# I KNOW THAT I KNOW NOTHING: NARROWING THE KNOWLEDGE-OUTPUT GAP IN LARGE LANGUAGE MODELS THROUGH EARLY HIGH-ENTROPY TOKEN INTERVENTION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Large language models can encode task-relevant information in their hidden states while failing to express it reliably during generation. We show that, in many incorrect cases, signals predictive of later failure are already present before decoding begins, yet these internal judgements do not consistently constrain the realised output. We identify structural compression in autoregressive decoding as a mechanism contributing to this knowledge-output gap: as generation proceeds, judgement-relevant information becomes progressively compressed along causal attention paths, weakening its influence on the final decision. To mitigate this effect, we propose a minimal test-time intervention that selectively targets a small number of early high-entropy tokens, where uncertainty and path selection concentrate. The intervention delays premature collapse into erroneous reasoning trajectories while leaving low-uncertainty decoding largely unchanged. Across benchmarks covering general knowledge, programming, mathematical reasoning, scientific reasoning, and multi-hop question answering, this strategy improves reasoning performance by 3.19 to 7.29 points and, in the inference-cost analysis, reduces generated-token usage by 25.4% to 38.9% while maintaining the highest accuracy.

## 1 INTRODUCTION

*"I know that I know nothing,"* a saying made famous by Socrates (Plato, 2019), captures a state in which knowledge is present yet only partially available to expression. It does not describe simple ignorance, but a form of self-awareness in which one can anticipate the limits of an answer before that answer can be fully articulated. Closely related phenomena have long been studied in cognitive science under the notion of the feeling of knowing (Koriat, 1993). For example, an experienced physician may sense that a patient's symptoms do not fit an initial diagnosis before identifying the underlying cause, and a software engineer may recognise that a system design will fail before formalising the flaw. In such cases, anticipation arises from prior knowledge (Allen, 2000): one understands enough to detect that the available expression is incomplete. Knowing and saying therefore come apart, revealing a gap between internal judgement and outward expression.

A closely analogous phenomenon can be observed in large language models (LLMs). Studies on factual question answering show that answer accuracy can be anticipated from question-only activations before generation begins (Cencerrado et al., 2025). Work on reasoning and difficulty estimation suggests that hidden representations encode internal assessments of problem difficulty (Zhu et al., 2025). Response-planning analyses further indicate that models form global expectations about future outputs before they are realised in text (Dong et al., 2025). Evidence on hidden factual knowledge likewise shows that task-relevant information may be present inside the model while remaining absent or distorted in the generated answer (Gekhman et al., 2025). These findings point to a knowledge–output gap, where internal knowledge and judgement are available but do not reliably shape the realised answer. Existing work has used related signals to organise training around problem difficulty (Ji et al., 2025), invoke additional search or reasoning when confidence is low (Lee et al., 2025), or modify model states during generation to reduce hallucinatory continuations (Simhi



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.

et al., 2024). Such approaches mainly examine how correct outputs can be predicted, elicited, or reinforced, leaving less explored why internally available judgements fail to constrain the final answer and how incorrect outputs emerge as generation unfolds.

Recent work suggests that information flow during language model generation is highly non-uniform, with a small number of influential positions shaping reasoning trajectories. Analyses of long-form reasoning show that extended traces often contain substantial redundancy, motivating attention to positions that determine subsequent generation Fatemi et al. (2025). Attribution studies identify thought anchors, such as planning, revision, and backtracking statements, that exert disproportionate influence on later reasoning Bogdan et al. (2025). Positional analyses further show that errors introduced early in reasoning can strongly affect the final prediction Liao et al. (2025). High-entropy minority tokens have been shown to function as reasoning forks that steer models towards different solution paths Li et al. (2026). Reasoning-distillation evidence suggests that early high-entropy tokens carry dense supervisory signals, with early token segments retaining much of the value of full reasoning traces Chen et al. (2025b). Future-token representation studies further show that coarse ordering information over upcoming tokens can improve language modelling, suggesting that early tokens can encode broad trajectory information before exact future tokens are fixed Zuhri et al. (2025). Together, these findings motivate the hypothesis that reasoning failures are likely to concentrate at early uncertain positions, especially early high-entropy tokens, where internal judgements begin to shape subsequent output trajectories.

Motivated by these observations, we revisit generation failures from a mechanistic perspective. We first ask whether incorrect answers are produced only after decoding begins, or whether failure-prone instances already carry distinguishable internal structure beforehand. Our probing results show that, before any output token is generated, instances that later lead to incorrect answers can already be separated from those that lead to correct ones. This suggests that many failures reflect a breakdown between internal judgement and realised expression, rather than a simple absence of task-relevant knowledge. We then analyse why such judgements fail to constrain generation and identify structural compression in autoregressive decoding as a contributing mechanism. As generation proceeds, decision-relevant information must be carried through increasingly constrained causal paths, causing its influence to weaken as the output trajectory becomes fixed. Earlier positions therefore offer greater leverage for reducing premature commitment to erroneous paths, while later positions mainly elaborate trajectories that have already been selected. This motivates our focus on early decoding steps, and on token-level predictive entropy as a signal of uncertainty and path selection. We show that early high-entropy tokens concentrate decision-critical information and account for a disproportionate share of generation errors. On this basis, we propose **EHETI** (**E**arly **H**igh-**E**ntropy **T**oken **I**ntervention), a minimal test-time intervention that acts only on a small number of these tokens, reducing the mismatch between internal judgements and realised outputs while leaving the rest of the generation process unchanged.

**Key Contributions:** (1) We show that generation failures exhibit pre-generation structure, with signals predictive of later errors already present before decoding begins and separable from those associated with correct outcomes. (2) We analyse why these internal judgements fail to constrain the realised answer, identifying structural compression in autoregressive decoding as a mechanism contributing to the knowledge–output gap. (3) We propose EHETI, a minimal sparse intervention on early high-entropy tokens, to mitigate premature commitment to erroneous reasoning paths without parameter updates or broad changes to the generation process. (4) Through extensive experiments on reasoning benchmarks spanning general knowledge, programming, mathematical reasoning, scientific reasoning, and multi-hop question answering, we show consistent performance gains together with substantial reductions in inference-time computational cost.

## 2 METHODOLOGY

As illustrated in Figure 1, our methodology proceeds through four successive stages. We first test whether later generation failures have measurable internal signatures before decoding. We then analyse how autoregressive decoding compresses decision-relevant information through causal computation paths. Next, we examine where uncertainty concentrates within generated reasoning trajectories. Finally, we apply EHETI, a minimal test-time intervention, at a small number of early high-entropy positions to reduce the gap between internal judgements and realised outputs.

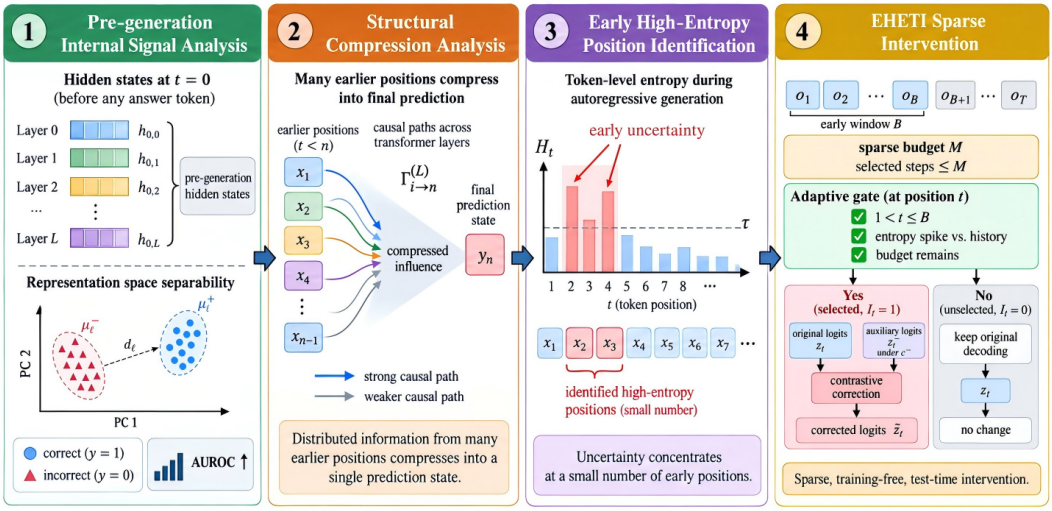


Figure 1: Overview of the proposed methodology. The workflow probes error-predictive internal signals, analyses structural compression in decoding, identifies early high-entropy decision points, and applies EHETI as a sparse test-time intervention.

### 2.1 Pre-Generation Separability of Error-Predictive Internal Signals

We first examine whether successful and failure-prone generations are already separable before any output token is produced. For instance $i$, let $\mathbf{h}_{0,\ell}^{(i)} \in \mathbb{R}^d$ denote the pre-generation hidden state at layer $\ell$, where $d$ is the hidden dimension and $t = 0$ denotes the state after processing the input but before generating the first answer token. Let $y_i \in \{0, 1\}$ denote the final-answer correctness label.

For each layer $\ell$, we split the dataset into correctly answered instances $\mathcal{D}^+$ and incorrectly answered instances $\mathcal{D}^-$. Their centroids in the pre-generation representation space are denoted by $\boldsymbol{\mu}_\ell^+$ and $\boldsymbol{\mu}_\ell^-$. The correctness direction is $\mathbf{d}_\ell = \boldsymbol{\mu}_\ell^+ - \boldsymbol{\mu}_\ell^-$. Each instance is scored by projecting its pre-generation representation onto this direction:

$$s_{i,\ell} = \frac{(\mathbf{h}_{0,\ell}^{(i)} - \boldsymbol{\mu}_\ell)^\top \mathbf{d}_\ell}{\|\mathbf{d}_\ell\|}, \quad \boldsymbol{\mu}_\ell = \frac{1}{2}(\boldsymbol{\mu}_\ell^+ + \boldsymbol{\mu}_\ell^-). \tag{1}$$

Here, $s_{i,\ell}$ is the correctness-direction score, and $\boldsymbol{\mu}_\ell$ is the midpoint between the correct and incorrect centroids. Layer-wise separability is measured by

$$\mathrm{AUROC}_\ell = \mathrm{AUROC}(\{s_{i,\ell}\}_{i=1}^N, \{y_i\}_{i=1}^N), \tag{2}$$

where $N$ is the number of evaluated instances. This analysis establishes the premise that later failures can already have measurable representation-level signatures before they appear in text.

Figure 2 shows that error-predictive signals are already separable before decoding across the evaluated models. During generation, the signal is relatively strong near the start, weakens at intermediate positions, and re-emerges near the end, indicating that its accessibility changes as the autoregressive trajectory unfolds. Detailed pre-generation and generation-time probing analyses are provided in Appendices E.1.1 and E.1.2. Appendix E.1.3 extends the analysis to output-relevant attributes, including response structure, content choices, answer confidence, and factual consistency. Appendix E.1.4 examines how these signals are organised across transformer layers.

### 2.2 Structural Compression in Autoregressive Decoding

The pre-generation analysis shows that error-predictive information can be present before decoding. We then analyse why such information may fail to reliably constrain the realised answer. In a decoder-only Transformer, the next-token distribution is produced from a terminal prediction state. Let $v_i^{(0)} \in \mathbb{R}^d$ denote the input-layer representation at position $i$, let $v_i^{(\ell)}$ denote its representation at

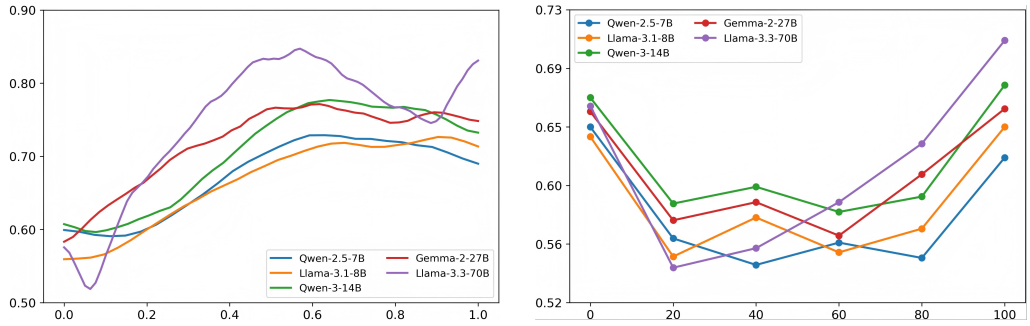


Figure 2: Pre-generation and generation-time dynamics of error-predictive internal signals across multiple LLMs on TriviaQA. **Left:** AUROC across normalised layers before decoding. **Right:** Binary F1 across normalised decoding positions.




Figure 3: Positional asymmetry in token-level copying across sequence lengths, including first-vs-last copying, hint effects on last-token copying, and interleaved binary sequences.

layer $\ell$, and let $y_n = \mathrm{norm}_3(v_n^{(L)})$ denote the final-token representation after $L$ decoder layers and final normalisation.

To quantify how strongly information from position $i$ can affect the final prediction state at position $n$, we analyse the sensitivity of $y_n$ to $v_i^{(0)}$. Let $\Gamma_{i\to n}^{(L)}$ denote the total attention-weighted mass of causal paths from position $i$ to position $n$ across $L$ layers. Under fixed forward-pass attention coefficients and bounded layer-normalisation effects, this sensitivity satisfies

$$\left\| \frac{\partial y_n}{\partial v_i^{(0)}} \right\| \le C\, \Gamma_{i\to n}^{(L)}. \tag{3}$$

Here, $C > 0$ collects layer-wise Lipschitz and normalisation constants, and $\|\cdot\|$ is a compatible vector norm. This bound formalises the structural-compression effect: information distributed across input positions must reach a single terminal prediction state through causal computation paths, and its influence can weaken when the available path mass is sparse or low weight.

Figure 3 provides a direct empirical counterpart to the causal-path analysis: first-token copying remains stable as sequence length increases, whereas last-token copying degrades substantially, with only partial recovery from explicit hints or interleaving. Appendix E.2.1 provides the full causal-path formulation and copying analysis. Appendix E.2.2 extends the same account from single-position recovery to multi-position accumulation. Appendix E.2.3 links aggregation failures to output-attractor formation through representational collapse and softmax normalisation. Appendix E.2.4 provides the corresponding per-regime output distributions.

### 2.3 Entropy Structure of Reasoning

Structural compression motivates attention to positions where the generation trajectory remains flexible. We therefore analyse where uncertainty concentrates during chain-of-thought generation. For an input $x$, let $o_{<t} = (o_1, \ldots, o_{t-1})$ be the generated prefix before step $t$, and let $\mathbf{z}_t \in \mathbb{R}^V$ be the

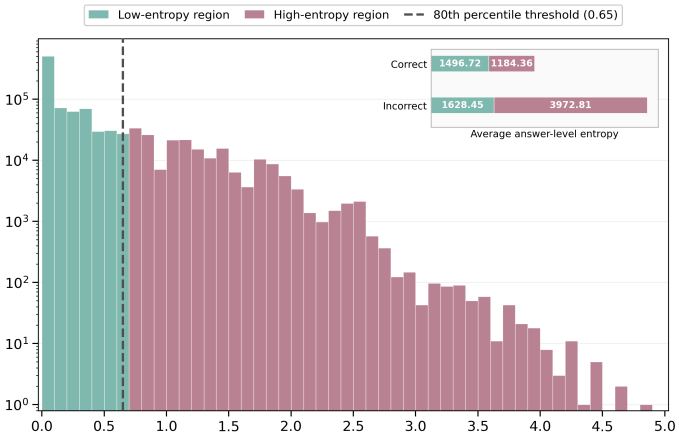


Figure 4: Token- and answer-level entropy during CoT generation. The main panel shows the token-level entropy distribution and empirical 80th-percentile split; the inset compares average answer-level entropy between correct and incorrect responses.

pre-softmax logit vector over a vocabulary of size $V$. With decoding temperature $T_{\text{dec}}$, the predictive distribution is $\mathbf{p}_t = \text{softmax}(\mathbf{z}_t/T_{\text{dec}})$. The scalar $p_{t,i}$ denotes the probability assigned to vocabulary item $i$. We measure token-level uncertainty as

$$H_t = -\sum_{i=1}^{V} p_{t,i} \log p_{t,i}. \tag{4}$$

Here, $H_t$ is tied to the predictive distribution at a particular decoding step, so the same surface token can have different entropy values in different reasoning contexts.

Let $\mathcal{H} = \{H_{n,t} \mid n = 1, \ldots, N,\ t = 1, \ldots, L_n\}$ be the pooled multiset of token entropy values over $N$ generated responses, where $H_{n,t}$ is the entropy at position $t$ in response $n$, and $L_n$ is the length of response $n$. We set an entropy cutoff $\tau = \text{Quantile}_{0.8}(\mathcal{H})$, and partition response $n$ into $\mathcal{T}^{(n)}_{\text{high}} = \{t \mid H_{n,t} > \tau\}$ and $\mathcal{T}^{(n)}_{\text{low}} = \{t \mid H_{n,t} \le \tau\}$. If $A_n = \sum_{t=1}^{L_n} H_{n,t}$ denotes cumulative answer-level entropy, its high- and low-entropy contributions are

$$A^{\text{high}}_n = \sum_{t \in \mathcal{T}^{(n)}_{\text{high}}} H_{n,t}, \quad A^{\text{low}}_n = \sum_{t \in \mathcal{T}^{(n)}_{\text{low}}} H_{n,t}. \tag{5}$$

This decomposition tests whether the uncertainty gap between correct and incorrect responses is concentrated in a small subset of positions or distributed across the full reasoning trace.

Figure 4 shows that most generated tokens have entropy close to zero, while a small minority carry substantially higher uncertainty. The answer-level gap between correct and incorrect responses is dominated by the high-entropy contribution, with comparatively little change in the low-entropy majority. Appendix E.3.1 provides the full entropy-concentration analysis, including token-level entropy statistics and answer-level decomposition by correctness. Appendix E.3.2 examines the semantic roles of high- and low-entropy token groups.

### 2.4 Sparse Test-Time Intervention

The preceding stages identify early high-entropy positions as sparse sites where local uncertainty can influence later reasoning. We use this structure to define EHETI, a training-free test-time intervention. At decoding step $t$, let $\mu_{t-1}$ and $\sigma_{t-1}$ be the running mean and standard deviation of entropy values observed before step $t$. Let $B$ denote the early-window length, $M$ the maximum number of intervention steps in one response, and $\alpha \ge 0$ the entropy-spike threshold. The intervention gate is

$$I_t = \mathbf{1}\Big[1 < t \le B,\ H_t > \mu_{t-1} + \alpha\sigma_{t-1}, \sum_{k=1}^{t-1} I_k < M\Big]. \tag{6}$$

Here, $I_t = 1$ indicates that the current position is selected for intervention, and $\mathbf{1}[\cdot]$ is the indicator function.

When $I_t = 0$, decoding uses the original logits $\mathbf{z}_t$. When $I_t = 1$, we compute auxiliary logits $\mathbf{z}_t^- \in \mathbb{R}^V$ under a lightweight contrastive condition $c^-$ representing unreliable reasoning behaviour. The corrected logits are

$$\widetilde{\mathbf{z}}_t = \mathbf{z}_t + \lambda I_t \left( \mathbf{z}_t - \mathbf{z}_t^- \right), \tag{7}$$

where $\lambda \geq 0$ controls intervention strength. The post-intervention distribution is $\widetilde{\mathbf{p}}_t = \mathrm{softmax}(\widetilde{\mathbf{z}}_t / T_{\mathrm{dec}})$. The auxiliary logits are computed only when $I_t = 1$, keeping the intervention sparse. Appendix E.4 provides the full intervention formulation and implementation details.

# 3 Experiments and Results

## 3.1 Experimental Setup

**Datasets:** We evaluate reasoning performance across general knowledge, programming, mathematical reasoning, scientific reasoning, and multi-hop QA. MMLU-Pro (Wang et al., 2024a) assesses broad multi-disciplinary knowledge and problem solving in a multiple-choice format. HumanEvalPlus (Liu et al., 2023) evaluates code generation correctness with augmented tests, while LiveCodeBench (Jain et al., 2024) measures programming ability in continuously updated evaluation settings. GPQA-Diamond (Rein et al., 2024) targets graduate-level scientific reasoning, and MATH500 (Lightman et al., 2023) evaluates competition-level mathematical problem solving requiring multi-step symbolic reasoning. 2WikiMultiHopQA (Ho et al., 2020) evaluates multi-hop question answering over natural-language evidence, where models must combine information across multiple passages to produce the final answer. For inference-cost analysis, we additionally use AIME 2024, AIME 2025, BRUMO 2025, and HMMT 2025, which are challenging competition-style mathematical reasoning benchmarks.

