Think When Unsure: Leveraging Model Confidence to Decide When to Use Chain-of-Thought

Samuel Lewis-Lim, Xingwei Tan, Zhixue Zhao, Nikolaos Aletras
School of Computer Science, University of Sheffield
United Kingdom
{slewis-lim1, xingwei.tan, zhixue.zhao, n.aletras}@sheffield.ac.uk

Abstract

Chain-of-thought (CoT) prompting improves performance on complex reasoning tasks, but often triggers unnecessary computation on simpler queries. A natural approach is to use the model's own confidence as a training-free signal to gate this decision, answering directly when confident and invoking CoT only when uncertain. Across four training-free confidence signals, three model families, and seven reasoning benchmarks, we find that the effectiveness of confidence-gated CoT varies, and it is unclear how to predict whether it will work for a new model. We address this with two key contributions. First, we identify that gating fails in the following circumstances: signal saturation, where confidence scores collapse to a narrow range and lose discriminative power; and selection bias, when confidence-gated routing sends the longest queries to CoT. Second, we propose a novel Pareto-optimal threshold selection method. This has the dual ability to preserve accuracy while reducing costs where the confidence signal is strong, and maintains accuracy where it is weak. Where the confidence signal is strong, gating reduces CoT usage by 30% without sacrificing accuracy. These findings provide practical guidance for evaluating and deploying models that selectively use CoT. [1]

1 Introduction

Chain-of-thought (CoT) prompting improves complex reasoning, including math and symbolic reasoning, and scientific question answering (Wei et al., 2022; Guo et al., 2025; Qwen Team, 2025). However, for tasks such as basic question answering and commonsense reasoning, CoT provides little benefit while substantially increasing token usage and latency (Liu et al., 2024; Sprague et al., 2025; Aggarwal et al., 2025).

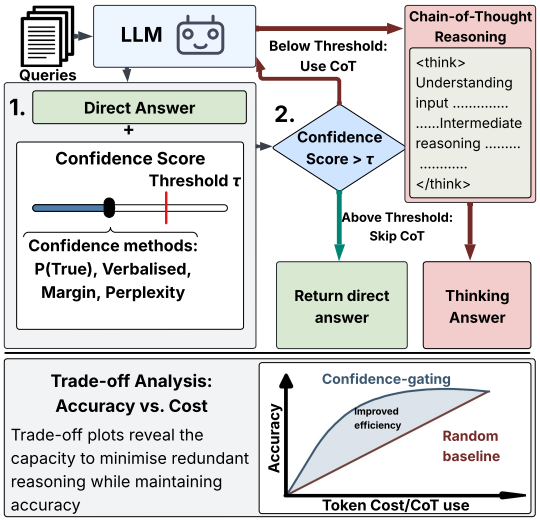


Figure 1: Confidence-gated CoT (top): the model produces a direct answer with a confidence estimate, triggering CoT only when confidence is low. The trade-off analysis (bottom) reveals how well each signal reduces unnecessary reasoning compared to random gating.

Recent models offer hybrid thinking modes (Qwen Team, 2025), while others provide separate instruct and thinking variants (Olmo et al., 2025). In both cases, users decide when using long CoT reasoning is appropriate. However, this still requires anticipating the necessity of CoT reasoning for each query. We refer to this as the *when-to-think* decision, in contrast to *how-long-to-think* methods that control the length of CoT once it has already been triggered. Most existing efficient reasoning methods address the *how-long-to-think* problem (Yang et al., 2025; Huang et al., 2025b; Liu et al., 2025); however, there is still comparatively little focus on the *when-to-think* setting, especially using training-free signals (Yue et al., 2025; Jiang et al., 2025; Chuang et al., 2025a).

Most prior when-to-think methods rely on reinforcement learning or classifiers, requiring additional training (Yue et al., 2025; Jiang et al., 2025;

[1] Our code is available at: https://github.com/samlewislim/cgr

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 369–396
November 6-10, 2026 ©2026 Association for Computational Linguistics

Chuang et al., 2025a). Training-free alternatives focus solely on perplexity (Lu et al., 2025). More broadly, confidence estimation methods, which use output probabilities or self-evaluation to assess prediction reliability (Kadavath et al., 2022; Tian et al., 2023; Farquhar et al., 2024), offer a natural signal for this decision and are effective for model routing (Ramirez et al., 2024; Chuang et al., 2025b). However, while confidence signals offer a natural heuristic, our evaluation demonstrates that their effectiveness in gating CoT varies across tasks and models. Furthermore, it remains unclear how to predict these inconsistencies or derive effective thresholds. To address this unpredictability, we make two key contributions. First, we identify that gating fails specifically due to signal saturation and selection bias. Second, we introduce a novel Pareto-optimal threshold selection method that reliably preserves accuracy while reducing unnecessary CoT usage, providing a practical method for deploying an efficient, selective-reasoning model.

Contributions:

- *An analysis of when and why confidence gating fails.* We identify how signal saturation and selection bias undermine discriminative power and prevent real-world token savings.
- *A novel Pareto-optimal threshold selection method* that reduces CoT usage by up to 30% without sacrificing accuracy. This approach outperforms CAR (Lu et al., 2025), the only prior confidence-based when-to-think method.
- *The first systematic evaluation of training-free confidence signals for compute-efficient reasoning*, scaling across four signals, three model families, and seven benchmarks to provide practical deployment guidance.

2 Related Work

Efficient and Adaptive Reasoning. Adaptive reasoning enables LLMs to dynamically adjust the depth or length of their reasoning process (Yue et al., 2025). Prior work either controls *how-long-to-think* once reasoning has started, or learn *when-to-think* policies that decide whether to invoke deeper reasoning. *How-long-to-think* approaches include early-exit methods using confidence or entropy monitors, probes, or decoding-time controls. (Yang et al., 2025; Zhang et al., 2025a; Huang et al., 2025b). Complementary work shortens reasoning through training on shorter CoT, or uses length-aware rewards that discourage redundant steps (Liu et al., 2025; Shen et al., 2025; Huang et al., 2025a).

When-to-think methods mostly rely on supervision or reinforcement learning to learn routing between reasoning modes, such as direct answering versus long CoT or fast versus slow thinking (Yue et al., 2025; Jiang et al., 2025; Chuang et al., 2025a). Unlike early-exit methods, which require generating partial reasoning before deciding to stop (Yang et al., 2025), these approaches skip reasoning entirely. Importantly, the two methods are orthogonal and complementary, since confidence gating selects when to trigger reasoning while early exiting can still be applied once CoT has been selected. Closest to our setting, Certainty-based Adaptive Reasoning (CAR) uses answer perplexity as a trigger for longer reasoning, applying Gaussian fitting to derive routing decisions (Lu et al., 2025). Confidence-based signals have also been used to route between models (Ramirez et al., 2024; Chuang et al., 2025b), and confidence tokens have been trained to route between reasoning modes (Chuang et al., 2025a). In contrast to these methods, we require no training and route between reasoning modes within a single model. We systematically compare multiple training-free confidence signals for this decision across models and tasks, and introduce a threshold selection method that jointly optimises accuracy and token cost, which we show outperforms CAR in §6.3. Parallel work has applied conformal risk control to provide formal risk guarantees for *how-long-to-think* early-exit thresholding (Wang et al., 2026). In our setting, we instead determine gating thresholds empirically from a calibration set to reliably bound downstream performance degradation.

Uncertainty Estimation in LLMs. Methods for estimating the reliability of LLM predictions derive confidence directly from model probabilities or logits. This includes perplexity, the difference between top token probabilities (Ramirez et al., 2024), and $P(\text{True})$ (Kadavath et al., 2022). Others prompt the LLM to verbalise a confidence score (Tian et al., 2023; Xiong et al., 2024; Yang et al., 2024; Zhou et al., 2025). Multi-sample methods generate multiple responses and measure agreement or semantic diversity (Kuhn et al., 2023; Farquhar et al., 2024). Finally, trained confidence predictors have also been proposed (Kossen et al., 2025; Chuang et al., 2025a; Damani et al., 2025; Stangel et al.,

2025). We focus on low-latency, training-free confidence signals, prioritising computational efficiency and low inference cost.

Model Cascades and Routing. These methods dynamically switch between models based on query difficulty, using trained routers (Ong et al., 2025; Feng et al., 2025) or confidence-based signals (Ramirez et al., 2024; Chuang et al., 2025b). While prior work routes queries across different models, we study an analogous setting between reasoning modes in a single model.

3 Confidence-Gated Chain-of-Thought

We define confidence-gated CoT, where a model selectively triggers reasoning based on its confidence estimate. Each query is first answered directly. If the confidence score is low, the model re-runs the query with CoT enabled.

Problem Definition More specifically, for each input x_i, a model, parametrised by θ, generates a direct answer and a confidence score $s(x_i; \theta)$. If the score is above a threshold τ, the direct answer is accepted; otherwise, the model answers the question with CoT enabled:

$$gate(x_i; \tau, \theta) = \begin{cases} \textsc{CoT}(x_i; \theta), & s(x_i; \theta) < \tau \\ \textsc{Direct}(x_i; \theta), & s(x_i; \theta) \geq \tau \end{cases}$$

CoT: The model generates an intermediate reasoning trace before emitting a final answer (Wei et al., 2022; Guo et al., 2025; Qwen Team, 2025).

Direct: The model is instructed to output only the final answer without generating intermediate reasoning. To achieve this, we append a concise instruction such as "*Answer:*" to the prompt, which elicits a short response with no CoT or explanation.

4 Trade-off Analysis and Threshold Selection

We first introduce a budget-based analysis to map CoT usage trade-offs and diagnose gating failures. Building on this, we propose a Pareto-optimal threshold selection method to identify decision boundaries for practical deployment.

4.1 CoT Budget

First, we evaluate accuracy and inference cost while limiting how frequently the model uses CoT using a predefined budget. We define the *CoT budget* as the fraction of input queries for which the model is permitted to invoke CoT, with all remaining queries answered directly. We sweep percentiles of the confidence score distribution to set these thresholds, allocating a fixed fraction of queries to CoT. This allows us to trace accuracy-efficiency trade-offs across budgets, plotting accuracy against average token cost or CoT usage. In addition to token cost, we also measure mean per-query wall-clock latency for each signal. As we use percentile thresholds, post-hoc (Guo et al., 2017) calibration methods are not applicable, as they adjust values but preserve their ranking.[2]

Identifying Budget Thresholds We consider both offline and online settings for obtaining percentile thresholds. In the *offline setting*, all direct answers and confidence scores are computed first, giving access to the full distribution of confidence scores before any decision about using CoT is made. This allows thresholds to be set exactly at chosen percentiles. In the *online setting*, queries arrive one at a time, so the full confidence score distribution is unavailable. Instead, we maintain a running score history and apply the dynamic percentile method of Ramirez et al. (2024). After each query t, its confidence score is added to the history and the threshold τ_t is set to the p-th percentile of $\{s(x_1), \ldots, s(x_{t-1})\}$, so that roughly $p\%$ of queries are routed to CoT. We randomise dataset order, use a short warm-up phase (the first 20 queries answered directly) to seed this history, and additionally test robustness to a worst-case difficulty-ordered stream in Appendix H.

4.2 Pareto-Optimal Threshold Selection

CoT budget-based curves characterise the full range of accuracy–cost trade-offs. *However, practically we often select a single threshold. Therefore, we propose a method for selecting Pareto-optimal gating thresholds from confidence scores.* Unlike prior work that optimises a decision boundary for correctness prediction (Lu et al., 2025), our method jointly optimises for accuracy and efficiency. A threshold is Pareto-optimal if no alternative threshold achieves equal or higher accuracy at a lower token cost. The set of such thresholds forms the Pareto front, which traces the best accuracy-cost trade-offs. From this front, we derive the threshold τ^* with the lowest token cost whose accuracy

[2]These methods would not affect the outcome of experiments since the decision based on percentile thresholds only depends on the ranking of queries.

remains within a tolerance ϵ of always using CoT:

$$\tau^* = \arg\min_\tau \text{Tok}(\tau) \quad \text{s.t. } \text{Acc}(\tau) \geq \text{Acc}_{\text{All-CoT}} - \epsilon.$$

To reflect realistic deployment, τ is estimated using a calibration set. Percentile thresholds are swept on this set to construct the Pareto front, after which the selected threshold is applied to the test set. To account for variability in calibration set sampling, we repeat this process using Monte Carlo cross-validation (Xu and Liang, 2001), reporting the mean and standard deviation of accuracy and token usage across runs. Although this uses a labelled calibration set, the method remains training-free in that no model parameters are updated. Given direct and CoT outputs, selecting τ^* is a simple percentile sweep that runs in seconds on a CPU.

5 Experimental Setup

5.1 Models

We use three open-weight models: Qwen3 (8B/32B) (Qwen Team, 2025) and GPT-OSS-20B (OpenAI, 2025), plus the closed-weight GPT-5.1 (OpenAI, 2025). All allow for direct answering and reasoning modes. Qwen3 provides non-thinking and thinking modes. GPT-OSS supports three reasoning effort levels (low, medium, high); for direct answering, we append the closing `</think>` tag to bypass reasoning, following the same convention Qwen3's non-thinking chat template applies automatically. Adding the "Answer:" instruction (§3) is done identically across all models. Unless otherwise stated, we use the *medium* setting for GPT-OSS, with results for other effort levels reported in Appendix M. GPT-5.1 similarly allows switching between direct generation and multiple reasoning effort levels.

