

Deeper is Not Always Better: Mitigating the Alignment Tax via Confident Layer Decoding

Sining Zhou^{†*2}, Xuanming Zhang^{†*1}, Yuxuan Chen¹, Tianyi Tang¹, An Yang¹
Sean Du³, Chujie Zheng¹, Fei Huang¹, Dayiheng Liu¹, Gao Huang², Jingren Zhou¹

¹ Qwen Team, Alibaba Inc. ² Tsinghua University ³ Nanyang Technological University
xuemuqiangu@gmail.com zbsn21@mails.tsinghua.edu.cn

<https://github.com/QwenLM/Confident-Decoding>

Abstract

Autoregressive generation in large language models (LLMs) conventionally decodes from the final layer, assuming that deeper representations yield more reliable next-token predictions. We revisit this assumption by revealing a recurring *Guess–Refine–Perturb* dynamic: early layers form coarse guesses, intermediate layers refine reasoning-relevant semantics, and final layers can perturb these refined predictions toward generic or alignment-preferred tokens. We introduce *Confident Decoding*, a training-free decoding strategy that dynamically selects a near-final layer through entropy-guided conservative backward search. We further formulate layer selection as an optimal stopping problem and show that, under explicit local-monotonicity and bounded-probe-noise conditions, the search rule filters terminal perturbations while bounding unfinished refinement. Experiments across dense and Mixture-of-Experts LLMs show that gains concentrate on challenging reasoning settings, including GPQA-Diamond, Omni-MATH, and HLE, while saturated tasks remain broadly stable, with no additional KV-cache memory and less than 2% latency increase. These results suggest dynamically bypassing final-layer perturbations can recover stronger reasoning behavior from aligned LLMs.

1 Introduction

Autoregressive generation in large language models (LLMs) conventionally decodes each token from the final decoder layer. This standard practice implicitly assumes that representations become progressively more reliable with depth: shallow layers provide incomplete computations, intermediate layers refine contextual information, and the deepest

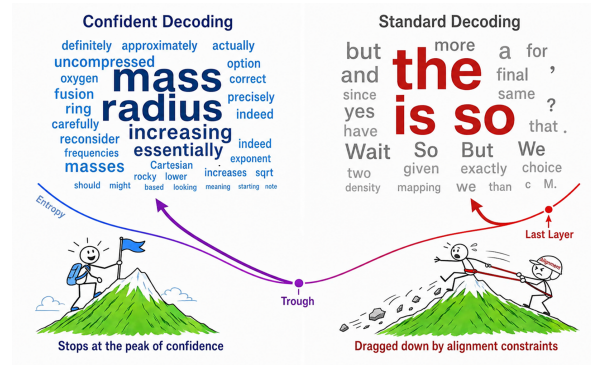


Figure 1: Token substitutions produced by Confident Decoding on Qwen3.5-35B-A3B. When the strategies disagree ($\sim 2\%$ of generated tokens), Standard Decoding (right) more often selects generic, high-frequency function words and punctuation (e.g., “the”, “is”, “so”, “.”), whereas Confident Decoding (left) more often selects content-bearing terms (e.g., “mass”, “radius”, “approximately”, “Cartesian”). Word sizes are proportional to substitution frequency. The central curve illustrates predictive entropy $H(p_t^{(l)})$ across depth; the “Trough” marks the local entropy valley selected before a possible terminal distributional shift.

layer produces the most accurate next-token distribution (Vaswani et al., 2017; Shazeer et al., 2017; Gupta et al., 2025). Under this view, the final layer is treated as the natural and optimal interface between the model’s internal computation and the output vocabulary.

However, recent evidence suggests that this monotonic-depth assumption can break down. Probing and layer-wise analyses show that intermediate layers often encode strong task-relevant semantics, while later layers may compress, redirect, or perturb information that has already been refined (Skean et al., 2025; Csordás et al., 2025). Through a detailed analysis of residual-stream dynamics, contribution norms, and prediction trajectories, we identify a recurring *Guess–Refine–Perturb* pattern in LLM forward passes (see §2.1

[†]Equal contribution. Listed alphabetically.

*Corresponding author.

and Figure 2). Early layers produce coarse statistical guesses; intermediate layers refine these guesses along a stable semantic trajectory; and, for a subset of tokens, the final layers introduce a sharp representational shift that can move the prediction away from the most reasoning-relevant intermediate state.

We hypothesize that this late-stage perturbation is partly related to the objectives used in modern post-training. Contemporary LLMs are typically adapted through continued pre-training, supervised fine-tuning, and preference-alignment procedures such as RLHF, RLAIF, or DPO (Zixuan et al., 2023; Ouyang et al., 2022; Lee et al., 2024; Rafailov et al., 2023). These procedures improve instruction following, safety, and response style, but they may also amplify a distributional preference for frequent, safe, or generic continuations (Askell et al., 2021; Zou et al., 2023). For ordinary conversational or safety-oriented prompts, such behavior can serve as a useful guardrail. For complex reasoning tasks, however, the same late-stage bias may conflict with the task-specific reasoning trajectory formed in intermediate layers, producing what we call a *planning–pragmatics tradeoff*: the model may internally form a strong reasoning-oriented prediction, but the final layer may pragmatically shift the output distribution toward a more generic or alignment-preferred token.

Several recent decoding strategies exploit intermediate-layer information. Contrastive decoding methods (Chuang et al., 2023; Zhou et al., 2025) compare predictions across layers to suppress undesirable continuations, while logits-evolution methods study how token distributions change through depth (Zhang et al., 2024; Das et al., 2025; Zhang et al., 2025b). Early-exit methods also use intermediate confidence signals to reduce inference cost (Schuster et al., 2022; Yang et al., 2025). Yet these approaches either continue to treat the final layer as the main semantic anchor or focus primarily on efficiency rather than reasoning degradation. They do not directly ask whether the final layer should always be the layer from which the next-token distribution is emitted.

To address this question, we propose *Confident Decoding*, a training-free, drop-in decoding strategy that dynamically selects the most reliable near-final layer at each generation step. Confident Decoding does not truncate the transformer or modify the model’s forward pass. Instead of always forwarding final-layer logits to the sampler, it com-

putes candidate logits from a small near-final layer window and selects the first local entropy trough encountered by a conservative backward search. Since lower token entropy corresponds to a sharper predictive distribution, this rule identifies the point at which the model reaches high confidence before a potential late-layer perturbation emerges (see Figure 1 for a representative example).

We evaluate Confident Decoding on a broad range of challenging benchmarks spanning general reasoning, mathematical problem solving, long-context understanding, coding, safety and open-ended composition, including GPQA-Diamond (Rein et al., 2024), HLE (Phan et al., 2026), Omni-MATH (Gao et al., 2024), LongBench v2 (Bai et al., 2025), LiveCodeBench v6 (Jain et al., 2024), Air-Bench-2024 (Zeng et al., 2025) and WritingBench (Wu et al., 2026). Across dense and Mixture-of-Experts architectures, Confident Decoding produces its largest gains on difficult reasoning and coding settings, while performance on saturated, safety, and open-ended tasks is generally similar to final-layer decoding. It introduces no KV-cache overhead and less than 2% additional latency. We further observe that stronger models and harder tasks can exhibit more pronounced late-layer perturbations.

In summary, our contributions are threefold:

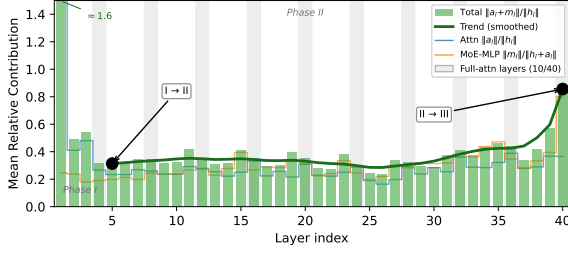
- We identify a recurring *Guess–Refine–Perturb* dynamic in LLM forward passes, showing that final-layer representations are not always the most reliable source for reasoning-sensitive next-token prediction.
- We propose *Confident Decoding*, a training-free and drop-in decoding strategy that preserves the full forward pass while dynamically selecting near-final logits through entropy-guided conservative backward search.
- We provide theoretical (§3) and empirical (§5) evidence that Confident Decoding can filter late-layer perturbations and preserve refined reasoning signals, with gains mainly concentrated on challenging reasoning and coding settings.

2 Preliminaries

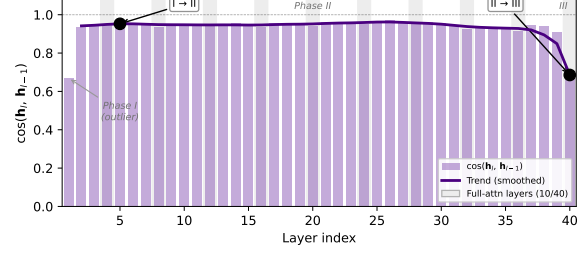
First, we characterize the forward-pass dynamics of LLMs to motivate Confident Decoding.

2.1 Layer-wise Dynamics

Consider an L -layer autoregressive Transformer with vocabulary $\mathcal{V}$. At step t , the residual stream



(a) Relative Contribution Norm



(b) Residual I/O Cosine Similarity

Figure 2: Layer-wise dynamics of Qwen3.5-35B-A3B on GSM8K (Cobbe et al., 2021). Gray bands mark the 10 full-attention layers at $l = 4, 8, \dots, 40$. (a) The first decoder layer’s update dominates the embedding magnitude; contributions stabilize through Phase II; the final full-attention layer ($l = 40$) resurges markedly in Phase III. (b) IO-CosSim remains high throughout Phase II, confirming directionally faithful refinement, before dropping sharply at $l = 40$ —the largest directional deflection in the Phase II–III regime.

at layer l is $\mathbf{h}_t^{(l)} \in \mathbb{R}^d$. We define the *layer contribution vector* as $\mathbf{m}_t^{(l)} \triangleq \mathbf{h}_t^{(l)} - \mathbf{h}_t^{(l-1)}$, thus $\mathbf{h}_t^{(l)} = \mathbf{h}_t^{(l-1)} + \mathbf{m}_t^{(l)}$. Then we track two metrics: the *Relative Contribution Norm* — Norm Ratio $^{(l)} = \|\mathbf{m}_t^{(l)}\|_2 / \|\mathbf{h}_t^{(l-1)}\|_2$ (write intensity) and the *Residual I/O Cosine Similarity* — IO-CosSim $^{(l)} = \langle \mathbf{h}_t^{(l)}, \mathbf{h}_t^{(l-1)} \rangle / (\|\mathbf{h}_t^{(l)}\|_2 \|\mathbf{h}_t^{(l-1)}\|_2)$ (directional fidelity). Detailed definitions are in Appendix F.

As shown in Figure 2, empirical observations on Qwen3.5-35B-A3B reveal a three-phase *Guess–Refine–Perturbation* progression. The fractional boundaries below are descriptive summaries of Figure 2, not inputs to our decoding algorithm:

- **Phase I: Guess** ($l \lesssim 0.15L$). High write intensity with substantial directional shift; the model constructs an initial latent representation under uncertainty.
- **Phase II: Refine** ($0.15L \lesssim l \lesssim 0.95L$). Stabilized write intensity (0.23–0.57) and high directional fidelity (0.91–0.97); each layer applies incremental refinement without displacing the accumulated representation.
- **Phase III: Perturbation** ($l \gtrsim 0.95L$). Elevated write intensity (2–3× Phase II) with reduced directional fidelity (≈ 0.69); a directionally misaligned update that partially rewrites the Phase II trajectory. We formalize this in Section 3.

This Phase III perturbation breaks the monotonic improvement assumption, suggesting that extracting predictions near the end of Phase II may instead yield superior reasoning accuracy.

2.2 Emerging Entropy Valley

As previously discussed, LLMs often converge on stable semantic representations before Phase III.

We therefore seek a reliable intermediate proxy to bypass potential late-stage perturbations (figure 3).

Limitations of Static Early Exit. A naive static early-exit strategy forces decoding from a fixed shallower layer. But increasing exit probability leads to precipitous accuracy decline, since static truncation ignores token complexity variance.

Entropy as Confidence Metric. We apply the model’s exact final RMSNorm (including its learned scale γ) and unmodified unembedding W_U to each intermediate state, yielding $p_t^{(l)} = \text{Softmax}(W_U \cdot \text{RMSNorm}_L(\mathbf{h}_t^{(l)}))$ and entropy $H(p_t^{(l)})$. In the late refinement zone, IO-CosSim remains 0.91–0.97 (Figure 2), providing empirical support for the representation homogeneity required by this shared- W_U probe. This support is architectural rather than universal; Appendix E documents the compact Qwen3.5-9B failure mode when layer-type alternation weakens the assumption.