**Comparative Models and Strategies:** We evaluate five reasoning models spanning different model families and parameter scales, including DeepSeek-R1-8B (Guo et al., 2025), Qwen3-32B (Yang et al., 2025a), GLM-5.1 (Zeng et al., 2026), MiniMax-M2.7 (Chen et al., 2026), and GPT-OSS-120B (Agarwal et al., 2025). For each model, we compare six inference strategies: Direct Inference (DI), MTI (Yang et al., 2025c), Self-Consistency (SC@32) (Wang et al., 2022), Self-Refine-2 (Madaan et al., 2023), Classifier-Free Guidance (CFG) (Sanchez et al., 2023), and EHETI. DI generates a single reasoning trajectory using standard decoding, without test-time intervention or output aggregation. MTI adjusts the generation process at selected decoding steps. SC@32 samples 32 independent reasoning trajectories and selects the final answer by answer-level majority voting. Self-Refine-2 generates an initial answer and performs two rounds of self-feedback and revision using the same model. CFG combines conditional and unconditional predictions during decoding to guide the generation distribution.

**Evaluation Metrics:** Predictions are evaluated using the official or commonly adopted metric for each benchmark, with accuracy (Acc.) used for MMLU-Pro, GPQA-Diamond, and MATH500, Pass@1 for HumanEvalPlus and LiveCodeBench, and answer EM for 2WikiMultiHopQA. In performance tables, *Avg.* denotes the averaged score across benchmarks. To quantify computational cost, we measure the total number of generated tokens during inference. For multi-sample or iterative strategies, token usage is accumulated across all generated trajectories, feedback, and revision steps. In cost tables, *Avg.* denotes the average of the comparison strategies, and the percentage after our token count reports the relative reduction against this average.

## 3.2 Overall Results

Table 1 presents the main comparison across five reasoning models and six benchmarks. EHETI obtains the highest average score for every model, with gains over Direct Inference ranging from 3.19 to 7.29 points. The improvement is largest on DeepSeek-R1-8B, suggesting that models with tighter representational and computational margins are more sensitive to early uncertain choices, where a local decision can prematurely narrow the subsequent trajectory and propagate errors through the rest of the response. The gains are smaller on stronger models such as GLM-5.1, MiniMax-M2.7, and GPT-OSS-120B, but remain consistent, indicating that scaling improves trajectory stability without

Table 1: Main performance comparison of inference strategies. Within each model block, green cells mark the best results, while blue cells mark cases where EHETI ranks second.

| Model | Strategy | MMLU-Pro | GPQA-Diamond | MATH500 | HumanEvalPlus | LiveCodeBench | 2WikiMultiHopQA | Avg. |
|---|---|---|---|---|---|---|---|---|
| | | Acc. | Acc. | Acc. | Pass@1 | Pass@1 | EM | Score |
| DeepSeek-R1-8B | DI | 50.13 | 49.06 | 89.04 | 66.52 | 39.57 | 56.18 | 58.42 |
| | MTI | 51.34 | 54.26 | 91.62 | 69.12 | 46.24 | 60.22 | 62.13 |
| | SC@32 | 52.72 | 57.36 | 90.88 | 68.96 | 42.12 | 64.26 | 62.72 |
| | Self-Refine-2 | 49.84 | 54.68 | 88.92 | 64.42 | 36.58 | 66.12 | 60.09 |
| | CFG | 49.82 | 50.08 | 87.46 | 65.73 | 41.28 | 56.46 | 58.47 |
| | **EHETI** | 53.84 | 59.48 | 93.26 | 71.92 | 51.04 | 64.72 | 65.71 |
| Qwen3-32B | DI | 76.76 | 61.62 | 96.94 | 97.56 | 91.02 | 76.77 | 83.45 |
| | MTI | 77.36 | 64.58 | 96.31 | 96.42 | 90.33 | 78.44 | 83.91 |
| | SC@32 | 78.86 | 67.24 | 97.82 | 98.94 | 93.52 | 82.68 | 86.51 |
| | Self-Refine-2 | 76.22 | 66.08 | 96.98 | 96.38 | 88.72 | 83.18 | 84.59 |
| | CFG | 74.96 | 56.57 | 94.06 | 95.73 | 87.25 | 76.76 | 80.89 |
| | **EHETI** | 78.06 | 68.74 | 98.73 | 98.23 | 93.62 | 82.46 | 86.64 |
| GLM-5.1 | DI | 84.18 | 86.16 | 98.56 | 90.54 | 73.96 | 81.23 | 85.77 |
| | MTI | 84.92 | 88.35 | 99.02 | 91.36 | 77.28 | 83.76 | 87.45 |
| | SC@32 | 85.72 | 90.38 | 98.83 | 91.62 | 76.88 | 86.18 | 88.27 |
| | Self-Refine-2 | 83.26 | 89.52 | 97.94 | 88.84 | 70.92 | 87.08 | 86.26 |
| | CFG | 84.04 | 86.38 | 97.64 | 89.67 | 75.62 | 81.42 | 85.80 |
| | **EHETI** | 85.64 | 91.86 | 99.58 | 92.16 | 82.24 | 86.88 | 89.73 |
| MiniMax-M2.7 | DI | 82.07 | 77.98 | 98.16 | 92.28 | 76.44 | 79.37 | 84.38 |
| | MTI | 82.82 | 80.76 | 98.72 | 93.46 | 79.64 | 81.86 | 86.21 |
| | SC@32 | 83.58 | 83.24 | 98.62 | 93.48 | 78.74 | 84.42 | 87.01 |
| | Self-Refine-2 | 81.34 | 82.12 | 97.82 | 91.24 | 73.76 | 84.12 | 85.07 |
| | CFG | 81.74 | 78.82 | 97.02 | 91.41 | 77.56 | 79.66 | 84.37 |
| | **EHETI** | 84.08 | 84.36 | 99.61 | 94.34 | 82.82 | 84.38 | 88.27 |
| GPT-OSS-120B | DI | 82.47 | 77.87 | 98.36 | 91.87 | 79.94 | 80.56 | 85.18 |
| | MTI | 83.22 | 80.24 | 98.93 | 92.72 | 82.86 | 82.63 | 86.77 |
| | SC@32 | 83.92 | 82.74 | 98.68 | 92.96 | 82.28 | 85.16 | 87.62 |
| | Self-Refine-2 | 81.78 | 81.24 | 98.02 | 90.68 | 77.82 | 85.38 | 85.82 |
| | CFG | 82.23 | 78.52 | 97.12 | 90.81 | 81.06 | 80.86 | 85.10 |
| | **EHETI** | 84.42 | 83.76 | 99.65 | 93.28 | 84.52 | 85.36 | 88.50 |

fully eliminating influential uncertain positions. Additional performance results on more models and inference strategies are reported in Appendices F.1 and F.2, respectively.

The comparison among inference strategies shows that different forms of test-time computation help in different ways. SC@32 is strong when repeated samples form a reliable majority, especially on knowledge-intensive and multi-hop tasks. Self-Refine-2 is particularly effective on 2WikiMultiHopQA, where explicit revision can help reconcile evidence across passages. These strategies are less stable on code and some symbolic tasks, where additional generation can introduce drift or disrupt an already correct solution. EHETI achieves the best average performance because it adjusts uncertain decision points while preserving stable continuation steps.

Across benchmarks, the largest gains appear on GPQA-Diamond, LiveCodeBench, and 2WikiMultiHopQA. These tasks require sequences of dependent decisions, so an early wrong turn can strongly constrain later reasoning or code construction. The improvement on LiveCodeBench is especially notable, since code generation contains many deterministic segments; the result suggests that EHETI does not broadly perturb such segments and mainly affects positions where the next step is genuinely ambiguous. The consistent gains on 2WikiMultiHopQA further indicate that the method can support evidence-dependent reasoning without relying on explicit answer revision.

Table 2 evaluates inference efficiency on competition-style reasoning benchmarks. The Avg. column summarises the competing inference strategies, while the bracketed values under EHETI report the relative token reduction against this average. Across all five models, EHETI uses fewer generated tokens than the comparison average and achieves the best average accuracy. At the model-average level, token reduction ranges from 25.4% to 38.9%, while accuracy improves by 1.6 to 3.6 points. This indicates that the method improves the cost–accuracy trade-off by modifying the reasoning trajectory at a small number of influential positions, with limited need for repeated full-response generation. Additional cost results on more models and inference strategies are reported in Appendices F.3 and F.4, respectively. Appendix F.5 further reports wall-clock latency, auxiliary forward passes, and peak GPU memory.

The efficiency gains are most pronounced on AIME2024 and AIME2025, where early mathematical choices often determine the remaining solution path. Competing strategies spend additional tokens by producing alternative trajectories or revision rounds, many of which repeat stable algebraic or explanatory segments after the decisive step has already been made. EHETI reduces this redundancy while preserving, and often improving, final-answer accuracy. The reduction is smaller on

Table 2: Main inference-cost comparison of token counts ($\times 10^8$) and accuracy (%) across competition-style reasoning benchmarks.

| Model | Dataset | MTI | | SC@32 | | Self-Refine-2 | | CFG | | Avg. | | EHETI | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Token | Acc. | Token | Acc. | Token | Acc. | Token | Acc. | Token | Acc. | Token | Acc. |
| DeepSeek-R1-8B | AIME2024 | 1.46 | 87.8 | 2.04 | 86.1 | 3.18 | 83.6 | 1.34 | 81.7 | 2.00 | 84.8 | **1.12** (-44.1%) | **89.6** |
| | AIME2025 | 2.10 | 82.8 | 2.34 | 81.7 | 3.76 | 80.1 | 1.96 | 78.4 | 2.54 | 80.8 | **1.81** (-28.7%) | **83.9** |
| | BRUMO2025 | 1.88 | 92.8 | 1.92 | 92.4 | 3.20 | 90.8 | 1.72 | 90.7 | 2.18 | 91.7 | **1.62** (-25.7%) | **93.3** |
| | HMMT2025 | 2.82 | 72.6 | 2.76 | 69.4 | 4.08 | 68.4 | 2.64 | 66.1 | 3.08 | 69.1 | **2.52** (-18.0%) | **73.8** |
| | Avg. | 2.06 | 84.0 | 2.26 | 82.4 | 3.56 | 80.7 | 1.92 | 79.2 | 2.45 | 81.6 | **1.77** (-27.9%) | **85.2** |
| Qwen3-32B | AIME2024 | 0.94 | 85.6 | 1.28 | 83.6 | 2.34 | 82.5 | 0.88 | 80.7 | 1.36 | 83.1 | **0.77** (-43.4%) | **88.0** |
| | AIME2025 | 1.54 | 79.6 | 1.56 | 79.0 | 2.82 | 78.0 | 1.41 | 75.6 | 1.83 | 78.0 | **1.38** (-24.7%) | **80.2** |
| | BRUMO2025 | 1.31 | 92.7 | 1.34 | 92.8 | 2.46 | 91.4 | 1.18 | 90.8 | 1.57 | 91.9 | **1.17** (-25.6%) | **92.9** |
| | HMMT2025 | 2.10 | 63.9 | **1.72** | 62.7 | 3.16 | 62.1 | 1.96 | 60.4 | 2.23 | 62.3 | 1.90 (-15.0%) | **64.1** |
| | Avg. | 1.47 | 80.5 | 1.48 | 79.5 | 2.69 | 78.5 | 1.36 | 76.9 | 1.75 | 78.8 | **1.30** (-25.4%) | **81.3** |
| GLM-5.1 | AIME2024 | 1.02 | 93.8 | 1.48 | 92.7 | 2.72 | 91.0 | 0.96 | 88.8 | 1.54 | 91.6 | **0.82** (-46.9%) | **94.5** |
| | AIME2025 | 1.24 | 90.7 | 1.74 | 89.8 | 3.12 | 88.9 | 1.18 | 86.6 | 1.82 | 89.0 | **1.06** (-41.8%) | **91.4** |
| | BRUMO2025 | 1.38 | 96.2 | 1.46 | 95.8 | 2.64 | 94.8 | 1.25 | 94.0 | 1.68 | 95.2 | **1.20** (-28.7%) | **96.8** |
| | HMMT2025 | 2.06 | 84.7 | 2.04 | 82.8 | 3.80 | 82.1 | 1.92 | 80.4 | 2.46 | 82.5 | **1.82** (-25.9%) | **85.6** |
| | Avg. | 1.43 | 91.3 | 1.68 | 90.3 | 3.07 | 89.2 | 1.33 | 87.5 | 1.88 | 89.6 | **1.23** (-34.7%) | **92.1** |
| MiniMax-M2.7 | AIME2024 | 1.08 | 91.5 | 1.56 | 90.8 | 2.86 | 88.6 | 1.02 | 87.8 | 1.63 | 89.7 | **0.88** (-46.0%) | **92.6** |
| | AIME2025 | 1.38 | 88.6 | 1.82 | 87.4 | 3.28 | 86.8 | 1.30 | 84.9 | 1.94 | 86.9 | **1.18** (-39.3%) | **89.8** |
| | BRUMO2025 | 1.42 | 94.9 | 1.52 | 94.5 | 2.76 | 93.7 | 1.28 | 92.8 | 1.74 | 94.0 | **1.24** (-28.9%) | **95.6** |
| | HMMT2025 | 2.18 | 81.4 | 2.18 | 80.2 | 3.94 | 79.6 | 2.04 | 77.3 | 2.58 | 79.6 | **1.94** (-25.0%) | **82.7** |
| | Avg. | 1.52 | 89.1 | 1.77 | 88.2 | 3.21 | 87.2 | 1.41 | 85.7 | 1.98 | 87.5 | **1.31** (-33.7%) | **90.2** |
| GPT-OSS-120B | AIME2024 | 1.05 | 96.3 | 1.78 | 95.9 | 2.96 | 95.4 | 1.10 | 94.0 | 1.72 | 95.4 | **0.87** (-49.5%) | **96.9** |
| | AIME2025 | 1.18 | 96.8 | 1.96 | 96.2 | 3.40 | 95.6 | 1.22 | 93.8 | 1.94 | 95.6 | **0.96** (-50.5%) | **97.5** |
| | BRUMO2025 | 1.44 | 83.2 | 1.62 | 82.7 | 2.88 | 81.6 | 1.35 | 80.4 | 1.82 | 82.0 | **1.27** (-30.3%) | **83.7** |
| | HMMT2025 | 2.16 | 92.4 | 2.38 | 91.5 | 4.08 | 91.2 | 2.02 | 89.7 | 2.66 | 91.2 | **1.88** (-29.3%) | **92.5** |
| | Avg. | 1.46 | 92.2 | 1.94 | 91.6 | 3.33 | 91.0 | 1.42 | 89.5 | 2.04 | 91.0 | **1.24** (-38.9%) | **92.7** |

HMMT2025, where longer and more difficult derivations leave more genuinely uncertain positions throughout the response; even there, the method maintains the best accuracy with substantially lower token usage than the comparison average.

The pattern also holds across model capacity. Stronger models such as GLM-5.1, MiniMax-M2.7, and GPT-OSS-120B already generate more stable trajectories, yet they still benefit from targeted intervention, suggesting that scale reduces redundant exploration without removing it. For DeepSeek-R1-8B and Qwen3-32B, the gains show two complementary cases: the former benefits from correcting fragile early decisions under tighter capacity margins, while the latter shows that high-capacity reasoning can still waste computation when uncertainty is handled by broad sampling. These results support the central claim that efficient reasoning depends on where computation is applied during decoding, not simply on how many complete trajectories are generated.

### 3.3 Ablation Study

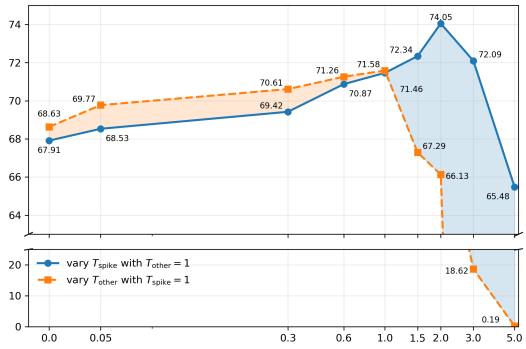


Figure 5: Temperature-sensitivity ablation on AIME. Temperature is varied for high-entropy and low-entropy positions, with the other group fixed at 1.

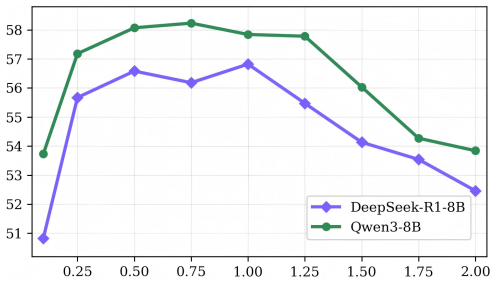


Figure 6: Sensitivity to a diagnostic fixed entropy cutoff $\tau$ on GPQA-Diamond. The x-axis shows the cutoff used to select candidate intervention positions, and the y-axis reports task accuracy.

**Temperature Sensitivity.** Figure 5 examines how reasoning accuracy changes when decoding temperature is applied to different entropy groups. We partition generated tokens by entropy and vary

$T_{\text{high}}$ with $T_{\text{low}} = 1$, and $T_{\text{low}}$ with $T_{\text{high}} = 1$. The two curves remain close at low temperatures, where both groups receive only mild perturbation and the model still follows largely stable continuations. A clear separation emerges after the mid-temperature range. Increasing $T_{\text{high}}$ continues to improve performance and reaches its peak at a substantially higher temperature, showing that high-entropy positions benefit from preserving a wider set of plausible reasoning continuations. These positions occur where the model has not yet committed to a single local direction, so moderate stochasticity can help avoid premature collapse into a weak path. In contrast, the low-entropy curve peaks near the default setting and then declines sharply. Once temperature is raised on low-entropy positions, the perturbation mainly disrupts deterministic continuation, including formula expansion, local arithmetic, and intermediate-step realisation, with little new reasoning flexibility gained. The ablation shows that temperature is useful when applied to uncertainty-bearing positions: high-entropy tokens support productive exploration, and low-entropy tokens mostly require stability. Detailed formulation and experimental procedure are provided in Appendix G.

**Fixed-Cutoff Sensitivity.** Figure 6 examines how performance changes on GPQA-Diamond using a diagnostic fixed entropy cutoff $\tau$, a static counterpart to the adaptive entropy-spike criterion used in EHETI. Here, $\tau$ selects candidate intervention positions according to token entropy, allowing us to isolate the effect of fixed cutoff choice. For both DeepSeek-R1-8B and Qwen3-8B, very small cutoffs include many low-entropy positions and weaken performance by modifying stable decoding steps. Performance improves in the mid-entropy range, where the selected positions better match uncertain reasoning decisions, but declines again when the cutoff becomes too restrictive and misses useful intervention sites. The different sensitivity patterns across models show that a single fixed cutoff is difficult to calibrate reliably, motivating our adaptive criterion, which normalises entropy against the running uncertainty profile of each generated response.

**Effective Modification Sites.** Figure 7 reports the entropy distribution of effective modification sites on the HumanEvalPlus dataset using DeepSeek-R1-8B. A site is counted as effective when the intervention changes the top-1 predicted token at that decoding step. Effective modification sites are sparse in the low-entropy region and concentrate in a bounded mid-to-high entropy interval, forming a clear peak. After the peak, the count decreases but remains non-negligible across the high-entropy tail, indicating that effective modification sites are concentrated in a broad high uncertainty range. This pattern supports entropy-guided targeting; low-entropy positions contribute few effective modification sites, while the mid-to-high and high-entropy regions contain most positions where intervention can alter the next-token decision. Appendix F.6 further reports response-level intervention frequency, positions, and final-answer transitions.

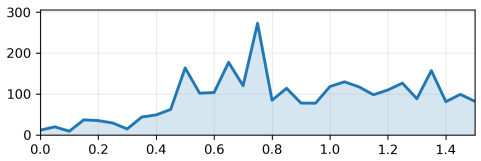


Figure 7: Entropy distribution of effective modification sites. The x-axis shows token entropy, and the y-axis reports the number of generated tokens whose top-1 prediction is changed by the intervention.

## 4 Additional Material

Additional material, including limitations, related work, implementation details, and case analysis, is provided in Appendices A, B, C, and H, respectively.

## 5 Conclusion

This work studies an LLM failure mode in which task-relevant information is internally encoded but not reliably expressed during generation. We find that failure-prone instances exhibit separable pre-generation signals, while autoregressive decoding compresses decision-relevant information through causal computation paths, weakening its influence on the realised answer. Uncertainty further concentrates at a small number of early high-entropy tokens, motivating EHETI, a minimal test-time intervention that improves reasoning performance across diverse reasoning benchmarks while reducing inference cost. Future work will explore richer signals for identifying influential generation positions and further examine how structural compression limits reliable expression in LLMs.