5.2 Datasets

We include seven benchmarks (details in Table 5, Appendix E) from four reasoning types following Sprague et al. (2025): (1) *commonsense reasoning* including CommonsenseQA (CSQA) (Talmor et al., 2019) and StrategyQA (Geva et al., 2021); (2) *knowledge-based reasoning* using MMLU-Redux (Gema et al., 2025); (3) *mathematical and scientific reasoning* on GPQA (Rein et al., 2024) and GSM8k (Cobbe et al., 2021); and (4) *soft reasoning* using LSAT-AGI (Zhong et al., 2024) and MUSR (Sprague et al., 2024). This diverse range of reasoning types allows us to test tasks where CoT has shown different levels of effectiveness.

5.3 Confidence Estimation Methods

We focus on confidence estimates that are produced by the model itself or based on its outputs without using an external predictor.

Perplexity: The perplexity of a generated answer measures the LLM's confidence in it, computed as the exponentiated average negative log-likelihood over the answer tokens (Lu et al., 2025).

$P(\textbf{True})$ **(Kadavath et al., 2022):** It first generates an answer via direct prompting. Then, it asks the LLM whether the answer is *(A) True* or *(B) False* in a second forward pass to extract the probability of generating the token "A". Full prompt details are found in Appendix F.

Margin Sampling: It measures the probability difference between the most likely and second most likely model predictions for a given input, typically used for model cascades (Ramirez et al., 2024).

Verbalised Confidence: This prompts the LLM to self-evaluate and express its confidence as part of its response (Yang et al., 2024). Following prior work (Yang et al., 2024; Tian et al., 2023), we ask the model to output a confidence score between 0 and 1 after its answer, which provides good calibration. Full prompt details are found in Appendix F.

5.4 Confidence Baselines

Expected Random Baseline. For a CoT budget $r \in [0, 1]$, we compute the expected accuracy and token cost as a weighted average of direct and CoT performance, with weights $1 - r$ and r, respectively. We compute these analytically rather than via random sampling, yielding a stable baseline. [3]

Oracle. We also include an oracle that triggers CoT whenever the direct answer is incorrect, acting as a perfect predictor of correctness. It represents the maximum performance that a confidence signal could achieve, serving as an upper bound baseline.

6 Results

We evaluate confidence-gated CoT across all models, signals, and tasks. We first characterise accuracy and efficiency trade-offs (§6.1), then analyse the conditions that determine gating success (§6.2),

[3] We also evaluate a keyword-based heuristic that triggers CoT based on input keywords in Appendix O. As it only produces a single fixed point, we use random for the main results.

and finally provide results in practical deployment settings (§6.3).

6.1 Accuracy–Efficiency Trade-offs

We evaluate accuracy–efficiency curves using percentile budgets as defined in §4.1. At each budget level, we report both accuracy and average token usage. Figure 2 shows aggregate results for GPT-OSS-20B, Qwen3-32B, Qwen3-8B, and GPT-5.1 comparing confidence-based gating against random selection and the oracle. For GPT-OSS-20B and Qwen3-32B, several confidence methods clearly outperform random gating. In particular, *margin* and *perplexity* are most effective for GPT-OSS-20B, while $P(\mathit{True})$ is best for Qwen3-32B. Using these signals, both models match the accuracy of always using CoT while reducing CoT usage by roughly 30%, saving an average of 70–100 tokens per query. In contrast, no confidence method consistently outperforms random gating for Qwen3-8B or GPT-5.1 across all budgets. Finally, the oracle performance indicates substantial headroom, e.g., GPT-OSS-20B could obtain higher accuracy while invoking CoT on fewer than half of the queries. *Overall, the best model and signal combinations match full-CoT accuracy while reducing CoT usage by around 30%, though a gap remains to oracle performance.* We next examine where these gains come from across tasks and models.

Commonsense, Soft reasoning, and Knowledge Tasks Benefit the Most. On MMLU, StrategyQA, and MUSR, confidence-gated CoT enables GPT-OSS and Qwen3-32B to match always-CoT accuracy while reducing token usage by 30 to 50%. Figure 3 shows representative examples. However, the oracle indicates room for improvement, using up to 75% less CoT for CSQA and StrategyQA. In contrast, mathematical and scientific tasks show limited benefit. For GSM8K, direct answering without CoT yields low accuracy, making token savings difficult without performance drops, a limitation reflected by the oracle. Similarly, on GPQA, certain confidence methods (e.g., perplexity for GPT-OSS-20B) outperform random gating, but efficiency gains remain small. While the oracle suggests potential improvements, current models lack sufficient discriminative ability on these challenging questions. Full results are in Appendix J. Overall, confidence-gating is most effective for tasks where high accuracy is possible with direct answer.

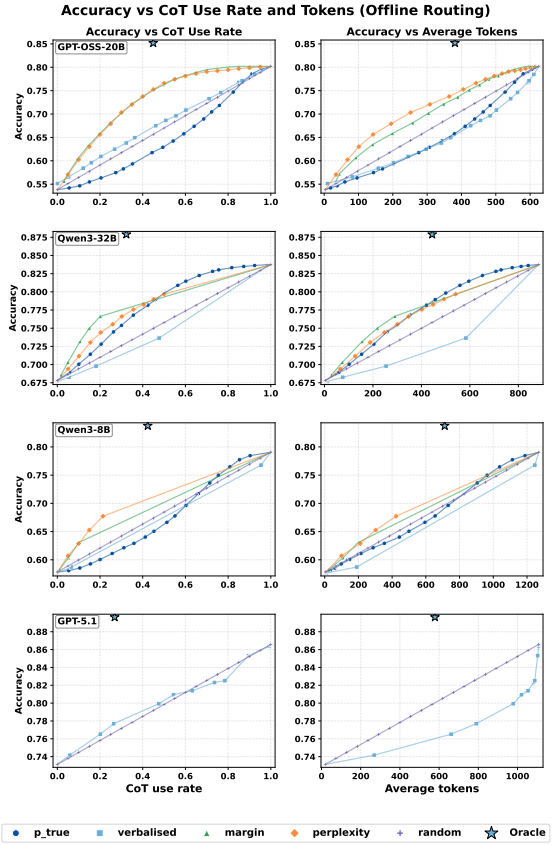


Figure 2: **Offline accuracy–efficiency trade-offs under percentile budgets.** Accuracy vs. CoT usage (left) and average tokens (right), aggregated over all datasets for all models. Curves show confidence signals vs. the random baseline; stars denote the oracle.

Model-Specific Confidence Effectiveness. Figure 2 reveals that the effectiveness of confidence signals is not uniform across models. A signal that acts as a strong predictor on one model (e.g., margin on GPT-OSS-20B) often performs worse than random on another. Conversely, $P(\mathrm{True})$ excels on Qwen3-32B but fails on GPT-OSS-20B. Furthermore, for models like Qwen3-8B and GPT-5.1, no tested method consistently outperforms random gating across all budgets. Ultimately, there is no universally effective confidence signal and utility is highly model-dependent. To understand this inconsistency, we must look at the underlying mechanics of how these signals behave.

6.2 When Does Confidence Gating Work?

Having established where confidence gating is effective, we now examine what determines its success. We find that performance is primarily driven by two factors: the discriminative power of the confidence signal and the impact of selection bias on the total token cost.

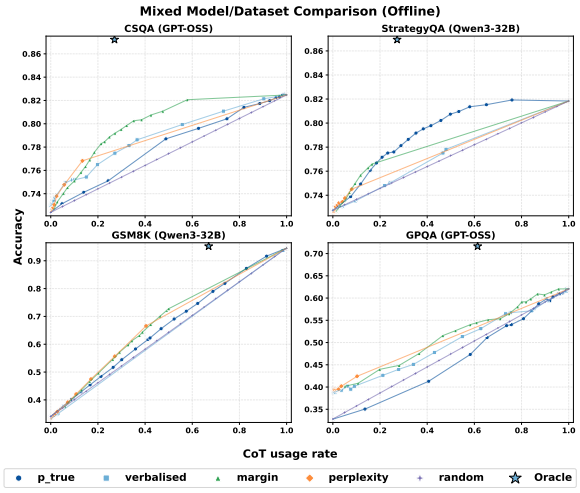


Figure 3: **Task-level accuracy–efficiency trade-offs** in CSQA, StrategyQA, GSM8K and GPQA, comparing confidence-gating to random and oracle across models.

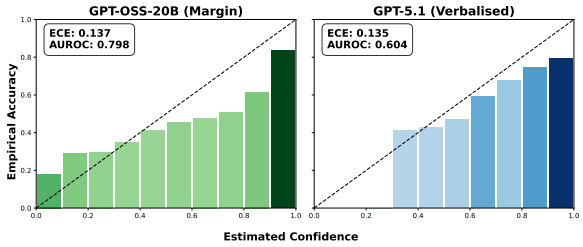


Figure 4: Reliability diagrams for GPT-OSS-20B (margin) and GPT-5.1 (verbalised). Both show similar ECE, whereas GPT-OSS has better AUROC.

Scale, Calibration, and Discriminative Power. We assess confidence signals using ECE for calibration (Guo et al., 2017; Pavlovic, 2025) and AUROC for discriminative power (Kadavath et al., 2022). Because gating relies on relative ranking rather than absolute probabilities, it depends fundamentally on discriminative power, which AUROC directly measures. As Jiang et al. (2024) demonstrate, ranking serves as a lower bound for calibration; as such, confidence scores can suffer from poor absolute ECE while still preserving the accurate rank ordering necessary for threshold-based routing. This dynamic is clearly evident in our results. For example, GPT-5.1 (verbalised) and GPT-OSS (margin) share nearly identical ECE scores (Figure 4), but GPT-OSS-20B achieves a higher AUROC and consequently delivers substantially better gating performance. More broadly, larger models generally attain higher AUROCs across signals, reinforcing the association between model scale and stronger discriminative power (Kadavath et al., 2022). Our full results are detailed in Appendix K. These results show that AUROC is a far more reliable predictor of gating success than ECE. Therefore, practitioners seeking to adopt confidence-based routing should prioritise AUROC when evaluating and selecting between available signals.

Signal Saturation. Even when a signal possesses discriminative power, its practical utility can be compromised by signal saturation. We define saturation as the phenomenon where a confidence method's score distribution collapses to a narrow range. For instance, Figure 2 shows that while margin on Qwen3-32B works well initially (even outperforming the other methods) it quickly saturates, with scores concentrating heavily near 1.0. When many queries share the exact same confidence value, they cannot be meaningfully separated. As the CoT budget increases, these identical scores force an all-or-nothing routing decision, flatlining the trade-off curve (as seen in Figure 2 for Qwen3-8B and 32B). This score saturation fundamentally limits the granularity of confidence-gating, requiring successful signals to maintain a wide distributional spread.

Selection Bias of Hard Queries. We also observe that a reduction in CoT frequency does not strictly guarantee proportional token savings. Due to selection bias, confidence gating inherently identifies and routes the hardest, most token-heavy queries to CoT, inflating the average cost per call compared to random selection. For example, while confidence gating for GPT-5.1 matches the CoT use rate of a random baseline, its average token usage is markedly worse. The gap between these two curves is itself evidence of selection bias, since it shows the queries routed to CoT are systematically longer than a random subset. We confirm this with a Spearman correlation of -0.64 between confidence and CoT length on GPT-5.1, the strongest negative correlation across our models. For confidence gating to generate token savings, the reduction in CoT frequency must be sufficient to offset this increased cost per call, a threshold achieved by Qwen3-32B ($P(\text{True})$) and GPT-OSS-20B (Margin/Perplexity) (Figure 2), but not by GPT-5.1. These results highlight that lower CoT usage does not guarantee efficiency; if confidence and length are negatively correlated, longer CoT responses can offset the gains from skipping shorter ones. When evaluating confidence gating, token and latency savings should therefore be measured directly rather than inferred from CoT usage reduction alone.

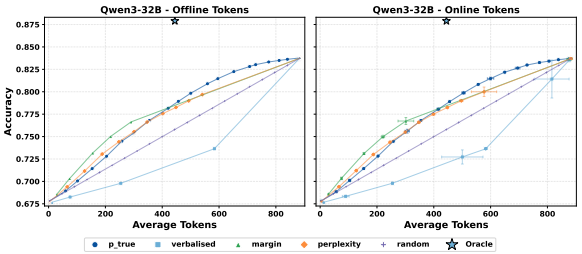


Figure 5: Online and offline budgeting for Qwen3-32B. Online performance broadly tracks the offline behaviour.

6.3 Performance in Practical Settings

To demonstrate practical deployment, we dynamically estimate thresholds using online budgeting, and apply our Pareto-optimal method to extract a single decision boundary that maximises savings without sacrificing accuracy.

Online CoT Budget–Accuracy Trade-offs Real-world systems process queries sequentially and cannot wait to observe a complete score distribution before making a routing decision. Our online setting simulates these live deployment constraints. Figure 5 demonstrates that for models with strong discriminative power (Qwen3-32B and GPT-OSS-20B), the dynamic online approach remains highly stable, closely tracking the offline ideal (full results in Appendix H). In contrast, the increased variability of *margin* and *perplexity* for Qwen3-8B at mid-to-high budgets highlights how signal saturation actively destabilizes threshold estimation. By showing that percentile thresholds can be estimated dynamically without a full confidence distribution, we confirm that budget-based gating is viable in a realistic deployment setting. This robustness also holds under distribution shift. When datasets are presented in a fixed order of increasing difficulty, the strongest signals continue to outperform random gating at every CoT budget (Appendix H).