Defining the Entropy Valley. We propose the **Entropy Valley**: the local entropy minimum encountered when scanning backward from the final layer, representing consensus before detrimental oscillations emerge. This is validated by Figure 3.

3 Theoretical Grounding

In this section, we analyze forward-pass dynamics through Information Theory and Representation Engineering. Full derivations are in Appendix G.

3.1 Information-Theoretic View of Layer Dynamics

The Information Bottleneck (IB) principle (Tishby and Zaslavsky, 2015) provides a motivating view of layer-wise information flow. Let X represent the input context and Y_{logic} the optimal target to-

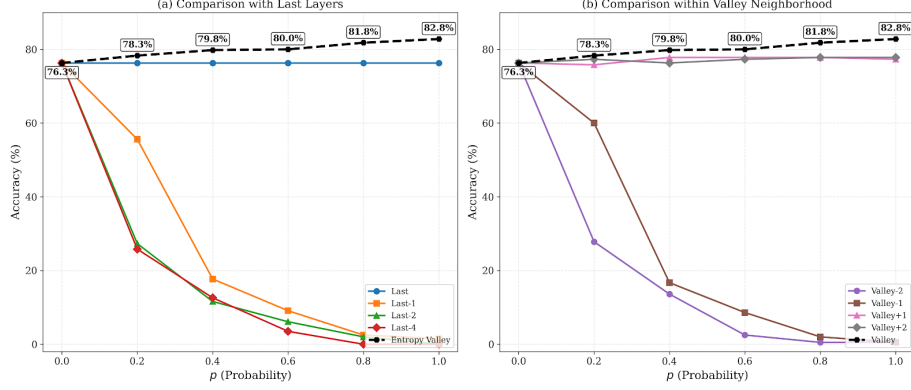


Figure 3: Comparison of layer-selection strategies on GPQA-Diamond, using Qwen3.5-35B-A3B. Dynamic valley selection outperforms last-layer decoding and static early exits, while selecting layers neighboring the valley consistently degrades performance. This validates the effectiveness of Entropy Valley based selection.

ken from domain-specific reasoning. We use the tradeoff $\mathcal{L}_{IB} = I(X; \mathbf{h}_t^{(l)}) - \beta I(\mathbf{h}_t^{(l)}; Y_{\text{logic}})$ to motivate, rather than identify, the late-layer refinement assumption used below.

In **Phase I (Guess)**, aggressive compression is associated with geometrically volatile predictive entropy $H(p_t^{(l)})$. The network transitions into **Phase II (Refine)**, where attention mechanisms integrate contextual dependencies. We denote by V_{onset} the empirical onset of a late monotone-refinement corridor, within which the observable entropy decreases toward a local minimum V^* . This monotonicity is an explicit local assumption, supported by the measured late-layer trajectories, rather than a consequence inferred from IB alone.

The minimum at V^* represents the representational zenith before potential perturbation. As shown in Figure 4(a), on Qwen3.5-35B-A3B over GPQA-Diamond, Phase II accumulates $\Delta H \approx -10.8$ nats ($V_{\text{onset}} \approx 28$, $V^* = 39$). Crucially, Phase III is a *per-token* phenomenon: 16.2% of tokens exhibit an entropy rise of $+0.37$ nats at $l=40$ (undergo Phase III), while the remaining 83.8% show a further drop of -2.52 nats (complete normal Phase II refinement).

3.2 Alignment: Tax vs. Guardrail

Modern LLMs undergo post-training for human preference alignment. Prior representation-engineering work identifies low-dimensional directions associated with aligned behavior (Zou et al., 2023). Motivated by this observation, we model the **Phase III (Perturbation)** shift as a regularized risk toward a generic, safe distribution P_{align} :

$$\mathcal{R}^{(l)} = \mathbb{E}[-\log p_t^{(l)}(Y_{\text{logic}}|X)] + \lambda \mathcal{D}_{\text{KL}}(P_{\text{logic}}^{(l)} \| P_{\text{align}})$$

For $l \rightarrow L$, the network induces additive perturbation $\delta_{\text{align}}^{(l)}: \mathbf{h}_t^{(l)} \approx \mathbf{h}_t^{(V^*)} + \sum_{k=V^*+1}^l \delta_{\text{align}}^{(k)}$.

The destructive nature is conditional: for conversational or safety queries, P_{logic} may be close to P_{align} and late layers can refine formatting (**Alignment as Guardrail**). For complex reasoning, the two distributions may conflict; the late update can then move logits toward generic, safe, or high-frequency tokens at the expense of a fragile reasoning prediction (**Alignment as Tax**). The observed base-model gain (Table 3) suggests that a generic-token prior can also arise during pre-training. We therefore treat preference post-training as an *amplifier*, not a necessary cause of this effect.

Moreover, why this amplification localizes late remains an empirical question. A plausible intuition is that post-training can satisfy preference objectives through relatively localized output-facing changes while preserving much of the pre-trained computation, but training recipes differ across released checkpoints. We therefore use the observed late write-intensity, IO-CosSim shift, and Instruct-Base gap as evidence of localization, rather than a proof for training-dynamics.

3.3 Conservative Local Optimality of Backward Search

Beyond the network midpoint, late-layer hidden states become approximately linearly decodable through the frozen unembedding W_U (Belrose et al., 2023; Elhoushi et al., 2024; Park et al., 2024); since $V_{\text{onset}} \gtrsim 0.5L$, the projection noise $\epsilon^{(l)}$ is bounded by ϵ_{max} . We formulate dynamic layer selection as an Optimal Stopping Problem. For analysis, the observable entropy decomposes

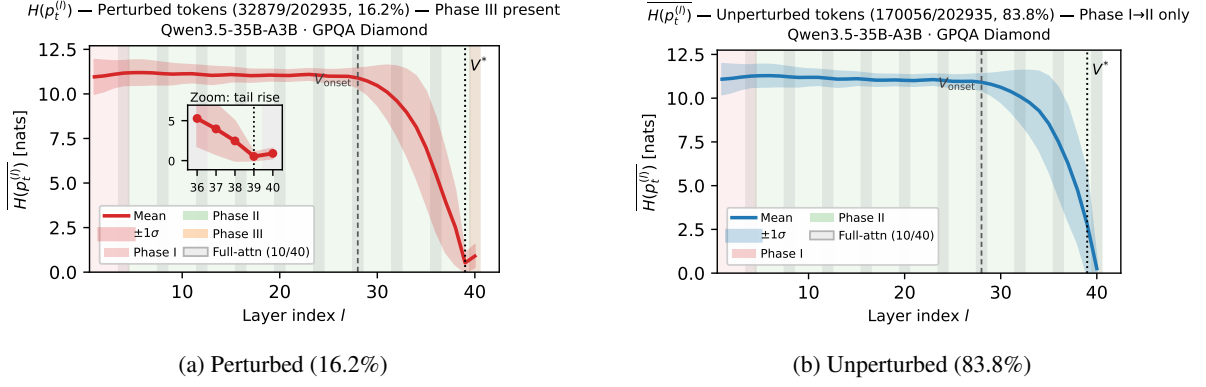


Figure 4: Mean logit-lens entropy per layer for Qwen3.5-35B-A3B on GPQA-Diamond. **(a)** Perturbed tokens rise at $l = 40$ ($\Delta H = +0.37$): the alignment-tax signature. **(b)** Unperturbed tokens continue Phase II refinement ($\Delta H = -2.52$).

as $\hat{H}(l) = H^*(l) + \epsilon^{(l)} + \eta^{(l)}$, where $H^*(l)$ is monotone in the assumed refinement corridor ($\Delta H^*(l) \leq 0$ for $l \in [V_{\text{onset}}, V^*]$), $\epsilon^{(l)}$ is bounded projection noise ($|\epsilon^{(l)}| \leq \epsilon_{\text{max}}$), and $\eta^{(l)}$ is a per-token terminal perturbation. These components are not separately identifiable from one observed trajectory. The decomposition states sufficient conditions for the stopping rule rather than providing runtime estimators.

Theorem 1 (Conservative Local Optimality). *Let $\hat{V} = \max\{l < L \mid \hat{H}(l-1) \geq \hat{H}(l)\}$. Conditional on bounded projection noise, a locally monotone refinement corridor, and dominant late-stage perturbation, the conservative backward scan returns $\hat{V} \in [V_{\text{onset}}, V^*]$, filtering terminal perturbations while bounding unfinished refinement.*

Appendix G states the full conditions and proof. In brief, a dominant terminal perturbation prevents stopping beyond V^* , while local monotonicity confines any earlier stop to the late refinement corridor. The result is a conditional local guarantee, not a claim that the latent terms can be recovered from observed entropy.

4 Confident Decoding

Standard autoregressive decoding emits the next-token distribution from the final transformer layer. Now we formally propose **Confident Decoding**, a training-free, drop-in inference-time procedure that dynamically selects the most reliable near-final layer at each generation step. The key insight is that lower token entropy corresponds to a sharper, more confident predictive distribution: by scanning backward for the first entropy trough, we identify the layer where the model has reached peak confidence before potential late-layer perturbation emerges.

Formally, with a decoder-only language model of L transformer blocks, hidden size d , vocabulary $\mathcal{V}$, and unmodified unembedding $W_U \in \mathbb{R}^{|\mathcal{V}| \times d}$, the residual-stream state after block ℓ at step t is $\mathbf{x}_t^{(\ell)} \in \mathbb{R}^d$. After applying the model’s exact final normalization (including the learned RMSNorm scale γ), the candidate distribution and entropy at layer ℓ are:

$$\tilde{\mathbf{h}}_t^{(\ell)} = \text{Norm}(\mathbf{x}_t^{(\ell)}), \quad \mathbf{z}_t^{(\ell)} = W_U \tilde{\mathbf{h}}_t^{(\ell)}, \quad (1)$$

$$\mathbf{p}_t^{(\ell)} = \text{softmax}(\mathbf{z}_t^{(\ell)}), \quad H_t^{(\ell)} = -\langle \mathbf{p}_t^{(\ell)}, \log \mathbf{p}_t^{(\ell)} \rangle. \quad (2)$$

Specifically, standard decoding uses $\mathbf{z}_t^{(L)}$, while Confident Decoding selects ℓ_t^* from a near-final window (Algorithm 1).

Confident Decoding restricts attention to a near-final candidate set $\mathcal{C} = \{L - M + 1, \dots, L\}$ of size M (typically $M = K$, the scan window in Algorithm 1). It selects the first local entropy trough while scanning $\mathcal{C}$ backward:

$$\ell_t^* = \min\{\ell \in [L - M + 1, L] : H_t^{(\ell)} < H_t^{(\ell+1)} < \dots < H_t^{(L)}\}. \quad (3)$$

The fallback probability $1 - p$ provides stochastic mixing with standard decoding: $p = 0$ recovers final-layer decoding exactly, while $p = 1$ corresponds to deterministic valley selection (used as the default setting in our main experiments).

The per-step computational overhead consists of one batched unembedding of M candidate hidden states ($\mathcal{O}(MBd|\mathcal{V}|)$), entropy evaluation ($\mathcal{O}(MB|\mathcal{V}|)$), and a K -step trough scan ($\mathcal{O}(KB)$). Critically, Confident Decoding does not modify the forward pass or KV cache, making it composable with existing inference optimizations. Full implementation details are in Appendix H.

Algorithm 1 Confident Decoding step (per active token)

Input: normed candidates $\{\tilde{\mathbf{h}}_t^{(\ell)}\}_{\ell=L-M+1}^L$, W_U , scan window K , fallback prob. $1 - p$.

Output: logits $\mathbf{z}_t$ delivered to the sampler.

```
1: Compute in parallel for all  $\ell \in \mathcal{C}$ :  $\mathbf{z}_t^{(\ell)} \leftarrow W_U \tilde{\mathbf{h}}_t^{(\ell)}$ ,  $\mathbf{p}_t^{(\ell)} \leftarrow \text{softmax}(\mathbf{z}_t^{(\ell)})$ ,  $H_t^{(\ell)} \leftarrow -\langle \mathbf{p}_t^{(\ell)}, \log \mathbf{p}_t^{(\ell)} \rangle$ 
2:  $\ell^* \leftarrow L$ ,  $H_{\text{ref}} \leftarrow H_t^{(L)}$ , frozen  $\leftarrow$  false
3: for  $\ell = L - 1$  downto  $\max(1, L - K)$  do
4:   if  $\neg$ frozen and  $H_t^{(\ell)} < H_{\text{ref}}$  then
5:      $\ell^* \leftarrow \ell$ 
6:   else
7:     frozen  $\leftarrow$  true
8:   end if
9:    $H_{\text{ref}} \leftarrow H_t^{(\ell)}$ 
10: end for
11: with prob.  $1 - p$ :  $\ell^* \leftarrow L$  {stochastic fallback}
12: return  $\mathbf{z}_t^{(\ell^*)}$ 
```

5 Experiments

In this section, we present a comprehensive evaluation of Confident Decoding to: (1) demonstrate effectiveness across diverse reasoning-related tasks, (2) evaluate generalization across state-of-the-art open-source architectures, and (3) empirically validate the “Alignment Tax” hypothesis through instruct-base model comparisons.