### AI USE STATEMENT

In this work, we used generative AI tools for brainstorming, sourcing and searching for information, and language editing and polishing. All AI-assisted work was reviewed by the authors, who take full responsibility for the final content of the paper.

### ETHICS STATEMENT

This work studies how internally encoded information is expressed during generation and introduces EHETI, a sparse inference-time intervention. The method uses no additional training data, does not update model parameters, and does not require user interaction, so it does not directly introduce new data-collection or privacy risks.

Improved reasoning reliability can still increase user trust in model outputs. Since language models remain fallible, especially outside the evaluated task distribution, the proposed intervention should not be treated as a guarantee of correctness. Human oversight and domain-specific safeguards remain necessary in high-stakes settings.

The method improves the practical efficiency and consistency of reasoning, which may increase the usefulness of models in sensitive applications such as automated decision support or large-scale content generation. These deployment risks depend on governance, access control, monitoring, and use constraints. The method also targets decoding-time expression failures, not representational bias, fairness, factual grounding, or value alignment. Improvements in reasoning performance therefore do not imply broader safety or alignment guarantees.

## REFERENCES

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Aisha Alansari and Hamzah Luqman. Large language models hallucination: A comprehensive survey. *arXiv preprint arXiv:2510.06265*, 2025.

Peter M Allen. Knowledge, ignorance, and learning. *Emergence, A Journal of Complexity Issues in Organizations and Management*, 2(4):78–103, 2000.

Md Rifat Arefin, Gopeshh Raaj Subbaraj, Nicolas Gontier, Yann LeCun, Irina Rish, Ravid Shwartz-Ziv, and Christopher Pal. Seq-vcr: Preventing collapse in intermediate transformer representations for enhanced reasoning. In *International Conference on Learning Representations*, volume 2025, pp. 72939–72960, 2025.

Yejin Bang, Ziwei Ji, Alan Schelten, Anthony Hartshorn, Tara Fowler, Cheng Zhang, Nicola Cancedda, and Pascale Fung. Hallulens: Llm hallucination benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24128–24156, 2025.

Paul C Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. Thought anchors: Which llm reasoning steps matter? *arXiv preprint arXiv:2506.19143*, 2025.

Iván Vicente Moreno Cencerrado, Arnau Padrés Masdemont, Anton Gonzalvez Hawthorne, David Demitri Africa, and Lorenzo Pacchiardi. No answer needed: Predicting llm answer accuracy from question-only linear probes. *arXiv preprint arXiv:2509.10625*, 2025.

Aili Chen, Aonian Li, Baichuan Zhou, Bangwei Gong, Binyang Jiang, Boji Dan, Changhao Zhang, Changqing Yu, Chao Wang, Cheng Ma, et al. The minimax-m2 series: Mini activations unleashing max real-world intelligence. *arXiv preprint arXiv:2605.26494*, 2026.

Justin Chen, Archiki Prasad, Swarnadeep Saha, Elias Stengel-Eskin, and Mohit Bansal. Magicore: Multi-agent, iterative, coarse-to-fine refinement for reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 32651–32674, 2025a.

Wei-Rui Chen, Vignesh Kothapalli, Ata Fatahibaarzi, Hejian Sang, Shao Tang, Qingquan Song, Zhipeng Wang, and Muhammad Abdul-Mageed. Distilling the essence: Efficient reasoning distillation via sequence truncation. *arXiv preprint arXiv:2512.21002*, 2025b.

Xiaoxue Cheng, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. Think more, hallucinate less: Mitigating hallucinations via dual process of fast and slow thinking. *arXiv preprint arXiv:2501.01306*, 2025.

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

Manuel Cossio. A comprehensive taxonomy of hallucinations in large language models. *arXiv preprint arXiv:2508.01781*, 2025.

Hanxing Ding, Liang Pang, Zihao Wei, Huawei Shen, and Xueqi Cheng. Rowen: Adaptive retrieval-augmented generation for hallucination mitigation in llms. In *Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pp. 12–21, 2025.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3029–3051, 2023.

Zhichen Dong, Zhanhui Zhou, Zhixuan Liu, Chao Yang, and Chaochao Lu. Emergent response planning in llms. *arXiv preprint arXiv:2502.06258*, 2025.

Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*, 2023.

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

Mehdi Fatemi, Banafsheh Rafiee, Mingjie Tang, and Kartik Talamadupula. Concise reasoning via reinforcement learning. *arXiv preprint arXiv:2504.05185*, 2025.

Susanne Förster and Yarden Skop. Between fact and fairy: tracing the hallucination metaphor in ai discourse. *AI & SOCIETY*, pp. 1–14, 2025.

Zorik Gekhman, Eyal Ben David, Hadas Orgad, Eran Ofek, Yonatan Belinkov, Idan Szpektor, Jonathan Herzig, and Roi Reichart. Inside-out: Hidden factual knowledge in llms. *arXiv preprint arXiv:2503.15299*, 2025.

Aman Goel, Daniel Schwartz, and Yanjun Qi. Zero-knowledge llm hallucination detection and mitigation through fine-grained cross-model consistency. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1982–1999, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Stevan Harnad. Language writ large: Llms, chatgpt, grounding, meaning and understanding. *arXiv preprint arXiv:2402.02243*, 2024.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

Chengsong Huang, Langlin Huang, Jixuan Leng, Jiacheng Liu, and Jiaxin Huang. Efficient test-time scaling via self-calibration. *arXiv preprint arXiv:2503.00031*, 2025a.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025b.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

Yunjie Ji, Sitong Zhao, Xiaoyu Tian, Haotian Wang, Shuaiting Chen, Yiping Peng, Han Zhao, and Xiangang Li. How difficulty-aware staged reinforcement learning enhances llms' reasoning capabilities: A preliminary experimental study. *arXiv preprint arXiv:2504.00829*, 2025.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

Hazel Kim, Tom A Lamb, Adel Bibi, Philip Torr, and Yarin Gal. Detecting llm hallucination through layer-wise information deficiency: Analysis of ambiguous prompts and unanswerable questions. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 32298–32310, 2025.

Asher Koriat. How do we know that we know? the accessibility model of the feeling of knowing. *Psychological review*, 100(4):609, 1993.

Sungjae Lee, Hyejin Park, Jaechang Kim, and Jungseul Ok. Semantic exploration with adaptive gating for efficient problem solving with language models. *arXiv preprint arXiv:2501.05752*, 2025.

Xianzhi Li, Ethan Callanan, Abdellah Ghassel, and Xiaodan Zhu. Entropy-gated branching for efficient test-time reasoning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5054–5069, 2026.

Baohao Liao, Xinyi Chen, Sara Rajaee, Yuhui Xu, Christian Herold, Anders Søgaard, Maarten de Rijke, and Christof Monz. Lost at the beginning of reasoning. *arXiv preprint arXiv:2506.22058*, 2025.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

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. *Advances in neural information processing systems*, 36:21558–21572, 2023.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023.

Omar Mahmoud, Ali Khalil, Thommen George Karimpanal, Buddhika Laknath Semage, and Santu Rana. The unintended trade-off of ai alignment: Balancing hallucination mitigation and safety in llms. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 1017–1037, 2026.

Negar Maleki, Balaji Padmanabhan, and Kaushik Dutta. Ai hallucinations: a misnomer worth clarifying. In *2024 IEEE conference on artificial intelligence (CAI)*, pp. 133–138. IEEE, 2024.

Timur Mudarisov, Mikhail Burtsev, Tatiana Petrova, and Radu State. Limitations of normalization in attention mechanism. *arXiv preprint arXiv:2508.17821*, 2025.

Yasumasa Onoe, Michael JQ Zhang, Eunsol Choi, and Greg Durrett. Creak: A dataset for commonsense reasoning over entity knowledge. *arXiv preprint arXiv:2109.01653*, 2021.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pp. 248–260. PMLR, 2022.

Bibek Paudel, Alexander Lyzhov, Preetam Joshi, and Puneet Anand. Hallucinot: Hallucination detection through context and common knowledge verification. *arXiv preprint arXiv:2504.07069*, 2025.

Plato. *Apology*. BookRix, 2019.

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

Guillaume Sanchez, Honglu Fan, Alexander Spangher, Elad Levi, Pawan Sasanka Ammanamanchi, and Stella Biderman. Stay on topic with classifier-free guidance. *ArXiv*, 2023.

Murray Shanahan. Talking about large language models. *Communications of the ACM*, 67(2):68–79, 2024.

Anqi Shao. New sources of inaccuracy? a conceptual framework for studying ai hallucinations. *Harvard Kennedy School Misinformation Review*, 2025.

Artem Shelmanov, Ekaterina Fadeeva, Akim Tsvigun, Ivan Tsvigun, Zhuohan Xie, Igor Kiselev, Nico Daheim, Caiqi Zhang, Artem Vazhentsev, Mrinmaya Sachan, et al. A head to predict and a head to question: Pre-trained uncertainty quantification heads for hallucination detection in llm outputs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 35700–35719, 2025.

Vaishnavi Shrivastava, Ananya Kumar, and Percy Liang. Language models prefer what they know: Relative confidence estimation via confidence preferences. *arXiv preprint arXiv:2502.01126*, 2025.

Adi Simhi, Jonathan Herzig, Idan Szpektor, and Yonatan Belinkov. Constructing benchmarks and interventions for combating hallucinations in llms. *arXiv preprint arXiv:2404.09971*, 2024.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Peiqi Sui, Eamon Duede, Sophie Wu, and Richard So. Confabulation: The surprising value of large language model hallucinations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14274–14284, 2024.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 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)*, pp. 4149–4158, 2019.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Pranav Narayanan Venkit, Tatiana Chakravorti, Vipul Gupta, Heidi Biggs, Mukund Srinath, Koustava Goswami, Sarah Rajtmajer, and Shomir Wilson. " confidently nonsensical?": A critical survey on the perspectives and challenges of'hallucinations' in nlp. *CoRR*, 2024.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *Advances in Neural Information Processing Systems*, 38:115452–115486, 2026.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024a.

Yuxia Wang, Minghan Wang, Muhammad Arslan Manzoor, Fei Liu, Georgi Nenkov Georgiev, Rocktim Jyoti Das, and Preslav Nakov. Factuality of large language models: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 19519–19529, 2024b.

Menghua Wu, Cai Zhou, Stephen Bates, and Tommi Jaakkola. Thought calibration: Efficient and confident test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 14302–14316, 2025.

Yihao Xue, Kristjan Greenewald, Youssef Mroueh, and Baharan Mirzasoleiman. Verify when uncertain: Beyond self-consistency in black box hallucination detection. *arXiv preprint arXiv:2502.15845*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025b.

Zhen Yang, Mingyang Zhang, Feng Chen, Ganggui Ding, Liang Hou, Xin Tao, and Ying-Cong Chen. Less is more: Improving llm reasoning with minimal test-time intervention. *arXiv preprint arXiv:2510.13940*, 2025c.

Gilad Yehudai, Haim Kaplan, Guy Dar, Royi Rassin, Asma Ghandeharioun, Mor Geva, and Amir Globerson. When can transformers count to n? *arXiv preprint arXiv:2407.15160*, 2024.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Luan Zhang, Dandan Song, Zhijing Wu, Yuhang Tian, Changzhi Zhou, Jing Xu, Ziyi Yang, and Shuhao Zhang. Detecting hallucination in large language models through deep internal representation analysis. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pp. 8357–8365, 2025a.

Wan Zhang and Jing Zhang. Hallucination mitigation for retrieval-augmented large language models: a review. *Mathematics*, 13(5):856, 2025.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. Siren's song in the ai ocean: A survey on hallucination in large language models. *Computational Linguistics*, 51(4):1373–1418, 2025b.

Yubo Zhu, Dongrui Liu, Zecheng Lin, Wei Tong, Sheng Zhong, and Jing Shao. The llm already knows: Estimating llm-perceived question difficulty via hidden representations. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1160–1176, 2025.

Zayd MK Zuhri, Erland Hilman Fuadi, and Alham Fikri Aji. Predicting the order of upcoming tokens improves language modeling. *arXiv preprint arXiv:2508.19228*, 2025.

# APPENDIX

## A LIMITATIONS

While EHETI improves reasoning performance across models and benchmarks, it has several limitations. First, the intervention relies on token-level entropy to locate influential generation positions. Entropy provides an interpretable signal of local uncertainty, but it may not capture all decision-critical positions, especially when influence is distributed across several tokens or expressed through low-entropy but semantically decisive continuations.

Second, our analysis focuses on autoregressive reasoning. The structural-compression account and the proposed intervention are studied under standard left-to-right decoding, and their behaviour under speculative decoding, non-autoregressive generation, or tool-augmented reasoning remains open. The experiments cover general knowledge, programming, mathematical reasoning, scientific reasoning, and multi-hop question answering, but tasks requiring long-horizon planning, external tool use, or interactive decision making may exhibit different uncertainty structure.

Finally, the method targets inference-time expression failures. It can reduce premature commitment to erroneous reasoning trajectories, but it does not remove the underlying structural-compression constraint or change the model's representational capacity. Addressing these deeper constraints may require architectural changes or training-time objectives, which are beyond the scope of this work.

## B RELATED WORK

**Definition and Conceptualisation:** Generation errors in large language models have been described using overlapping terms such as hallucination, confabulation, fabrication, factuality error, faithfulness error, and grounding failure. Among these, *hallucination* is the most widely used, although its scope remains inconsistent across NLP and broader AI discourse (Venkit et al., 2024; Maleki et al., 2024). It generally refers to plausible yet unsupported, fabricated, or factually incorrect outputs (Huang et al., 2025b; Wang et al., 2024b), and has motivated taxonomies distinguishing factuality from faithfulness, intrinsic from extrinsic errors, and hallucination from related failures in grounding, calibration, and inference (Zhang et al., 2025b; Cossio, 2025; Alansari & Luqman, 2025). Similar surface errors can nevertheless arise from different conditions, including inconsistency with world knowledge, input context, or task constraints (Bang et al., 2025). Alternative terms such as *confabulation* emphasise that erroneous generations may remain coherent and plausible despite being unsupported (Farquhar et al., 2024; Sui et al., 2024). More broadly, several studies caution that psychological terms such as hallucination, belief, and knowing may encourage anthropomorphic interpretations of language models (Harnad, 2024; Shanahan, 2024), motivating greater precision in how such failures are described (Förster & Skop, 2025; Shao, 2025).

**Detection and Mitigation:** Methods for detecting and mitigating generation errors commonly rely on external verification, uncertainty estimation, consistency checking, and internal-state analysis. Evidence-grounded methods validate generated claims against retrieved documents, provided context, or external knowledge (Paudel et al., 2025; Zhang & Zhang, 2025), while uncertainty-based approaches estimate error risk from model confidence or specialised uncertainty signals (Shrivastava et al., 2025; Shelmanov et al., 2025). Consistency-based methods compare responses, models, or reasoning traces to identify unstable claims (Goel et al., 2025; Xue et al., 2025), whereas representation-based work shows that hidden states and layer-wise information patterns can reveal error-related signals before they appear in the final output (Zhang et al., 2025a; Kim et al., 2025). Mitigation methods further improve reliability through retrieval (Ding et al., 2025), confidence-guided allocation of inference-time computation (Huang et al., 2025a; Wu et al., 2025), and self-evaluation or multi-agent refinement (Cheng et al., 2025; Chen et al., 2025a; Mahmoud et al., 2026). These approaches largely act once an answer, uncertainty signal, or detectable risk pattern is already available.

# C Implementation Details

We use greedy decoding as the default generation strategy and require final answers to be enclosed in a box for automatic parsing. For generation-time probing, layer-specific one-hidden-layer MLP probes are trained with ReLU activation, with the hidden size selected from the candidate values $1, 2, 4, 8, 16, 32, 64, 128, 256, 512$, and $1024$. The probing data are split into $60\%/20\%/20\%$ train/validation/test subsets. Probes are trained for $400$ epochs with cross-entropy loss, and results are averaged over three random seeds. For output-attribute probing, prompt construction follows Table 3. For pre-generation correctness-direction probing, residual stream activations are extracted at the final input token before any answer token is generated. For layer-wise scans, activations are collected every 2 layers for models below 10B parameters and every $4$ layers for larger models. For structural-compression experiments, token-copying prompts follow Table 4, while aggregation and output-attractor prompts follow Table 5. For entropy-structure analysis, Qwen3-8B is run in thinking mode with decoding temperature $T_{\text{dec}} = 1.0$, and entropy statistics are computed over more than $10^6$ generated CoT tokens. For the semantic-role word clouds, token types with fewer than $100$ occurrences are filtered out before ranking by average entropy. The complete intervention settings and validation grid are provided in Appendix E.4. Experiments are run on an NVIDIA B200 cluster.

# D Prompt Design and Templates

## D.1 Templates for Output-Relevant Attribute Probing

Table 3: Prompt templates for output-relevant attribute probing. Each template instantiates the prompt distribution $p_a(x)$ for attribute $a$. The generated response $y$ is mapped to a regression target or classification label through the attribute rule $g_a(y)$.

| **Attribute** | **Prompt schema** | **Target** $g_a(y)$ |
|---|---|---|
| Response length | {data} | Generated token count, excluding special tokens. |
| Reasoning steps | Provide step-by-step solution, starting with `Step 1:`. Problem: {data} | Number of explicit step markers. |
| Character choice | Here's the first sentence of a story: {data}. Continue this story with one sentence that introduces a new animal character. | Animal character introduced in the continuation. |
| Multiple-choice answer | Before choosing your answer, briefly explain why in one short sentence. Then select from the options: {data} | Selected option among the valid choices. |
| Answer confidence | Please first write analysis of the problem, then select the single correct answer to the following question from the options, and state your option explicitly: {data} | Whether the selected option matches the gold answer. |
| Factual consistency | First discuss this statement briefly in one short sentence, then explicitly state that you are in favour of the statement or explicitly state if you feel unsure: {data} | Whether the model stance matches the statement truth label. |

This subsection specifies how the prompt distribution $p_a(x)$ is instantiated for the output-relevant attribute probing experiments, as summarised in Table 3. For each attribute $a$, we pair the corresponding dataset with a task-specific prompt template. The templates are designed to elicit responses whose structural properties, content choices, answer correctness, or factual stance can be mapped to attribute targets by $g_a(y)$. These templates are used in the multi-dataset probing analysis in Appendix E.1.3 and the layer-wise analysis in Appendix E.1.4.

## D.2 Templates for Structural-Compression Experiments

This subsection specifies the prompt templates used for the structural-compression experiments in Section 2.2 and Appendices E.2.1, E.2.2, E.2.3, and E.2.4. Table 4 presents the templates for the token-copying experiments, while Table 5 presents those used for the aggregation and output-attractor analyses. For all templates, {seq} denotes the generated input sequence, {target} de-

Table 4: Prompt templates for the token-copying experiments. The first-token and last-token settings differ only in the queried position, while the hinted and interleaved settings modify the prompt or sequence structure used for last-token copying.

| Experiment | Setting | Prompt template |
|---|---|---|
| First-token copying | Direct | Consider the following sequence: {seq}. What is the first digit in this sequence? Please answer exactly as "The answer to your question is: <ANSWER>". |
| Last-token copying | Direct | Consider the following sequence: {seq}. What is the last digit in this sequence? Please answer exactly as "The answer to your question is: <ANSWER>". |
| Last-token copying with hint | Hinted Direct | Consider the following sequence: {seq}. What is the last digit in this sequence? Hint: it is not necessarily 1; check carefully. Please answer exactly as "The answer to your question is: <ANSWER>". |
| Interleaved binary copying | Direct | Consider the following sequence: {seq}. What is the last digit in this sequence? Please answer exactly as "The answer to your question is: <ANSWER>". |

Table 5: Prompt templates for aggregation experiments. The Direct regime requests only the final answer in a fixed format. Zero-shot CoT asks the model to decompose the computation before producing the final answer. Few-shot CoT additionally provides in-context examples illustrating grouping and counting operations.

| Task | Prompting regime | Prompt template |
|---|---|---|
| Summation | Direct | Please perform the following sum: {seq}. Please give the answer on the final line exactly as "The final answer to your maths question is: <ANSWER>". |
| Homogeneous counting | Direct | Please count the number of ones in the following sequence: {seq}. Please give the answer on the final line exactly as "The final answer to your maths question is: <ANSWER>". |
| Interleaved binary counting | Direct | Please count the number of ones in the following sequence: {seq}. Please give the answer on the final line exactly as "The final answer to your maths question is: <ANSWER>". |
| Word counting | Direct | Please count the number of times {target} appears in the following sentence: "{seq}". Please give the answer on the final line exactly as "The final answer to your maths question is: <ANSWER>". |
| All aggregation tasks | Zero-shot CoT | Let's think step by step and break down the computation before giving the final answer. {task prompt} Please give the answer on the final line exactly as "The final answer to your maths question is: <ANSWER>". |
| All aggregation tasks | Few-shot CoT | Let's think step by step and break down the computation before giving the final answer. Here are a few examples: {few-shot examples}. Now solve the following problem: {task prompt} Please give the answer on the final line exactly as "The final answer to your maths question is: <ANSWER>". |

notes the queried word when applicable, and <ANSWER> denotes the parsed numerical or symbolic answer.