Pareto-Optimal Thresholds We evaluate our Pareto-optimal threshold selection method from §4.1 using a combined 10% calibration split drawn uniformly across combined datasets with the error tolerance ϵ set to 1%. Appendix N reports a sensitivity analysis over $\epsilon \in \{0.5\%, 1\%, 1.5\%, 2\%\}$. To provide reliable estimates of performance variance, we report the mean and standard deviation across 100 Monte Carlo repeats (Xu and Liang, 2001). Table 1 summarises the results. For models with reliable confidence scores, our method successfully reduces token usage while sacrificing minimal accuracy. For Qwen3-32B, using $P(True)$ saves an average of 171 tokens per query, where a 26% reduction in CoT usage translates to a 23% reduction in latency. Meanwhile, GPT-OSS-20B saves roughly 65 tokens per query via *margin* or *perplexity* (reducing total CoT usage by roughly 30%), which translates to a latency reduction of up to 20%. Even with the weaker discriminative power of Qwen3-8B, the method secures a 10% usage reduction (roughly 87 tokens per query), translating to an 8% latency improvement. Average wall clock timing is also included in Appendix D. The method also acts as a safeguard when confidence signals fail. For GPT-5.1, where the model lacks the discriminative power to support token reduction, the

	Method	Acc. ↑	ΔAcc ↑	CoT (%) ↓	Avg Time ↓	Tok. saved ↑
GPT-OSS-20B	All CoT	79.9	0.0	100.0	7.7	0.0
	Pareto (ours)					
	P(True)	79.2 $_{0.5}$	-0.7	95.5 $_{2.3}$	7.4 $_{0.1}$	15.3 $_{8.9}$
	Verbalised	79.7 $_{0.1}$	-0.2	99.2 $_{0.0}$	7.6 $_{0.0}$	1.1 $_{3.1}$
	Margin	79.1 $_{0.4}$	-0.8	**68.1** $_{3.8}$	**6.1** $_{0.2}$	65.0 $_{12.1}$
	Perplexity	79.0 $_{0.5}$	-1.0	70.6 $_{7.9}$	6.5 $_{0.4}$	**65.7** $_{22.0}$
	CAR					
	P(True)	75.9 $_{0.8}$	-4.1	77.6 $_{7.8}$	6.3 $_{0.4}$	83.4 $_{32.3}$
	MLP Probe					
		77.2 $_{0.5}$	-2.7	83.0 $_{1.8}$	6.5 $_{0.1}$	70.9 $_{9.9}$
	Oracle	85.0	+5.1	45.9	4.3	187.2
Qwen3-32B	All CoT	83.8	0.0	100.0	46.3	0.0
	Pareto (ours)					
	P(True)	82.8 $_{0.5}$	-0.9	**73.8** $_{5.6}$	**35.5** $_{2.7}$	**170.9** $_{45.1}$
	Verbalised	83.7 $_{0.1}$	-0.1	98.9 $_{0.0}$	46.1 $_{0.2}$	3.8 $_{4.1}$
	Margin	83.8 $_{0.1}$	0.0	100.0 $_{0.0}$	46.3 $_{0.2}$	0.0 $_{4.1}$
	Perplexity	83.8 $_{0.1}$	0.0	100.0 $_{0.0}$	46.3 $_{0.2}$	0.0 $_{4.1}$
	CAR					
	P(True)	76.4 $_{0.4}$	-7.3	34.2 $_{1.4}$	16.2 $_{0.9}$	543.8 $_{13.4}$
	MLP Probe					
		79.1 $_{0.5}$	-4.6	44.9 $_{2.3}$	24.2 $_{1.7}$	401.2 $_{28.7}$
	Oracle	87.9	+4.1	32.2	20.9	446.7
Qwen3-8B	All CoT	79.1	0.0	100.0	23.5	0.0
	Pareto (ours)					
	P(True)	78.4 $_{0.5}$	-0.7	**90.8** $_{4.9}$	**21.7** $_{1.1}$	**86.6** $_{54.9}$
	Verbalised	79.0 $_{0.3}$	0.0	100.0 $_{0.5}$	23.4 $_{0.1}$	0.2 $_{6.2}$
	Margin	79.1 $_{0.2}$	0.0	100.0 $_{0.0}$	23.5 $_{0.1}$	0.0 $_{5.6}$
	Perplexity	79.1 $_{0.2}$	0.0	100.0 $_{0.0}$	23.5 $_{0.1}$	0.0 $_{5.6}$
	CAR					
	P(True)	76.9 $_{0.2}$	-2.2	82.2 $_{0.3}$	19.4 $_{0.1}$	209.2 $_{7.5}$
	MLP Probe					
		76.6 $_{0.5}$	-2.5	75.6 $_{1.6}$	20.6 $_{0.3}$	189.5 $_{15.2}$
	Oracle	83.8	+4.7	42.2	11.9	563.8
GPT-5.1	All CoT	86.6	0.0	100.0	-	0.0
	Pareto (ours)					
	Verbalised	86.1 $_{0.5}$	-0.4	97.0 $_{4.2}$	-	1.5 $_{10.2}$
	CAR					
	Verbalised	77.7 $_{0.2}$	-8.9	26.4 $_{0.9}$	-	320.6 $_{19.4}$
	Oracle	89.6	+3.1	26.9	-	528.6

Table 1: Performance with Pareto-optimal thresholds ($\epsilon = 1\%$) and highest-accuracy CAR result per model, across datasets. CoT usage, average time, and tokens saved per query. Full CAR results in Appendix L.

method yields negligible savings (averaging 1.5 tokens). Rather than forcing arbitrary CoT reductions that degrade performance, the thresholding safely defaults toward full CoT. This ensures that even when a signal cannot support efficiency gains, the method strictly preserves baseline accuracy.

OOD Calibration Set. To examine how well thresholds generalise, we extend our analysis beyond the mixed calibration setup. By using each individual dataset to calibrate thresholds for the others, we assess out-of-distribution (OOD) transfer across tasks (Appendix I). Challenging and diverse datasets, such as MMLU-Redux and GPQA, yield efficiency gains comparable to the combined split. For example, calibrating $P(\textit{True})$ on MMLU-Redux limits the accuracy drop to 0.8% across all other datasets while reducing CoT usage to 85%. Conversely, simpler datasets like CSQA consistently overestimate how much CoT can be removed, resulting in larger accuracy drops. Because CSQA questions are significantly easier, they produce relaxed thresholds that fail on more difficult tasks. Ultimately, calibration sets must reflect the difficulty and diversity of the target tasks for the Pareto method to transfer effectively. In practice, a broad, moderately difficult held-out set is sufficient for reliable threshold transfer.

Comparison to Routing Baselines. We compare our approach against two baselines. First, to benchmark against prior confidence-based routing, we evaluate CAR (Lu et al., 2025), which fits a Gaussian to perplexity scores to derive thresholds. We apply this to all the confidence signals. Second, because our method relies on a calibration set, we compare against a minimally supervised alternative, an MLP probe (Azaria and Mitchell, 2023; Stacey et al., 2026) trained on middle-layer hidden states to predict correctness of direct answers. Both baselines use the same 10% calibration split setting as our method. For CAR, we apply the Gaussian fit across all four signals. For the probe, we retrain 50 times with randomised samples to estimate variance, full training details can be found in Appendix G. As shown in Table 1, while both baselines reduce token cost, they suffer substantial accuracy loss. On Qwen3-32B, CAR's best signal loses 7.3% accuracy. The probe performs better but it still loses between 2.5% and 4.6% across models. In contrast, our Pareto-optimal method strictly limits accuracy degradation to 1%, ensuring efficiency gains without these performance tradeoffs.

Overall, practical claims about confidence-gated CoT should be understood conditionally. Efficiency gains are available when the chosen model and signal combination shows strong discriminative power (§6.2) and on tasks where high accuracy is possible with a direct answer (§6.1). For other combinations, such as Qwen3-8B or GPT-5.1, the value of our approach lies in the safe default of the Pareto method, which preserves full-CoT accuracy when the signal cannot support savings.

7 Qualitative Analysis

To understand where our Pareto-optimal thresholds succeed and where they leave efficiency on the table, we classify every routing decision into one of five outcomes: **CoT Fixed** (CoT corrects a wrong direct answer), **Direct** (tokens saved by correctly skipping CoT), **Excess CoT** (redundant reasoning for an already correct direct answer), **Missed Fix** (wrong direct answer, CoT skipped), and **Both Fail**. Table 2 shows the distribution. Because routing requires strong discriminative power, the *Direct* outcome closely tracks AUROC, rising from 7.1% on Qwen3-8B to 23.5% on Qwen3-32B and 27.2% on GPT-OSS-20B. Our method is safe, holding *Missed Fixes* near 1% while *CoT Fixed* correctly triggers on 19–30% of queries. However, *Excess CoT* exposes a major bottleneck, forming the largest share for most models. In these cases, the direct answer is correct, but weak confidence prevents the model from clearing the ϵ threshold. Example 1 shows a successful *Direct* bypass, while Example 2 illustrates *Excess CoT* triggered by low confidence ($P(\text{True}) = 0.59$). Ultimately, further efficiency is limited by the models' tendency to under-report confidence on correct answers.

Example 1

Question: Would a Nike shoebox be too small to fit a swan in?
Direct answer: *Yes* ($P(\text{True}) = 0.99$).
Ground Truth: *Yes.*

Example 1: *Direct*. This represents a positive case where accuracy is preserved and tokens are saved by directly answering. Confidence-gated CoT saves 284 tokens by choosing to skip CoT.

Example 2: *Excess CoT*. In this example, the direct answer was already correct, but the policy still used CoT, leading to redundant tokens.
These examples highlight both the effectiveness and the limitations of confidence-gated CoT. The

Example 2

Question: Where would you put a glass after drinking from it?

Answer choices: (A) ocean, (B) water cooler, (C) cabinet, (D) dishwasher, (E) dining room.

Direct answer: *(D)* ($P(\text{True}) = 0.59$).

CoT reasoning (excerpt): "Option A doesn't make sense… Option D, dishwasher, is correct. Therefore, the answer is D."

Category	Qwen8B	Qwen32B	GPT-20B	GPT-5.1
CoT Fixed	24.7 $_{0.8}$	18.8 $_{0.6}$	29.9 $_{0.6}$	16.0 $_{0.5}$
Direct	7.1 $_{3.6}$	23.5 $_{4.6}$	27.2 $_{2.5}$	2.4 $_{3.5}$
Excess CoT	50.8 $_{3.6}$	44.4 $_{4.6}$	26.6 $_{2.5}$	70.7 $_{3.5}$
Missed Fix	1.2 $_{0.8}$	1.3 $_{0.6}$	1.3 $_{0.6}$	0.5 $_{0.5}$
Both fail	16.2 $_{0.1}$	12.1 $_{0.1}$	15.0 $_{0.1}$	10.4 $_{0.1}$

Table 2: Distribution of outcome types. Values denote %, averaged over calibration runs (std dev in subscript).

substantial portion of *CoT Fixed* cases, combined with the minimal rate of *Missed Fix* (approx. 1%), confirms that confidence signals effectively identify necessary reasoning. However, the frequency of *Excess CoT* shows that models often lack the confidence to skip reasoning even when correct. This analysis highlights that while current signals can be effective, further gains can be unlocked by better distinguishing correct from incorrect answers.

8 Conclusion

We propose an evaluation framework for confidence-gated CoT and conduct the first systematic study of training-free confidence signals for efficient reasoning. While the effectiveness of confidence-gated routing varies across models and signals, we demonstrate how to systematically diagnose and manage this instability. We identify signal saturation and selection bias as primary failure modes, establishing AUROC as a better predictor of gating success than calibration error. Providing a practical solution, we introduce a Pareto-optimal thresholding method that outperforms the baselines evaluated, CAR and a supervised probe. This approach preserves accuracy while driving end-to-end latency reductions for strong signals, and safely maintains baseline performance for weak ones. Ultimately, our study provides a foundation for evaluating and improving confidence signals for efficient reasoning.

Limitations

We employ a standard prompting strategy for verbalised confidence consistent with prior work (Yang et al., 2024), future work could extend this analysis to various prompting strategies. Additionally, future work could explore confidence estimation methods that require minimal sampling (Zhou et al., 2025; Kuhn et al., 2023) or models that are trained to produce better verbalised confidence estimates (Stangel et al., 2025; Zhang et al., 2025b), both of which can be studied within our framework. Comparisons to fully trained when-to-think models (Jiang et al., 2025; Chuang et al., 2025a) and adaptations of conformal risk control (Wang et al., 2026) are left to future work; these methods require model training or target a different reasoning-control setting, and our claims of outperformance are relative to the baselines evaluated here. Our Pareto threshold selection also requires a labelled calibration set with both direct and CoT outputs, so it is not directly applicable where ground-truth labels are unavailable.

Ethical Consideration

Language models can generate content that is harmful (Weidinger et al., 2022). Our contribution focuses on efficiency without training and as such will not affect the existing risks present in each model. All datasets are MIT-licensed, apart from GPQA, which is released under a CC-BY 4.0 license. We use these datasets to evaluate NLP models, which is in line with their intended purpose. To the best of our knowledge there is no PII or offensive content in these datasets.

