply Learn-then-Test to a confidence-based rule that chooses how many recognition hypotheses to pass to LLM-based error correction, bounding the expected relative word-error-rate degradation with high probability; EoE-Safe instead selects among rules from different expert families with simultaneous bounds.

E.3 Threshold Provenance for the Fixed-Threshold Evaluation

Deployment thresholds for the fixed-threshold evaluation are a priori constants fixed in the serving layer before benchmark evaluation (e.g., 0.999 for max-softmax, 0.005 for JSD, and 0.05 for the hidden residual); a source audit confirms that they were committed once and never revised, so these thresholds were not tuned on benchmark outcomes. On Ouro, the same values are points of the prespecified grids of the policy analysis (Appendix E); on both models, the hidden residual and top-1 agreement require two consecutive satisfied comparisons. On Huginn, the output-based signals are computed from a pre-coda proxy distribution, the final norm and output head applied to each step's recurrent state, which avoids running the coda at every step; the hidden residual compares recurrent states directly. All 32 steps are executed and each token is read out through the coda at its selected step, so, as in our harness, the reported depth cuts are analytical. For Ouro, the fixed-threshold values come from the policy analysis on the reference checkpoints at the same thresholds; the Ouro serving wrapper is not used, because in a same-protocol GSM8K check the reference implementation did not reproduce its early-exit scores.

E.4 Statement and Proof Sketch of Proposition 2

Proposition 2 *Suppose the per-step output distributions converge geometrically, $D\big(p_k, p_\infty\big) \le C\rho^{\,k}$ for some $\rho \in (0,1)$ and a divergence D that satisfies the triangle-type bound $D(p_k, p_{k-1}) \le C'\rho^{\,k-1}$ and dominates the sup-norm difference, $\|p-q\|_\infty \le D(p,q)$ (total variation does; the thresholded JSD signal of the study instantiates the template approximately, since JSD itself is not a metric). Consider a convergence candidate that exits at the first k with $D(p_k, p_{k-1}) \le \tau$, and a confidence candidate that exits at the first k with $\max_y p_k(y) \ge \tau_c$. Then: (i) the convergence candidate fires by step $k^*(\tau) = \max\big(2, \lceil \log(C'/\tau)/\log(1/\rho) \rceil + 1\big)$ (consecutive distributions can first be compared at k=2), which does not depend on K; provided $K \ge k^*(\tau)$, its saving $K - k^*(\tau)$ grows linearly in K; (ii) if the limiting confidence $\max_y p_\infty(y)$ of a token is below $\tau_c - C\rho^{\,k}$ for all $k \le K$, the confidence candidate never fires on that token at any depth.*

Table 6: Per-task expansion of Table 1a: per-family calibrated selection on Ouro under the EoE-SAFE max-t certificate (three seeds, ϵ=0.01, α=0.05; band computed over each selector's own search pool). #L is analytic recurrent steps per token; MATH500 uses a 512-token cap. Bold best, underline second best, marked on the AVG columns only.

	GSM8K		MATH500		HumanEval+		MMLU		AVG.	
Selector	Acc↑	#L↓	Acc↑	#L↓	Acc↑	#L↓	Acc↑	#L↓	Acc↑	#L↓
Ouro-1.4B										
full-depth decoding (K=4)	76.07	4.00	50.13	4.00	44.72	4.00	65.76	4.00	59.17	4.00
Fixed-depth	76.07	4.00	50.13	4.00	51.22	3.33	65.76	4.00	**60.80** (↑1.63)	3.83 (↓4.2%)
Trained gate	75.92	3.51	50.13	4.00	51.22	1.82	65.25	3.34	<u>60.63</u> (↑1.46)	3.17 (↓20.8%)
Max-softmax	76.07	3.20	50.53	2.20	43.90	1.32	65.19	3.41	58.93 (↓0.24)	2.53 (↓36.6%)
Entropy	76.07	4.00	50.13	4.00	44.72	3.08	65.61	3.76	59.13 (↓0.04)	3.71 (↓7.3%)
Margin	76.07	4.00	50.13	4.00	44.72	3.10	65.48	3.56	59.10 (↓0.07)	3.67 (↓8.4%)
Convergence	76.07	4.00	49.47	3.50	47.56	2.22	65.38	3.54	59.62 (↑0.45)	3.32 (↓17.1%)
Committee	76.07	4.00	50.13	4.00	44.72	2.15	65.32	3.24	59.06 (↓0.11)	3.35 (↓16.3%)
EoE-Free (ours)	76.07	3.20	50.53	2.20	44.31	1.29	65.32	3.24	59.06 (↓0.11)	**2.48** (↓37.9%)
EoE-All (ours)	76.07	3.20	50.53	2.20	44.31	1.29	65.32	3.24	59.06 (↓0.11)	**2.48** (↓37.9%)
Ouro-2.6B										
full-depth decoding (K=4)	87.76	4.00	59.73	4.00	40.24	4.00	72.97	4.00	65.18	4.00
Fixed-depth	87.76	4.00	59.73	4.00	39.84	3.67	72.97	4.00	65.08 (↓0.10)	3.92 (↓2.1%)
Trained gate	87.66	3.17	59.73	4.00	40.24	3.39	72.44	3.40	65.02 (↓0.16)	3.49 (↓12.7%)
Max-softmax	87.51	2.91	59.73	4.00	39.84	1.35	72.92	3.56	65.00 (↓0.18)	2.96 (↓26.1%)
Entropy	87.76	4.00	59.73	4.00	40.24	4.00	72.69	3.58	65.11 (↓0.07)	3.90 (↓2.6%)
Margin	87.76	4.00	59.73	4.00	40.24	4.00	72.83	3.25	<u>65.14</u> (↓0.04)	3.81 (↓4.7%)
Convergence	87.76	4.00	59.73	4.00	41.46	3.30	72.47	3.48	**65.36** (↑0.18)	3.70 (↓7.6%)
Committee	87.76	4.00	59.73	4.00	40.24	2.08	72.68	3.09	65.10 (↓0.07)	3.29 (↓17.7%)
EoE-Free (ours)	87.51	2.91	59.73	4.00	39.84	1.35	72.68	3.09	64.94 (↓0.24)	<u>2.84</u> (↓29.1%)
EoE-All (ours)	87.35	2.89	59.73	4.00	39.84	1.35	72.68	3.09	64.90 (↓0.28)	**2.83** (↓29.2%)