For Few-shot CoT, the examples are matched to the target task type. In summation, examples show grouping repeated ones into small blocks before computing the total. In homogeneous and interleaved binary counting, examples show grouping the sequence and counting only positions containing 1. In word counting, examples show scanning the sentence and accumulating occurrences of {target}. The output-attractor distributions in Appendix E.2.3 and Appendix E.2.4, shown in Figure 11 and Figure 12, are computed from the parsed final answers generated under the same aggregation templates.

# E Detailed Methodological Analyses

This appendix provides the detailed analyses supporting the methodology in Section 2. Appendix E.1 expands the internal-signal analysis. Appendix E.2 provides the structural-compression analyses. Appendix E.3 details the entropy-structure analysis. Appendix E.4 provides the complete intervention formulation.

## E.1 Pre-Generation Internal Signal Analysis

### E.1.1 Pre-Generation Correctness-Direction Probing

Section 2.1 tests whether successful and failure-prone generations are already separable in hidden representations before decoding. As shown in the left panel of Figure 2, probing experiments on TriviaQA (Joshi et al., 2017) show that pre-generation representations contain signals predictive of later answer failure before any output token is produced. The left panel reports AUROC from pre-generation representations $\mathbf{h}_{0,\ell}$ across normalised layer indices $\ell/L$. These signals give internal judgements a measurable representation-level form: the model already encodes information associated with later success or failure before the answer is realised in text.

We use pre-generation correctness-direction probing to measure how separability develops across network depth before any output token is generated. For each layer $\ell$, we split the dataset into correctly answered instances $\mathcal{D}^+$ and incorrectly answered instances $\mathcal{D}^-$. We then compute the corresponding centroids $\boldsymbol{\mu}_\ell^+$ and $\boldsymbol{\mu}_\ell^-$ in the pre-generation representation space. The layer-wise correctness direction is

$$\mathbf{d}_\ell = \boldsymbol{\mu}_\ell^+ - \boldsymbol{\mu}_\ell^-. \tag{8}$$

For instance $i$, we score its pre-generation representation $\mathbf{h}_{0,\ell}^{(i)}$ by projecting it onto this direction:

$$s_{i,\ell} = \frac{(\mathbf{h}_{0,\ell}^{(i)} - \boldsymbol{\mu}_\ell)^\top \mathbf{d}_\ell}{\|\mathbf{d}_\ell\|}, \qquad \boldsymbol{\mu}_\ell = \frac{1}{2}(\boldsymbol{\mu}_\ell^+ + \boldsymbol{\mu}_\ell^-). \tag{9}$$

Here, $s_{i,\ell}$ is the correctness-direction score, and $\boldsymbol{\mu}_\ell$ is the midpoint between the correct and incorrect centroids at layer $\ell$. Layer-wise separability is measured by

$$\mathrm{AUROC}_\ell = \mathrm{AUROC}(\{s_{i,\ell}\}_{i=1}^N, \{y_i\}_{i=1}^N), \tag{10}$$

where $N$ is the number of evaluated instances and $y_i \in \{0, 1\}$ denotes the final-answer correctness label for instance $i$. Intermediate layers progressively separate error-prone instances from correct ones before decoding begins, indicating that correctness-related structure is already organised during internal computation.

This probing analysis establishes the representation-level basis for the first stage. Later failures are associated with measurable differences in hidden states before generation, indicating that error-predictive structure is already present before it becomes visible in the final answer.

### E.1.2 Generation-Time Evolution of Error-Predictive Signals

Section 2.1 shows that pre-generation representations already contain signals predictive of later answer failure. This appendix tracks how the same type of error-predictive signal evolves after decoding begins. As shown in the right panel of Figure 2, generation-time probing on TriviaQA (Joshi et al., 2017) reveals that correctness-related information remains accessible during generation, while its visibility changes across decoding positions. The right panel reports binary F1 from generation-time hidden states $\mathbf{h}_{t,\ell}$ across normalised decoding positions $t$ (%).

Let $\mathbf{h}_{t,\ell} \in \mathbb{R}^d$ denote the hidden state at decoding position $t$ and layer $\ell$, where $d$ is the hidden dimension and $t = 0$ denotes the pre-generation state after the input has been processed. For responses of different lengths, decoding positions are mapped to a normalised generation timeline and evaluated at equidistant relative positions. At each normalised decoding position, we apply a layer-specific MLP probe $q_\theta^{(\ell)}$ to $\mathbf{h}_{t,\ell}$, where $\theta$ denotes the probe parameters, and predict the final-answer correctness label. Probe performance is evaluated using binary F1 score.

High F1 at early positions indicates that the model state already contains information about whether the forthcoming answer is likely to succeed or fail. The signal is relatively strong near the start of generation ($t \approx 0$), weakens during intermediate decoding, and re-emerges near the end. This non-monotonic trajectory suggests that error-predictive information is not simply absent during generation. Its accessibility varies as the autoregressive trajectory unfolds, consistent with the view that internal judgements can be present while their influence on the realised answer becomes unstable.

### E.1.3 Multi-Dataset Predictability of Output-Relevant Internal Signals

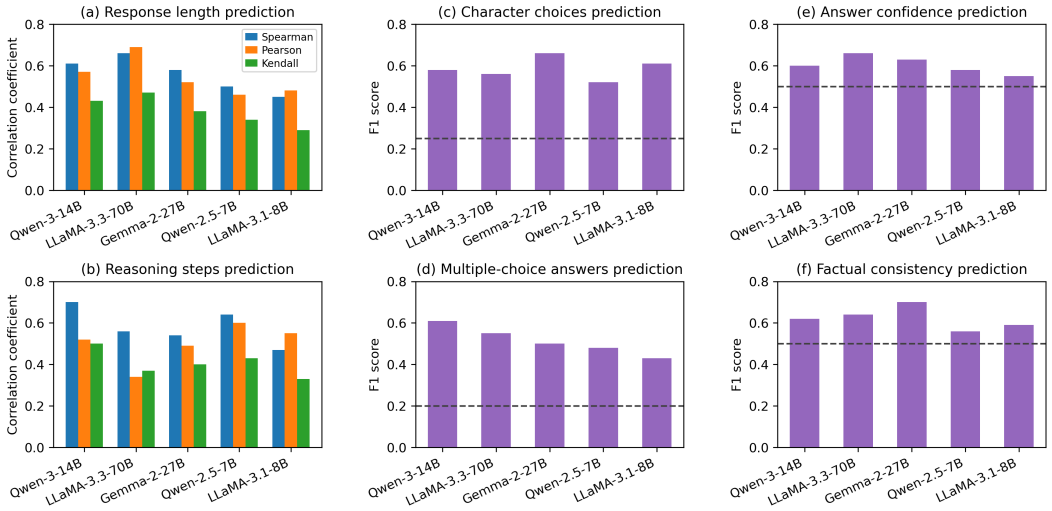


Figure 8: Predictability of planning-related output attributes from pre-generation hidden states. Each panel corresponds to one attribute prediction task. The horizontal axis lists evaluated LLMs, and the vertical axis reports probe performance, using correlation coefficients for continuous attributes and F1 score for discrete attributes. Dashed horizontal lines in the discrete-attribute panels indicate random-baseline F1 scores.

Figure 8 extends the pre-generation probing analysis in Section 2.1 beyond final-answer correctness. The main analysis shows that failure-prone instances can already be separated from successful ones before decoding begins. Here, we ask whether the same pre-generation representations also encode broader properties of the forthcoming response. To make this question concrete, we probe a set of planning-related output attributes, including response structure, content choices, answer confidence, and factual consistency. These attributes provide observable targets for testing whether internal judgements are reflected in model states beyond a binary correctness label.

The probing experiments cover different model families and parameter scales, including Qwen2.5-7B (Yang et al., 2025b), Llama-3.1-8B (Grattafiori et al., 2024), Qwen3-14B (Yang et al., 2025a), Gemma-2-27B (Team et al., 2024), and Llama-3.3-70B[1].

We formulate each attribute prediction problem as $T_a = (p_a(x), g_a(y))$, where $a$ indexes the analysed attribute, $p_a(x)$ is the prompt distribution, and $g_a(y)$ maps a generated response $y$ to its target attribute. For each attribute, $p_a(x)$ is instantiated using the task-specific prompt schema in Appendix D.1, paired with the corresponding dataset. The analysed attributes include continuous structural properties, namely response length and reasoning steps, and discrete decision- or behaviour-related properties, namely character choice, multiple-choice answer, answer confidence, and factual consistency. We evaluate these attributes on their corresponding datasets: response length on UltraChat (Ding et al., 2023), reasoning steps on GSM8K (Cobbe et al., 2021), character choice on TinyStories (Eldan & Li, 2023), multiple-choice answering on CommonsenseQA (Talmor et al., 2019), answer confidence on MedMCQA (Pal et al., 2022), and factual consistency on CREAK (Onoe et al., 2021).

[1] https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct

For each attribute $a$, we sample prompts $x_i \sim p_a(x)$, where $i$ indexes an instance, and generate responses $y_i$ using greedy decoding. The target attribute is $z_i^{(a)} = g_a(y_i)$, where $z_i^{(a)}$ denotes the target value or label for instance $i$ under attribute $a$. For response length, $g_a(y_i)$ counts generated tokens; for reasoning steps, it counts explicit step markers; for character choice and multiple-choice answering, it extracts the selected category or option; for answer confidence, it records whether the selected option matches the gold label; and for factual consistency, it records whether the model's stated stance matches the truth label of the statement.

For each prompt $x_i$, we extract the pre-generation hidden state $\mathbf{h}_\ell^{(i)} \in \mathbb{R}^d$ at layer $\ell$, where $d$ is the hidden dimension and $\ell \in \{1, \dots, L\}$ indexes the $L$ transformer layers. For attribute $a$ and layer $\ell$, the probing dataset is $\mathcal{D}_{a,\ell} = \{(\mathbf{h}_\ell^{(i)}, z_i^{(a)})\}_{i=1}^{N_a}$, where $N_a$ is the number of examples for attribute $a$.

The MLP probe takes $\mathbf{h}_\ell^{(i)}$ as input. For each attribute $a$, layer $\ell$, and MLP hidden dimension $r$, the probe is defined as

$$q_{a,\ell,r}(\mathbf{h}) = \mathbf{W}_2 \,\mathrm{ReLU}(\mathbf{W}_1 \mathbf{h} + \mathbf{b}_1) + \mathbf{b}_2. \tag{11}$$

Here, $q_{a,\ell,r}$ denotes the probe for the corresponding attribute, layer, and hidden dimension. The parameters $\mathbf{W}_1 \in \mathbb{R}^{r \times d}$ and $\mathbf{b}_1 \in \mathbb{R}^r$ map the model representation into the MLP hidden layer. The output parameters are $\mathbf{W}_2$ and $\mathbf{b}_2$, whose shapes depend on the prediction type. For continuous attributes, $\mathbf{W}_2 \in \mathbb{R}^{1 \times r}$ and $\mathbf{b}_2 \in \mathbb{R}$, so the probe outputs a scalar. For discrete attributes with $K_a$ classes, $\mathbf{W}_2 \in \mathbb{R}^{K_a \times r}$ and $\mathbf{b}_2 \in \mathbb{R}^{K_a}$, so the probe outputs class logits.

We select $r$ from $\mathcal{R} = \{1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024\}$. For each attribute, the probing data are split into training, validation, and test subsets using a 60:20:20 split. Each probe is trained for 400 epochs. We search over both the transformer layer $\ell$ and the MLP hidden dimension $r$, select the configuration with the best validation performance, and report held-out test performance averaged over three random seeds.

For continuous attributes $a \in \{\textsc{Length}, \textsc{Steps}\}$, the probe prediction is $\hat{z}_i^{(a)} = q_{a,\ell,r}(\mathbf{h}_\ell^{(i)})$, and the training objective is mean squared error:

$$\mathcal{L}_{\mathrm{reg}} = \frac{1}{N_a} \sum_{i=1}^{N_a} (\hat{z}_i^{(a)} - z_i^{(a)})^2. \tag{12}$$

For discrete attributes, the predicted class distribution is $\hat{\mathbf{p}}_i^{(a)} = \mathrm{softmax}(q_{a,\ell,r}(\mathbf{h}_\ell^{(i)}))$. With class label $c_i^{(a)} \in \{1, \dots, K_a\}$, the probe is trained with cross-entropy:

$$\mathcal{L}_{\mathrm{cls}} = -\frac{1}{N_a} \sum_{i=1}^{N_a} \log \hat{\mathbf{p}}_{i,c_i^{(a)}}^{(a)}. \tag{13}$$

For continuous attributes, performance is evaluated using Pearson, Spearman, and Kendall correlation coefficients. Let $\hat{\mathbf{z}}$ denote the vector of predictions, $\mathbf{z}$ denote the vector of targets, and $\bar{\hat{z}}$ and $\bar{z}$ denote their sample means. Pearson correlation measures linear agreement:

$$\rho_{\mathrm{Pearson}} = \frac{\mathrm{cov}(\hat{\mathbf{z}}, \mathbf{z})}{\sigma_{\hat{z}} \sigma_z}. \tag{14}$$

Here, $\mathrm{cov}(\hat{\mathbf{z}}, \mathbf{z})$ is the sample covariance, and $\sigma_{\hat{z}}$ and $\sigma_z$ are the sample standard deviations of predictions and targets. Spearman correlation is computed as $\rho_{\mathrm{Spearman}} = \rho_{\mathrm{Pearson}}(\mathrm{rank}(\hat{\mathbf{z}}), \mathrm{rank}(\mathbf{z}))$, measuring rank-level consistency. Kendall's $\tau$ measures ordinal agreement from pairwise comparisons:

$$\tau = \frac{n_c - n_d}{\frac{1}{2} N(N-1)}. \tag{15}$$

Here, $N$ is the number of evaluated examples, while $n_c$ and $n_d$ denote the numbers of concordant and discordant prediction-target pairs. These three metrics jointly test whether pre-generation representations encode approximate attribute values and preserve the relative ordering of instances.

For discrete attributes, performance is evaluated using F1 score. Let TP, FP, and FN denote true positives, false positives, and false negatives for a class. Precision and recall are $P = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$ and $R = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$. The F1 score is

$$\mathrm{F1} = \frac{2PR}{P + R}. \tag{16}$$

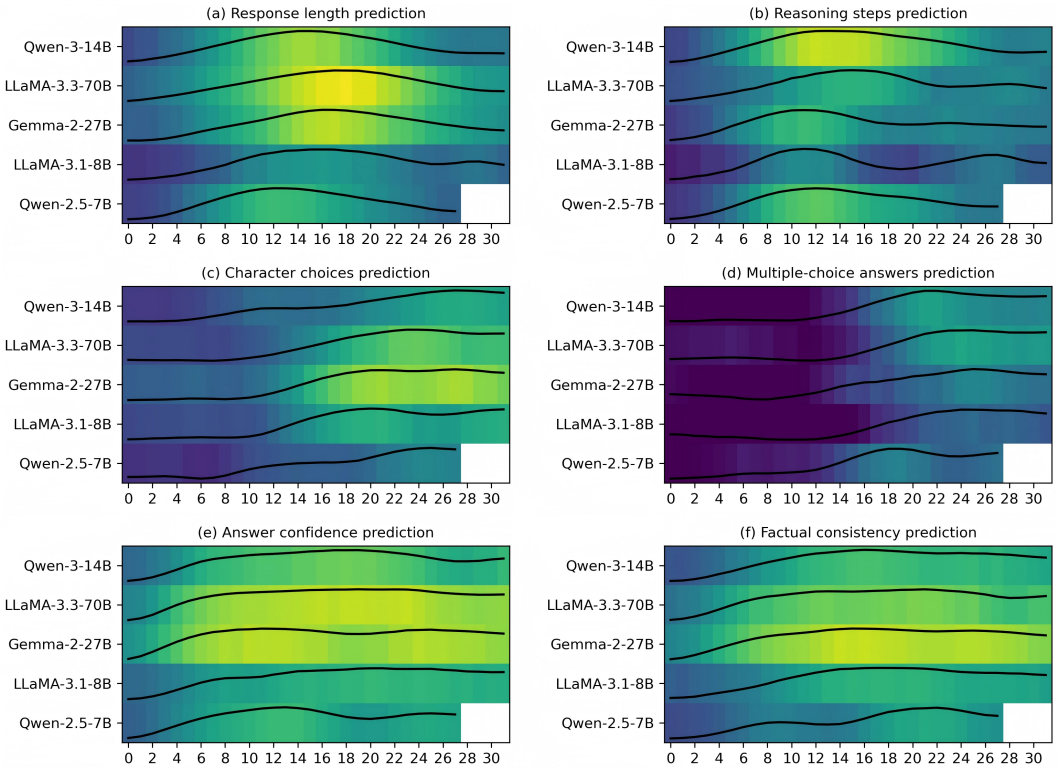


Figure 9: Layer-wise predictability of planning-related output attributes. Each subplot corresponds to one attribute prediction task. The horizontal axis shows transformer layers from early to late, and the vertical axis lists models. Heatmap colours indicate absolute probe performance, using Spearman correlation for response length and reasoning steps and F1 score for discrete attributes, with lighter colours denoting higher predictability. Black curves show row-wise normalised layer-wise trajectories, highlighting relative depth-wise trends within each model.

For multi-class attributes, we report the class-averaged F1 score over the valid label set.

Across the six attributes, pre-generation hidden states reliably predict planning-related properties of later responses. Response length and reasoning steps exhibit consistently positive Pearson, Spearman, and Kendall correlations, showing that structural properties of the forthcoming answer are already partially organised before decoding. Character choice, multiple-choice answer, answer confidence, and factual consistency also achieve F1 scores above random baselines, indicating that pre-generation states contain information about future content choices and behaviour-related outcomes. The strength of these signals varies across both attributes and models, with structural properties, confidence, and factual consistency generally more recoverable than decision-intensive attributes such as multiple-choice answering. Within our framework, these results extend the main error-separability analysis from correctness prediction to a broader set of future-response properties. The multi-dataset evidence shows that internal judgements can be expressed as structured, output-relevant information in the pre-generation state before the final answer is realised.

### E.1.4 Layer-wise Organisation of Output-Relevant Internal Signals

Building on the multi-dataset probing analysis in Appendix E.1.3, Figure 9 examines how the same planning-related output attributes are organised across transformer depth. The previous analysis asks whether pre-generation representations encode properties of the forthcoming response. Here, we retain the same prompt templates, datasets, target attributes, generated responses, and probing targets, and change the unit of analysis from the best pre-generation representation to the full layer-wise trajectory. This allows us to examine where the measurable forms of internal judgement, reflected in output-relevant hidden-state signals, become most detectable inside the network. For consistency

across models, we evaluate the first $L_{\text{eval}} = 32$ transformer layers in all layer-wise plots, except for Qwen2.5-7B, which has 28 transformer layers in total. For each model $m$, attribute $a$, and transformer layer $\ell$, we extract the pre-generation hidden state $\mathbf{h}_{\ell}^{(i,m)} \in \mathbb{R}^{d_m}$ for instance $i$, where $d_m$ is the hidden dimension of model $m$. The layer index satisfies $\ell \in \{1, \ldots, L_{\text{eval}}\}$. The probing target for instance $i$ is denoted by $z_i^{(a)}$, where continuous attributes use scalar targets and discrete attributes use class labels.