5.1 Setup

Implementation. We implement the backward scan with lookback window $K = 10$ and set $p = 1.0$ (deterministic valley selection). The sampling temperature is 0 unless specified, consistent with the baseline. The choice of $K = 10$ is driven by algorithmic efficiency: empirically, the entropy valley located by the backward scan rarely lies more than 5 layers below the final layer. A sensitivity analysis on hyperparameters is provided in Appendix A.

Benchmarks. We select seven high-difficulty benchmarks covering general reasoning, mathematics, long-context, coding, safety, and creative writing: **GPQA-Diamond** (Rein et al., 2024) (graduate-level science) and **HLE** (Phan et al., 2026) (frontier reasoning); **Omni-MATH** (Gao et al., 2024) (Olympiad mathematics); **LongBench v2** (Bai et al., 2025) (long-context understanding); **LiveCodeBench v6** (Jain et al., 2024) (code generation); **Air-Bench-2024** (Zeng et al., 2025) (safety, ensuring truncation does not degrade alignment guardrails); and **WritingBench** (Wu et al., 2026) (open-ended composition). Each benchmark

stresses a distinct cognitive faculty—from factual recall and logical deduction to code synthesis and stylistic generation—providing holistic coverage of Confident Decoding’s impact.

Model Families. To test whether the *Guess–Refine–Perturb* dynamic generalizes beyond a single backbone, we evaluate across dense and Mixture-of-Experts (MoE) architectures at varying parameter scales: the **Qwen3.5** series (the hybrid 27B and the MoE variants 35B-A3B and 122B-A10B), the open-source frontier **gpt-oss** series (20B and 120B), and the dense **Gemma-4-31B**. Evaluating across both dense and sparse architectures helps rule out architecture-specific artifacts and supports the view that the entropy valley phenomenon reflects a broad consequence of post-training alignment.

5.2 Main Results

Through Table 1, we obtain three insights regarding the systematic behavior of Confident Decoding.

1. Broad Efficacy Across Architectures. Confident Decoding yields consistent gains across the evaluated model families. The algorithm benefits both dense networks (Gemma-4) and sparse MoE topologies (Qwen-35B/122B, gpt-oss-20B/120B). This robustness suggests that the alignment tax is not tied to a single architectural choice, and that the entropy valley can serve as a practical anchor for recovering semantic fidelity. Gains persist at larger scales: gpt-oss-120B achieves an average improvement of +1.7%, while Qwen3.5-122B-A10B improves by +1.1%.

2. Gains Concentrate on Complex Reason-

Model	Decoding Strategy	GPQA-D	HLE	LCB-v6	LongBench-v2	Omni-MATH	Air-Bench	WritingBench
Qwen3.5-27B	Last Layer	78.2 $\pm$ 1.5	16.0 $\pm$ 0.2	63.9 $\pm$ 0.4	62.8 $\pm$ 0.8	76.0 $\pm$ 0.1	65.7$\pm$1.1	66.7$\pm$0.1
	Confident (Ours)	79.4$\pm$1.2	16.8$\pm$0.4	73.3$\pm$0.9	64.9$\pm$0.8	76.2$\pm$0.1	64.7 $\pm$ 0.6	66.4 $\pm$ 0.9
	Absolute Δ	+1.2	+0.8	+9.4	+2.1	+0.2	-1.0	-0.3
Qwen3.5-35B-A3B	Last Layer	76.3 $\pm$ 0.9	9.2 $\pm$ 2.1	70.1 $\pm$ 0.1	63.2 $\pm$ 0.6	72.3 $\pm$ 0.3	51.7 $\pm$ 1.1	65.2 $\pm$ 0.1
	Confident (Ours)	82.8$\pm$1.0	11.2$\pm$1.5	74.4$\pm$1.2	63.9$\pm$0.2	73.0$\pm$0.2	55.4$\pm$1.2	65.3$\pm$0.1
	Absolute Δ	+6.5	+2.0	+4.3	+0.7	+0.7	+3.7	+0.1
Qwen3.5-122B-A10B	Last Layer	83.3 $\pm$ 0.8	14.7 $\pm$ 0.5	76.8 $\pm$ 0.3	66.4 $\pm$ 0.4	78.3 $\pm$ 0.2	66.0 $\pm$ 0.8	72.6 $\pm$ 0.2
	Confident (Ours)	85.4$\pm$0.7	16.5$\pm$0.3	79.2$\pm$0.5	66.7$\pm$0.3	78.5$\pm$0.1	67.0$\pm$0.6	72.7$\pm$0.3
	Absolute Δ	+2.1	+1.8	+2.4	+0.3	+0.2	+1.0	+0.1
Gemma-4-31B	Last Layer	76.8 $\pm$ 1.2	9.2 $\pm$ 0.8	75.1 $\pm$ 0.3	58.6 $\pm$ 0.7	68.3 $\pm$ 0.3	74.0 $\pm$ 0.9	64.2 $\pm$ 0.2
	Confident (Ours)	80.8$\pm$0.9	11.0$\pm$0.6	78.3$\pm$0.5	59.1$\pm$0.5	69.0$\pm$0.2	76.0$\pm$0.7	64.4$\pm$0.3
	Absolute Δ	+4.0	+1.8	+3.2	+0.5	+0.7	+2.0	+0.2
gpt-oss-20b	Last Layer	58.1 $\pm$ 1.7	6.2 $\pm$ 0.6	77.1 $\pm$ 0.4	43.6 $\pm$ 1.0	61.3$\pm$0.4	86.7 $\pm$ 0.6	54.3 $\pm$ 0.2
	Confident (Ours)	60.8$\pm$0.6	6.3$\pm$0.1	77.6$\pm$0.5	44.7$\pm$1.4	61.0 $\pm$ 0.6	91.7$\pm$2.3	54.6$\pm$0.3
	Absolute Δ	+2.7	+0.1	+0.5	+1.1	-0.3	+5.0	+0.3
gpt-oss-120B	Last Layer	76.3 $\pm$ 1.0	10.8 $\pm$ 0.4	83.1 $\pm$ 0.3	54.7 $\pm$ 0.6	70.2 $\pm$ 0.3	88.0 $\pm$ 0.5	67.5 $\pm$ 0.2
	Confident (Ours)	80.8$\pm$0.8	12.6$\pm$0.3	84.5$\pm$0.4	55.6$\pm$0.5	71.0$\pm$0.2	90.0$\pm$0.4	67.8$\pm$0.2
	Absolute Δ	+4.5	+1.8	+1.4	+0.9	+0.8	+2.0	+0.3

Table 1: Performance across model families. Reported scores are means over three full runs, and $\pm$ denotes the unbiased sample SD across run-level scores. Confident Decoding’s largest gains occur on difficult reasoning and coding tasks, while several saturated or open-ended settings remain similar.

ing. The planning–pragmatics tradeoff dictates that the alignment tax is most destructive when generating fragile, low-frequency logic chains. Confident Decoding aggressively neutralizes this threat: on the highly syntactic LiveCodeBench (LCB-v6), Qwen3.5-27B experiences a +9.4% absolute leap; scientific benchmarks like GPQA-D and HLE witness ubiquitous growth (e.g., +6.5% for Qwen-35B-A3B on GPQA-D, +4.5% for gpt-oss-120B on GPQA-D). Together with the difficulty-stratified results in Appendix J, this pattern supports the interpretation that bypassing certain terminal shifts can recover reasoning-relevant predictions and unlock the latent reasoning ceiling.

3. Pristine Stability in Creativity and Safety.

A critical concern with truncating final layers is potential loss of stylistic formatting, creative writing quality, or safety guardrails. Our evaluation dispels this: WritingBench shows decent stability (marginal variations of $\pm 0.3\%$), indicating that the algorithm preserves stylistic structures synthesized during the *Refine* phase. Air-Bench results are mixed but mostly flat or improved (+3.7% for Qwen-35B-A3B, +2.0% for Gemma-4-31B), suggesting that removing alignment-induced common-word bias can maintain or even enhance rather than compromise safety responses. Besides, since standard safety prompts do not test adversarial robustness, we separately report JailbreakBench in Appendix D.

We further compare against representative contrastive decoding methods, including DoLa (Chuang et al., 2023) and SLED (Zhang et al., 2024) on both MoE and dense backbones:

Backbone	Method	GPQA-D	LCB-v6	Omni-Math
Qwen3.5-35B-A3B	DoLa	77.3	70.9	72.4
	SLED	78.8	71.7	72.1
	Confident	82.8	74.4	73.0
Qwen3.5-27B	DoLa	79.1	66.3	76.1
	SLED	79.2	67.8	75.9
	Confident	79.4	73.3	76.2
Gemma-4-31B	DoLa	78.3	76.4	68.7
	SLED	78.7	77.2	68.4
	Confident	80.8	78.3	69.0

Table 2: Baseline comparison for DoLa and SLED on Qwen3.5-35B-A3B, Qwen3.5-27B, and Gemma-4-31B (three-run means). Full benchmark results for Qwen3.5-35B-A3B are in Appendix B.

As an outcome-level check of the entropy proxy, Appendix C shows that the selected layer assigns greater normalized probability to the golden GPQA-D option than the final layer on the non-final-selection subset, further supporting the interpretation that the entropy valley is a practical anchor for recovering reasoning-relevant fidelity.

5.3 Instruct vs. Base: Validating the Alignment Tax

Through Table 3 we obtain two observations: (1) the larger average gain for Instruct (+2.6 vs. +1.1 points for Base) is consistent with post-training amplifying a generic-token bias that is already present after pre-training; and (2) LongBench-v2 changes little ($\Delta \approx 0$), consistent with dense context anchoring P_{logic} . These accuracy-level observations are corroborated at the token level. On the Instruct model, the backward scan identifies a non-trivial entropy valley for 12.8% of tokens, compared to 10.4% on the Base model. Token-level substitu-

Model	Decoding Strategy	GPQA-D	HLE	LCB-v6	LongBench-v2	Omni-MATH	Air-Bench	WritingBench	Avg.
Qwen3.5-35B-A3B-Base	Last Layer	70.3	8.0	55.0	59.0	61.3	67.3	64.0	55.0
	Confident (Ours)	72.2	9.0	57.4	58.9	61.1	70.2	63.8	56.1
	<i>Absolute Δ</i>	+1.9	+1.0	+2.4	-0.1	-0.2	+2.9	-0.2	+1.1
Qwen3.5-35B-A3B	Last Layer	76.3	9.2	70.1	63.2	72.3	51.7	65.2	58.3
	Confident (Ours)	82.8	11.2	74.4	63.9	73.0	55.4	65.3	60.9
	<i>Absolute Δ</i>	+6.5	+2.0	+4.3	+0.7	+0.7	+3.7	+0.1	+2.6

Table 3: **Base vs. Instruct comparison.** The larger average intervention gain after instruction tuning is consistent with post-training amplifying, rather than uniquely creating, the late generic-token bias.

tion rates are 2.60% (Instruct) vs. 2.36% (Base), with mean entropy gap at substitution positions of $\Delta H = 3.48 \times 10^{-2}$ vs. 3.34×10^{-2} . This consistent amplification across rebound rate, substitution rate, and entropy magnitude indicates that the accuracy advantage reflects a proportionally higher rate of corrective intervention on the Instruct model, as predicted by the Alignment Tax formulation.

6 Discussion and Analysis

In this section, we provide further analysis concerning Confident Decoding’s behavioral nuances, performance scaling, and computational overhead.

6.1 Entropy Valley Dynamics

Pooled across all seven benchmarks on Qwen3.5-35B-A3B, the backward scan selects a non-final layer for 11.5% of tokens. For the remaining 88.5%, it retains the last layer. Among non-final selections, 21.4% alter the argmax prediction, yielding a pooled substitution rate of 2.47%. Note that these pooled rates differ from the GPQA-D Instruct rates (12.8% non-final selection; 2.60% substitution) in Section 5.3. The 16.2% in Section 3.1 instead measures GPQA-D tokens with a final-layer entropy rise, a looser event that need not change the selected layer or argmax. The intervention rate exhibits temporal structure: non-trivial valley rate rises from 9.7% (first decile) to 13.4% (60–70% decile), then declines to 8.0% (final decile). This is consistent with *Contextual Saturation*: the alignment tax is most acute where the chain-of-thought diverges maximally from P_{align} , and is progressively suppressed as accumulating context re-anchors the distribution. Extended visualizations are in Appendix K.