References

Pranjal Aggarwal, Seungone Kim, Jack Lanchantin, Sean Welleck, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. 2025. Optimalthinkingbench: Evaluating over and underthinking in llms. *Preprint*, arXiv:2508.13141.

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

Yu-Neng Chuang, Prathusha Kameswara Sarma, Parikshit Gopalan, John Boccio, Sara Bolouki, Xia Hu, and Helen Zhou. 2025a. Learning to route LLMs with confidence tokens. In *Forty-second International Conference on Machine Learning*.

Yu-Neng Chuang, Leisheng Yu, Guanchu Wang, Lizhe Zhang, Zirui Liu, Xuanting Cai, Yang Sui, Vladimir Braverman, and Xia Hu. 2025b. Confident or seek stronger: Exploring uncertainty-based on-device llm routing from benchmarking to generalization. *Preprint*, arXiv:2502.04428.

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

Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. 2025. Beyond binary rewards: Training lms to reason about their uncertainty. *Preprint*, arXiv:2507.16806.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Tao Feng, Yanzhen Shen, and Jiaxuan You. 2025. Graphrouter: A graph-based router for LLM selections. In *The Thirteenth International Conference on Learning Representations*.

Aryo Pradipta Gema and 1 others. 2025. Are we done with MMLU? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5069–5096, Albuquerque, New Mexico. Association for Computational Linguistics.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1321–1330. JMLR.org.

Daya Guo and 1 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Chengyu Huang, Zhengxin Zhang, and Claire Cardie. 2025a. Hapo: Training language models to reason concisely via history-aware policy optimization. *Preprint*, arXiv:2505.11225.

Yao Huang, Huanran Chen, Shouwei Ruan, Yichi Zhang, Xingxing Wei, and Yinpeng Dong. 2025b. Mitigating overthinking in large reasoning models via manifold steering. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Lingjie Jiang, Xun Wu, Shaohan Huang, Qingxiu Dong, Zewen Chi, Li Dong, Xingxing Zhang, Tengchao Lv, Lei Cui, and Furu Wei. 2025. Think only when you need with large hybrid-reasoning models. *Preprint*, arXiv:2505.14631.

Zhengping Jiang, Anqi Liu, and Benjamin Van Durme. 2024. Addressing the binning problem in calibration assessment through scalar annotations. *Transactions of the Association for Computational Linguistics*, 12:120–136.

Saurav Kadavath and 1 others. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth A Malik, and Yarin Gal. 2025. Semantic entropy probes: Robust and cheap hallucination detection in LLMs.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Ryan Liu, Jiayi Geng, Addison J. Wu, Ilia Sucholutsky, Tania Lombrozo, and Thomas L. Griffiths. 2024. Mind your step (by step): Chain-of-thought can reduce performance on tasks where thinking makes humans worse. *Preprint*, arXiv:2410.21333.

Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. 2025. Learn to reason efficiently with adaptive length-based reward shaping. *Preprint*, arXiv:2505.15612.

Jinghui Lu and 1 others. 2025. Prolonged reasoning is not all you need: Certainty-based adaptive routing for efficient llm/mllm reasoning. *Preprint*, arXiv:2505.15154.

Team Olmo, :, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, and 50 others. 2025. Olmo 3. *Preprint*, arXiv:2512.13961.

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

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

OpenAI. 2025. Introducing gpt-5.1 for developers. Accessed: 2025-12-24.

Maja Pavlovic. 2025. Understanding model calibration - a gentle introduction and visual exploration of calibration and the expected calibration error (ece). In *ICLR Blogposts 2025*. Https://iclr-blogposts.github.io/2025/blog/calibration/.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Guillem Ramirez, Alexandra Birch, and Ivan Titov. 2024. Optimising calls to large language models with uncertainty-based two-tier selection. In *Proceedings of the 2024 Conference on Language Modeling*. Conference on Language Modeling, COLM 2024 ; Conference date: 07-10-2024 Through 09-10-2024.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, Zhaoxiang Liu, and Shiguo Lian. 2025. DAST: Difficulty-adaptive slow-thinking for large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 2322–2331, Suzhou (China). Association for Computational Linguistics.

Zayne Rea Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. 2024. MuSR: Testing the limits of chain-of-thought with multistep soft reasoning. In *The Twelfth International Conference on Learning Representations*.

Zayne Rea Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2025. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. In *The Thirteenth International Conference on Learning Representations*.

Joe Stacey, Hadas Orgad, Kentaro Inui, Benjamin Heinzerling, and Nafise Sadat Moosavi. 2026. Hidden failures in robustness: Why supervised uncertainty quantification needs better evaluation. *Preprint*, arXiv:2604.11662.

Paul Stangel, David Bani-Harouni, Chantal Pellegrini, Ege Özsoy, Kamilia Zaripova, Matthias Keicher, and Nassir Navab. 2025. Rewarding doubt: A reinforcement learning approach to calibrated confidence expression of large language models. *Preprint*, arXiv:2503.02623.

Alon Talmor, Jonathan Herzig, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.

Xi Wang, Anushri Suresh, Alvin Zhang, Rishi More, William Jurayj, Benjamin Van Durme, Mehrdad Farajtabar, Daniel Khashabi, and Eric Nalisnick. 2026. Conformal thinking: Risk control for reasoning on a compute budget. *Preprint*, arXiv:2602.03814.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, Courtney Biles, Sasha Brown, Zac Kenton, Will Hawkins, Tom Stepleton, Abeba Birhane, Lisa Anne Hendricks, Laura Rimell, William Isaac, and 4 others. 2022. Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 214–229, New York, NY, USA. Association for Computing Machinery.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*.

Qing-Song Xu and Yi-Zeng Liang. 2001. Monte carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, 56(1):1–11.

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

Daniel Yang, Yao-Hung Hubert Tsai, and Makoto Yamada. 2024. On verbalized confidence scores for llms. *Preprint*, arXiv:2412.14737.

Linan Yue, Yichao Du, Yizhi Wang, Weibo Gao, Fangzhou Yao, Li Wang, Ye Liu, Ziyu Xu, Qi Liu, Shimin Di, and Min-Ling Zhang. 2025. Don't overthink it: A survey of efficient r1-style large reasoning models. *Preprint*, arXiv:2508.02120.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. 2025a. Reasoning models know when they're right: Probing hidden states for self-verification. In *Second Conference on Language Modeling*.

Caiqi Zhang, Xiaochen Zhu, Chengzu Li, Nigel Collier, and Andreas Vlachos. 2025b. Reinforcement learning for better verbalized confidence in long-form generation. *Preprint*, arXiv:2505.23912.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2024. AGIEval: A human-centric benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2299–2314, Mexico City, Mexico. Association for Computational Linguistics.

Ziang Zhou, Tianyuan Jin, Jieming Shi, and Li Qing. 2025. Steerconf: Steering LLMs for confidence elicitation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

A LLM Assistant Use

The writing of this paper received proofreading and language polishing suggestions using LLMs. In addition, parts of our experimental code were drafted or refactored with the assistance of GitHub Copilot; all final text and code was manually reviewed and verified by the authors.

B Practical Guidance

Confidence gating is most effective when direct answering already achieves high accuracy, e.g., commonsense and knowledge-based reasoning. For tasks where direct accuracy is low, e.g., mathematical reasoning, gating provides limited benefit as most queries genuinely require CoT. The best signal is model family dependent: margin and perplexity are effective for GPT-OSS-20B, while P(True) is best for Qwen3-32B, so it is important to test a range of confidence methods for new models. When adopting a new method or model, practitioners should prioritise AUROC over ECE as a predictor of gating success, check for score saturation that limits separating power, and verify token savings directly rather than inferring them from CoT usage reduction due to selection bias. These insights apply to any confidence signal, whether training-free or learned. Finally, our Pareto-optimal selection procedure identifies reliable thresholds that preserve accuracy while reducing cost. This requires only a small labelled calibration set that is broadly representative of the target tasks; as shown in §6.3, a broad, moderately difficult held-out set is sufficient for reliable threshold transfer.

C Model Inference Settings

For GPT-5.1 we use greedy decoding for direct answers and the default temperature of 1.0 when using the high reasoning effort. We use Hugging Face Transformers for inference on Qwen and GPT-OSS models. For Qwen models (8B and 32B), we follow the recommended decoding settings from the model cards, using a temperature of 0.6 and top-p of 0.95 to avoid degenerate repetition. For GPT-OSS-20B, we use the default sampling configuration with temperature 1.0 and top-p 1.0. In all settings, we set a maximum limit of 7000 thinking tokens and insert text that prompts the model to answer after this limit has been reached.

D Timing Results

Dataset	CoT Low	CoT Med	CoT High	Dir. Margin	Dir. P(T)	Dir. Verb.
csqa	1.6	4.0	4.7	0.0	0.1	0.1
gpqa	26.1	47.3	55.5	0.2	0.3	0.2
gsm8k	1.5	3.9	6.0	0.1	0.1	0.1
lsat	11.0	15.8	18.5	0.2	0.4	0.2
mmlu-redux	2.8	6.5	8.6	0.1	0.2	0.1
musr	4.8	8.0	10.4	0.5	1.0	0.6
strategyqa	1.0	1.9	2.7	0.0	0.0	0.1
ALL	$6.9_{\pm 9.1}$	$12.5_{\pm 16.0}$	$15.2_{\pm 18.5}$	$0.1_{\pm 0.2}$	$0.3_{\pm 0.3}$	$0.2_{\pm 0.2}$

Table 3: Mean inference time (seconds/question) for GPT-OSS-20B across CoT budget levels (Low/Med/High) and direct methods. *ALL* reports macro mean ± std across datasets.

	Dataset	CoT	Dir. Margin	Dir. P(T)	Dir. Verb.
Qwen3-32B	csqa	13.3	0.1	0.1	0.2
	gpqa	247.7	0.2	0.4	0.3
	gsm8k	28.1	0.1	0.2	0.3
	lsat	77.7	0.2	0.4	0.3
	mmlu-redux	49.9	0.1	0.3	0.3
	musr	59.7	0.4	0.8	0.6
	strategyqa	12.1	0.1	0.1	0.2
	ALL	$69.8_{\pm 82.1}$	$0.2_{\pm 0.1}$	$0.3_{\pm 0.2}$	$0.3_{\pm 0.1}$
Qwen3-8B	csqa	6.4	0.0	0.0	0.0
	gpqa	67.3	0.1	0.1	0.1
	gsm8k	29.8	0.1	0.0	0.0
	lsat	34.8	0.1	0.1	0.1
	mmlu-redux	40.2	0.0	0.1	0.0
	musr	23.0	0.1	0.2	0.1
	strategyqa	5.5	0.0	0.0	0.0
	ALL	$29.6_{\pm 21.3}$	$0.0_{\pm 0.0}$	$0.1_{\pm 0.1}$	$0.1_{\pm 0.0}$

Table 4: Mean inference time (seconds/question) for Qwen3-32B and Qwen3-8B. *ALL* reports macro mean ± std across datasets.

E Dataset Statistics

Dataset	# Samples
CommonsenseQA (CSQA)	1221
StrategyQA	2290
MMLU-Redux	3000
GSM8K	1319
GPQA	448
LSAT-AGI	1009
MUSR	756

Table 5: Dataset statistics.

F Prompts

Verbalised Prompt

Please directly provide your best guess of the answer to the question and give the probability that you think it is correct (0.0 to 1.0). Take your uncertainty in the prompt, the task difficulty, your knowledge availability, and other sources of uncertainty into account. Give only the guess and probability, with no other words or explanation.

Format your final response as:
Answer: <your_best_guess>.
Probability: <score between 0.0 and 1.0>

P(True) Prompt

User:
Is this answer:
(A) True
(B) False

Assistant:
The answer is:

G Supervised MLP Probe Training Details

Activation extraction. For each question we run a single forward pass with a direct-answer prompt (chain-of-thought disabled) which ends with `Answer:(`. We extract the residual-stream hidden state at the final token position from the middle transformer layer ($\ell = \lfloor L/2 \rfloor$) on the first token of the answer.

Architecture and training. We train a four-layer MLP ($d \to 256 \to 128 \to 64 \to 1$, ReLU activations) on standardised activations to predict binary answer correctness, optimised with binary cross-entropy and Adam (lr $= 10^{-3}$, batch size 64, 200 epochs). All hyperparameters are held constant across models. We retrain 50 times with independently randomised train/test splits, each using 10% of data for training to match the calibration budget of other methods (Table 1). Reported figures are the mean and standard deviation across splits.

H Online Budget Results

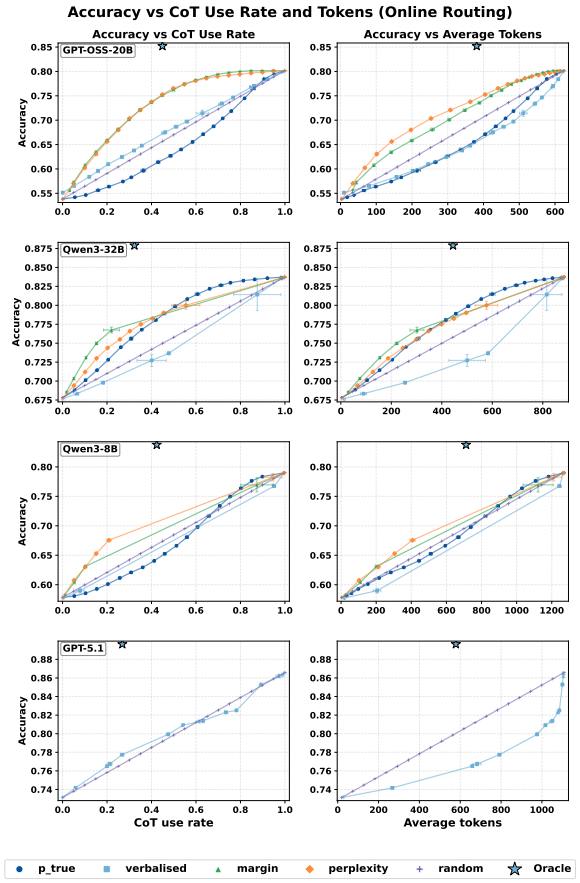


Figure 6: Online Accuracy vs. CoT use rate (left) and average tokens (right) across all datasets in the **online** setting. Stars show Oracle performance.