At each layer $\ell$, we train a separate one-hidden-layer MLP probe on the corresponding pre-generation representations. For an MLP hidden dimension $r$, the probe is defined as

$$q_{m,a,\ell,r}(\mathbf{h}) = \mathbf{W}_2 \mathrm{ReLU}(\mathbf{W}_1 \mathbf{h} + \mathbf{b}_1) + \mathbf{b}_2. \tag{17}$$

Here, $q_{m,a,\ell,r}$ denotes the probe for model $m$, attribute $a$, layer $\ell$, and hidden dimension $r$. The parameters $\mathbf{W}_1 \in \mathbb{R}^{r \times d_m}$ and $\mathbf{b}_1 \in \mathbb{R}^r$ map the model representation into the probe hidden layer. The output parameters $\mathbf{W}_2$ and $\mathbf{b}_2$ depend on the prediction type. For continuous attributes, $\mathbf{W}_2 \in \mathbb{R}^{1 \times r}$ and $\mathbf{b}_2 \in \mathbb{R}$, so the probe outputs a scalar. For a discrete attribute with $K_a$ classes, where $K_a$ is the number of valid labels for attribute $a$, $\mathbf{W}_2 \in \mathbb{R}^{K_a \times r}$ and $\mathbf{b}_2 \in \mathbb{R}^{K_a}$, so the probe outputs $K_a$ logits.

The MLP hidden dimension is selected from $\mathcal{R} = \{1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024\}$. For each $(m, a, \ell)$, we train probes under all candidate values $r \in \mathcal{R}$. The probing data are split into training, validation, and test subsets using a 60:20:20 split. Each probe is trained for 400 epochs. Continuous attributes are trained with mean squared error, and discrete attributes are trained with cross-entropy. The hidden dimension is chosen by validation performance:

$$r^{\star}_{m,a,\ell} = \arg\max_{r \in \mathcal{R}} \mathrm{Val}_a(q_{m,a,\ell,r}). \tag{18}$$

Here, $\mathrm{Val}_a(\cdot)$ denotes the validation metric for attribute $a$, using Spearman correlation for continuous attributes and F1 score for discrete attributes. The corresponding held-out score is then recorded as

$$S_{m,a,\ell} = \mathrm{Test}_a\left(q_{m,a,\ell,r^{\star}_{m,a,\ell}}\right). \tag{19}$$

The score $S_{m,a,\ell}$ is the absolute layer-wise predictability shown by the heatmap, with lighter colours indicating higher predictability. For response length and reasoning steps, $S_{m,a,\ell}$ is Spearman correlation. For character choice, multiple-choice answer, answer confidence, and factual consistency, $S_{m,a,\ell}$ is F1 score. Results are averaged over three random seeds.

To visualise relative depth-wise trends independently of absolute model strength, we compute a row-wise normalised trajectory for each model and attribute:

$$\widetilde{S}_{m,a,\ell} = \frac{S_{m,a,\ell} - \min_{\ell'} S_{m,a,\ell'}}{\max_{\ell'} S_{m,a,\ell'} - \min_{\ell'} S_{m,a,\ell'} + \epsilon}. \tag{20}$$

Here, $\widetilde{S}_{m,a,\ell} \in [0, 1]$ is the normalised score at layer $\ell$, $\ell'$ ranges over the evaluated layers, and $\epsilon = 10^{-8}$ is used only for numerical stability when the layer-wise scores have near-zero range. The black curves plot these normalised trajectories within each model row. They therefore indicate where an attribute is relatively strongest across depth, while the heatmap colours preserve absolute probe performance.

The layer-wise trajectories reveal distinct organisations across attribute types. Structural attributes, including response length and reasoning steps, tend to strengthen from early layers and peak in intermediate layers, suggesting that coarse response structure is formed during intermediate computation before final decoding states are reached. Content attributes, including character choice and multiple-choice answer, show stronger later-layer concentration, consistent with their dependence on semantic integration and answer selection. Behavioural attributes, including answer confidence and factual consistency, become detectable early and remain comparatively stable across later layers, indicating that correctness- and consistency-related signals are available before final output formation. These patterns extend the first-stage evidence from separability to depth-wise organisation: output-relevant internal signals, as measurable forms of internal judgement, are already distributed across model depth before the final response is realised.

### E.2 Structural Compression and Output Attractors

#### E.2.1 Causal Path Analysis for Token Copying

Using GPT-5.2 (Singh et al., 2025), we analyse simple copying settings in which the relevant information is explicitly present in the prompt, but must be preserved through the causal computation graph before it can affect the final prediction. Figure 3 examines copying in binary sequences where the correct answer is determined by a single target token. The panels compare first-token and last-token copying, last-token copying with and without an explicit hint, and last-token copying in interleaved binary sequences. The horizontal axis shows sequence length, and the vertical axis indicates whether the copied token is correct. Let $x = (x_1, \ldots, x_n)$ denote an input sequence of length $n$, where $x_i$ is the token at position $i$. We compare two cases: copying the first token $x_1$ from a sequence of the form $(0, 1, \ldots, 1)$, and copying the last token $x_n$ from a sequence of the form $(1, \ldots, 1, 0)$. In both cases, the target token is 0, so the task differs only in the position from which the relevant information must be recovered. The model reliably copies the first token across increasing sequence lengths, while accuracy for the last token drops sharply once the sequence becomes moderately long. The remaining panels further show that explicit hints and interleaved binary patterns improve last-token copying, but the improvement is partial: the same length-dependent instability reappears when the sequence becomes longer.

This asymmetry reflects how information propagates in a decoder-only Transformer. Let $v_i^{(0)} \in \mathbb{R}^d$ denote the input-layer representation of token position $i$, where $d$ is the hidden dimension. Let $v_i^{(\ell)}$ denote the representation at layer $\ell$, where $\ell \in \{0, \ldots, L\}$, and $L$ is the total number of decoder layers. The final representation used for next-token prediction is $y_n = \mathrm{norm}_3(v_n^{(L)})$, where $\mathrm{norm}_3(\cdot)$ is the final normalisation applied after the last Transformer block. To quantify how strongly position $i$ can affect the final prediction state, we analyse the sensitivity $\partial y_n / \partial v_i^{(0)}$.

To make the sensitivity analysis tractable, we summarise layer-normalisation effects with abstract constant factors and treat the forward-pass attention coefficients as fixed with respect to the derivative. This does not assign numerical values to the attention weights; it keeps the weights obtained in the forward pass unchanged while bounding how information propagates along the causal attention graph. Let $\alpha_{j,i}^{(\ell)}$ denote the attention coefficient from source position $i$ to target position $j$ at layer $\ell$. We write $\bar{\alpha}_{j,i}^{(\ell)}$ for the corresponding normalised coefficient after incorporating the residual path and layer-normalisation factor.

Let $\mathcal{P}_{i \to n}^{(L)}$ denote the set of causal paths from input position $i$ to final position $n$ across $L$ layers:

$$\mathcal{P}_{i \to n}^{(L)} = \{(\pi_0, \ldots, \pi_L) \mid \pi_0 = i,\ \pi_L = n, \\ \pi_{\ell+1} \geq \pi_\ell,\ \ell = 0, \ldots, L-1\}. \tag{21}$$

Here, $\pi_0, \ldots, \pi_L$ are intermediate token positions along a causal path. The condition $\pi_{\ell+1} \geq \pi_\ell$ follows from the causal attention mask, since information can only flow from earlier positions to the same or later positions.

The total attention-weighted path mass from position $i$ to position $n$ is

$$\Gamma_{i \to n}^{(L)} = \sum_{(\pi_0, \ldots, \pi_L) \in \mathcal{P}_{i \to n}^{(L)}} \prod_{\ell=0}^{L-1} \bar{\alpha}_{\pi_{\ell+1}, \pi_\ell}^{(\ell)}. \tag{22}$$

Under the assumptions above, the sensitivity of $y_n$ to $v_i^{(0)}$ satisfies

$$\left\| \frac{\partial y_n}{\partial v_i^{(0)}} \right\| \leq C\, \Gamma_{i \to n}^{(L)}. \tag{23}$$

Here, $C > 0$ collects layer-wise Lipschitz and normalisation constants, and $\|\cdot\|$ denotes a compatible vector norm. Eq. (23) shows that sensitivity depends on the total weight of causal paths linking an input position to the final prediction state. Earlier positions admit many possible paths through the triangular causal graph, while a late-position target has fewer opportunities to affect $y_n$. The copying results therefore expose a structural bottleneck in decoding: the relevant token may be present in the prompt, yet its influence on the final prediction can become weak when the available causal paths are sparse or low weight.

### E.2.2 AGGREGATION FAILURE

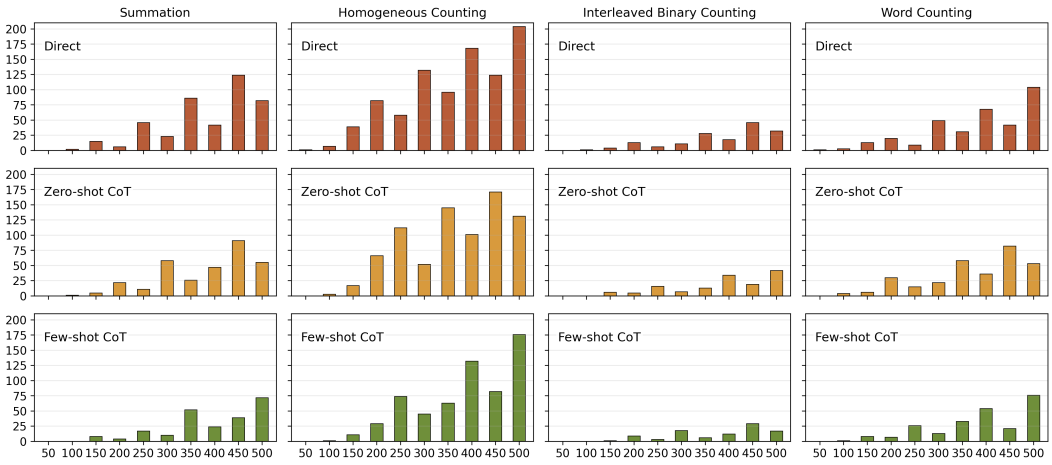


Figure 10: Failure of aggregation under increasing sequence length. Columns correspond to summation, homogeneous counting, interleaved binary counting, and word counting. Rows correspond to Direct answering, Zero-shot CoT prompting, and Few-shot CoT prompting. The horizontal axis shows sequence length, and the vertical axis reports absolute error between the parsed model prediction and the gold answer.

Figure 10 extends the structural-compression analysis from single-token copying to aggregation tasks. Copying isolates whether information at one position can reach the final prediction state. Aggregation requires a stronger operation: information from many positions must be accumulated into a single numerical answer. We evaluate four tasks following this principle: summation, homogeneous counting, interleaved binary counting, and word counting.

Concretely, the summation task asks the model to compute an expression of the form $1 + \cdots + 1$. Homogeneous counting asks the model to count the number of $1$ tokens in a sequence containing only 1s. Interleaved binary counting asks the model to count the number of $1$ tokens in an alternating binary sequence. Word counting asks the model to count the number of occurrences of a specified target word in a generated sentence. These tasks differ in surface form, but all require the model to preserve count-relevant information across many input positions and return a single numerical answer.

We use three prompting regimes, with the corresponding templates reported in Table 5. In the Direct regime, the model is instructed to output only the final answer in a fixed answer format. In the Zero-shot CoT regime, where CoT denotes chain-of-thought prompting, the model is asked to break down the computation before giving the final answer. In the Few-shot CoT regime, the prompt additionally includes in-context examples showing how to group and count elements before producing the final answer.

Let $n$ denote the sequence length and let $\hat{y}_n$ denote the model prediction extracted from the generated answer. The gold answer is denoted by $y_n^*$. For summation and homogeneous counting, the correct answer grows linearly with $n$. For binary counting, if $x_i \in \{0, 1\}$ is the token at position $i$, the correct answer is

$$y_n^* = \sum_{i=1}^{n} \mathbb{I}[x_i = 1]. \tag{24}$$

Here, $\mathbb{I}[\cdot]$ is the indicator function, which equals $1$ when the condition is true and $0$ otherwise. In homogeneous counting, $x_i = 1$ for all $i$, so $y_n^* = n$. In interleaved binary counting, the sequence follows an alternating binary pattern, and the target remains the exact number of positions with $x_i = 1$. For word counting, $x_i$ indexes a word occurrence in the sentence, and the same indicator form counts occurrences of the target word.

For each generated response, we parse the final numerical answer and compute the absolute error

$$e_n = |\hat{y}_n - y_n^*|. \tag{25}$$

Here, $e_n$ is the absolute deviation between the parsed model prediction and the gold answer at sequence length $n$. The results show a broad degradation trend as $n$ increases, while the trajectories remain locally non-monotonic. Local drops and jumps appear across tasks and prompting regimes, reflecting cases where model outputs fall near preferred numerical modes, intermediate reasoning temporarily stabilises the count, or CoT introduces additional grouping and reasoning drift. Homogeneous counting exhibits the largest errors because repeated inputs are especially vulnerable to compression into a small set of numerical outputs. Interleaved binary counting remains more stable, suggesting that alternating structure helps preserve the relevant signal. Word counting shows intermediate behaviour, with additional irregularity from word-boundary and parsing ambiguity. Zero-shot CoT and Few-shot CoT often reduce error, especially at shorter and mid-range lengths, while their advantages are setting-specific. Direct answering can remain stronger at some lengths, and Few-shot CoT does not uniformly dominate Zero-shot CoT.

The output-attractor analysis characterises how these aggregation errors appear in the generated answers. Let $\mathcal{V}$ be the set of numerical values appearing as parsed model outputs, and let $N_{\text{out}}$ be the number of evaluated generations. The empirical frequency of a predicted value $v \in \mathcal{V}$ is

$$f(v) = \sum_{j=1}^{N_{\text{out}}} \mathbb{I}[\hat{y}^{(j)} = v]. \tag{26}$$

Here, $\hat{y}^{(j)}$ is the parsed prediction for the $j$-th generated response. A mechanically counting model would produce values that follow the true targets across sequence lengths. The empirical frequency distribution instead concentrates around a small number of irregular preferred values, indicating that many distinct inputs are mapped to repeated or neighbouring numerical outputs.

A representation-based explanation follows from the same representational-collapse effect discussed in Section 2.2. Let $r(x) = \text{norm}_3(v_n^{(L)})$ denote the final-token representation for input sequence $x$, where $v_n^{(L)}$ is the last-token hidden state after $L$ decoder layers. Let $x^*$ be a related sequence that differs from $x$ by one additional repetition of the final token, and let $r(x^*) = \text{norm}_3(v_{n+1}^{*(L)})$. For a small $\varepsilon > 0$, if $\|r(x) - r(x^*)\|_1 < \varepsilon$, their unnormalised logits $\ell(x) = Wr(x) + b$ and $\ell(x^*) = Wr(x^*) + b$ satisfy

$$\begin{aligned} \|\ell(x) - \ell(x^*)\|_\infty &= \|W(r(x) - r(x^*))\|_\infty \\ &\leq \|W\|_{\infty,1} \|r(x) - r(x^*)\|_1. \end{aligned} \tag{27}$$

Here, $W$ and $b$ are the output projection matrix and bias, $\|\cdot\|_1$ is the vector $L_1$ norm, $\|\cdot\|_\infty$ is the maximum absolute coordinate, and $\|W\|_{\infty,1}$ is the induced matrix norm mapping $L_1$ representation differences to $L_\infty$ logit differences. When the right-hand side becomes small, two inputs requiring different numerical answers induce nearly indistinguishable next-token distributions. This makes exact aggregation increasingly fragile under finite precision.

The aggregation experiments support the same structural-compression account as the copying experiment. As sequence length grows, the model must preserve more position-specific information while still producing the answer from a compressed terminal state. CoT creates intermediate reasoning tokens that can stabilise some shorter and mid-length cases, but it can also introduce grouping errors or reasoning drift, and the final answer remains constrained by how much count-relevant information survives into the prediction state.

### E.2.3 Mechanistic Interpretation of Output Attractors

**Output Attractors.** Figure 11 analyses the distribution of parsed final answers from the aggregation tasks by counting how often each numerical value is produced under each prompting regime. The Gold distribution gives the reference pattern of true counts required by the task, while the remaining distributions show where model predictions actually fall. Under Direct answering, predictions concentrate around a small number of irregular preferred values, forming pronounced output attractors. Zero-shot CoT prompting and Few-shot CoT prompting spread the predictions more widely and preserve more mass near the Gold targets, especially around lower and mid-range target values. Nevertheless, both CoT regimes still show residual concentration around dominant numerical modes at higher target values. Appendix E.2.4 provides the corresponding per-regime distributions, separating the Gold, Direct, Zero-shot CoT, and Few-shot CoT outputs for clearer comparison.

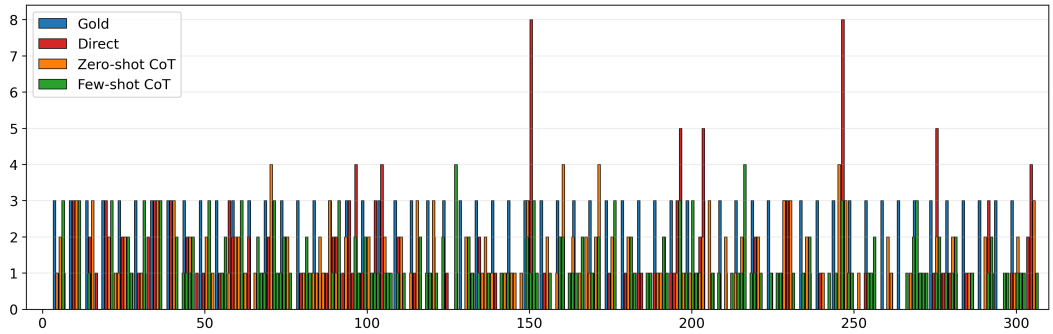


Figure 11: Output attractors in long-sequence aggregation. The distribution of parsed numerical outputs is compared against the Gold reference distribution under Direct answering, Zero-shot CoT prompting, and Few-shot CoT prompting. The horizontal axis shows answer value, and the vertical axis shows frequency.

This behaviour reflects a representational-collapse effect (Arefin et al., 2025), where distinct long inputs are mapped to increasingly similar final-token representations. In aggregation, this effect is further shaped by softmax normalisation (Yehudai et al., 2024), since attention updates preserve relative proportions such as $n_1/n_0$ more directly than absolute counts, making scale-sensitive numerical answers difficult to maintain (Mudarisov et al., 2025). A simplified two-token setting illustrates this ratio dependence. Let $n_0$ and $n_1$ denote the numbers of tokens of type 0 and 1. Let $q_0$ be the query associated with a token of type 0, $k_0$ and $k_1$ the keys for token types 0 and 1, and $v_0$ and $v_1$ their value vectors. Define

$$\begin{aligned} \eta_0 &= \exp(q_0^\top k_0), \\ \eta_1 &= \frac{n_1}{n_0} \exp(q_0^\top k_1). \end{aligned} \tag{28}$$

To isolate the effect of softmax normalisation, we omit the MLP and normalisation terms in this simplified update. The first-layer attention update for a token of type 0, denoted $z_0^{(1)}$, is

$$z_0^{(1)} = \frac{\eta_0}{\eta_0 + \eta_1} v_0 + \frac{\eta_1}{\eta_0 + \eta_1} v_1 + v_0. \tag{29}$$

The update depends on the ratio $n_1/n_0$, with absolute count absent from the update. This ratio dependence makes it difficult to preserve the magnitude information required for counting, especially when different sequences share similar proportions but require different numerical answers. In long contexts, the final prediction can therefore be shaped by compressed global statistics and a small number of dominant output modes.

The copying and attractor analyses identify a decoding-side mechanism behind the gap observed in the first stage. Error-predictive internal signals may already be present before generation, yet autoregressive decoding compresses position-specific and count-sensitive information into a single final prediction state. The resulting bottleneck makes realised outputs sensitive to causal path structure, softmax normalisation, and attractor-like output preferences.

### E.2.4 Per-Regime Output Attractor Distributions

Figure 12 decomposes the parsed output distributions from the aggregation tasks into separate regimes. The Gold distribution gives the reference pattern of true numerical targets, while the remaining panels show parsed model predictions under Direct answering, Zero-shot CoT prompting, and Few-shot CoT prompting. Since each target value appears three times in the evaluation set, the Gold panel has a uniform frequency of three at every target value. Frequencies below this reference indicate that some generations move away from the exact gold answer, while frequencies above it indicate that multiple distinct targets collapse onto the same predicted value. Separating the four distributions makes these concentration patterns easier to inspect than the grouped comparison in Figure 11.