6.2 Task Difficulty vs. Algorithm Efficacy

We stratified MATH and Omni-MATH into four difficulty tiers according to the baseline’s Pass@1 rate (see detailed results in Appendix J). It is observed that Confident Decoding’s benefit grows

substantially as task complexity increases:

- **Level 1 (Easy):** Near-perfect baseline accuracy, where Confident Decoding performs marginally below baseline (e.g., -0.4%).
- **Levels 3–4 (Hard):** Fragile, low-frequency logic chains diverge from P_{align} . For Level 4 of Omni-MATH where gpt-oss-20b’s baseline reasoning collapses to 1.1%, Confident Decoding delivers up to +22.4 points.

6.3 Architectural Robustness

Confident Decoding yields positive average gains across diverse architectures (Table 1), although compact Qwen3.5-9B exhibits task-dependent regressions (Appendix E). The entropy decomposition $\hat{H}(l) = H^*(l) + \epsilon^{(l)} + \eta^{(l)}$ provides a conditional analysis when the local-monotonicity and probe-noise assumptions hold. This suggests that late perturbations are not specific to either dense or sparse architectures.

Why existing contrastive methods underperform? Both DoLa and SLED yield smaller gains than Confident Decoding on hybrid MoE architectures (Appendix B). DoLa assumes early and late layers occupy comparable subspaces differing only in reasoning depth—a condition broken by layer-type alternation in hybrid architectures. SLED requires smooth, monotonic logit evolution, disrupted by non-monotonic entropy profiles from expert switching. Confident Decoding avoids these limitations by independently evaluating entropy at each layer rather than contrasting across layers. Extended degradation analysis across model depths and hybrid architectures is in Appendix E.

6.4 Computational Overhead

Confident Decoding reuses the full forward-pass KV cache, so its overhead is confined to additional unembedding projections. Evaluating all $K=10$ candidates for every token incurs a deliberately conservative worst-case increase of 74.6% FLOPs. In empirical lazy execution, however, 88.5% of

tokens require no extra projection and the mean is 0.116 extra projections per token. Table 4 reports the resulting +0.87% empirical FLOPs, zero additional KV-cache memory, and < 2% end-to-end latency. Because the added vocabulary projections are independent of context length while Transformer attention cost grows with sequence length, their relative overhead decreases for longer sequences. Full benchmarks are in Appendix M.

Component	FLOPs	Relative
Full Forward (Base)	5,212M	100.00%
Mean Extra Projections	45M	+0.87%
KV-Cache Memory	0 MB	+0.00%
Wall-clock (vLLM)	< 2%	

Table 4: **Computational overhead on Qwen3.5-35B-A3B.** Empirical FLOPs increase is bounded to < 1%.

7 Related Work

Layer-wise Dynamics and Mechanistic Interpretability. Standard autoregressive generation assumes token representations improve monotonically with depth (Csordás et al., 2025). However, intermediate states often crystallize confident, task-relevant predictions well before the final layer (Belrose et al., 2023; Skean et al., 2025; Liu et al., 2026; Wang and Zhou, 2024). This aligns with the Guess-then-Refine account (Gupta et al., 2025)—demonstrated via TunedLens and activation patching—and findings that mid-layer feed-forward networks progressively localize domain concepts and factual knowledge (Geva et al., 2022, 2023; Meng et al., 2022). Building on these insights, our work focuses on the conditional Phase III where final-layer predictions themselves degrade, motivating an efficacy-driven bypass rather than an efficiency-oriented early exit.

Contrastive and Adaptive Decoding. Contrastive methods reweight token probabilities to exploit latent layer-wise knowledge. Li et al. (2023b) improve open-ended generation by contrasting expert against amateur logits, while DoLa (Chuang et al., 2023) contrasts final-layer logits with early layers. Subsequent variants refine this via gradient corrections (Zhang et al., 2024), adaptive weighting (Zhou et al., 2025), entropy guidance (Das et al., 2025; Tan et al., 2026), or token-layer synchronization (Zhang et al., 2025a; Zhang, 2025; Zhu et al., 2025). Other methods shift activations (Li et al.,

2023a) or amplify context (Shi et al., 2024). Despite their efficacy, these approaches universally anchor on the final layer as the reference distribution, inherently inheriting late-stage perturbations. Confident Decoding discards this dependency, utilizing a low-entropy local valley to isolate generation from corrupted terminal distributions.

Test-Time Computation and Optimal Stopping.

Halting computation at intermediate layers originated as an efficiency optimization (Dehghani et al., 2019). Methods like CALM (Schuster et al., 2022) and LayerSkip (Elhoushi et al., 2024) target latency through token-wise exits, but often require intermediate-layer training or threshold calibration, and pure latency-driven exits face diminishing returns (Wei et al., 2026; Luo et al., 2025a,b; Fan et al., 2025; Yang et al., 2025; Chen et al., 2025). In contrast to self-consistency (Wang et al., 2023), which scales multi-path compute, we preserve the full forward pass and optimize the readout layer of a single path. Recent studies reveal that RLHF-induced degradation structurally concentrates in upper layers (Lin et al., 2024; Qi et al., 2025). By bypassing this alignment tax, Confident Decoding repurposes structural truncation as an efficacy-driven vertical test-time compute (TTC) scaling paradigm (Snell et al., 2025), demonstrating that optimizing *where* to stop inside the network is as vital as how long to think outside it.

8 Conclusion

In this paper, we challenge the pervasive assumption that final-layer representations in Large Language Models inherently yield the optimal semantic state. By formalizing the *Guess-Refine-Perturbation* layer-wise dynamics, we show that alignment-induced distributional shifts at late-stage (the "Alignment Tax") can perturb carefully constructed reasoning chains. To navigate this planning-pragmatics tradeoff, we introduce Confident Decoding, a training-free, risk-averse optimal stopping algorithm. By dynamically locating the monotonic entropy valley during a backward scan, our method filters alignment-induced perturbations under explicit local-monotonicity and probe-noise conditions. Extensive evaluations across dense and MoE architectures show that Confident Decoding delivers significant performance gains on frontier reasoning benchmarks (e.g., GPQA-Diamond, Omni-MATH, HLE) with negligible, bounded extra computational overhead.

Limitations

Confident Decoding is constrained by the structural alignment of the unembedding matrix W_U with intermediate residual states. Late-layer IO-CosSim supports this assumption on Qwen3.5-35B-A3B, but the mixed Qwen3.5-9B results show that layer-type probe mismatch can violate it on compact hybrid models. Furthermore, our approach mitigates symptoms of late distributional bias during decoding rather than resolving its training-time causes. It is also not a safety-centric method and should not replace safety fine-tuning or deployment-time safeguards. We discuss further directions in Appendix N.

Ethical Considerations

Given that all of our experiments are based on publicly available LLM datasets (broadly used benchmarks), there are no adverse social impacts associated with the data. Furthermore, the method we propose is a general-purpose LLM decoding algorithm that does not involve fine-tuning of the LLM. Our JailbreakBench evaluation finds low attack success rates on the two tested Qwen3.5 models, but this limited test does not mean that the method can be substituted for existing safety safeguards.

References

- Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, and 1 others. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.
- Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, and 1 others. 2025. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3639–3664.
- Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.
- Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwal, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. 2024. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. In *NeurIPS Datasets and Benchmarks Track*.
- Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. Do NOT think that much for $2+3=?$ on the overthinking of o1-like LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 9487–9499. PMLR.
- Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Róbert Csordás, Christopher D Manning, and Christopher Potts. 2025. Do language models use their depth efficiently? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Souvik Das, Lifeng Jin, Linfeng Song, Haitao Mi, Baolin Peng, and Dong Yu. 2025. Entropy guided extrapolative decoding to improve factuality in large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6589–6600.
- Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. 2019. Universal transformers. In *International Conference on Learning Representations*.
- Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, and 1 others. 2024. Layerskip: Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12622–12642.
- Siqi Fan, Xin Jiang, Xiang Li, Xuying Meng, Peng Han, Shuo Shang, Aixin Sun, Yequan Wang, and Zhongyuan Wang. 2025. Not all layers of LLMs are necessary during inference. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 5083–5091.
- Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, and 1 others. 2024. Omni-math: A universal olympiad level mathematic benchmark for large language models. In *The Thirteenth International Conference on Learning Representations*.
- Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceed-*

- ings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 12216–12235.
- Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1321–1330. PMLR.
- Akshat Gupta, Jay Yeung, Gopala Anumanchipalli, and Anna Ivanova. 2025. How do llms use their depth? *arXiv preprint arXiv:2510.18871*.
- Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Live-codebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.
- Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and 1 others. 2024. Rlaif vs. rlhf: scaling reinforcement learning from human feedback with ai feedback. In *Proceedings of the 41st International Conference on Machine Learning*, pages 26874–26901.
- Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023a. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, volume 36.
- Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023b. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312.
- Yong Lin, Hangyu Lin, Wei Xiong, Shizhe Diao, Jianmeng Liu, Jipeng Zhang, Rui Pan, Haoxiang Wang, Wenbin Hu, Hanning Zhang, Hanze Dong, Renjie Pi, Han Zhao, Nan Jiang, Heng Ji, Yuan Yao, and Tong Zhang. 2024. Mitigating the alignment tax of RLHF. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 580–606.
- Xukai Liu, Ye Liu, Jipeng Zhang, Yanghai Zhang, Kai Zhang, and Qi Liu. 2026. Layer-order inversion: Rethinking latent multi-hop reasoning in large language models. *arXiv preprint arXiv:2601.03542*.
- Xuan Luo, Weizhi Wang, and Xifeng Yan. 2025a. Adaptive layer-skipping in pre-trained llms. In *Second Conference on Language Modeling*.
- Xuan Luo, Weizhi Wang, and Xifeng Yan. 2025b. Diff-skip: Differential layer skipping in large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7221–7231.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 39643–39666.
- Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, and 1 others. 2026. A benchmark of expert-level academic questions to assess ai capabilities. *Nature*, 649(8099):1139–1146.
- Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. 2025. Safety alignment should be made more than just a few tokens deep. In *The Thirteenth International Conference on Learning Representations*.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First conference on language modeling*.
- Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. 2022. Confident adaptive language modeling. *Advances in Neural Information Processing Systems*, 35:17456–17472.
- Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarczyk, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.
- Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 783–791.
- Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Nikul Patel, Jalal Naghiyev, Yann Lecun, and Ravid Shwartz-Ziv. 2025. Layer by layer: Uncovering hidden representations in language models. In *International Conference on Machine Learning*, pages 55854–55875. PMLR.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling llm test-time compute optimally can be more effective than scaling model parameters. In *The Thirteenth International Conference on Learning Representations*.
- Wenhui Tan, Fiorenzo Parascandolo, Enver Sangineto, Jianzhong Ju, Zhenbo Luo, Qian Cao, Rita Cucchiara, Ruihua Song, and Jian Luan. 2026. Restoring exploration after post-training: Latent exploration decoding for large reasoning models. *arXiv preprint arXiv:2602.01698*.
- Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. In *2015 IEEE information theory workshop (itw)*, pages 1–5. Ieee.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*.
- Xuezhi Wang and Denny Zhou. 2024. Chain-of-thought reasoning without prompting. In *Advances in Neural Information Processing Systems*, volume 37.
- Rui Wei, Rui Du, Hanfei Yu, Devesh Tiwari, Jian Li, Zhaozhao Xu, and Hao Wang. 2026. The diminishing returns of early-exit decoding in modern llms. *arXiv preprint arXiv:2603.23701*.
- Yuning Wu, Jiahao Mei, Ming Yan, Chenliang Li, Shaopeng Lai, Yuran Ren, Zijia Wang, Ji Zhang, Mengyue Wu, Qin Jin, and 1 others. 2026. Writing-bench: A comprehensive benchmark for generative writing. *Advances in Neural Information Processing Systems*, 38.
- Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. 2025. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*.
- Yi Zeng, Yu Yang, Andy Zhou, Jeffrey Ziwei Tan, Yuheng Tu, Yifan Mai, Kevin Klyman, Minzhou Pan, Ruoxi Jia, Dawn Song, and 1 others. 2025. Air-bench 2024: A safety benchmark based on regulation and policies specified risk categories. In *The Thirteenth International Conference on Learning Representations*.
- Hongxiang Zhang, Hao Chen, Muhao Chen, and Tianyi Zhang. 2025a. Active layer-contrastive decoding reduces hallucination in large language model generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3028–3046.
- Jianyi Zhang, Da-Cheng Juan, Cyrus Rashtchian, Chun-Sung Ferng, Heinrich Jiang, and Yiran Chen. 2024. Sled: Self logits evolution decoding for improving factuality in large language models. *Advances in Neural Information Processing Systems*, 37:5188–5209.
- Xuanming Zhang. 2025. Generalization or memorization: Dynamic decoding for mode steering. In *Socially Responsible and Trustworthy Foundation Models at NeurIPS 2025*.
- Xuanming Zhang, Yuxuan Chen, Samuel Yeh, and Sharon Li. 2025b. Cognition-of-thought elicits social-aligned reasoning in large language models. In *Socially Responsible and Trustworthy Foundation Models at NeurIPS 2025*.
- Yuechi Zhou, Chuyue Zhou, Jianxin Zhang, Juntao Li, and Min Zhang. 2025. Alw: Adaptive layer-wise contrastive decoding enhancing reasoning ability in large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8506–8524.
- Jingze Zhu, Yongliang Wu, Wenbo Zhu, Jiawang Cao, Yanqiang Zheng, Jiawei Chen, Xu Yang, Bernt Schiele, Jonas Fischer, and Xinting Hu. 2025. Layercake: Token-aware contrastive decoding within large language model layers. *arXiv preprint arXiv:2507.04404*.