Difficulty-Ordered Distribution Shift. The main online results randomise dataset order. To test robustness to non-stationary traffic, we re-run the online simulation with datasets presented in a fixed order of increasing difficulty: CommonsenseQA → StrategyQA → GSM8K → MuSR → LSAT → MMLU-Redux → GPQA. This is a worst case for the running-percentile estimator, since early high-confidence queries could set thresholds that route too few later, harder queries to CoT. Figure 7 shows that performance is not meaningfully degraded. For Qwen3-32B, $P(\text{True})$ and perplexity remain the strongest signals and outperform random gating at every CoT budget (e.g. ∼79% and ∼80% accuracy respectively at a 50% CoT budget), consistent with the randomised setting. For GPT-OSS-20B, the difficulty-ordered online curve is nearly indistinguishable from the offline curve, with the same signal ranking (margin and perplexity strongest, $P(\text{True})$ weakest).

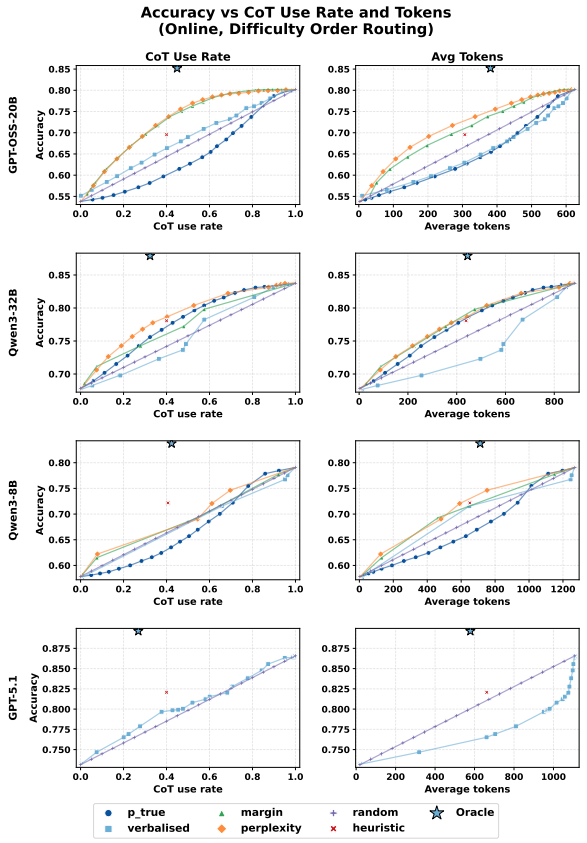


Figure 7: Online accuracy–efficiency trade-offs under a difficulty-ordered query stream for Qwen3-8B, Qwen3-32B, GPT-OSS-20B and GPT-5.1.

I OOD Pareto Results

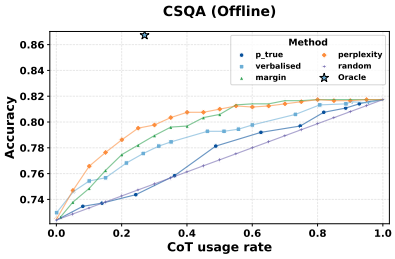
	Method	Acc. ↑	ΔAcc ↑	CoT (%) ↓	Avg. Tok. saved ↑
CSQA	All CoT	83.4	0.0	100.0	0.0
	All Direct	65.5	-17.8	0.0	943.1
	P(True)	70.4	-13.0	19.5	746.7
	Verbalised	65.2	-18.2	0.0	935.3
	Margin	83.4	0.0	100.0	0.0
	Perplexity	73.3	-10.1	22.2	658.0
GPQA	All CoT	84.9	0.0	100.0	0.0
	All Direct	69.1	-15.8	0.0	749.5
	P(True)	84.4	-0.4	81.2	105.8
	Verbalised	84.8	-0.1	98.8	4.0
	Margin	84.9	0.0	100.0	0.0
	Perplexity	84.9	0.0	100.0	0.0
GSM8K	All CoT	82.1	0.0	100.0	0.0
	All Direct	72.9	-9.2	0.0	919.4
	P(True)	82.1	0.0	98.2	9.5
	Verbalised	82.1	0.0	98.9	3.1
	Margin	82.1	0.0	100.0	0.0
	Perplexity	82.1	0.0	100.0	0.0
LSAT	All CoT	83.1	0.0	100.0	0.0
	All Direct	67.1	-16.0	0.0	811.1
	P(True)	83.1	0.0	98.5	6.5
	Verbalised	83.0	-0.1	98.7	4.3
	Margin	83.1	0.0	100.0	0.0
	Perplexity	83.1	0.0	100.0	0.0
MMLU-REDUX	All CoT	83.0	0.0	100.0	0.0
	All Direct	63.8	-19.2	0.0	907.4
	P(True)	82.2	-0.8	84.8	116.9
	Verbalised	82.8	-0.1	98.5	5.0
	Margin	83.0	0.0	100.0	0.0
	Perplexity	83.0	0.0	100.0	0.0
MUSR	All CoT	85.2	0.0	100.0	0.0
	All Direct	68.9	-16.4	0.0	859.2
	P(True)	80.9	-4.3	47.0	407.4
	Verbalised	74.4	-10.9	45.7	306.5
	Margin	85.2	0.0	100.0	0.0
	Perplexity	85.2	0.0	100.0	0.0
STRATEGYQA	All CoT	84.3	0.0	100.0	0.0
	All Direct	66.4	-17.9	0.0	1014.3
	P(True)	78.3	-6.0	43.8	499.0
	Verbalised	84.2	-0.1	99.6	1.9
	Margin	84.3	0.0	100.0	0.0
	Perplexity	84.3	0.0	100.0	0.0

Table 6: Results for Qwen3-32B by dataset with Pareto-optimal thresholds ($\epsilon = 1\%$).

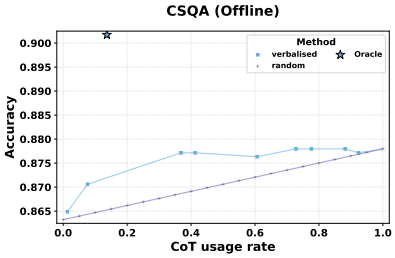
	Method	Acc. ↑	ΔAcc ↑	CoT (%) ↓	Avg. Tok. saved ↑
CSQA	All CoT	79.7	0.0	100.0	0.0
	All Direct	51.2	-28.5	0.0	514.0
	P(True)	68.7	-11.0	70.9	135.3
	Verbalised	71.9	-7.7	70.8	59.8
	Margin	78.8	-0.9	68.7	68.6
	Perplexity	76.9	-2.8	54.7	126.0
GPQA	All CoT	80.6	0.0	100.0	0.0
	All Direct	54.8	-25.9	0.0	408.3
	P(True)	80.6	-0.1	99.1	2.2
	Verbalised	80.4	-0.3	99.2	1.2
	Margin	80.1	-0.6	68.9	60.1
	Perplexity	77.5	-3.2	48.4	140.6
GSM8K	All CoT	77.8	0.0	100.0	0.0
	All Direct	60.0	-17.8	0.0	515.9
	P(True)	77.6	-0.1	98.7	3.0
	Verbalised	77.8	-0.0	100.0	0.0
	Margin	75.8	-2.0	53.6	108.1
	Perplexity	74.7	-3.0	45.6	149.5
LSAT	All CoT	79.8	0.0	100.0	0.0
	All Direct	54.0	-25.8	0.0	417.4
	P(True)	77.4	-2.4	89.2	40.2
	Verbalised	72.2	-7.6	68.6	61.3
	Margin	79.3	-0.5	69.8	54.6
	Perplexity	78.7	-1.1	66.6	69.3
MMLU-REDUX	All CoT	78.2	0.0	100.0	0.0
	All Direct	49.8	-28.4	0.0	519.2
	P(True)	78.2	0.0	100.0	0.0
	Verbalised	73.7	-4.5	84.3	25.4
	Margin	77.9	-0.3	77.2	45.1
	Perplexity	77.0	-1.2	67.8	72.5
MUSR	All CoT	81.4	0.0	100.0	0.0
	All Direct	54.1	-27.3	0.0	480.0
	P(True)	75.5	-5.9	82.0	79.8
	Verbalised	67.8	-13.6	44.1	150.3
	Margin	78.6	-2.8	53.4	111.5
	Perplexity	78.8	-2.6	53.3	116.5
STRATEGYQA	All CoT	81.7	0.0	100.0	0.0
	All Direct	51.6	-30.1	0.0	567.9
	P(True)	81.2	-0.5	98.2	8.0
	Verbalised	79.7	-2.0	94.3	7.6
	Margin	81.2	-0.5	71.1	61.1
	Perplexity	81.7	-0.1	86.4	31.0

Table 7: Results for GPT-OSS-20B by dataset with Pareto-optimal thresholds ($\epsilon = 1\%$).

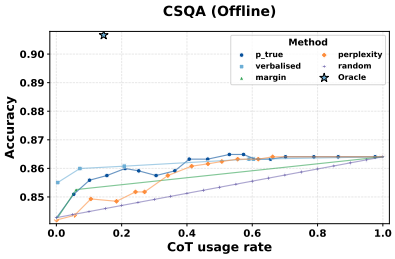
J Per Dataset Trade-off Plots



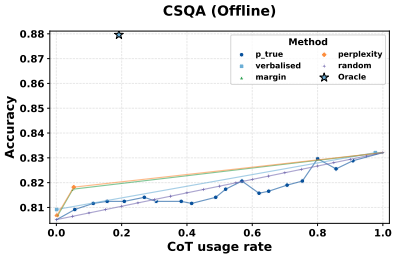
(a) GPT-OSS Medium - CoT Use



(b) GPT-5 - CoT Use

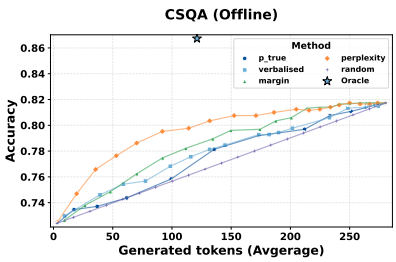


(c) Qwen3-32B - CoT Use

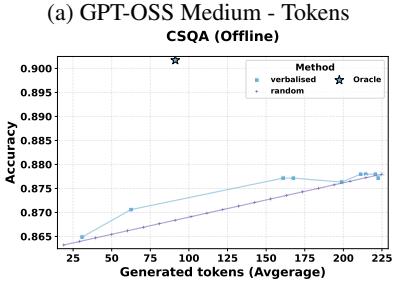


(d) Qwen3-8B - CoT Use

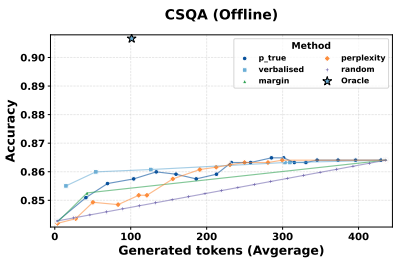
Figure 8: **CSQA (Part 1): Accuracy vs. CoT Use.**



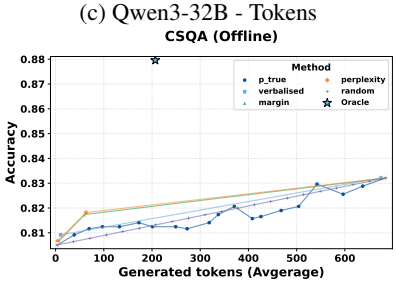
(a) GPT-OSS Medium - Tokens



(b) GPT-5 - Tokens

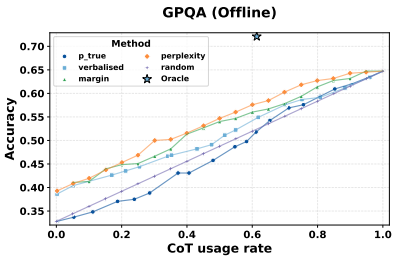


(c) Qwen3-32B - Tokens

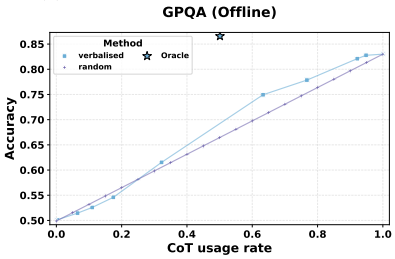


(d) Qwen3-8B - Tokens

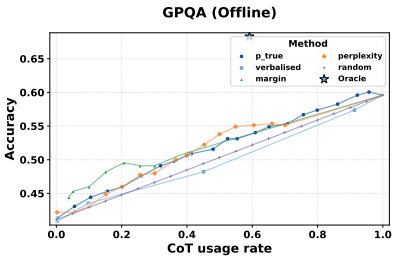
Figure 9: **CSQA (Part 2): Average Tokens vs. Accuracy.**