The Direct-answering distribution exhibits the sharpest concentration around a small number of irregular preferred values. Different input instances requiring distinct numerical answers are mapped

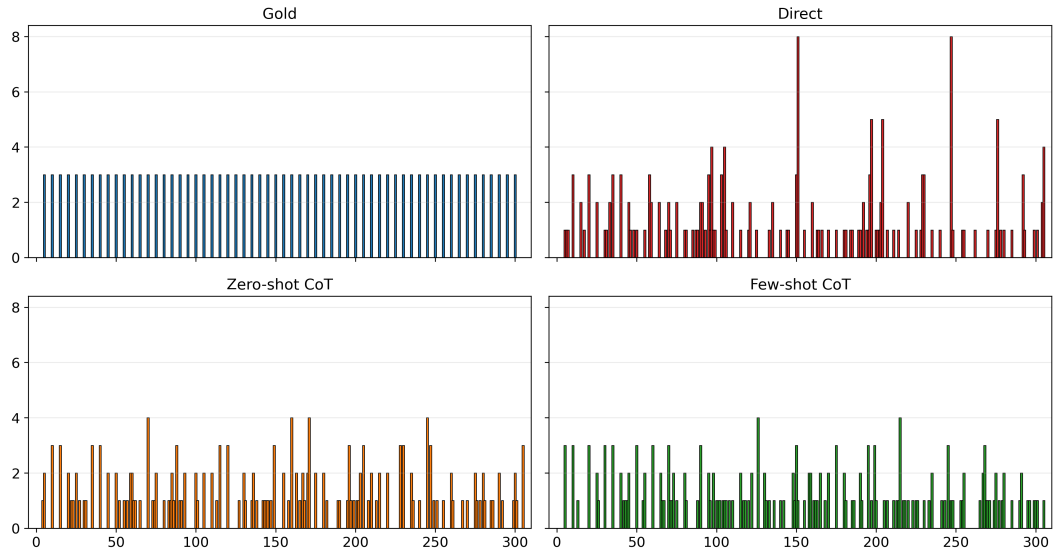


Figure 12: Per-regime output-attractor distributions. The four panels show the Gold reference distribution, Direct answering, Zero-shot CoT prompting, and Few-shot CoT prompting separately. The horizontal axis shows answer value, and the vertical axis shows frequency.

to repeated or neighbouring output values. Zero-shot CoT prompting spreads the distribution more widely and preserves more mass near the Gold targets, but it still shows local concentrations around dominant numerical modes. Few-shot CoT prompting further stabilises many lower and mid-range targets, although it does not uniformly dominate Zero-shot CoT. In several regions, the few-shot examples appear to introduce local grouping biases, causing predictions to move towards nearby anchor values.

These per-regime distributions support the same structural-compression account as the main output-attractor plot. CoT can make some intermediate computations explicit and reduce the sharpness of the attractors, but the final numerical answer remains produced from the terminal prediction state. As a result, the distribution becomes less concentrated under CoT, while residual attractor behaviour persists in the form of irregular numerical modes and local anchor values.

### E.3 Entropy Structure and Semantic Roles

#### E.3.1 Entropy Concentration Details

The entropy analysis in Section 2.3 examines where uncertainty concentrates within the generated reasoning trajectory. This analysis asks whether uncertainty is distributed across the whole chain of thought or concentrated at a small number of decision-sensitive positions. We use mathematical reasoning problems from AIME 2024 and AIME 2025 and evaluate Qwen3-8B (Yang et al., 2025a) in thinking mode. The model generates chain-of-thought responses with decoding temperature $T_{\text{dec}} = 1.0$. For every generated response, we record the output tokens and the pre-softmax logits at each decoding step.

For an input question $x$, let $o_{<t} = (o_1, \ldots, o_{t-1})$ denote the partial response generated before step $t$, where $o_i$ is the token generated at position $i$. At step $t$, the model produces a logit vector $\mathbf{z}_t \in \mathbb{R}^V$, where $V$ is the vocabulary size. The decoding temperature $T_{\text{dec}}$ controls the sharpness of the next-token distribution, giving

$$\mathbf{p}_t = \text{softmax}(\mathbf{z}_t / T_{\text{dec}}). \tag{30}$$

The scalar $p_{t,i}$ denotes the probability assigned to vocabulary item $i$ at step $t$. We measure the uncertainty of this predictive distribution using token-level entropy:

$$H_t = -\sum_{i=1}^{V} p_{t,i} \log p_{t,i}. \tag{31}$$

Here, $H_t$ is attached to the predictive distribution at a particular decoding step, so it depends on both the emitted token and its reasoning context. The same surface token can therefore receive different entropy values when it appears in different contexts.

Figure 4 summarises the resulting entropy statistics by combining the token-level entropy structure with an answer-level decomposition by correctness. In the main panel, the horizontal axis gives token-level entropy and the vertical axis gives token count on a logarithmic scale; the dashed line marks the empirical 80th percentile used for the high/low split. The inset compares average answer-level entropy between correct and incorrect responses, decomposed into low- and high-entropy contributions. The token-level distribution is strongly long-tailed. Most generated tokens have entropy close to zero, while a small minority have substantially higher uncertainty. This pattern suggests that much of the chain of thought is generated through stable local continuation, while a smaller set of positions carries concentrated uncertainty.

Let $\mathcal{H}$ denote the multiset of token entropy values collected from all generated responses:

$$\mathcal{H} = \{H_{n,t} \mid n = 1, \ldots, N,\ t = 1, \ldots, L_n\}. \tag{32}$$

Here, $H_{n,t}$ is the entropy at position $t$ in response $n$, $N$ is the number of responses, and $L_n$ is the length of response $n$. Following prior work on high-entropy reasoning tokens (Wang et al., 2026), we set the cutoff $\tau$ to the empirical 80th percentile of the pooled entropy distribution:

$$\tau = \mathrm{Quantile}_{0.8}(\mathcal{H}). \tag{33}$$

For each response $n$, decoding positions are partitioned into high- and low-entropy sets:

$$\mathcal{T}_{\mathrm{high}}^{(n)} = \{t \mid H_{n,t} > \tau\}, \quad \mathcal{T}_{\mathrm{low}}^{(n)} = \{t \mid H_{n,t} \leq \tau\}. \tag{34}$$

Here, $\mathcal{T}_{\mathrm{high}}^{(n)}$ contains the most uncertain decoding positions in response $n$, and $\mathcal{T}_{\mathrm{low}}^{(n)}$ contains the remaining positions.

To relate token uncertainty to answer correctness, we parse the final answer of each generated response and compare it with the gold answer. A response is assigned to the correct set $\mathcal{C}$ if its parsed final answer matches the gold answer, and to the incorrect set $\mathcal{I}$ otherwise. For response $n$, answer-level entropy is defined as the cumulative uncertainty along its reasoning trajectory:

$$A_n = \sum_{t=1}^{L_n} H_{n,t}. \tag{35}$$

Here, $A_n$ measures the total uncertainty accumulated across the generated reasoning trajectory for response $n$. We further decompose $A_n$ according to the entropy partition:

$$A_n^{\mathrm{high}} = \sum_{t \in \mathcal{T}_{\mathrm{high}}^{(n)}} H_{n,t}, A_n^{\mathrm{low}} = \sum_{t \in \mathcal{T}_{\mathrm{low}}^{(n)}} H_{n,t}. \tag{36}$$

Here, $A_n^{\mathrm{high}}$ is the cumulative entropy contributed by high-entropy positions in response $n$, and $A_n^{\mathrm{low}}$ is the corresponding contribution from low-entropy positions. By construction, $A_n = A_n^{\mathrm{high}} + A_n^{\mathrm{low}}$. The inset reports group averages of these quantities for correct and incorrect responses. Incorrect responses exhibit substantially higher average answer-level entropy than correct responses. The low-entropy contribution changes only slightly between correct and incorrect responses, while the gap between correct and incorrect responses is dominated by the high-entropy contribution. This pattern indicates that answer failure is associated with localised instability at a small subset of uncertain positions, with limited change across the low-entropy majority of the reasoning trajectory.

### E.3.2 Semantic Roles of Entropy-Grouped Tokens

Section 2.3 shows that the answer-level entropy gap between correct and incorrect responses is dominated by high-entropy decoding positions, while low-entropy positions form the stable majority of the reasoning trajectory. To examine the functional roles of these positions, we analyse the token strings generated at different entropy ranges. For a token type $w$, let $\mathcal{O}(w) = \{(n, t) \mid o_{n,t} = w\}$ be the set of decoding positions where $w$ appears, where $o_{n,t}$ is the token generated at position $t$ in




Figure 13: Word clouds of frequent tokens grouped by average entropy during CoT generation. The left panel shows high-entropy tokens, and the right panel shows low-entropy tokens. Word size reflects average token entropy across occurrences.

response $n$. To reduce noise from rare tokens, we retain only token types whose frequency $|\mathcal{O}(w)|$ exceeds a fixed threshold. The average entropy of token type $w$ is

$$\bar{H}(w) = \frac{1}{|\mathcal{O}(w)|} \sum_{(n,t)\in\mathcal{O}(w)} H_{n,t}. \tag{37}$$

We rank frequent token types by $\bar{H}(w)$ and visualise the highest- and lowest-entropy groups using word clouds in Figure 13. Word size reflects the average entropy of the corresponding token type across its occurrences.

Low-entropy tokens are generated in a highly deterministic manner. They mainly support path execution, including grammatical completion, fixed expression expansion, formula continuation, and the articulation of intermediate results along an already established reasoning path. High-entropy tokens show a different profile. They are more often associated with logical transitions, hypothesis formulation, condition introduction, contrast, revision, and reasoning direction selection. These positions tend to occur where several plausible continuations remain available, so the selected token can shape the subsequent trajectory of the chain of thought.

The entropy analysis identifies a sparse uncertainty structure in reasoning generation. High-entropy positions form a minority of generated tokens, yet they account for most of the answer-level entropy gap between correct and incorrect responses and align semantically with reasoning transitions and branching decisions. Low-entropy positions form the deterministic majority and mainly execute an already selected path. This provides a token-level account of how reasoning failures can arise from localised unstable positions within otherwise fluent chains of thought.

### E.4 Intervention Details

This appendix provides the detailed formulation of EHETI introduced in Section 2.4. Given an input question $x$, let $o_{<t} = (o_1, \ldots, o_{t-1})$ denote the generated prefix before decoding step $t$, where $o_i$ is the token generated at position $i$. Let $\pi_\theta$ denote the language model with parameters $\theta$, which maps the input and generated prefix to a next-token predictive distribution. At step $t$, the model produces a logit vector $\mathbf{z}_t \in \mathbb{R}^V$, where $V$ is the vocabulary size. The corresponding predictive distribution is

$$\mathbf{p}_t = \mathrm{softmax}(\mathbf{z}_t / T_{\mathrm{dec}}), \tag{38}$$

where $T_{\mathrm{dec}}$ is the decoding temperature. We compute the token entropy $H_t$ using Eq. (4).

We identify decision-sensitive positions by comparing the current token entropy with the uncertainty profile established by the preceding decoding steps. For $t > 1$, let $\mu_{t-1}$ and $\sigma_{t-1}$ denote the running mean and standard deviation of entropy values observed before step $t$:

$$\mu_{t-1} = \frac{1}{t-1}\sum_{k=1}^{t-1} H_k, \quad \sigma_{t-1} = \sqrt{\frac{1}{t-1}\sum_{k=1}^{t-1}(H_k - \mu_{t-1})^2}. \tag{39}$$

Here, $\mu_{t-1}$ and $\sigma_{t-1}$ characterise the uncertainty profile established by the preceding decoding steps. The current entropy $H_t$ is therefore evaluated against the trajectory history without contributing to its own reference threshold. This adaptive criterion accounts for the fact that different questions can induce different entropy scales.

Let $B$ denote the maximum number of early decoding positions eligible for intervention, and let $M$ denote the maximum number of intervention steps allowed in one response. We define the binary intervention gate $I_t$ as

$$I_t = \mathbf{1}\Big[1 < t \leq B,\ H_t > \mu_{t-1} + \alpha\sigma_{t-1}, \sum_{k=1}^{t-1} I_k < M\Big]. \tag{40}$$

Here, $\mathbf{1}[\cdot]$ is the indicator function, which equals 1 when the condition is true and 0 otherwise. The parameter $\alpha \geq 0$ controls how large an entropy increase must be before intervention is triggered. The early-window constraint $t \leq B$ follows from the structural-compression analysis: earlier uncertain decisions can redirect later reasoning, while later positions more often refine a path already selected. The budget $M$ keeps the procedure sparse and prevents repeated modification of the whole trajectory.

When $I_t = 0$, the model decodes from the original logits $\mathbf{z}_t$. When $I_t = 1$, we apply a lightweight contrastive correction to the logits at that step. Let $c^-$ denote a lightweight contrastive condition that represents unreliable or erroneous reasoning behaviour and is used only to compute an auxiliary logit signal. In our implementation, $c^-$ is instantiated with an auxiliary instruction such as *"Continue the current reasoning in a hasty and unreliable way, following the first plausible path and skipping verification of logical, factual, arithmetic, or programming constraints."* Let $\mathbf{z}_t^- \in \mathbb{R}^V$ denote the auxiliary logits produced by the same model under $c^-$, using the original input and the same generated prefix $o_{<t}$. The corrected logits are

$$\widetilde{\mathbf{z}}_t = \mathbf{z}_t + \lambda I_t \left(\mathbf{z}_t - \mathbf{z}_t^-\right). \tag{41}$$

Here, $\widetilde{\mathbf{z}}_t$ denotes the corrected logits, and $\lambda \geq 0$ controls the intervention strength. If $I_t = 0$, Eq. (41) reduces to $\widetilde{\mathbf{z}}_t = \mathbf{z}_t$, so stable low-entropy steps remain unchanged. If $I_t = 1$, the update moves the prediction away from tokens supported by the contrastive condition and increases the relative preference for tokens supported by the original conditional context.

The final decoding distribution is

$$\widetilde{\mathbf{p}}_t = \mathrm{softmax}\left(\widetilde{\mathbf{z}}_t / T_{\mathrm{dec}}\right). \tag{42}$$

Here, $\widetilde{\mathbf{p}}_t$ is the post-intervention predictive distribution at step $t$. The next token $o_t$ is selected from $\widetilde{\mathbf{p}}_t$ using the same decoding rule as the base model. The auxiliary logits $\mathbf{z}_t^-$ are computed only when $I_t = 1$, so each selected position requires one additional forward pass and each response uses at most $M$ auxiliary forward passes.

Table 6: EHETI implementation settings and validation grids. The selected values are fixed across all models and benchmarks.

| Parameter | Default | Validation grid |
|---|---|---|
| $B$ | 128 | $\{64, 128, 256\}$ |
| $M$ | 4 | $\{2, 4, 8\}$ |
| $\alpha$ | 1.0 | $\{0.5, 1.0, 1.5\}$ |
| $\lambda$ | 0.5 | $\{0.25, 0.5, 0.75, 1.0\}$ |
| $T_{\mathrm{dec}}$ | 1.0 | fixed |

Table 6 summarises the implementation settings and validation grids. The values of $B$, $M$, $\alpha$, and $\lambda$ are selected on validation data and then fixed for all reported models and benchmarks. We use $T_{\mathrm{dec}} = 1.0$ when computing token entropy and the post-intervention distribution.

This intervention matches the mechanism identified in the previous stages. It targets early high-entropy spikes where reasoning paths are still flexible and local uncertainty can propagate, while preserving deterministic path-execution tokens. The procedure requires no parameter updates, no additional training data, and no broad modification of the model's decoding behaviour.

# F Additional Experimental Results

## F.1 Additional Model-Scale Performance Analysis

Table 7 extends the main comparison to additional model scales under the same inference strategies. The results reinforce the main finding: EHETI achieves the highest average score for every

Table 7: Additional model-scale performance analysis across the same reasoning benchmarks and inference strategies used in the main comparison. Within each model block, green cells mark the best results, while blue cells mark cases where EHETI ranks second.

| Model | Strategy | MMLU-Pro | GPQA-Diamond | MATH500 | HumanEvalPlus | LiveCodeBench | 2WikiMultiHopQA | Avg. |
|---|---|---|---|---|---|---|---|---|
| | | Acc. | Acc. | Acc. | Pass@1 | Pass@1 | EM | Score |
| Qwen3-8B | DI | 70.52 | 57.07 | 96.82 | 67.07 | 79.44 | 68.48 | 73.23 |
| | MTI | 71.22 | 59.06 | 95.94 | 66.02 | 87.11 | 70.56 | 74.98 |
| | SC@32 | 73.18 | 63.28 | 97.58 | 69.82 | 81.16 | 74.82 | 76.64 |
| | Self-Refine-2 | 69.86 | 61.42 | 96.34 | 65.74 | 76.52 | 75.28 | 74.19 |
| | CFG | 70.19 | 57.58 | 92.56 | 65.85 | 84.75 | 69.04 | 73.33 |
| | **EHETI** | 73.46 | 64.06 | 98.32 | 74.84 | 91.36 | 75.84 | 79.65 |
| Qwen3-14B | DI | 75.61 | 57.58 | 96.46 | 70.12 | 85.94 | 72.73 | 76.41 |
| | MTI | 75.04 | 61.09 | 95.93 | 74.54 | 89.42 | 74.78 | 78.47 |
| | SC@32 | 78.18 | 64.68 | 97.36 | 72.74 | 87.96 | 79.14 | 80.01 |
| | Self-Refine-2 | 74.82 | 62.18 | 96.22 | 69.58 | 83.28 | 79.52 | 77.60 |
| | CFG | 74.12 | 60.12 | 94.76 | 69.51 | 88.96 | 73.42 | 76.81 |
| | **EHETI** | 76.62 | 67.28 | 98.16 | 77.24 | 93.18 | 79.88 | 82.06 |
| DeepSeek-R1-14B | DI | 63.07 | 55.02 | 93.98 | 75.98 | 57.93 | 63.46 | 68.24 |
| | MTI | 64.42 | 60.03 | 95.24 | 78.12 | 63.48 | 67.84 | 71.52 |
| | SC@32 | 66.12 | 62.86 | 95.18 | 78.34 | 60.36 | 70.06 | 72.15 |
| | Self-Refine-2 | 62.46 | 61.28 | 93.84 | 75.08 | 55.42 | 70.84 | 69.82 |
| | CFG | 62.52 | 55.88 | 92.56 | 74.93 | 59.26 | 63.98 | 68.19 |
| | **EHETI** | 66.84 | 65.18 | 97.42 | 80.12 | 67.56 | 70.96 | 74.68 |
| DeepSeek-R1-32B | DI | 70.03 | 60.98 | 95.94 | 84.07 | 72.04 | 69.84 | 75.48 |
| | MTI | 71.18 | 64.88 | 97.12 | 86.02 | 76.46 | 73.48 | 78.19 |
| | SC@32 | 72.82 | 67.62 | 96.94 | 86.12 | 74.22 | 75.04 | 78.79 |
| | Self-Refine-2 | 69.22 | 66.46 | 95.72 | 83.48 | 69.64 | 74.86 | 76.56 |
| | CFG | 69.64 | 61.92 | 94.48 | 82.89 | 73.06 | 70.14 | 75.36 |
| | **EHETI** | 73.42 | 69.18 | 98.62 | 87.34 | 80.48 | 75.82 | 80.81 |
| DeepSeek-R1-70B | DI | 74.96 | 64.96 | 97.04 | 88.02 | 78.02 | 74.27 | 79.55 |
| | MTI | 76.06 | 68.42 | 98.02 | 89.43 | 82.16 | 77.48 | 81.93 |
| | SC@32 | 77.18 | 71.36 | 97.86 | 89.54 | 79.96 | 79.46 | 82.56 |
| | Self-Refine-2 | 73.62 | 69.86 | 96.78 | 87.46 | 76.12 | 79.44 | 80.55 |
| | CFG | 74.58 | 65.78 | 95.66 | 86.85 | 79.03 | 74.46 | 79.39 |
| | **EHETI** | 77.58 | 72.08 | 99.08 | 90.42 | 84.26 | 79.86 | 83.88 |

additional model, with gains over Direct Inference ranging from 4.34 to 6.44 points. The largest gains appear on Qwen3-8B and DeepSeek-R1-14B, where limited representational and computational margins make early uncertain decisions more likely to constrain the remaining trajectory. The gains become smaller at higher capacity, but they remain consistent, suggesting that scaling improves the stability of reasoning paths without removing the local points at which trajectories can still be redirected.

The table also reveals local capacity reversals. In several comparisons, a smaller model equipped with EHETI approaches or exceeds the Direct Inference result of a larger model in the same family, such as Qwen3-8B relative to Qwen3-14B and DeepSeek-R1-32B relative to DeepSeek-R1-70B. The reversal is not global; it appears most clearly on tasks where early choices strongly shape later reasoning or code structure, including GPQA-Diamond, MATH500, and LiveCodeBench. This suggests that part of the scale gap comes from trajectory stability during decoding, not only from the amount of knowledge or computation encoded in model parameters.