Ke Zixuan, Shao Yijia, Lin Haowei, Konishi Tatsuya, Kim Gyuhaek, Liu Bing, and 1 others. 2023. Continual pre-training of language models. In *Proceedings of The Eleventh International Conference on Learning Representations (ICLR-2023)*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xu Wang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, and 1 others. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

A Hyperparameters and Configurations

We organize our discussion around two groups of hyperparameters. The first group is intrinsic to CONFIDENT DECODING and controls the entropy-valley search itself: the lookback window K and the fallback (valley-selection) probability p . The second group consists of the standard sampling parameters (temperature T and nucleus probability top- p) together with benchmark-specific decoding settings, where, to ensure a fair comparison with the greedy baseline and to remain faithful to community-reported numbers, we follow the official evaluation repositories of each benchmark.

Algorithm-Specific Hyperparameters (K and p). The lookback window K specifies how many transformer layers the backward scan can traverse, while p governs whether decoding uses the selected valley (p) or the standard final layer ($1 - p$). We use $K=10$ and $p=1.0$ as fixed defaults in the main experiments. Across the seven benchmarks and evaluated model scales, selected valleys rarely lie more than 5 layers below the final layer (Figures 5–10); $K=10$ therefore provides substantial slack while bounding projection cost. The phase boundaries $0.15L$ and $0.95L$ in Section 2.1 are descriptive labels and are not algorithm inputs. Table 6 is a robustness check for the fixed defaults, not a benchmark-specific search used to select them.

Sampling Parameters (T and top- p). For all benchmarks except WritingBench ($T=0.7$ for creative writing), we use greedy-style decoding with temperature $T=0.0$ and top- $p=1.0$. This mirrors the default configuration used by the corresponding official evaluation repositories (GPQA-Diamond, HLE, LiveCodeBench, LongBench v2, Omni-MATH, Air-Bench 2024 and WritingBench). This choice serves two purposes. First, it makes our greedy baseline comparable to community-reported scores. Second, it isolates the contribution of CONFIDENT DECODING: any performance change we observe must come from the layer at

which the next-token distribution is read, not from sampling-induced diversity. The ablation in Table 6 further supports this design choice.

Benchmark-Specific Settings. Table 5 summarizes the prompt template and decoding configuration we use for each benchmark, all of which are inherited from the corresponding official codebase with only the model-serving backend swapped to a vLLM-hosted OpenAI-compatible endpoint: Unless otherwise noted, performance tables report means over three full runs. Where shown, $\pm$ is the unbiased sample SD across run-level scores; it captures operational variability from parallel vLLM execution, judge variability for judged tasks, and sampling variability for WritingBench. Compact Tables 3 and 7 omit SDs for space.

GPQA-Diamond (Rein et al., 2024)

System Prompt:

(none — zero-shot setting)

User Prompt Template:

What is the correct answer to this question:
 $\hookrightarrow$ {question}

Choices:

- (A) {choice1}
- (B) {choice2}
- (C) {choice3}
- (D) {choice4}

Format your response as follows: 'The
 $\hookrightarrow$ correct answer is (insert answer here)'

Decoding Settings: $T=0.0$, top- $p=1.0$, max tokens 32,768.

Grading: Exact-match extraction of the answer letter from the model output.

HLE — Humanity’s Last Exam (Phan et al., 2026)

System Prompt:

Your response should be in the following
 $\hookrightarrow$ format:

Explanation: {your explanation for your
 $\hookrightarrow$ answer choice}

Answer: {your chosen answer}

Confidence: {your confidence score between
 $\hookrightarrow$ 0% and 100% for your answer}

User Prompt:

The raw question text from the HLE dataset (supports multi-modal with optional image URLs).

Decoding Settings: $T=0.0$, top- $p=1.0$, max tokens 32,768.

Grading: LLM-as-a-judge with gpt-5.4 evaluating Explanation/Answer/Confidence against ground truth.

LiveCodeBench v6 — Code Generation (Jain et al., 2024)

System Prompt:

You are an expert Python programmer. You
→ will be given a question (problem
→ specification) and will
generate a correct Python program that
→ matches the specification and passes all
→ tests.

User Prompt Template:

```
### Question:  
{question_content}
```

```
### Format:  
{starter_code_message}
```

```
### Answer: (use the provided format with  
→ backticks)
```

Decoding Settings: $T=0.0$, $\text{top-}p=1.0$, max tokens 32,768, $n=1$ generation per problem.

Grading: Official functional-correctness evaluator (release v6) running the generated code against test cases.

LongBench v2 (Bai et al., 2025)

System Prompt:

(none — in-context only)

User Prompt Template:

Please read the following text and answer
→ the questions below.

```
<text>  
$DOC$  
</text>
```

What is the correct answer to this question:

```
→ $Q$  
Choices:  
(A) $C_A$  
(B) $C_B$  
(C) $C_C$  
(D) $C_D$
```

Please think step by step.

Decoding Settings: $T=0.0$, $\text{top-}p=1.0$, CoT phase max tokens 32,768, answer phase max tokens 16,384. Long inputs truncated by head–tail concatenation to fit the model’s context window (official truncation strategy).

Grading: Exact-match of the extracted answer letter.

Omni-MATH (Gao et al., 2024)

System Prompt:

You are a helpful and harmless assistant.
→ You should think step-by-step and put
→ your final answer
within \boxed{ }.

User Prompt:

The problem statement (field question or problem in the dataset JSONL).

Decoding Settings: $T=0.0$, $\text{top-}p=1.0$, max tokens 32,768, $n=1$.

Grading: Official rule-based math evaluator — `parse_ground_truth + run_execute + math_equal`.

Air-Bench 2024 (Zeng et al., 2025)

System Prompt:

You are a helpful assistant.

User Prompt:

Question text sampled from the official taxonomy (5 samples per leaf category, `sample_size=5`).

Decoding Settings: $T=0.0$, $\text{top-}p=1.0$, max tokens 512.

Grading: Official judge prompts evaluated by gpt-5.4 via LLM-as-a-judge; the judge outputs a short reasoning and a numerical score.

WritingBench (Wu et al., 2026)

System Prompt: (none — direct user query)

User Prompt: Raw writing query from the official benchmark (1,000 queries spanning 6 domains: Academic & Engineering, Finance & Business, Politics & Law, Literature & Arts, Education, Advertising & Marketing).

Decoding Settings: $T=0.7$, $\text{top-}p=0.8$, $\text{top-}k=20$, max tokens 16,000. These generation parameters follow the official WritingBench leaderboard configuration.

Scoring: Two-stage pipeline: (1) candidate model generates a response; (2) a judge model (default: gpt-5.4) scores the response on a 1–10 scale against five instance-specific criteria, each requiring a score and a textual justification. The system prompt for the judge is “*You are an expert evaluator with extensive experience in evaluating response of given query.*”, and the scoring prompt follows the official rubric with scoring bands 1–2 (critical deficiencies), 3–4 (noticeable shortcomings), 5–6 (adequate baseline), 7–8 (strong with minor refinements), and 9–10 (exceptional). The final WritingBench score is the mean over all criteria across all queries.

Sensitivity Analysis on p and T . We ablate the two most influential decoding knobs—the valley-selection probability p and the temperature T —on GPQA-Diamond using **Qwen3.5-35B-A3B**. Results are reported in Table 6.

Two trends emerge clearly. First, fixing $p=1.0$ and increasing the temperature from 0.0 to 1.0 degrades accuracy almost monotonically (from 82.8% to 80.8%). This indicates that, once the decoding layer is correctly chosen, additional stochasticity in the softmax draws back in the very alignment-tax noise that CONFIDENT DECODING suppresses, and yields no benefit. Second, fixing $T=0.0$ and decreasing p from 1.0 down to 0.0 produces a strict, monotone collapse from 82.8% to 76.3%. Note that the lower endpoint $p=0.0$ exactly recovers standard greedy decoding from the final layer, and our deterministic valley-selection setting ($p=1.0$) improves over it by +6.5 absolute points. Intermediate values of p smoothly interpolate between the two regimes, which directly mirrors the theoretical picture in Section 3.2: each unit of probability mass placed back on the perturbed

Benchmark	Eval. size	Temp.	Max Tokens	Judge	Prompt Source
GPQA-Diamond	198	0.0	32,768	–	official zero-shot
HLE	2,500	0.0	32,768	gpt-5.4	official system prompt
LiveCodeBench v6	1,055	0.0	32,768	–	official release_v6
LongBench v2	503	0.0	32,768 (CoT)	–	official 0shot_cot
Omni-MATH	2,821	0.0	32,768	rule-based	official math prompt
Air-Bench 2024	5/category	0.0	512	gpt-5.4	official taxonomy + judge prompts
WritingBench	1,000	0.7	16,000	gpt-5.4	official queries + judge prompts

Table 5: Benchmark-specific evaluation settings. “Eval. size” is the number of evaluated items, except Air-Bench, which follows the official sampling rule shown. “Judge” denotes the auxiliary model used for LLM-as-a-judge scoring.

Temperature T	p	GPQA-Diamond Acc. (%)
0.0	1.0	82.8
0.1	1.0	80.0
0.4	1.0	81.8
0.7	1.0	81.3
1.0	1.0	80.8
0.0	0.8	81.8
0.0	0.6	80.0
0.0	0.4	79.8
0.0	0.2	78.3
0.0	0.0	76.3

Table 6: Ablation of p and temperature on GPQA-Diamond accuracy with Qwen3.5-35B-A3B. The default setting ($T=0.0$, $p=1.0$) achieves the best accuracy. Top block: vary T at $p=1.0$; bottom block: vary p at $T=0.0$ (where $p=0.0$ recovers standard greedy decoding).

final layer reintroduces a proportional fraction of the KL penalty $\mathcal{D}_{\text{KL}}(P_{\text{logic}}||P_{\text{align}})$. Consequently, the deterministic-valley regime ($p=1.0$, $T=0.0$) is not merely a tuned best operating point—it is the configuration the theory predicts, and we adopt it as our default in all main experiments. Note that though these results support the fixed deterministic setting on these benchmarks, we do not interpret this table as a universal hyperparameter optimum.

B More Details of the Baselines

We compare Confident Decoding against two representative contrastive decoding baselines: Decoding by Contrasting Layers (DoLa) (Chuang et al., 2023) and Self Logits Evolution Decoding (SLED) (Zhang et al., 2024). Both methods were originally developed and evaluated on standard dense Transformers (e.g., LLaMA-family models), and their official implementations rely on layer-indexing conventions and residual-stream

access patterns specific to homogeneous dense architectures. Consequently, neither can be directly applied to modern hybrid or MoE backbones such as Qwen3.5-35B-A3B, where interleaved DeltaNet/full-attention layers and sparse expert routing fundamentally alter the layer-wise representation geometry.

To enable a fair comparison, we re-implemented both DoLa and SLED within the same inference framework used for Confident Decoding, adapting them to operate correctly on the hybrid MoE architecture. We first validated our re-implementations on LLaMA-family models, confirming that they reproduce the originally reported results. Table 7 reports the results on Qwen3.5-35B-A3B. Both contrastive baselines yield modest improvements over greedy decoding, but their gains fall substantially short of those achieved by Confident Decoding.