Proof sketch. (i) By assumption $D(p_k, p_{k-1}) \le C'\rho^{k-1}$; setting this at most τ and solving for k gives $k^*(\tau)$ independent of K, and the saving is $K - k^*$. (ii) By convergence, $\max_y p_k(y)$ differs from the limiting confidence by at most $C\rho^k$, so if the limiting confidence is below τ_c minus that bound, no step fires. □

E.5 Paired Uncertainty and Test-Score Resolution

For binary correctness scores, a paired difference D_i lies in $\{-1, 0, 1\}$. Let $p_{\text{flip}} = \Pr(D_i \neq 0)$ and $\Delta = \mathbb{E}[D_i]$; then $\mathrm{Var}(D_i) = p_{\text{flip}} - \Delta^2$. For independent validation examples and Δ near zero, a one-sided normal approximation gives the single-candidate uncertainty margin

$$m_{\text{pair}} \;\approx\; z_{1-\alpha}\sqrt{\frac{p_{\text{flip}}}{n}}, \tag{4}$$

where n is the validation size and $z_{1-\alpha}$ is the standard-normal quantile. At p_{flip}=0.03, n=660, and α=0.05, this margin is about 0.011. It describes an approximate uncertainty scale, not a universal minimum quality budget: acceptance also depends on the estimated quality difference, and simultaneous calibration uses a different critical value.

Test-score resolution is a separate issue. On HumanEval+, a test half has 82 examples, so a one-example change in correctness changes the score by $1/82 \approx 1.22$ percentage points, more than the one-point budget. Table 1b retains these trials in all overall counts. Its supplementary Res. column excludes the 40 HumanEval+ trials and reports the remaining 120 trials whose one-example score increment stays within the budget.

E.6 Per-Task Selection Results on Ouro

Table 6 expands the per-model averages of Table 1a into per-task scores.

E.7 PER-SEED PONDERLM-2 SELECTION AUDIT

Table 7 reports every (scale, seed) selection on LAMBADA behind the PonderLM-2 results of Section 7, recomputed under the same max-t protocol as the Ouro tables (ϵ=0.01, α=0.05): the selected candidate, its validation lower bound, and the held-out delta, steps, and cut. No seed falls back, and every held-out delta lies within one budget of full depth. Compared with per-candidate (marginal) calibration, the simultaneous band rejects the two aggressive fixed-depth-2 picks on 1.4B (seeds 1–2, 50% cuts) in favor of the deeper KL-convergence candidate, making the selected expert identity unanimous across scales and seeds.

Table 7: PonderLM-2 selection audit on LAMBADA (K=4; validation/test halves $2576/2577$; EOE-SAFE max-t, ϵ=0.01, α=0.05; Val. $L(\pi)$ is the simultaneous validation lower bound of the selected candidate, all clearing $-\epsilon = -1$pp). Full-depth test scores: 52.6–52.9% (410M), 59.9–61.3% (1.4B).

Scale	Seed	Selected	Val. $L(\pi)$	Test Δ	#L	Cut
410M	0	`kl@.05`	-0.80pp	$+0.00$pp	2.27	43.2%
410M	1	`kl@.05`	-0.79pp	-0.04pp	2.28	43.1%
410M	2	`kl@.05`	-0.63pp	-0.12pp	2.27	43.2%
1.4B	0	`kl@.05`	-0.33pp	$+0.54$pp	2.63	34.3%
1.4B	1	`kl@.05`	-0.28pp	$+0.54$pp	2.62	34.6%
1.4B	2	`kl@.05`	-0.04pp	$+0.27$pp	2.62	34.5%