Across tasks, the appendix table follows the same broad structure as the main results while making the scale effect clearer. GPQA-Diamond and LiveCodeBench show the strongest sensitivity to strategy choice, which is expected for tasks where a small early decision can determine a scientific inference chain or a program structure. On 2WikiMultiHopQA, EHETI achieves the highest score across all additional models, indicating that targeted intervention can also support evidence reconciliation across passages without relying on a task-specific revision process. Its consistent performance across these heterogeneous reasoning tasks drives the highest average score for every model.

### F.2 Additional Inference-Strategy Performance Analysis

Table 8 reports additional inference strategies across all evaluated models, complementing the main and scale-analysis tables. SC@16, SC@64, and SC@512 instantiate Self-Consistency with different sampling budgets, where multiple reasoning trajectories are generated and the final answer is selected by answer-level majority voting. Best-of-N (BoN@32) (Snell et al., 2024) samples 32 candidate responses and selects the final answer using the same model as a self-verifier. These strategies cover repeated sampling and candidate selection, while EHETI performs targeted intervention during the primary generation trajectory.

Table 8: Additional inference-strategy performance comparison across all evaluated models and benchmarks, covering alternative sampling budgets, candidate-selection strategies, and EHETI. Within each model block, green cells mark the best results, while blue cells mark cases where EHETI ranks second.

| Model | Strategy | MMLU-Pro | GPQA-Diamond | MATH500 | HumanEvalPlus | LiveCodeBench | 2WikiMultiHopQA | Avg. |
|---|---|---|---|---|---|---|---|---|
| | | Acc. | Acc. | Acc. | Pass@1 | Pass@1 | EM | Score |
| DeepSeek-R1-8B | SC@16 | 51.83 | 54.12 | 90.34 | 67.28 | 40.86 | 61.74 | 61.03 |
| | SC@64 | 52.38 | 58.74 | 89.82 | 67.54 | 40.94 | 63.48 | 62.15 |
| | SC@512 | 51.92 | 60.16 | 89.26 | 66.84 | 39.72 | 64.74 | 62.11 |
| | BoN@32 | 52.46 | 61.82 | 91.14 | 69.64 | 40.38 | 66.88 | 63.72 |
| | **EHETI** | 53.84 | 59.48 | 93.26 | 71.92 | 51.04 | 64.72 | 65.71 |
| Qwen3-32B | SC@16 | 78.12 | 64.38 | 97.34 | 98.51 | 92.66 | 80.02 | 85.17 |
| | SC@64 | 78.24 | 66.42 | 97.96 | 98.38 | 94.16 | 81.36 | 86.09 |
| | SC@512 | 77.72 | 68.56 | 97.28 | 98.04 | 92.84 | 83.04 | 86.25 |
| | BoN@32 | 78.34 | 69.86 | 97.72 | 98.42 | 90.86 | 84.92 | 86.69 |
| | **EHETI** | 78.06 | 68.74 | 98.73 | 98.23 | 93.62 | 82.46 | 86.64 |
| GLM-5.1 | SC@16 | 85.24 | 88.02 | 98.68 | 90.82 | 75.46 | 84.26 | 87.08 |
| | SC@64 | 85.18 | 91.24 | 98.46 | 91.08 | 78.02 | 85.04 | 88.17 |
| | SC@512 | 84.64 | 92.36 | 98.18 | 90.56 | 75.86 | 86.52 | 88.02 |
| | BoN@32 | 85.58 | 93.14 | 98.94 | 91.81 | 75.32 | 88.36 | 88.86 |
| | **EHETI** | 85.64 | 91.86 | 99.58 | 92.16 | 82.24 | 86.88 | 89.73 |
| MiniMax-M2.7 | SC@16 | 83.06 | 80.42 | 98.36 | 92.72 | 77.66 | 82.04 | 85.71 |
| | SC@64 | 83.04 | 84.16 | 98.18 | 92.84 | 80.16 | 83.24 | 86.94 |
| | SC@512 | 83.26 | 85.42 | 97.94 | 93.08 | 78.62 | 83.66 | 87.00 |
| | BoN@32 | 83.36 | 86.54 | 98.73 | 93.68 | 77.86 | 85.64 | 87.64 |
| | **EHETI** | 84.08 | 84.36 | 99.61 | 94.34 | 82.82 | 84.38 | 88.27 |
| GPT-OSS-120B | SC@16 | 83.18 | 80.06 | 98.54 | 92.21 | 81.24 | 83.42 | 86.44 |
| | SC@64 | 83.56 | 83.36 | 98.43 | 92.42 | 83.26 | 84.44 | 87.58 |
| | SC@512 | 83.06 | 84.18 | 98.16 | 91.84 | 81.76 | 85.02 | 87.34 |
| | BoN@32 | 83.74 | 85.62 | 98.82 | 93.15 | 81.18 | 86.76 | 88.21 |
| | **EHETI** | 84.42 | 83.76 | 99.65 | 93.28 | 84.52 | 85.36 | 88.50 |
| Qwen3-8B | SC@16 | 72.04 | 60.16 | 97.21 | 68.44 | 80.24 | 72.14 | 75.04 |
| | SC@64 | 72.46 | 62.22 | 97.06 | 68.74 | 82.34 | 73.86 | 76.11 |
| | SC@512 | 71.84 | 64.48 | 96.68 | 67.86 | 80.88 | 74.36 | 76.02 |
| | BoN@32 | 72.72 | 65.92 | 97.64 | 70.28 | 80.12 | 76.42 | 77.18 |
| | **EHETI** | 73.46 | 64.06 | 98.32 | 74.84 | 91.36 | 75.84 | 79.65 |
| Qwen3-14B | SC@16 | 76.92 | 61.04 | 96.88 | 71.16 | 86.74 | 76.62 | 78.23 |
| | SC@64 | 77.22 | 65.42 | 96.72 | 71.42 | 89.12 | 77.86 | 79.63 |
| | SC@512 | 76.58 | 66.84 | 97.18 | 70.66 | 87.44 | 78.42 | 79.52 |
| | BoN@32 | 77.54 | 68.36 | 97.48 | 73.12 | 86.54 | 80.64 | 80.61 |
| | **EHETI** | 76.62 | 67.28 | 98.16 | 77.24 | 93.18 | 79.88 | 82.06 |
| DeepSeek-R1-14B | SC@16 | 64.86 | 59.12 | 94.62 | 76.68 | 59.22 | 67.48 | 70.33 |
| | SC@64 | 65.24 | 64.08 | 94.34 | 77.26 | 61.48 | 69.42 | 71.97 |
| | SC@512 | 64.48 | 65.62 | 94.05 | 76.42 | 59.88 | 70.28 | 71.79 |
| | BoN@32 | 65.78 | 66.74 | 95.32 | 78.86 | 59.18 | 71.82 | 72.95 |
| | **EHETI** | 66.84 | 65.18 | 97.42 | 80.12 | 67.56 | 70.96 | 74.68 |
| DeepSeek-R1-32B | SC@16 | 71.68 | 64.46 | 96.43 | 84.72 | 73.08 | 72.86 | 77.20 |
| | SC@64 | 71.86 | 68.36 | 96.32 | 84.96 | 75.26 | 74.32 | 78.51 |
| | SC@512 | 71.04 | 69.82 | 96.01 | 84.47 | 73.66 | 74.68 | 78.28 |
| | BoN@32 | 72.36 | 70.94 | 97.14 | 86.46 | 73.42 | 76.24 | 79.43 |
| | **EHETI** | 73.42 | 69.18 | 98.62 | 87.34 | 80.48 | 75.82 | 80.81 |
| DeepSeek-R1-70B | SC@16 | 76.14 | 68.04 | 97.48 | 88.46 | 78.82 | 77.14 | 81.01 |
| | SC@64 | 76.36 | 72.22 | 97.28 | 88.54 | 81.06 | 78.68 | 82.36 |
| | SC@512 | 75.58 | 73.04 | 97.05 | 88.06 | 79.62 | 78.92 | 82.05 |
| | BoN@32 | 76.92 | 74.12 | 98.04 | 89.98 | 79.28 | 80.62 | 83.16 |
| | **EHETI** | 77.58 | 72.08 | 99.08 | 90.42 | 84.26 | 79.86 | 83.88 |

The Self-Consistency results show that increasing the sampling budget does not consistently improve performance. Larger budgets often help on GPQA-Diamond and multi-hop QA, where alternative trajectories can expose useful answer diversity, but the same trend is less stable on MATH500, HumanEvalPlus, and LiveCodeBench, where additional samples may reinforce common symbolic errors or introduce invalid code variants. BoN@32 is the strongest additional strategy in many model blocks, especially on GPQA-Diamond and 2WikiMultiHopQA, where answer-level plausibility comparison is useful. Its advantage is less consistent on code tasks, where same-model verification may favour coherent explanations over functional correctness.

EHETI achieves the best average score in most model blocks and remains close to the strongest strategy in the remaining cases. Broad sampling and verification can be effective, but their gains depend heavily on whether extra computation produces genuinely useful alternatives. The method can act before an erroneous trajectory fully unfolds, which is especially useful when later candidates differ in surface form while sharing an early commitment. This explains its strong performance on MATH500, HumanEvalPlus, and LiveCodeBench, where local decisions often determine the subsequent reasoning or program structure. The results indicate that effective test-time computation

is shaped by where intervention is applied within the generation trajectory, especially while the trajectory is still malleable.

### F.3 Additional Model-Scale Cost Analysis

Table 9: Additional model-scale cost comparison of token counts ($\times 10^8$) and accuracy (%) across competition-style reasoning benchmarks, using the same inference strategies as the main cost table.

| Model | Dataset | MTI | | SC@32 | | Self-Refine-2 | | CFG | | Avg. | | EHETI | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Token | Acc. | Token | Acc. | Token | Acc. | Token | Acc. | Token | Acc. | Token | Acc. |
| Qwen3-8B | AIME2024 | 1.38 | 78.4 | 1.96 | 78.0 | 3.04 | 75.8 | 1.30 | 74.6 | 1.92 | 76.7 | **1.05** (-45.3%) | **82.7** |
| | AIME2025 | 1.98 | 75.6 | 2.22 | 74.5 | 3.58 | 73.1 | 1.84 | 70.6 | 2.41 | 73.5 | **1.68** (-30.1%) | **77.4** |
| | BRUMO2025 | 1.72 | 90.6 | 1.82 | 90.0 | 2.98 | 88.4 | 1.60 | 87.8 | 2.03 | 89.2 | **1.43** (-29.6%) | **91.2** |
| | HMMT2025 | 2.64 | 67.4 | 2.62 | 65.7 | 3.86 | 64.8 | 2.50 | 62.7 | 2.91 | 65.2 | **2.34** (-19.4%) | **68.6** |
| | Avg. | 1.93 | 78.0 | 2.16 | 77.0 | 3.37 | 75.5 | 1.81 | 73.9 | 2.32 | 76.1 | **1.62** (-29.8%) | **80.0** |
| Qwen3-14B | AIME2024 | 1.16 | 84.6 | 1.68 | 83.2 | 2.72 | 81.3 | 1.08 | 79.9 | 1.66 | 82.3 | **0.92** (-44.6%) | **86.4** |
| | AIME2025 | 1.72 | 78.8 | 1.90 | 77.5 | 3.16 | 76.6 | 1.60 | 74.2 | 2.10 | 76.8 | **1.51** (-27.9%) | **79.8** |
| | BRUMO2025 | 1.52 | 92.2 | 1.62 | 92.4 | 2.68 | 90.5 | 1.42 | 89.8 | 1.81 | 91.2 | **1.31** (-27.6%) | **92.8** |
| | HMMT2025 | 2.32 | 69.6 | 2.34 | 68.4 | 3.42 | 67.1 | 2.20 | 65.4 | 2.57 | 67.6 | **2.13** (-17.1%) | **70.4** |
| | Avg. | 1.68 | 81.3 | 1.89 | 80.4 | 3.00 | 78.9 | 1.58 | 77.3 | 2.04 | 79.5 | **1.47** (-27.8%) | **82.3** |
| DeepSeek-R1-14B | AIME2024 | 1.42 | 86.8 | 1.98 | 85.0 | 3.08 | 82.6 | 1.34 | 81.0 | 1.96 | 83.9 | **1.08** (-44.8%) | **88.2** |
| | AIME2025 | 2.02 | 81.6 | 2.28 | 80.4 | 3.64 | 79.2 | 1.88 | 76.8 | 2.46 | 79.5 | **1.74** (-29.1%) | **83.1** |
| | BRUMO2025 | 1.82 | 94.0 | 1.90 | 93.6 | 3.16 | 91.8 | 1.68 | 90.9 | 2.14 | 92.6 | **1.52** (-29.0%) | **94.5** |
| | HMMT2025 | 2.76 | 72.8 | 2.72 | 70.8 | 4.02 | 69.8 | 2.58 | 66.9 | 3.02 | 70.1 | **2.46** (-18.5%) | **73.9** |
| | Avg. | 2.00 | 83.8 | 2.22 | 82.5 | 3.48 | 80.8 | 1.87 | 78.9 | 2.39 | 81.5 | **1.70** (-28.9%) | **84.9** |
| DeepSeek-R1-32B | AIME2024 | 1.20 | 90.3 | 1.76 | 89.4 | 2.84 | 87.5 | 1.12 | 85.4 | 1.73 | 88.2 | **0.96** (-44.5%) | **91.8** |
| | AIME2025 | 1.62 | 86.1 | 2.02 | 85.2 | 3.22 | 84.0 | 1.50 | 81.2 | 2.09 | 84.1 | **1.42** (-32.1%) | **87.3** |
| | BRUMO2025 | 1.54 | 95.1 | 1.68 | 94.8 | 2.86 | 92.9 | 1.44 | 91.7 | 1.88 | 93.6 | **1.35** (-28.2%) | **95.4** |
| | HMMT2025 | 2.38 | 76.9 | 2.44 | 75.4 | 3.70 | 74.1 | 2.22 | 70.8 | 2.69 | 74.3 | **2.16** (-19.6%) | **78.1** |
| | Avg. | 1.69 | 87.1 | 1.98 | 86.2 | 3.16 | 84.6 | 1.57 | 82.3 | 2.10 | 85.1 | **1.47** (-29.8%) | **88.2** |
| DeepSeek-R1-70B | AIME2024 | 1.08 | 92.8 | 1.64 | 92.3 | 2.62 | 90.6 | 1.02 | 88.8 | 1.59 | 91.1 | **0.86** (-45.9%) | **93.7** |
| | AIME2025 | 1.36 | 89.7 | 1.84 | 89.5 | 3.02 | 87.9 | 1.28 | 85.7 | 1.88 | 88.2 | **1.18** (-37.1%) | **90.5** |
| | BRUMO2025 | 1.48 | 96.0 | 1.58 | 95.7 | 2.70 | 93.8 | 1.36 | 92.6 | 1.78 | 94.5 | **1.26** (-29.2%) | **96.2** |
| | HMMT2025 | 2.18 | 81.6 | 2.28 | 80.7 | 3.46 | 79.5 | 2.04 | 75.6 | 2.49 | 79.4 | **1.96** (-21.3%) | **82.4** |
| | Avg. | 1.53 | 90.0 | 1.83 | 89.5 | 2.95 | 88.0 | 1.43 | 85.7 | 1.94 | 88.3 | **1.31** (-32.0%) | **90.7** |

Table 9 extends the main cost analysis to additional model scales while keeping the same comparison strategies. The results show that the efficiency pattern is not limited to the five main models. Across Qwen3 and DeepSeek-R1 variants, EHETI consistently achieves the best average accuracy with the lowest average token usage. The token reduction relative to the comparison average ranges from 27.8% to 32.0%, while accuracy improves by 2.4 to 3.9 points. This indicates that the cost advantage does not come from sacrificing correctness; it comes from avoiding redundant full-trajectory computation once the decisive parts of the reasoning path have been stabilised.

The model-scale trend is informative. Smaller variants such as Qwen3-8B and DeepSeek-R1-14B show larger accuracy gains, which is consistent with their tighter representational and computational margins. In these settings, an early unstable choice can consume a large portion of the model's effective reasoning capacity, so reducing trajectory drift has a stronger effect. Larger variants require fewer tokens on average and start from more stable reasoning behaviour, yet the reduction remains substantial. This suggests that scaling compresses part of the uncertainty, but it does not remove the inefficiency of generating multiple complete solutions or repeated revisions.

The task pattern mirrors the main cost table. AIME2024 and AIME2025 show the strongest token reductions, since many solutions hinge on a few early mathematical commitments followed by relatively stable derivations. HMMT2025 yields smaller reductions because harder instances contain more sustained uncertainty across the solution, leaving less purely redundant generation to remove. The same behaviour appears across both Qwen3 and DeepSeek-R1 families, supporting the view that efficient test-time reasoning depends on matching computation to the structure of uncertainty, not simply increasing the number of generated trajectories.

### F.4 Additional Inference-Strategy Cost Analysis

Table 10 extends the cost analysis to additional inference strategies across all evaluated models. SC@16, SC@64, and SC@512 vary the Self-Consistency sampling budget, while BoN@32 selects from 32 sampled candidates using self-verification. These strategies increase test-time computation

Table 10: Additional inference-cost comparison of token counts ($\times 10^8$) and accuracy (%) across all evaluated models and competition-style reasoning benchmarks, covering alternative sampling budgets and candidate-selection strategies.