The limited effectiveness of contrastive baselines on hybrid MoE architectures is structurally predictable. For example, DoLa dynamically selects a premature layer that maximally diverges from the final layer (measured by Jensen–Shannon divergence) and decodes from the contrastive log-probability difference between the two, implicitly assuming that early and late layers occupy a representationally comparable subspace differing only in depth of reasoning. On hybrid backbones, the interleaving of DeltaNet and full-attention layers introduces discontinuous shifts in representation geometry (as formalized by the $\epsilon_{\text{type}}^{(l)}$ term in our degradation analysis), breaking this subspace-comparability assumption. The contrastive signal is therefore diluted by architectural noise, reducing its ability to isolate the intended factuality gradient. SLED applies gradient-style corrections based on the evolution of logits across layers, requiring the logit trajectory to be locally smooth and monotonically informative. The non-monotonic entropy

Method	GPQA-D	HLE	LCB-v6	LongBench-v2	Omni-Math	Air-Bench	WritingBench
Last Layer	76.3	9.2	70.1	63.2	72.3	51.7	65.2
DoLa	77.3	7.8	70.9	63.4	72.4	52.0	65.2
SLED	78.8	7.4	71.7	63.2	72.1	53.0	65.3
Confident (Ours)	82.8	11.2	74.4	63.9	73.0	55.4	65.3

Table 7: Comparison of contrastive decoding baselines on Qwen3.5-35B-A3B. DoLa and SLED are re-implemented within our inference framework to support the hybrid MoE architecture. Both contrastive baselines yield smaller gains than Confident Decoding on this backbone.

profile induced by layer-type alternation and expert switching corrupts the logit-evolution signal, limiting the effectiveness of the corrections.

Both baselines share a common structural limitation: they assume representational homogeneity across the layers being contrasted—a condition that modern hybrid and MoE architectures do not satisfy. Confident Decoding mitigates this limitation through a fundamentally different evaluation strategy: rather than contrasting logit distributions across layers (which requires representational comparability between the contrasted pair), it independently evaluates the predictive entropy at each candidate layer and selects the one with minimal uncertainty. Although it relies on the same unembedding projection W_U and is therefore not immune to probe mismatch (as analyzed in the degradation analysis below), the per-layer independent evaluation avoids amplifying structured noise through cross-layer subtraction, making it less sensitive to the heterogeneous geometry of modern architectures.

C Outcome-Level Check of Layer Confidence

For open-ended reasoning, a unique “gold next token” is generally undefined. We therefore follow outcome-oriented calibration practice (Guo et al., 2017; Kadavath et al., 2022) and perform a direct-answer check on GPQA-Diamond. Given the forced cue The correct answer is (, we normalize the logits over the four option-letter tokens and compare the golden-option mass $P_4(\text{golden})$ at the selected entropy-valley layer ℓ^* and final layer L using the same forward pass.

For Qwen3.5-9B, the full-set averages are 0.464 at ℓ^* and 0.441 at L (+0.023); on the subset where $\ell^* \neq L$, they are 0.412 and 0.384 (+0.028). For Qwen3.5-35B-A3B, the corresponding non-final-selection subset gives 0.358 and 0.350 (+0.008). This targeted check indicates that valley selection tracks increased probability mass on the correct out-

come under direct-answer prompting. It is not intended as a general calibration guarantee for open-ended chains of thought.

D Adversarial Safety Check

Air-Bench uses standard prompts and does not directly test adversarial bypass. We therefore additionally evaluate the 100 misuse behaviors from JailbreakBench (Chao et al., 2024), using its official Llama-3-70B binary judge. Attack success rate (ASR) remains low for both tested models:

Model	Last Layer	Confident	Δ
Qwen3.5-35B-A3B	2%	3%	+1 point
Qwen3.5-27B	1%	1%	0 points

Table 8: JailbreakBench ASR on 100 misuse behaviors.

The 27B result is unchanged, while the 35B-A3B result differs by one judged example. Given the small evaluation set and binary judge, this check basically rules out a large degradation in this setting. We emphasize that Confident Decoding is not a safety-centric method and should not replace safety fine-tuning or deployment-time safeguards.

E Degradation Analysis

Table 9 reveals that Confident Decoding does not uniformly benefit all backbones. MoE models (e.g., Qwen3.5-35B-A3B) and larger Transformer-based models (e.g., Qwen3.5-27B, gpt-oss-20b) generally benefit or maintain performance across benchmarks, while the compact hybrid Qwen3.5-9B presents a mixed picture: it suffers notable regressions on certain benchmarks (e.g., GPQA-D: 64.6 $\rightarrow$ 62.1, Omni-MATH: 49.1 $\rightarrow$ 47.1) yet simultaneously achieves clear improvements on others (LCB-v6: 41.1 $\rightarrow$ 47.7, Air-Bench: 53.0 $\rightarrow$ 56.0, HLE: 5.2 $\rightarrow$ 6.1). This section analyzes the degradation through two interacting factors—*architectural heterogeneity* and *model depth*—and

discusses what structural properties make certain backbones more amenable to Confident Decoding.

Factor I: Hybrid architecture compresses the refinement corridor. Qwen3.5 family employs a hybrid design. For example, Qwen3.5-9B interleaves 24 linear-attention (DeltaNet) layers with 8 full-attention layers ($L=32$). DeltaNet layers perform recurrent-style state updates that produce representations with a stronger “process state” character, while full-attention layers periodically reorganize global context. Because adjacent layers may belong to fundamentally different computational paradigms, the late *refinement corridor*—the band of layers where the model’s token-level prediction monotonically sharpens—is substantially compressed. Consequently, even a single-layer roll-back in the backward scan can cross the decision boundary between the final full-attention consolidation layer and a pre-final DeltaNet state that has not yet completed global-context integration.

The root mechanism is a *probe mismatch* induced by layer-type alternation. The backward scan evaluates entropy by projecting each intermediate hidden state through the shared final unembedding matrix W_U —a projection that is well-calibrated only when the probed layer’s representation lies in roughly the same geometric subspace as the final layer’s. In the hybrid architecture, however, successive layers alternate between linear-attention recurrence and full-attention recomposition, causing the representation geometry to shift at each type boundary. The effective probe error $\epsilon^{(l)}$ therefore acquires a structured, layer-type-dependent component. Decomposing the projection noise $\epsilon^{(l)}$ from Section 3.3 into a smooth baseline term and a layer-type term:

$$\hat{H}(l) = H^*(l) + \epsilon_{\text{probe}}^{(l)} + \epsilon_{\text{type}}^{(l)} + \eta^{(l)}, \quad (4)$$

where $\epsilon_{\text{type}}^{(l)}$ exhibits discontinuous jumps at DeltaNet $\leftrightarrow$ full-attention transitions. When this structured noise dominates the true entropy gradient $\Delta H^*(l)$, the observed entropy valley $\arg \min_l \hat{H}(l)$ can deviate from the true semantic optimum $\arg \min_l H^*(l)$, causing the backward scan to commit tokens to pre-convergent representations. Notably, this probe mismatch is *task-dependent*: on benchmarks where the entropy valley signal is strong and concentrated in the final few layers (e.g., code generation in LCB-v6), the structured noise is insufficient to derail the scan, and even the 9B backbone achieves a clear gain

(+6.6). The degradation concentrates on tasks such as GPQA-D and Omni-MATH, where the entropy gradient in the refinement corridor is shallower and more susceptible to being masked by $\epsilon_{\text{type}}^{(l)}$.

Factor II: Model depth widens the refinement corridor. Network depth modulates the degradation through a straightforward mechanism: deeper networks allocate more consecutive layers of the same type to the late refinement zone, widening the corridor over which $|\Delta H^*(l)| \gg |\Delta \epsilon^{(l)}|$ holds and providing a larger margin for the backward scan to locate a reliable valley. Qwen3.5-9B ($L=32$) has only 8 full-attention layers spread across the entire stack, leaving the final refinement zone with very few homogeneous layers. By contrast, Qwen3.5-35B-A3B ($L=40$) and Qwen3.5-27B have substantially deeper stacks, and the wider corridor trivially satisfies the local-monotonicity precondition. This is consistent with the data: Qwen3.5-27B exhibits no degradation on GPQA-D (78.2 $\rightarrow$ 79.4) and even gains +9.4 on LCB-v6. We note that depth and architecture are partially confounded in our model suite—the deeper models also differ in parameter count and expert routing—so we cannot fully disentangle the contribution of depth alone from other architectural factors. Nonetheless, the pattern is consistent: shallower networks with narrower corridors are more vulnerable to probe-error-induced misselection.

Why MoE architectures appear robust. Qwen3.5-A series—an MoE backbone that also employs hybrid attention—achieves significant Confident Decoding gains. Two properties may contribute beyond their depth. First, sparse expert routing concentrates task-relevant updates in specialized sub-networks, which may amplify the true refinement signal $|\Delta H^*(l)|$ in the late corridor and make it easier to dominate probe noise. Second, because each token activates only a small subset of experts, the effective representation trajectory across adjacent layers is smoother than in a dense hybrid backbone of equivalent depth, potentially reducing the type-switching probe error $\epsilon_{\text{type}}^{(l)}$. However, we emphasize that in our current experimental setup, depth and MoE routing are confounded: disentangling their individual contributions would require comparing models that differ only in one of these axes, which we leave to future work.

Scope delimitation. The degradation on Qwen3.5-9B does not invalidate the entropy-valley hypothesis. Rather, the failure arises when deter-

ministic valley selection ($p=1.0$) is applied to a backbone whose refinement corridor is too narrow to tolerate universal rollback on certain task types. The core precondition for Confident Decoding—as formalized in the dominance condition of Theorem 1 ($\Delta\eta^{(l)} > 2\epsilon_{\max}$)—is that the late refinement corridor must be sufficiently wide for the true entropy gradient to dominate the total probe error: $|\Delta H^*(l)| \gg |\Delta\epsilon^{(l)}|$, where $\epsilon^{(l)} = \epsilon_{\text{probe}}^{(l)} + \epsilon_{\text{type}}^{(l)}$. This condition is more readily satisfied on deeper networks, but can be violated on compact hybrid architectures where layer-type alternation introduces structured probe noise and limited depth compresses the safe corridor.

We also note that even models for which Confident Decoding is broadly beneficial can exhibit task-specific regressions. For instance, Qwen3.5-27B—a deeper model that gains strongly on reasoning benchmarks—shows a -1.0 drop on AirBench ($65.7 \rightarrow 64.7$). This suggests that the alignment-tax mechanism and the probe-mismatch mechanism can interact differently depending on the task’s reliance on final-layer formatting versus intermediate-layer reasoning. A complete characterization of these per-task dynamics is an important direction for future work.

These findings delineate two axes—*architectural homogeneity* and *model depth*—along which the applicability of Confident Decoding varies.

F Layer-wise Dynamics: Formal Definitions and Extensions

In this appendix we provide the full sublayer-level definitions underlying the layer contribution analysis in Section 2.

In the standard pre-norm formulation, the residual stream at layer l is updated via attention and feed-forward sublayers:

$$\tilde{\mathbf{h}}_t^{(l)} = \mathbf{h}_t^{(l-1)} + f_{\text{Attn}}^{(l)}(\mathbf{h}_t^{(l-1)}), \quad (5)$$

$$\mathbf{h}_t^{(l)} = \tilde{\mathbf{h}}_t^{(l)} + f_{\text{FFN}}^{(l)}(\tilde{\mathbf{h}}_t^{(l)}), \quad (6)$$

where $f_{\text{Attn}}^{(l)}$ denotes the attention sublayer (full or linear) and $f_{\text{FFN}}^{(l)}$ the feed-forward sublayer (MLP or MoE). The total layer contribution vector is:

$$\begin{aligned} \mathbf{m}_t^{(l)} &\triangleq \mathbf{h}_t^{(l)} - \mathbf{h}_t^{(l-1)} \\ &= f_{\text{Attn}}^{(l)}(\mathbf{h}_t^{(l-1)}) + f_{\text{FFN}}^{(l)}(\tilde{\mathbf{h}}_t^{(l)}) \end{aligned}$$

This definition naturally extends to parallel-attention variants where both sublayers receive

$\mathbf{h}_t^{(l-1)}$ directly. The two metrics used in Section 2 are:

$$\begin{aligned} \text{Norm Ratio}^{(l)} &= \frac{\|\mathbf{m}_t^{(l)}\|_2}{\|\mathbf{h}_t^{(l-1)}\|_2}, \\ \text{IO-CosSim}^{(l)} &= \frac{\mathbf{h}_t^{(l)} \cdot \mathbf{h}_t^{(l-1)}}{\|\mathbf{h}_t^{(l)}\|_2 \|\mathbf{h}_t^{(l-1)}\|_2} \end{aligned}$$

Under the residual stream decomposition $\mathbf{h}_t^{(l)} = \mathbf{h}_t^{(l-1)} + \mathbf{m}_t^{(l)}$, $\text{Norm Ratio}^{(l)}$ characterizes the *write intensity*. When $\text{Norm Ratio}^{(l)} \gg 1$, the contribution dominates the existing state ($\mathbf{h}_t^{(l)} \approx \mathbf{m}_t^{(l)}$), effectively overwriting the accumulated representation; when $\text{Norm Ratio}^{(l)} \ll 1$, the layer applies incremental corrections without displacing prior content; and when $\text{Norm Ratio}^{(l)} \lesssim 1$, the new write is comparable in magnitude to the carried residual. In this intermediate regime, $\text{Norm Ratio}^{(l)}$ alone cannot distinguish constructive reinforcement from disruptive rewriting, because the net effect depends on the update direction. $\text{IO-CosSim}^{(l)}$ therefore characterizes *directional fidelity*: in the high-dimensional representation space where direction encodes semantic content, a value near 1 indicates the layer preserves and refines the existing semantic trajectory, while lower values indicate rotation into a semantically distinct subspace (Mikolov et al., 2013).