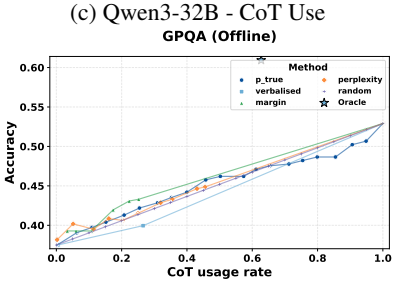
(a) GPT-OSS Medium - CoT Use



(b) GPT-5 - CoT Use

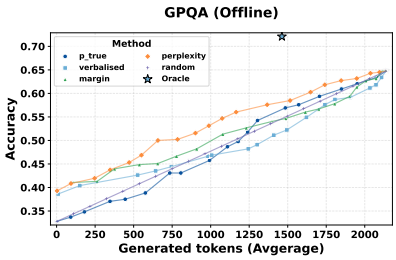


(c) Qwen3-32B - CoT Use

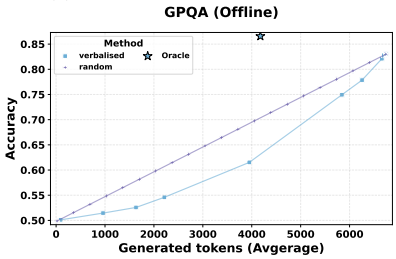


(d) Qwen3-8B - CoT Use

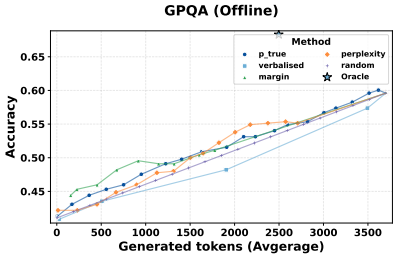
Figure 10: **GPQA (Part 1): Accuracy vs. CoT Use.**



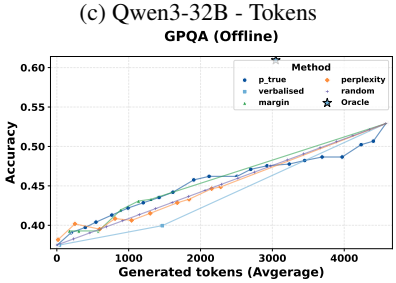
(a) GPT-OSS Medium - Tokens



(b) GPT-5 - Tokens

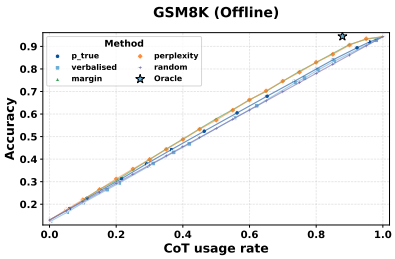


(c) Qwen3-32B - Tokens

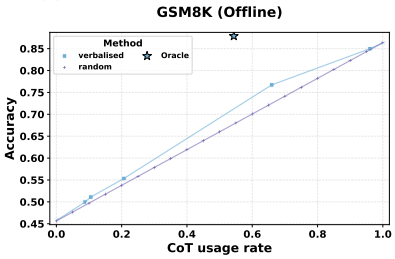


(d) Qwen3-8B - Tokens

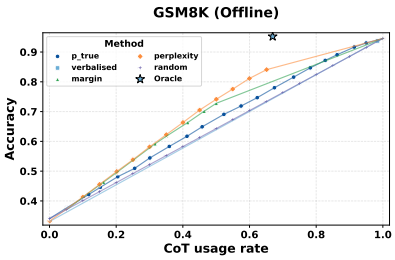
Figure 11: **GPQA (Part 2): Average Tokens vs. Accuracy.**



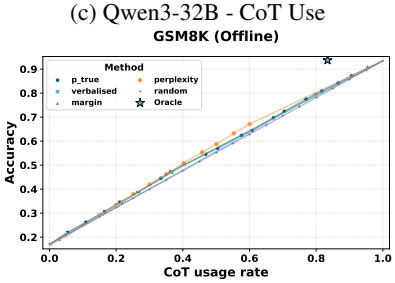
(a) GPT-OSS Medium - CoT Use



(b) GPT-5 - CoT Use

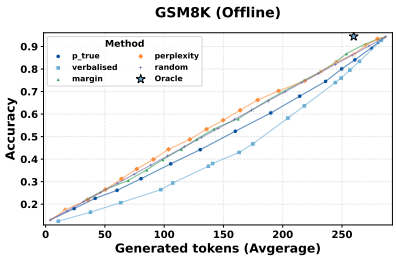


(c) Qwen3-32B - CoT Use

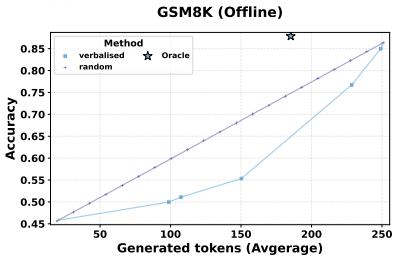


(d) Qwen3-8B - CoT Use

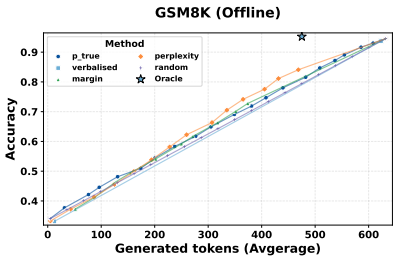
Figure 12: **GSM8K (Part 1): Accuracy vs. CoT Use.**



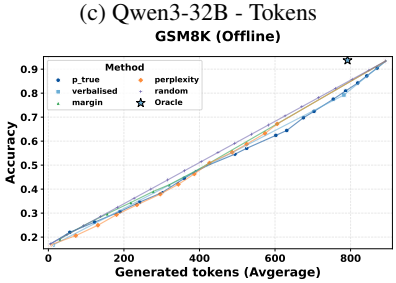
(a) GPT-OSS Medium - Tokens



(b) GPT-5 - Tokens

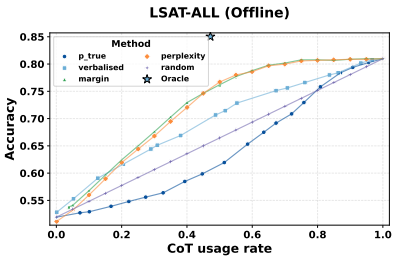


(c) Qwen3-32B - Tokens

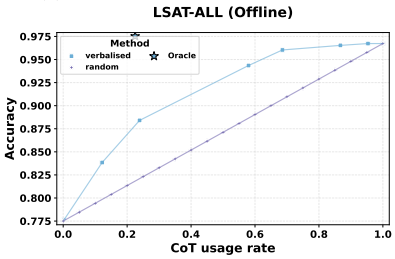


(d) Qwen3-8B - Tokens

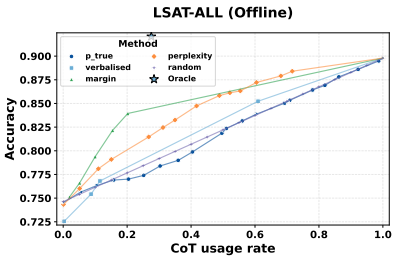
Figure 13: **GSM8K (Part 2): Average Tokens vs. Accuracy.**



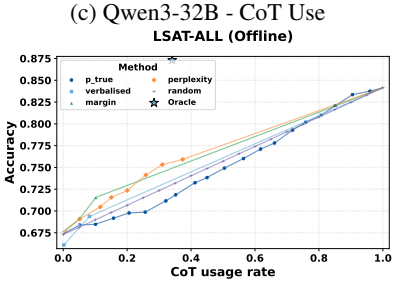
(a) GPT-OSS Medium - CoT Use



(b) GPT-5 - CoT Use

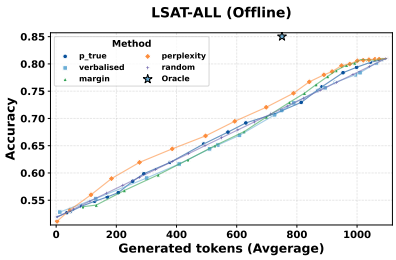


(c) Qwen3-32B - CoT Use

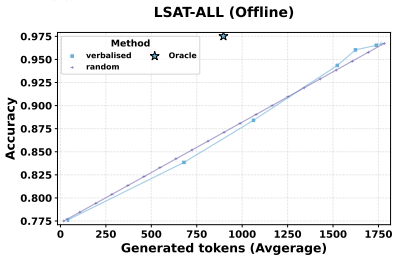


(d) Qwen3-8B - CoT Use

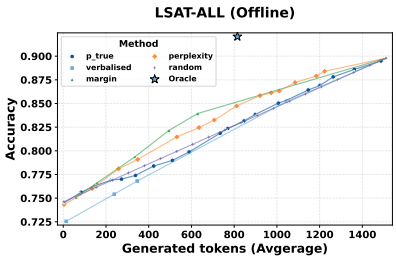
Figure 14: **LSAT-All (Part 1): Accuracy vs. CoT Use.**



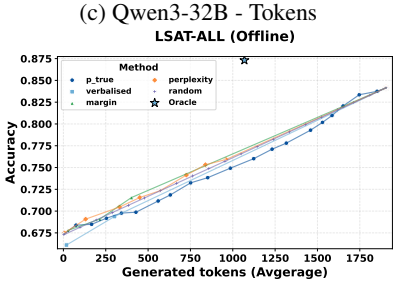
(a) GPT-OSS Medium - Tokens



(b) GPT-5 - Tokens

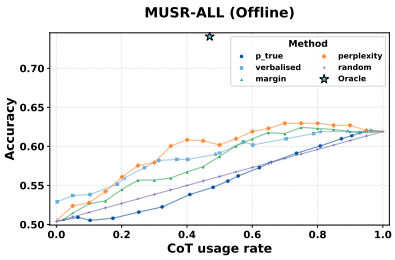


(c) Qwen3-32B - Tokens

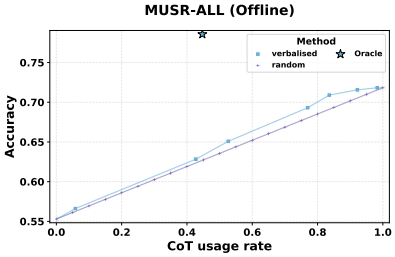


(d) Qwen3-8B - Tokens

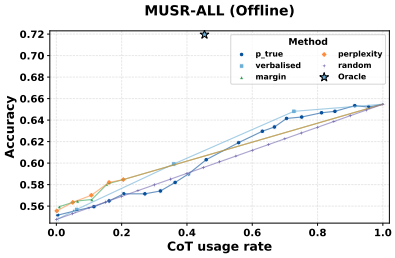
Figure 15: **LSAT-All (Part 2): Average Tokens vs. Accuracy.**



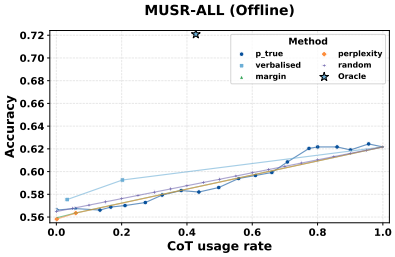
(a) GPT-OSS Medium - CoT Use



(b) GPT-5 - CoT Use

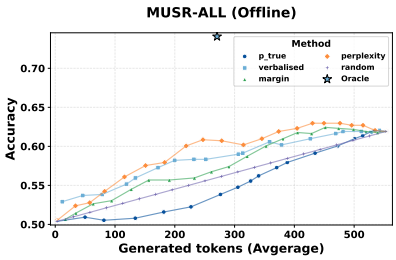


(c) Qwen3-32B - CoT Use

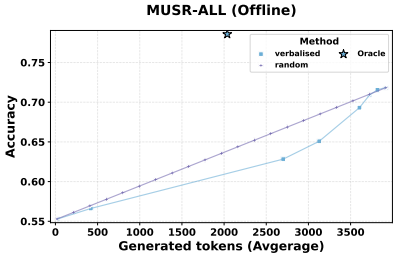


(d) Qwen3-8B - CoT Use

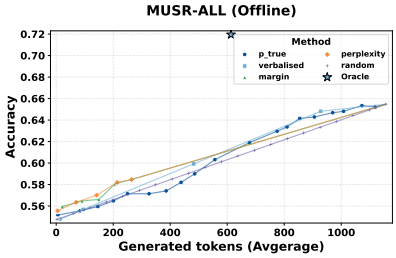
Figure 16: **MuSR-All (Part 1): Accuracy vs. CoT Use.**