| Model | Dataset | SC@16 | | SC@64 | | SC@512 | | BoN@32 | | Avg. | | EHETI | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Token | Acc. | Token | Acc. | Token | Acc. | Token | Acc. | Token | Acc. | Token | Acc. |
| DeepSeek-R1-8B | AIME2024 | 1.62 | 84.9 | 2.46 | 85.8 | 3.55 | 86.7 | 2.38 | 88.2 | 2.50 | 86.4 | **1.12** (-55.2%) | **89.6** |
| | AIME2025 | 1.88 | 79.6 | 2.88 | 82.4 | 4.01 | 82.3 | 2.92 | **84.2** | 2.92 | 82.1 | **1.81** (-38.1%) | 83.9 |
| | BRUMO2025 | **1.54** | 91.6 | 2.28 | 92.1 | 3.56 | 93.3 | 2.38 | **93.6** | 2.44 | 92.7 | 1.62 (-33.6%) | 93.3 |
| | HMMT2025 | **2.18** | 67.9 | 3.24 | 70.5 | 4.49 | 69.8 | 3.46 | 73.2 | 3.34 | 70.4 | 2.52 (-24.6%) | **73.8** |
| | Avg. | 1.81 | 81.0 | 2.71 | 82.7 | 3.90 | 83.0 | 2.79 | 84.8 | 2.80 | 82.9 | **1.77** (-36.9%) | **85.2** |
| Qwen3-32B | AIME2024 | 1.03 | 82.9 | 1.56 | 84.1 | 2.00 | 84.8 | 1.64 | 86.1 | 1.56 | 84.5 | **0.77** (-50.6%) | **88.0** |
| | AIME2025 | **1.22** | 78.2 | 1.92 | 79.4 | 2.43 | 80.1 | 2.02 | **80.9** | 1.90 | 79.7 | 1.38 (-27.3%) | 80.2 |
| | BRUMO2025 | **1.06** | 92.1 | 1.68 | 93.0 | 2.17 | 93.3 | 1.78 | **93.5** | 1.67 | 93.0 | 1.17 (-30.0%) | 92.9 |
| | HMMT2025 | **1.38** | 61.8 | 2.08 | 63.2 | 2.76 | 63.4 | 2.28 | **64.5** | 2.13 | 63.2 | 1.90 (-10.6%) | 64.1 |
| | Avg. | **1.17** | 78.8 | 1.81 | 79.9 | 2.34 | 80.4 | 1.93 | 81.2 | 1.81 | 80.1 | 1.30 (-28.0%) | **81.3** |
| GLM-5.1 | AIME2024 | 1.20 | 91.6 | 1.82 | 93.4 | 2.46 | 94.1 | 1.92 | 94.3 | 1.85 | 93.4 | **0.82** (-55.7%) | **94.5** |
| | AIME2025 | 1.36 | 88.4 | 2.18 | 90.6 | 2.88 | 91.8 | 2.28 | **92.0** | 2.18 | 90.7 | **1.06** (-51.3%) | 91.4 |
| | BRUMO2025 | **1.12** | 95.3 | 1.84 | 96.0 | 2.55 | **97.1** | 1.98 | 97.0 | 1.87 | 96.4 | 1.20 (-35.9%) | 96.8 |
| | HMMT2025 | **1.68** | 81.6 | 2.52 | 83.6 | 3.42 | 84.2 | 2.78 | 85.0 | 2.60 | 83.6 | 1.82 (-30.0%) | **85.6** |
| | Avg. | 1.34 | 89.2 | 2.09 | 90.9 | 2.83 | 91.8 | 2.24 | **92.1** | 2.13 | 91.0 | **1.23** (-42.3%) | **92.1** |
| MiniMax-M2.7 | AIME2024 | 1.25 | 89.4 | 1.90 | 91.2 | 2.58 | 91.7 | 2.02 | 92.0 | 1.94 | 91.1 | **0.88** (-54.6%) | **92.6** |
| | AIME2025 | 1.42 | 86.1 | 2.24 | 88.2 | 3.02 | 89.5 | 2.38 | **90.3** | 2.27 | 88.5 | **1.18** (-47.9%) | 89.8 |
| | BRUMO2025 | **1.18** | 93.6 | 1.92 | 95.2 | 2.66 | 96.0 | 2.06 | **96.2** | 1.96 | 95.3 | 1.24 (-36.6%) | 95.6 |
| | HMMT2025 | **1.78** | 78.8 | 2.68 | 81.0 | 3.60 | 81.5 | 2.92 | 82.1 | 2.75 | 80.9 | 1.94 (-29.3%) | **82.7** |
| | Avg. | 1.41 | 87.0 | 2.19 | 88.9 | 2.96 | 89.7 | 2.34 | **90.2** | 2.23 | 89.0 | **1.31** (-41.1%) | **90.2** |
| GPT-OSS-120B | AIME2024 | 1.38 | 94.8 | 2.08 | 96.2 | 2.66 | 96.7 | 2.30 | 96.8 | 2.11 | 96.1 | **0.87** (-58.7%) | **96.9** |
| | AIME2025 | 1.52 | 95.0 | 2.46 | 96.6 | 3.23 | 97.1 | 2.62 | 97.4 | 2.46 | 96.5 | **0.96** (-60.9%) | **97.5** |
| | BRUMO2025 | **1.24** | 82.0 | 2.02 | 83.1 | 2.68 | 83.8 | 2.14 | **84.0** | 2.02 | 83.2 | 1.27 (-37.1%) | 83.7 |
| | HMMT2025 | **1.86** | 90.3 | 2.94 | 92.1 | 4.09 | 92.8 | 3.12 | **93.1** | 3.00 | 92.1 | 1.88 (-37.4%) | 92.5 |
| | Avg. | 1.50 | 90.5 | 2.38 | 92.0 | 3.17 | 92.6 | 2.54 | **92.8** | 2.40 | 92.0 | **1.24** (-48.0%) | 92.7 |
| Qwen3-8B | AIME2024 | 1.56 | 76.2 | 2.42 | 78.7 | 3.35 | 80.1 | 2.26 | 81.0 | 2.40 | 79.0 | **1.05** (-56.2%) | **82.7** |
| | AIME2025 | 1.76 | 72.8 | 2.74 | 75.2 | 3.84 | 75.9 | 2.78 | 77.2 | 2.78 | 75.3 | **1.68** (-39.6%) | **77.4** |
| | BRUMO2025 | 1.44 | 89.1 | 2.16 | 90.8 | 3.29 | 91.7 | 2.22 | **92.0** | 2.28 | 90.9 | **1.43** (-37.2%) | 91.2 |
| | HMMT2025 | **2.08** | 64.0 | 3.12 | 66.4 | 4.18 | 66.2 | 3.34 | 68.1 | 3.18 | 66.2 | 2.34 (-26.4%) | **68.6** |
| | Avg. | 1.71 | 75.5 | 2.61 | 77.8 | 3.67 | 78.5 | 2.65 | 79.6 | 2.66 | 77.9 | **1.62** (-38.9%) | **80.0** |
| Qwen3-14B | AIME2024 | 1.34 | 81.7 | 2.02 | 83.6 | 2.80 | 83.9 | 2.08 | 85.1 | 2.06 | 83.6 | **0.92** (-55.3%) | **86.4** |
| | AIME2025 | **1.50** | 76.2 | 2.36 | 78.4 | 3.12 | 78.6 | 2.42 | **80.1** | 2.35 | 78.3 | 1.51 (-35.7%) | 79.8 |
| | BRUMO2025 | **1.28** | 91.4 | 2.00 | 92.7 | 2.78 | 93.1 | 2.16 | **93.4** | 2.06 | 92.7 | 1.31 (-36.3%) | 92.8 |
| | HMMT2025 | **1.86** | 66.8 | 2.86 | 69.2 | 3.54 | 67.9 | 2.98 | 70.2 | 2.81 | 68.5 | 2.13 (-24.2%) | **70.4** |
| | Avg. | 1.50 | 79.0 | 2.31 | 81.0 | 3.06 | 80.9 | 2.41 | 82.2 | 2.32 | 80.8 | **1.47** (-36.7%) | **82.3** |
| DeepSeek-R1-14B | AIME2024 | 1.58 | 83.4 | 2.42 | 85.8 | 3.42 | 85.4 | 2.34 | 87.5 | 2.44 | 85.5 | **1.08** (-55.7%) | **88.2** |
| | AIME2025 | 1.82 | 78.9 | 2.80 | 81.7 | 3.88 | 82.0 | 2.86 | **83.4** | 2.84 | 81.5 | **1.74** (-38.7%) | 83.1 |
| | BRUMO2025 | **1.50** | 92.5 | 2.30 | 94.2 | 3.44 | 94.9 | 2.38 | **95.1** | 2.41 | 94.2 | 1.52 (-36.8%) | 94.5 |
| | HMMT2025 | **2.14** | 69.3 | 3.22 | 71.6 | 4.32 | 71.2 | 3.40 | 73.5 | 3.27 | 71.4 | 2.46 (-24.8%) | **73.9** |
| | Avg. | 1.76 | 81.0 | 2.69 | 83.3 | 3.77 | 83.4 | 2.75 | **84.9** | 2.74 | 83.2 | **1.70** (-37.9%) | **84.9** |
| DeepSeek-R1-32B | AIME2024 | 1.40 | 88.2 | 2.14 | 89.9 | 2.92 | 90.2 | 2.20 | 91.0 | 2.17 | 89.8 | **0.96** (-55.7%) | **91.8** |
| | AIME2025 | 1.58 | 83.7 | 2.46 | 86.0 | 3.31 | 86.1 | 2.58 | **87.6** | 2.48 | 85.9 | **1.42** (-42.8%) | 87.3 |
| | BRUMO2025 | **1.32** | 93.8 | 2.06 | 95.2 | 2.95 | 95.6 | 2.18 | **95.8** | 2.13 | 95.1 | 1.35 (-36.5%) | 95.4 |
| | HMMT2025 | **1.94** | 73.8 | 2.96 | 76.2 | 3.78 | 75.7 | 3.04 | 78.0 | 2.93 | 75.9 | 2.16 (-26.3%) | **78.1** |
| | Avg. | 1.56 | 84.9 | 2.41 | 86.8 | 3.24 | 86.9 | 2.50 | 88.1 | 2.43 | 86.7 | **1.47** (-39.3%) | **88.2** |
| DeepSeek-R1-70B | AIME2024 | 1.30 | 91.4 | 2.00 | 92.7 | 2.68 | 93.3 | 2.10 | 93.5 | 2.02 | 92.7 | **0.86** (-57.4%) | **93.7** |
| | AIME2025 | 1.44 | 88.2 | 2.26 | 90.0 | 3.04 | 90.8 | 2.42 | **91.1** | 2.29 | 90.0 | **1.18** (-48.5%) | 90.5 |
| | BRUMO2025 | **1.24** | 94.8 | 1.94 | 96.1 | 2.70 | 96.7 | 2.08 | **96.8** | 1.99 | 96.1 | 1.26 (-36.7%) | 96.2 |
| | HMMT2025 | **1.82** | 78.8 | 2.76 | 81.4 | 3.56 | 81.6 | 2.86 | 82.3 | 2.75 | 81.0 | 1.96 (-28.7%) | **82.4** |
| | Avg. | 1.45 | 88.3 | 2.24 | 90.0 | 3.00 | 90.6 | 2.37 | **90.9** | 2.27 | 90.0 | **1.31** (-41.9%) | 90.7 |

in different ways: SC expands the number of full reasoning trajectories, while BoN spends extra computation on candidate generation and selection.

The cost pattern clarifies the limitation of broad test-time scaling. Increasing the Self-Consistency budget usually improves accuracy on difficult mathematical benchmarks, especially AIME2025 and HMMT2025, but the gains are uneven and come with rapidly increasing token usage. SC@512 and BoN@32 often achieve strong accuracy, yet their advantage is purchased through many complete trajectories, including long stretches where the model is already following a stable continuation.

Against these additional strategies, EHETI gives a different cost profile. It usually reduces token usage substantially relative to the average comparison strategy, while reaching the best or near-best accuracy across models. The advantage is clearest when the competing methods spend many tokens searching over entire solutions, as in AIME2024 and AIME2025. On harder cases such as HMMT2025, the reduction becomes smaller because uncertainty persists across more of the derivation, yet the method still preserves a favourable cost-accuracy balance. This supports the broader

claim from the main cost table: efficient reasoning depends less on generating more complete candidates and more on applying computation where the trajectory is still genuinely unstable.

### F.5 Practical Inference-Cost Analysis

Generated-token counts do not capture the auxiliary computation introduced by the intervention. We therefore measure end-to-end wall-clock latency, the number of auxiliary forward passes, and peak GPU memory on the same NVIDIA B200 cluster used for the main experiments. The analysis covers DeepSeek-R1-8B, Qwen3-32B, and GPT-OSS-120B on the four competition-style reasoning benchmarks used in the main cost comparison. Each activated intervention step requires one auxiliary forward pass, so the average number of auxiliary passes directly reflects the average intervention count per response. Tables 11 and 12 report the latency and memory results, respectively.

Table 11: End-to-end wall-clock latency in seconds per response on the NVIDIA B200 cluster. Lower values are better, and the best result in each row is shown in bold.

| Model | Dataset | MTI | SC@32 | Self-Refine-2 | SC@512 | BoN@32 | EHETI |
|---|---|---|---|---|---|---|---|
| DeepSeek-R1-8B | AIME2024 | 10.1 | 18.9 | 20.7 | 286.4 | 21.6 | **8.2** |
| | AIME2025 | 15.6 | 28.1 | 29.4 | 425.8 | 31.9 | **12.6** |
| | BRUMO2025 | 14.2 | 25.4 | 26.8 | 386.9 | 29.2 | **11.7** |
| | HMMT2025 | 22.7 | 40.8 | 39.5 | 619.7 | 46.5 | **18.3** |
| Qwen3-32B | AIME2024 | 15.2 | 25.6 | 29.8 | 389.7 | 29.3 | **12.9** |
| | AIME2025 | 25.8 | 42.9 | 48.7 | 652.3 | 49.2 | **21.4** |
| | BRUMO2025 | 22.1 | 36.4 | 41.5 | 553.8 | 41.8 | **18.1** |
| | HMMT2025 | 37.4 | 59.7 | 66.8 | 907.6 | 68.5 | **31.5** |
| GPT-OSS-120B | AIME2024 | 21.7 | 35.9 | 42.4 | 547.8 | 41.2 | **18.8** |
| | AIME2025 | 25.0 | 41.3 | 48.8 | 630.2 | 47.6 | **21.3** |
| | BRUMO2025 | 34.6 | 56.2 | 66.9 | 857.1 | 64.7 | **29.0** |
| | HMMT2025 | 50.7 | 81.5 | 94.2 | 1243.6 | 93.4 | **43.7** |

Table 12: Average auxiliary (Aux.) forward passes for EHETI and peak GPU memory usage in GiB. Lower memory usage is better, and the best memory result in each row is shown in bold.

| Model | EHETI Aux. Passes | MTI | SC@32 | Self-Refine-2 | SC@512 | BoN@32 | EHETI |
|---|---|---|---|---|---|---|---|
| DeepSeek-R1-8B | 3.2 | **23.7** | 28.8 | 25.3 | 29.1 | 30.5 | 26.9 |
| Qwen3-32B | 2.6 | **74.9** | 83.5 | 77.4 | 84.2 | 85.7 | 79.1 |
| GPT-OSS-120B | 2.1 | 241.6 | 251.3 | 244.9 | 252.4 | 254.8 | **238.7** |

The intervention requires only 2.1–3.2 auxiliary forward passes per response on average, below the maximum budget $M = 4$. Despite this additional computation, EHETI achieves the lowest wall-clock latency in all reported settings. Peak GPU memory remains comparable to the lightweight baselines and below the broad sampling strategies for the larger models, showing that the generated-token reductions in the main analysis translate into practical inference efficiency.

### F.6 Response-Level Intervention Analysis

We next examine intervention behaviour at the response level on three representative models and three benchmarks. For each response, coverage records whether at least one intervention is triggered, and the intervention count records the number of activated steps. Responses with no activated intervention are assigned a count of zero and are included when computing the mean and standard deviation. P25, P50, and P75 denote the 25th, 50th, and 75th percentiles of the absolute decoding-token positions of the interventions. Rel. P50 reports the median relative (Rel.) intervention position after normalising by the total generated response length. DI and EHETI report accuracy for MMLU-Pro and MATH500 and Pass@1 for LiveCodeBench, all in percentages. Table 13 summarises these response-level statistics.

To measure whether these local interventions alter final outcomes, we pair the Direct Inference and EHETI results for each evaluated instance and classify the transition in final-answer correctness. $W$

Table 13: Response-level intervention statistics. DI and EHETI performance, intervention coverage, and relative median position are reported as percentages; the intervention count is reported as mean $\pm$ standard deviation. The better DI/EHETI performance is shown in bold.

| Model | Dataset | Performance | | Intervention frequency | | Intervention position | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | DI | EHETI | Coverage | Mean count | P25 | P50 | P75 | Rel. P50 |
| DeepSeek-R1-8B | MMLU-Pro | 50.13 | **53.84** | 94.17 | $3.42 \pm 0.99$ | 13 | 36 | 74 | 5.21 |
| | MATH500 | 89.04 | **93.26** | 98.36 | $3.76 \pm 0.64$ | 11 | 32 | 67 | 1.43 |
| | LiveCodeBench | 39.57 | **51.04** | 97.12 | $3.69 \pm 0.77$ | 14 | 39 | 81 | 2.41 |
| Qwen3-32B | MMLU-Pro | 76.76 | **78.06** | 90.48 | $3.08 \pm 1.13$ | 17 | 44 | 86 | 5.74 |
| | MATH500 | 96.94 | **98.73** | 95.63 | $3.47 \pm 0.90$ | 14 | 40 | 78 | 1.67 |
| | LiveCodeBench | 91.02 | **93.62** | 93.86 | $3.36 \pm 1.00$ | 16 | 46 | 90 | 2.68 |
| GPT-OSS-120B | MMLU-Pro | 82.47 | **84.42** | 86.79 | $2.84 \pm 1.21$ | 20 | 51 | 95 | 6.13 |
| | MATH500 | 98.36 | **99.65** | 92.54 | $3.26 \pm 1.06$ | 17 | 45 | 84 | 1.79 |
| | LiveCodeBench | 79.94 | **84.52** | 91.28 | $3.18 \pm 1.11$ | 19 | 50 | 97 | 2.87 |

Table 14: Final-answer correctness transitions from Direct Inference to EHETI. Values are percentages of evaluated instances.

| Model | Dataset | $\mathbf{W} \to \mathbf{R}$ | $\mathbf{R} \to \mathbf{W}$ | $\mathbf{W} \to \mathbf{W}$ | $\mathbf{R} \to \mathbf{R}$ |
|---|---|---|---|---|---|
| DeepSeek-R1-8B | MMLU-Pro | 5.02 | 1.31 | 44.85 | 48.82 |
| | MATH500 | 4.89 | 0.67 | 6.07 | 88.37 |
| | LiveCodeBench | 13.08 | 1.61 | 47.35 | 37.96 |
| Qwen3-32B | MMLU-Pro | 1.98 | 0.68 | 21.26 | 76.08 |
| | MATH500 | 2.02 | 0.23 | 1.04 | 96.71 |
| | LiveCodeBench | 3.07 | 0.47 | 5.91 | 90.55 |
| GPT-OSS-120B | MMLU-Pro | 2.57 | 0.62 | 14.96 | 81.85 |
| | MATH500 | 1.41 | 0.12 | 0.23 | 98.24 |
| | LiveCodeBench | 5.12 | 0.54 | 14.94 | 79.40 |

and $R$ denote wrong and right final answers, respectively. The four transition categories partition the evaluated instances, and Table 14 reports their percentages.

Across these settings, intervention coverage ranges from $86.79\%$ to $98.36\%$, while the average intervention count remains between $2.84$ and $3.76$ per response. The median relative intervention position lies within the first $1.43\%$–$6.13\%$ of the generated response, confirming that the intervention remains concentrated near the beginning of generation. In every setting, the $W \to R$ rate exceeds the $R \to W$ rate, showing that beneficial final-answer changes consistently outweigh harmful reversals.

## G Entropy-Grouped Temperature Sensitivity

This appendix describes the diagnostic temperature-sensitivity experiment reported in Figure 5. The experiment is conducted with DeepSeek-R1-8B and uses group-specific temperature perturbation to compare how different entropy groups affect reasoning accuracy. Model parameters are fixed, and no logit correction is applied. The only change is the temperature assigned to each entropy group during decoding.

For each decoding step $t$, let $\mathbf{z}_t \in \mathbb{R}^V$ denote the original logits, where $V$ is the vocabulary size. We first compute the pre-perturbation predictive distribution,

$$\mathbf{p}_t = \mathrm{Softmax}\left(\frac{\mathbf{z}_t}{T_{\mathrm{dec}}}\right). \tag{43}$$

The corresponding token entropy is

$$H_t = -\sum_{i=1}^{V} p_{t,i} \log p_{t,i}. \tag{44}$$

We define the entropy threshold $h_{\mathrm{th}}$ as the empirical $80$th percentile of token entropy values. Positions with $H_t > h_{\mathrm{th}}$ are treated as high-entropy positions, while the remaining positions are treated as lower-entropy positions.

To isolate the effect of perturbing each group, we replace the default decoding temperature with a group-specific temperature,

$$T'_t = \begin{cases} T_{\text{high}}, & H_t > h_{\text{th}}, \\ T_{\text{low}}, & H_t \le h_{\text{th}}. \end{cases} \tag{45}$$

The perturbed decoding distribution is then

$$\mathbf{p}'_t = \text{Softmax}\left(\frac{\mathbf{z}_t}{T'_t}\right). \tag{46}$$

The $T = 0$ setting is implemented as the argmax limit of temperature sampling.

We consider two sweeps. In the first, $T_{\text{low}}$ is fixed to 1 and $T_{\text{high}}$ is varied. In the second, $T_{\text{high}}$ is fixed to 1 and $T_{\text{low}}$ is varied. All other decoding settings remain unchanged. Evaluation is conducted on AIME 2024 and AIME 2025, and we report the average accuracy. Since entropy groups are determined before group-specific temperature scaling, the experiment isolates whether reasoning performance is more sensitive to perturbations at high-entropy or lower-entropy positions.

## H Case Analysis

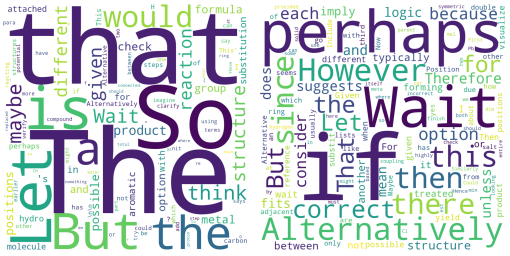


Figure 14: Word clouds of generated tokens before and after intervention on GPQA-Diamond. The left panel shows the original generation, and the right panel shows the generation after applying EHETI.

Figure 14 compares token word clouds before and after applying EHETI on GPQA-Diamond using DeepSeek-R1-8B under the same inference setting as the main experiments. We focus on instances where the original generation is incorrect but the intervention leads to the correct answer, so the comparison highlights lexical changes associated with successful trajectory correction. The left panel corresponds to the original generation, while the right panel shows the generation after intervention.

In the original setting, the word cloud is relatively dominated by generic function words such as *the*, *is*, *but*, and *that*. These tokens are syntactically valid, but they provide limited information about reasoning structure and often reflect continuation along an already selected path. After intervention, structure-indicating reasoning tokens become more visible, including *if*, *however*, *perhaps*, *wait*, and *alternatively*. These tokens are associated with conditional reasoning, uncertainty expression, reconsideration, and path adjustment, indicating a more active reorganisation of the generation process around critical positions.

This contrast suggests that the intervention does not simply inject arbitrary variation. By selectively influencing key generation positions, it encourages the model to move away from inert continuation and towards language patterns that better support reasoning revision and exploration of alternative paths.