Figure 2 shows the empirical profiles for Qwen3.5-35B-A3B on GSM8K (Cobbe et al., 2021), where the three-phase Guess–Refine–Perturbation structure is clearly visible.

G Full Theoretical Analysis

This appendix provides complete derivations and extended discussion for the theoretical framework sketched in Section 3.

G.1 Information Bottleneck View of Layer Dynamics

The Information Bottleneck (IB) principle (Tishby and Zaslavsky, 2015) motivates our view of layer-wise compression and refinement, but does not by itself establish monotonic inference-time dynamics. Let X denote the input context and Y_{logic} an idealized target token derived from domain-specific reasoning. We use the following tradeoff as a conceptual model:

$$\mathcal{L}_{\text{IB}} = I(X; \mathbf{h}_t^{(l)}) - \beta I(\mathbf{h}_t^{(l)}; Y_{\text{logic}})$$

Model	Method	GPQA-D	HLE	LCB-v6	LongBench-v2	Omni-Math	Air-Bench	WritingBench
Qwen3.5-0.8B	Last Layer	0.0	0.0	0.0	0.0	0.0	52.0	10.0
	Confident (Ours)	0.0	0.0	0.0	0.0	0.0	54.0	10.1
Qwen3.5-9B	Last Layer	64.6	5.2	41.1	62.0	49.1	53.0	60.7
	Confident (Ours)	62.1	6.1	47.7	60.5	47.1	56.0	59.5
Qwen3.5-9B-Base	Last Layer	67.7	5.4	49.1	54.4	54.7	76.0	61.1
	Confident (Ours)	65.7	6.3	50.5	56.1	53.0	76.0	63.0
Qwen3.5-27B	Last Layer	78.2	16.0	63.9	62.8	76.0	65.7	66.7
	Confident (Ours)	79.4	16.8	73.3	64.9	76.2	64.7	66.4
Qwen3.5-35B-A3B	Last Layer	76.3	9.2	70.1	63.2	72.3	51.7	65.2
	Confident (Ours)	82.8	11.2	74.4	63.9	73.0	55.4	65.3
Qwen3.5-122B-A10B	Last Layer	83.3	14.7	76.8	66.4	78.3	66.0	72.6
	Confident (Ours)	85.4	16.5	79.2	66.7	78.5	67.0	72.7
Gemma-4-31B	Last Layer	76.8	9.2	75.1	58.6	68.3	74.0	64.2
	Confident (Ours)	80.8	11.0	78.3	59.1	69.0	76.0	64.4
gpt-oss-20b	Last Layer	58.1	6.2	77.1	43.6	61.3	86.7	54.3
	Confident (Ours)	60.8	6.3	77.6	44.7	61.0	91.7	54.6
gpt-oss-120B	Last Layer	76.3	10.8	83.1	54.7	70.2	88.0	67.5
	Confident (Ours)	80.8	12.6	84.5	55.6	71.0	90.0	67.8

Table 9: Full results for final-layer and Confident Decoding across model backbones. Values are means over three full runs.

During **Phase I (Guess)**, the network aggressively compresses the superficial input X (minimizing $I(X; \mathbf{h})$). This severe dimensionality reduction induces geometric volatility, rendering the predictive entropy $H(p_t^{(l)})$ highly unstable and susceptible to degenerate unigram biases. Critically, the residual influence of this compression noise carries over into the early portion of Phase II, leaving the predictive entropy on a high-entropy plateau that is empirically indistinguishable from Phase I, as shown in Figure 4.

The network then transitions into **Phase II (Refine)**, where successive attention mechanisms rigorously integrate contextual dependencies, dominating the flow to maximize $I(\mathbf{h}_t^{(l)}; Y_{\text{logic}})$. Within Phase II, we identify an internal inflection layer V_{onset} at which the carry-over of Phase I dissipates and the network enters its monotone refinement regime. Within $l \geq V_{\text{onset}}$, the predictive entropy $H(p_t^{(l)})$ serves as an empirical upper bound for the true conditional entropy $H(Y_{\text{logic}}|\mathbf{h}_t^{(l)})$. As mutual information monotonically increases across this late-refine window, $H(p_t^{(l)})$ follows a strict monotone downward trajectory.

The minimum at V^* represents the exact representational zenith before potential perturbation sets in. As shown in Figure 4(a), on Qwen3.5-35B-A3B over GPQA-Diamond generating 4,096 tokens per prompt, Phase II accumulates $\Delta H \approx -10.8$ nats across 202,935 tokens ($V_{\text{onset}} \approx 28$, $V^* = 39$). Crucially, Phase III is not a fixed architectural re-

gion but a *per-token* phenomenon determined by whether the final layer perturbs a token’s prediction: 16.2% of tokens exhibit an entropy rise of $+0.37$ nats at $l = 40$ —these tokens undergo Phase III. The remaining 83.8% of tokens show a further entropy drop of -2.52 nats—these tokens simply complete their normal Phase I→II refinement through the final layer.

The two-zone structure of Phase II merits additional discussion. In the early portion ($l \lesssim 0.6L$, context assembly), the interleaved full-attention layers progressively build global token context; the decrease in $H^*(l)$ is slow and is masked by the projection noise $\epsilon^{(l)}$ from applying W_U to not-yet-converged intermediate representations. In the late portion ($0.6L \lesssim l \lesssim 0.95L$, convergent refinement), once global context is sufficiently assembled, remaining layers perform precision refinement. The refinement signal now dominates the projection noise, so $\Delta H^*(l) < 0$ holds strictly in this sub-zone. Per-prompt Spearman analysis over the late-refine window yields $\rho \approx -0.87$ (100% of prompts, $\rho < 0$), validating the strict monotone decrease required by Theorem 1.

G.2 Alignment Tax: Guardrail vs. Tax Duality

Modern LLMs undergo rigorous post-training to align with human preferences. Recent findings in Representation Engineering (Zou et al., 2023) reveal that such alignment behaviors are governed by low-rank steering vectors, predominantly activated

in the final layers to modulate outputs without overwriting pre-trained knowledge. Thus, in **Phase III (Perturbation)**, the latent representations optimize a regularized risk toward a generic, safe distribution P_{align} :

$$\mathcal{R}^{(l)} = \mathbb{E}[-\log p_t^{(l)}(Y_{\text{logic}}|X)] + \lambda \mathcal{D}_{\text{KL}}(P_{\text{logic}}^{(l)} \| P_{\text{align}})$$

where λ governs the steering intensity. Let V^* be the oracle transition boundary between Phase II and Phase III. For $l \rightarrow L$, the network induces an additive perturbation $\delta_{\text{align}}^{(l)}$ to minimize the KL divergence: $\mathbf{h}_t^{(l)} \approx \mathbf{h}_t^{(V^*)} + \sum_{k=V^*+1}^l \delta_{\text{align}}^{(k)}$.

The destructive nature of this perturbation is strictly conditional:

- **Alignment as a Guardrail (Safety Tasks):**
For standard conversational or safety queries, $P_{\text{logic}} \approx P_{\text{align}}$. The KL divergence is negligible. The late-stage layers refine formatting without semantic disruption.
- **Alignment as a Tax (Complex Reasoning):**
For domains like rigorous mathematics and science, the specialized logic distribution P_{logic} sharply conflicts with the generic P_{align} . The additive perturbation forcibly steers the latent state away from the reasoning subspace, mathematically manifesting as an unnatural entropy oscillation that derails the established logic chain.

A critical subtlety: we do *not* assume every alignment-induced update is harmful, nor that each individual $\delta_{\text{align}}^{(k)}$ must increase entropy token by token. The claim is narrower and empirically grounded: on tasks where the task-optimal distribution is sharper and more domain-specific than P_{align} , the *net effect* of the late alignment correction is to push the final logits away from the task-optimal mode, increasing task-conditional uncertainty. For $l > V^*$ we require only that the cumulative perturbation term dominates locally in aggregate.

This formulation also explains why Confident Decoding removes mainly the *negative* effect of alignment while preserving its *positive* effect. The positive consequences of post-training—instruction following, discourse coherence, style regularization, factual caution, response formatting—are distributed through the whole fine-tuned network and are already encoded in the mature Phase II representation $\mathbf{h}_t^{(V^*)}$. By contrast, the harmful effect for hard reasoning tasks is concentrated at the very end as a late distributional push

toward safer or more generic tokens. Decoding at V^* therefore retains alignment as a *representational competence* but removes alignment as an *overriding last-mile bias*.

This “surgical separation” is also consistent with the ablations in Figure 3: if our method simply removed alignment altogether, static exits such as $V^* - 1$, $V^* - 2$, or $0.75L$ should work similarly. They do not—those strategies often underperform because they cut into the genuinely beneficial refinement computation of late Phase II. The success of the dynamic valley layer supports the more specific interpretation: the method is not “less model” but “less harmful terminal perturbation.”

G.3 Full Optimal Stopping Theorem and Proof

Empirical evidence from prior work consistently shows that, beyond roughly the network midpoint, late-layer hidden states become approximately linearly decodable through the frozen unembedding W_U (Belrose et al., 2023; Elhoushi et al., 2024; Park et al., 2024); since our empirical V_{onset} falls well within this regime ($V_{\text{onset}} \gtrsim 0.5L$), the projection noise $\epsilon^{(l)}$ for $l \geq V_{\text{onset}}$ is correspondingly small and bounded by a tight ϵ_{max} .

We formulate dynamic layer selection as an Optimal Stopping Problem. For analysis, the observable entropy $\hat{H}(l)$ decomposes into:

$$\hat{H}(l) = H^*(l) + \epsilon^{(l)} + \eta^{(l)}$$

where:

1. $H^*(l)$ is the monotonic true entropy. It may be irregular in shallow layers, but in the late-Phase II corridor it decreases strictly and reaches its minimum at V^* . $\Delta H^*(l) \leq 0$ for $l \in [V_{\text{onset}}, V^*]$.
2. $\epsilon^{(l)}$ is the zero-mean projection noise caused by the unembedding matrix W_U acting on intermediate representations. It captures basis mismatch and local micro-oscillation, and is bounded as $|\epsilon^{(l)}| \leq \epsilon_{\text{max}}$.
3. $\eta^{(l)}$ is the per-token alignment perturbation. $\eta^{(l)} \approx 0$ for $l \leq V^*$; for tokens where alignment correction conflicts with the task-optimal prediction, $\Delta \eta^{(l)} > 2\epsilon_{\text{max}}$ in the final layers).

The three components are not separately identifiable from a single entropy trajectory and are not estimated by the algorithm. They define sufficient

conditions under which the backward search is locally reliable. Table 10 summarizes the operational notation.

Symbol	Definition
V_{onset}	onset of the assumed monotone corridor
V^*	oracle endpoint before terminal perturbation
$\hat{V}$	layer returned by the backward scan
ϵ_{max}	bound on intermediate-probe noise
M	number of candidate hidden states
K	maximum backward scan distance
p	probability of using the selected valley

Table 10: Notation used in the stopping analysis and algorithm.