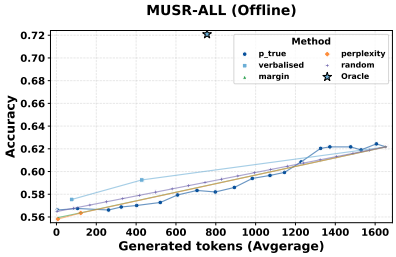
(a) GPT-OSS Medium - Tokens



(b) GPT-5 - Tokens

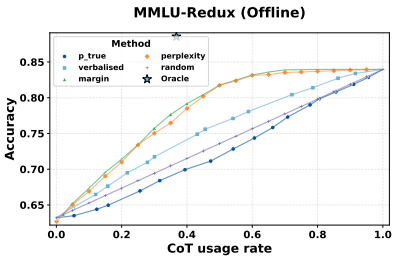


(c) Qwen3-32B - Tokens

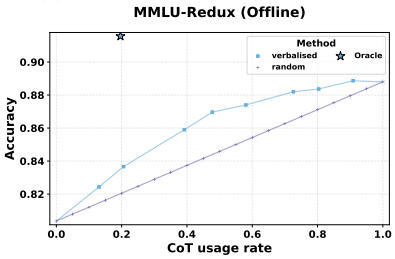


(d) Qwen3-8B - Tokens

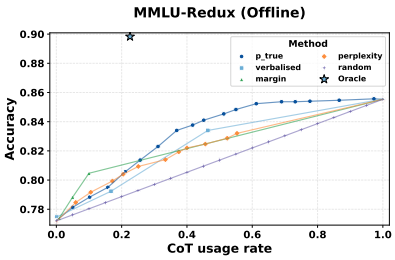
Figure 17: **MuSR-All (Part 2): Average Tokens vs. Accuracy.**



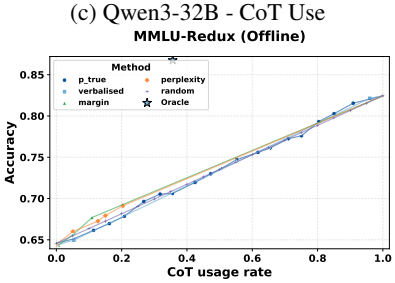
(a) GPT-OSS Medium - CoT Use



(b) GPT-5 - CoT Use

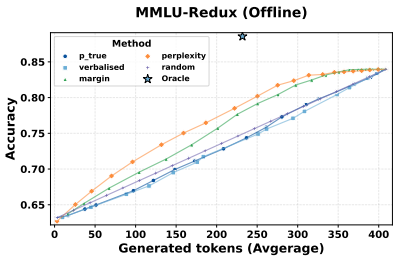


(c) Qwen3-32B - CoT Use

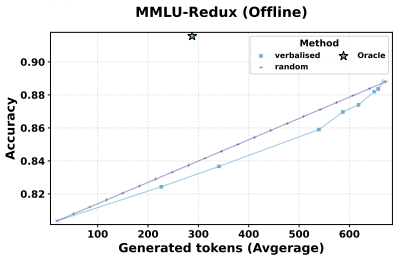


(d) Qwen3-8B - CoT Use

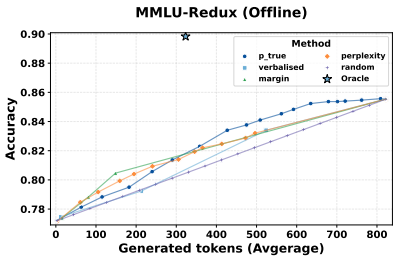
Figure 18: **MMLU-Redux (Part 1): Accuracy vs. CoT Use.**



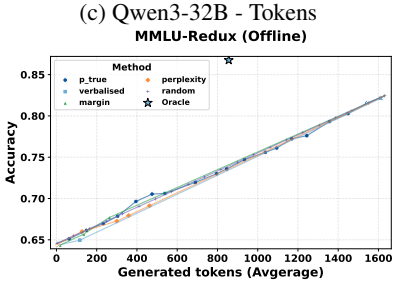
(a) GPT-OSS Medium - Tokens



(b) GPT-5 - Tokens

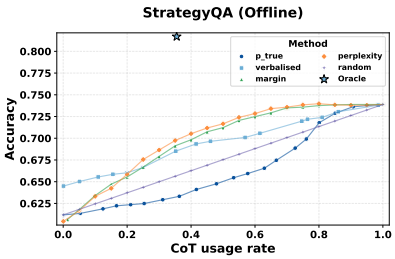


(c) Qwen3-32B - Tokens

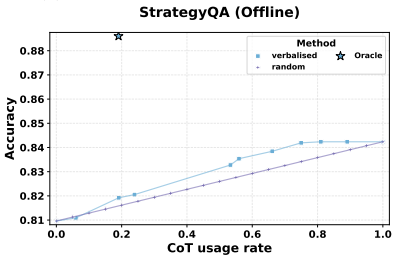


(d) Qwen3-8B - Tokens

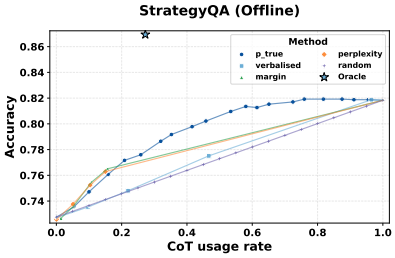
Figure 19: **MMLU-Redux (Part 2): Average Tokens vs. Accuracy.**



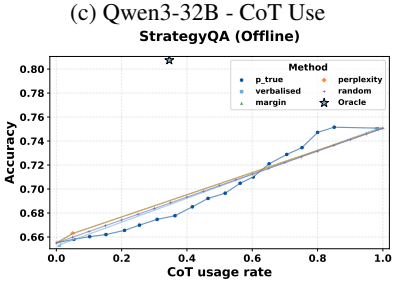
(a) GPT-OSS Medium - CoT Use



(b) GPT-5 - CoT Use

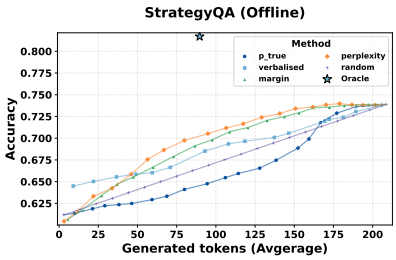


(c) Qwen3-32B - CoT Use

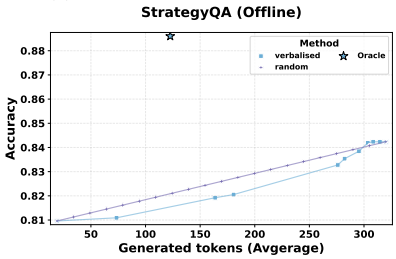


(d) Qwen3-8B - CoT Use

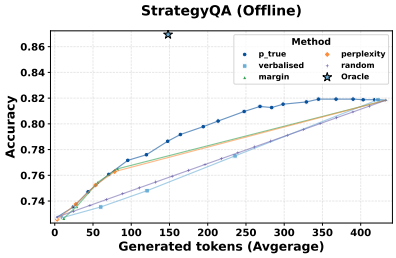
Figure 20: **StrategyQA (Part 1): Accuracy vs. CoT Use.**



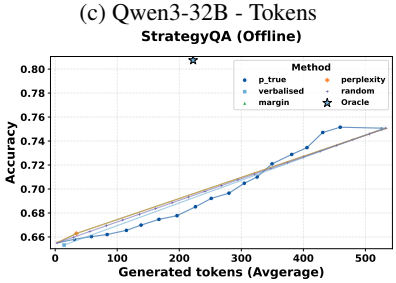
(a) GPT-OSS Medium - Tokens



(b) GPT-5 - Tokens



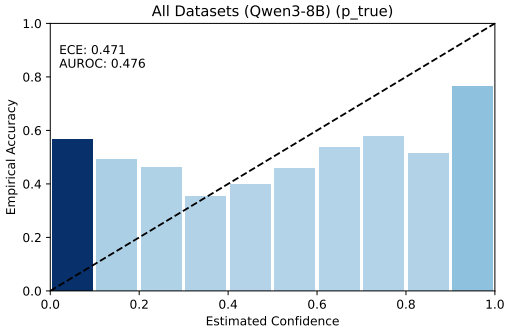
(c) Qwen3-32B - Tokens



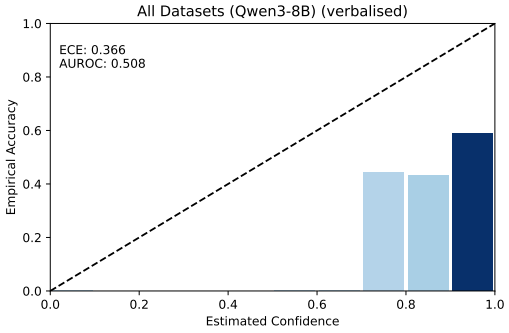
(d) Qwen3-8B - Tokens

Figure 21: **StrategyQA (Part 2): Average Tokens vs. Accuracy.**

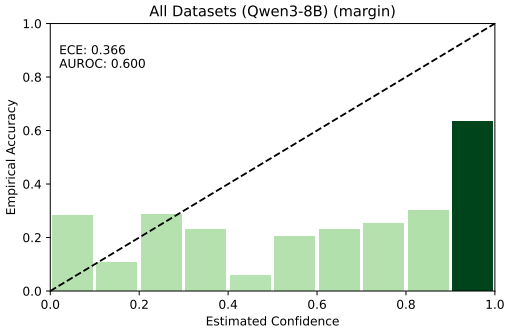
K Reliability Diagrams



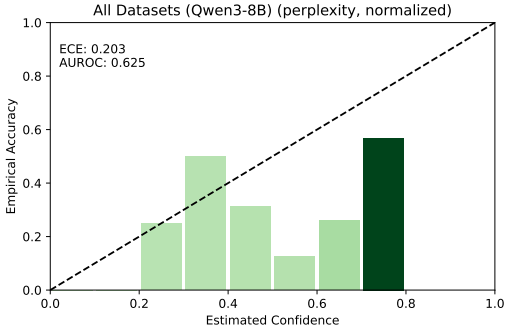
(a) p_true



(b) Verbalised

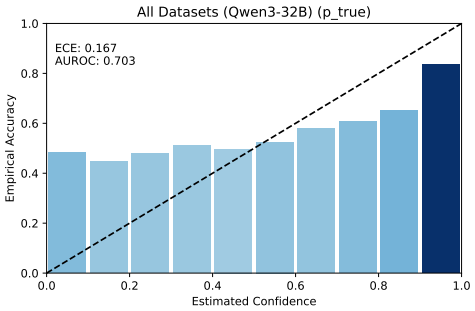


(c) Margin

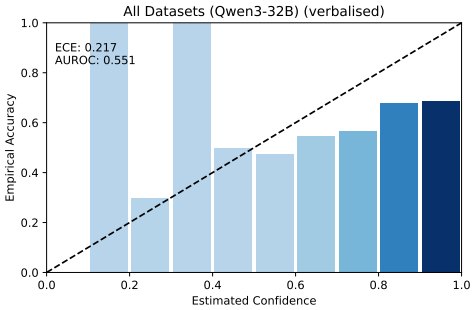


(d) Perplexity (norm)

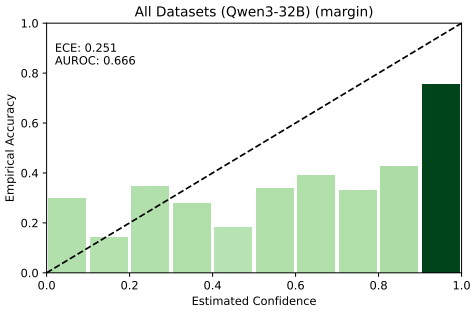
Figure 22: Reliability: **Qwen3-8B**. Bars darken with bin count; dashed line is perfect calibration.



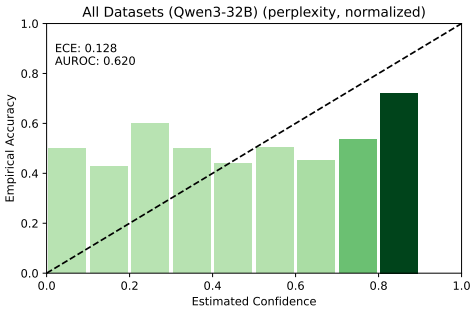
(a) p_true



(b) Verbalised

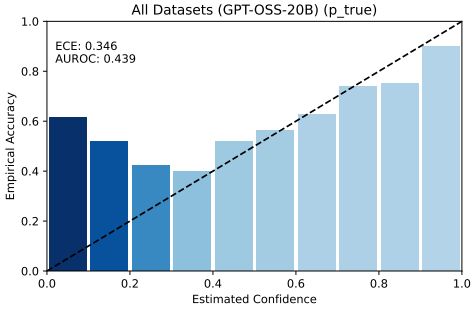


(c) Margin

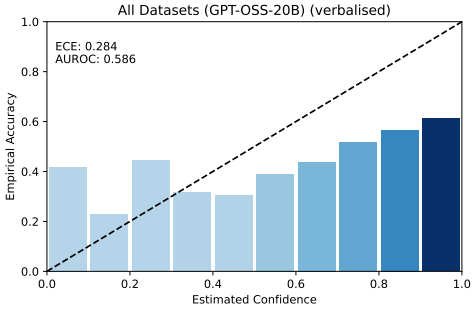


(d) Perplexity (norm)

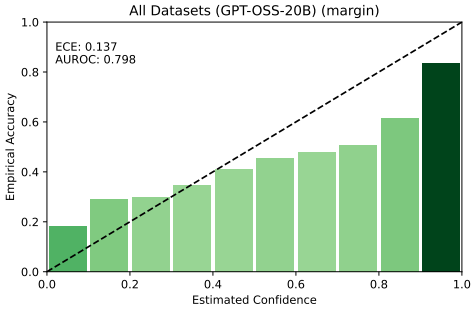
Figure 23: Reliability diagrams for **Qwen3-32B**. Bars darken with bin count; dashed line is perfect calibration.



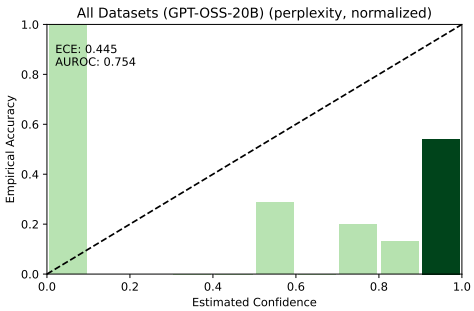
(a) p_true



(b) Verbalised



(c) Margin



(d) Perplexity (norm)

Figure 24: Reliability diagrams for **GPT-OSS-20B**. Bars darken with bin count; dashed line is perfect calibration.

L Full CAR Results

Method	Acc. ↑	ΔAcc ↑	CoT (%) ↓	Tok. saved ↑
GPT-OSS				
CAR P(True)	75.9 $_{0.8}$	-4.1	77.6 $_{7.8}$	83.4 $_{32.3}$
CAR Verbalised	62.8 $_{0.7}$	-17.1	28.8 $_{2.8}$	235.5 $_{17.4}$
CAR Margin	74.4 $_{0.2}$	-5.5	42.7 $_{0.5}$	166.3 $_{4.0}$
CAR Perplexity	63.1 $_{4.4}$	-16.8	19.0 $_{21.6}$	391.4 $_{105.2}$
Qwen32B				
CAR P(True)	76.4 $_{0.4}$	-7.3	34.2 $_{1.4}$	543.8 $_{13.4}$
CAR Verbalised	70.6 $_{1.2}$	-13.1	23.6 $_{9.7}$	565.1 $_{106.9}$
CAR Margin	74.8 $_{0.5}$	-8.9	14.8 $_{1.1}$	668.3 $_{16.0}$
CAR Perplexity	72.7 $_{2.2}$	-11.0	16.9 $_{16.5}$	691.5 $_{136.2}$
Qwen8B				
CAR P(True)	76.9 $_{0.2}$	-2.2	82.2 $_{0.3}$	209.2 $_{7.5}$
CAR Verbalised	61.3 $_{2.2}$	-17.8	12.8 $_{11.2}$	1038.8 $_{121.4}$
CAR Margin	63.8 $_{0.3}$	-15.2	11.5 $_{0.6}$	1048.1 $_{11.6}$
CAR Perplexity	63.0 $_{0.5}$	-16.1	9.9 $_{1.0}$	1066.0 $_{22.7}$
GPT-5.1				
CAR Verbalised	77.7 $_{0.2}$	-8.9	26.4 $_{0.9}$	320.6 $_{19.4}$

Table 8: Full CAR (Lu et al., 2025) results across all confidence signals and models. CAR fits a Gaussian to confidence scores to derive routing thresholds. Best accuracy result per model is reported in Table 1.

M GPT-OSS full results

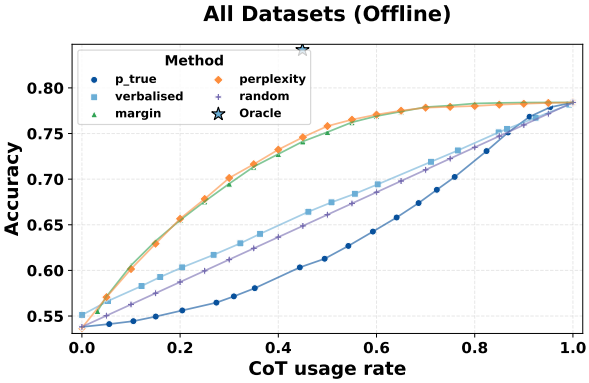


Figure 25: GPT-OSS-20B Accuracy vs. CoT usage results (Low reasoning effort)

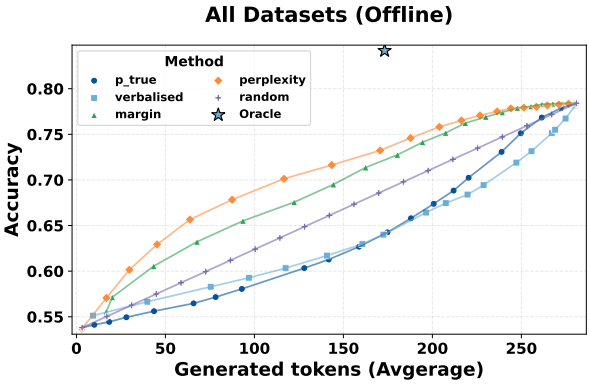


Figure 26: GPT-OSS-20B Accuracy vs. Average Tokens results (Low reasoning effort)

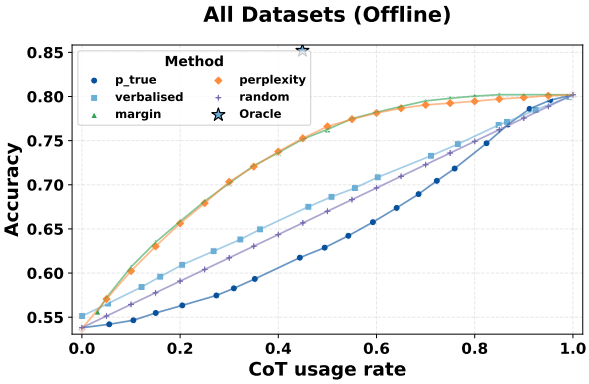


Figure 27: GPT-OSS-20B Accuracy vs. CoT usage results (High reasoning effort)

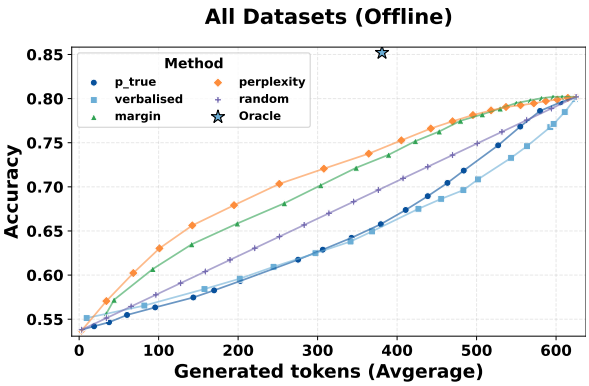


Figure 28: GPT-OSS-20B Accuracy vs. Average Tokens results (High reasoning effort)

N Sensitivity to Accuracy Tolerance ϵ

Table 9 reports Pareto-optimal threshold results across $\epsilon \in \{0.5\%, 1\%, 1.5\%, 2\%\}$. As ϵ increases, token savings increase for signals with strong discriminative power, while the realised accuracy drop closely tracks the chosen tolerance. Signals with weak discriminative power or saturated scores (e.g. margin and perplexity on Qwen3-32B and Qwen3-8B) default to full CoT at every tolerance.

ϵ	Signal	Acc.	ΔAcc	CoT (%)	Tok. saved
GPT-OSS-20B (All CoT: 79.9)					
0.5%	P(True)	79.7	-0.2	98.5	5.2
	Verbalised	79.7	-0.2	99.3	1.1
	Margin	79.5	-0.5	72.5	52.3
	Perplexity	79.4	-0.5	80.6	40.2
1%	P(True)	79.2	-0.7	95.5	15.3
	Verbalised	79.7	-0.2	99.2	1.1
	Margin	79.1	-0.8	68.1	65.0
	Perplexity	79.0	-1.0	70.6	65.7
1.5%	P(True)	78.9	-1.0	94.2	20.6
	Verbalised	79.5	-0.5	98.2	2.5
	Margin	78.7	-1.2	64.6	76.1
	Perplexity	78.6	-1.4	64.2	84.7
2%	P(True)	78.5	-1.4	92.6	28.4
	Verbalised	78.9	-1.1	95.5	6.2
	Margin	78.2	-1.7	60.4	90.4
	Perplexity	78.1	-1.8	59.8	100.1
Qwen3-32B (All CoT: 83.8)					
0.5%	P(True)	83.2	-0.5	81.4	115.3
	Verbalised	83.7	-0.1	98.9	3.8
	Margin	83.8	0.0	100.0	0.0
	Perplexity	83.8	0.0	100.0	0.0
1%	P(True)	82.8	-0.9	73.8	170.9
	Verbalised	83.7	-0.1	98.9	3.8
	Margin	83.8	0.0	100.0	0.0
	Perplexity	83.8	0.0	100.0	0.0
1.5%	P(True)	82.4	-1.4	68.7	212.4
	Verbalised	83.7	-0.1	98.9	3.8
	Margin	83.8	0.0	100.0	0.0
	Perplexity	83.8	0.0	100.0	0.0
2%	P(True)	82.0	-1.8	64.5	249.3
	Verbalised	83.7	-0.1	98.9	3.8
	Margin	83.8	0.0	100.0	0.0
	Perplexity	83.8	0.0	100.0	0.0
Qwen3-8B (All CoT: 79.1)					
0.5%	P(True)	78.8	-0.3	95.8	37.4
	Verbalised	79.1	0.0	100.0	0.0
	Margin	79.1	0.0	100.0	0.0
	Perplexity	79.1	0.0	100.0	0.0
1%	P(True)	78.4	-0.7	90.8	86.6
	Verbalised	79.0	0.0	100.0	0.2
	Margin	79.1	0.0	100.0	0.0
	Perplexity	79.1	0.0	100.0	0.0
1.5%	P(True)	78.0	-1.1	87.1	130.4
	Verbalised	79.0	-0.1	99.8	1.2
	Margin	79.1	0.0	100.0	0.0
	Perplexity	79.1	0.0	100.0	0.0
2%	P(True)	77.6	-1.4	85.3	158.7
	Verbalised	78.5	-0.5	98.9	5.4
	Margin	79.1	0.0	100.0	0.0
	Perplexity	79.1	0.0	100.0	0.0
GPT-5.1 (verbalised only; All CoT: 86.6)					
0.5%	Verbalised	86.4	-0.2	99.0	0.4
1%	Verbalised	86.1	-0.4	97.0	1.5
1.5%	Verbalised	85.5	-1.1	91.5	4.5
2%	Verbalised	85.3	-1.2	89.8	5.4

Table 9: Pareto-optimal threshold results across accuracy tolerances ϵ. Means over 100 Monte Carlo calibration repeats. The $\epsilon = 1\%$ rows correspond to Table 1.

O Heuristic Baseline

We include a keyword-based heuristic baseline that triggers CoT whenever task-relevant keywords or phrases are detected in the input query. The full keyword list is shown in Table 10. If any keyword is present, the query is routed to CoT; otherwise, the model answers directly. This produces a single fixed point on the graph rather than a full trade-off curve. For GPT-OSS-20B, the heuristic operates at approximately 0.4 CoT usage with 0.70 accuracy, outperforming random at this point but falling below margin and perplexity at the same CoT usage rate. For the Qwen models it performs similarly or better to the confidence methods at the same CoT usage rate. Results are shown in Figure 29.

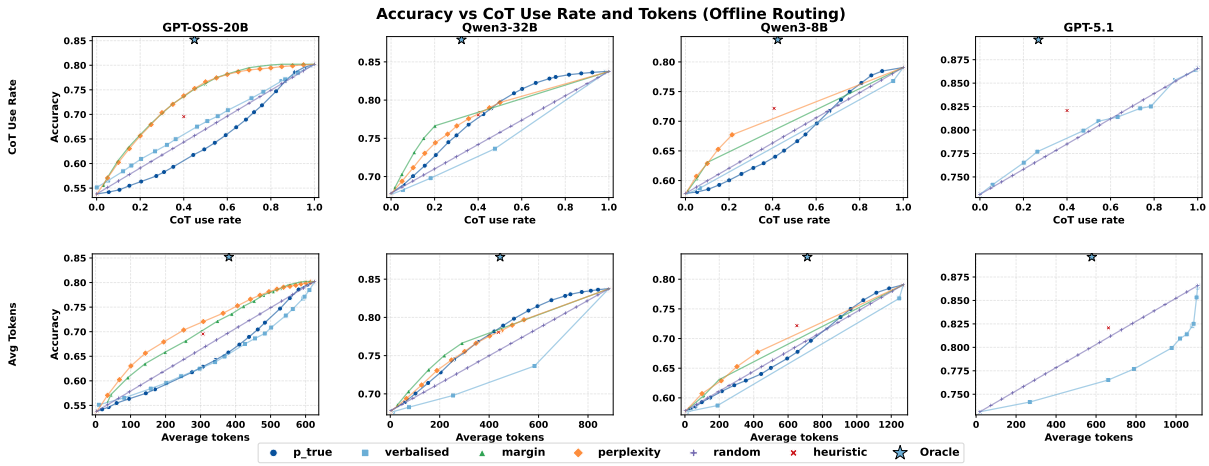


Figure 29: Offline accuracy–efficiency trade-offs with keyword heuristic baseline. The heuristic produces a single fixed operating point, compared to the random baseline which traces full trade-off curves.

Category	**Keywords / Phrases**
Mathematical operations	*calculate, compute, solve, evaluate, determine*
Quantitative indicators	*how many, how much, what is the, find the*
Problem-solving	*figure out, work out, estimate, approximate*
Logical reasoning	*prove, justify, explain why, deduce, infer, conclude, implies, therefore, because*
Comparisons	*compare, contrast, which is greater, which is less*
Optimization	*maximize, minimize, optimal, best option*
Arithmetic operations	*sum, total, difference, product*

Table 10: Heuristic keyword list used for routing decisions. A question is routed to CoT if it contains any of these terms.