Theorem 1 (Conservative Local Optimality of Backward Search). *Let $\hat{V}$ be the first local minimum encountered when scanning backward from L , i.e.,*

$$\hat{V} = \max\{l < L \mid \hat{H}(l-1) \geq \hat{H}(l)\}.$$

Assume that there exists an oracle boundary V^* such that: (a) for $l \in [V_0, V^*]$ with some $V_0 < V^*$, $H^*(l)$ is strictly decreasing; (b) for $l > V^*$, the cumulative perturbation increment dominates the local probe noise, i.e., $\Delta\eta^{(l)} > 2\epsilon_{\text{max}}$. Then the backward search never stops in Phase III, returns some $\hat{V} \in [V_0, V^*]$, and the excess risk relative to decoding at V^* is bounded by the residual decrease of $H^*(l)$ over $[\hat{V}, V^*]$.

Proof sketch.

1. *Excluding Phase III.* For any $l > V^*$,

$$\Delta\hat{H}(l) = \Delta H^*(l) + \Delta\epsilon^{(l)} + \Delta\eta^{(l)}.$$

Because $H^*(l)$ has already bottomed out and the cumulative perturbation grows beyond the boundary, the assumed dominance $\Delta\eta^{(l)} > 2\epsilon_{\text{max}}$ implies $\Delta\hat{H}(l) > 0$. Hence the entropy strictly decreases when scanning backward through Phase III, so the stopping rule cannot trigger there.

2. *Entering the safe corridor.* Once the scan crosses into $l \leq V^*$, the perturbation term vanishes and the process is governed by $H^*(l) + \epsilon^{(l)}$. Since $H^*(l)$ is strictly decreasing on $[V_0, V^*]$, the first reversal can only occur inside this corridor, and therefore the selected layer satisfies $\hat{V} \leq V^*$.

3. *Why local optimality is enough.* If the algorithm stops exactly at V^* , the choice is oracle-optimal. If it stops slightly earlier because the final few improvements are smaller than the probe fluctuation, then the loss is only the remaining refinement gain between $\hat{V}$ and V^* . Since this resid-

ual segment lies within the already-converged late-Phase II corridor, that gain is small by construction. Thus the method trades a tiny amount of unfinished refinement for complete removal of the much riskier post- V^* perturbation.

Conclusion. Conservative backward search is a *risk-dominant local stopping rule*: it sacrifices at most a small tail of refinement while avoiding the larger failure mode introduced by terminal alignment drift. ■

Remark 1 (When the corridor is weak or absent). For easy tokens or tasks naturally aligned with P_{align} , the perturbation term may be negligible and the late monotone corridor may collapse into the final layer. In this case the algorithm stops at L or $L-1$, effectively reducing to standard decoding. Thus the method does not require the beneficial structure to hold universally; it only exploits it when it is present.

H Systems Implementation Details

A correct implementation of Confident Decoding inside a production inference engine such as vLLM (Kwon et al., 2023) is nontrivial, because the engine combines tensor parallelism, continuous batching, torch.compile-style graph compilation, and CUDA graph replay. Naive in-graph mutation of Python attributes or dynamically reallocated buffers causes stale state during graph replay and silent correctness regressions. Our implementation rests on the following principles.

Unmodified forward pass. Confident Decoding never truncates the transformer. The model executes all L blocks at every step, so the KV cache, attention kernels, prefix caching, and continuous-batching scheduler operate identically to standard decoding. Only the layer whose logits feed the sampler is changed. The method is therefore composable with the engine’s existing speculative-decoding, multimodal, and tensor-parallel infrastructure.

Graph-safe candidate extraction. The compiled inner model collects the residual-stream tensors $\{\mathbf{x}_t^{(\ell)}\}_{\ell \in C}$ into a Python list returned alongside the final hidden states. Crucially, no normalization, unembedding, entropy computation, or attribute mutation is performed inside the compiled region. The eager wrapper applies the model’s exact final RMSNorm, including its learned scale γ , to every candidate and uses the same unmodified W_U as final-layer decoding. It then stacks candidates

into a tensor of shape $[M, S, d]$, where S is the captured token count of the current CUDA graph replay. Restricting all stateful logic to the eager scope prevents buffers updated during one replay from leaking into another.

Shape-aware buffering under continuous batching. Logits computing consumes a sliced hidden-state tensor of shape $[B, d]$ that contains positions that truly require sampling, where B is the number of positions remaining after slicing. With $B \leq S$, mapping the sliced positions back to the full-forward candidate states requires the engine’s pre-slice token count S and the slicing indices. The wrapper maintains a buffer indexed by S , retrieves the entry corresponding to the active replay shape, and slices it with the same indices used for the final-layer hidden states. A consume-once protocol clears these per-step indices after logits are computed, preventing stale values from leaking into subsequent calls.

Vectorized logits and entropy. The implementation is fully vectorized over both the candidate window and active tokens. Entropy and the trough scan are expressed as fused tensor operations using a per-token frozen mask, so the back-to-front search collapses to a sequence of K element-wise updates rather than a Python loop over tokens.

Cost analysis. Compared to standard decoding, the extra per-step overhead is: one batched unembedding of M candidate hidden states ($\mathcal{O}(MBd|\mathcal{V}|)$), an entropy evaluation ($\mathcal{O}(MB|\mathcal{V}|)$), and a K -step trough scan ($\mathcal{O}(KB)$), where B is typically far smaller than the pre-filled context length. The candidate window M thus directly controls the worst-case overhead. The overall cost is $\mathcal{O}(MBd|\mathcal{V}|)$, generally not greater than the regular decoding cost per step of $\mathcal{O}(12LSd^2 + 2LSdT + Bd|\mathcal{V}|)$ (suppose feed-forward dim $d_{ff} = 3d$). As the sequence length increases, the proportion of additional computational burden relative to regular decoding overhead decreases. Memory overhead is negligible since Confident Decoding only needs to cache the hidden states of a single forward step.

I Architectural Robustness Analysis

Table 9 spans three model families (Qwen, gpt-oss, Gemma), covering dense, hybrid-attention, and Mixture-of-Experts architectures across nine backbones. Confident Decoding yields positive average gains on every backbone with $L \geq 24$, confirming

that the entropy-valley signal is not an artifact of a single architecture or training recipe.

The decomposition $\hat{H}(l) = H^*(l) + \epsilon^{(l)} + \eta^{(l)}$ is not tied to dense or sparse feed-forward layers, but its stopping guarantee remains conditional on local monotonicity and bounded probe noise. The empirical results suggest that these conditions can hold across both dense and MoE backbones, but they need not hold for every architecture or token.

The tested MoE backbones show positive average changes comparable to the larger dense models. This pattern is consistent with late perturbations arising in both dense and sparse architectures, but does not isolate post-training from other architectural or training factors.

Factor I: Hybrid architecture compresses the refinement corridor. Qwen3.5 family employs a hybrid design (e.g., Qwen3.5-9B interleaves 24 linear-attention (DeltaNet) layers with 8 full-attention layers). Because adjacent layers may belong to fundamentally different computational paradigms, the late refinement corridor is substantially compressed. A single-layer rollback can cross the decision boundary between the final full-attention consolidation layer and a pre-final DeltaNet state that has not yet completed global-context integration.

The root mechanism is a *probe mismatch* induced by layer-type alternation. The backward scan evaluates entropy by projecting each intermediate hidden state through the shared final unembedding matrix W_U —a projection that is well-calibrated only when the probed layer’s representation lies in roughly the same geometric subspace as the final layer. In hybrid architectures, successive layers alternate between linear-attention recurrence and full-attention recomposition, causing the representation geometry to shift at each type boundary. The effective probe error $\epsilon^{(l)}$ therefore acquires a structured, layer-type-dependent component $\epsilon_{\text{type}}^{(l)}$ that exhibits discontinuous jumps at DeltaNet $\leftrightarrow$ full-attention transitions. When this structured noise dominates the true entropy gradient $\Delta H^*(l)$, the observed entropy valley can deviate from the true semantic optimum.

Factor II: Model depth widens the refinement corridor. Deeper networks allocate more consecutive layers of the same type to the late refinement zone, widening the corridor over which $|\Delta H^*(l)| \gg |\Delta \epsilon^{(l)}|$ holds. Qwen3.5-9B ($L=32$) has only 8 full-attention layers spread across the

entire stack, leaving the final refinement zone with very few homogeneous layers. By contrast, Qwen3.5-35B-A3B ($L=40$) and Qwen3.5-27B have substantially deeper stacks, and the wider corridor trivially satisfies the local-monotonicity precondition. Depth and architecture are partially confounded in our model suite, but the pattern is consistent: shallower networks with narrower corridors are more vulnerable to probe-error-induced misselection.

Why MoE architectures appear robust. MoE backbones that also employ hybrid attention (Qwen3.5-A series) still achieve significant Confident Decoding gains. First, sparse expert routing concentrates task-relevant updates in specialized sub-networks, which may amplify the true refinement signal $|\Delta H^*(l)|$ while reducing the structured probe noise $\epsilon_{\text{type}}^{(l)}$. Second, because each token activates only a small subset of experts, the effective representation trajectory across adjacent layers is smoother than in a dense hybrid backbone, potentially reducing the type-switching probe error. However, depth and MoE routing are confounded in our setup; disentangling their individual contributions requires comparing models differing in only one axis.

Scope delimitation. The degradation on Qwen3.5-9B does not invalidate the entropy-valley hypothesis. The failure arises when deterministic valley selection ($p=1.0$) is applied to a backbone whose refinement corridor is too narrow. The core precondition for Confident Decoding—as formalized in Theorem 1 ($\Delta\eta^{(l)} > 2\epsilon_{\text{max}}$)—requires that the late refinement corridor is sufficiently wide: $|\Delta H^*(l)| \gg |\Delta\epsilon^{(l)}|$, where $\epsilon^{(l)} = \epsilon_{\text{probe}}^{(l)} + \epsilon_{\text{type}}^{(l)}$. This condition is more readily satisfied on deeper networks.

Even models broadly benefiting from Confident Decoding can exhibit task-specific regressions: Qwen3.5-27B shows -1.0 drop on AirBench ($65.7 \rightarrow 64.7$). The alignment-tax mechanism and the probe-mismatch mechanism can interact differently depending on the task’s reliance on final-layer formatting versus intermediate-layer reasoning.

J Difficulty-Stratified Results

Tables 12 and 13 report full difficulty-stratified results for gpt-oss-20b and Qwen3.5-35B-A3B on MATH and Omni-MATH.

The scaling law is consistent across both model

Computational Component	FLOPs	Relative Overhead
Theoretical Foundations		
Full Forward Pass (Base Model)	5,212M	100.00%
Single Unembedding Projection (W_U)	389M	+7.46%
Worst-case Scan ($K = 10$ Projections)	3,890M	+74.64%
Empirical Execution (Ours)		
Mean Extra Projections (0.116 /token)	45M	+0.87%
Incremental KV-Cache Memory Cost	0 MB	+0.00%
<i>End-to-End Wall-clock (vLLM)</i>		< 2% per token

Table 11: **Detailed computational overhead on Qwen3.5-35B-A3B** ($L=40$, $d=2560$, $|V|=151,936$, $K=10$). The empirical FLOPs increase is tightly bounded to $< 1\%$, maintaining a negligible memory footprint and deployment latency.

families: Confident Decoding’s benefit grows monotonically with task difficulty, with the most dramatic gains on Level 4 problems where baseline performance has essentially collapsed.

K Extended Entropy Valley Dynamics

This section provides extended visualizations and analysis of entropy valley dynamics.

Figure 1 (in the main text) shows representative token substitutions produced by Confident Decoding on Qwen3.5-35B-A3B. When the two decoding strategies disagree ($\sim 2\%$ of generated tokens), Standard Decoding selects generic, high-frequency function words and punctuation (e.g., “the”, “is”, “so”, “.”), while Confident Decoding recovers domain-specific, semantically precise terminology (e.g., “mass”, “radius”, “approximately”, “Cartesian”).

The backward scan exploits per-token heterogeneity: for perturbed tokens (16.2%) it selects V^* to bypass Phase III, and for unperturbed tokens (83.8%) it naturally selects L to utilize full refinement. This adaptive behavior is confirmed by the temporal structure of the intervention rate: the valley rate peaks at the 60–70% decile of generation, where the chain-of-thought diverges maximally from P_{align} , and converges to $\sim 8\%$ at the end of generation as context anchors the distribution.

Table 14 provides a fine-grained token-level view of this phenomenon, listing the 20 most frequently substituted tokens for both the Base and Instruct variants of Qwen3.5-35B-A3B on GPQA-Diamond. In both variants, the tokens replaced by Standard Decoding (last layer) are dominated by function words and punctuation (Base: 39% function, 22% punctuation; Instruct: 27% function, 43% punctuation), while Confident Decoding selects content-rich, domain-specific tokens instead

Decoding Strategy	MATH (Difficulty Level)				Omni-MATH (Difficulty Level)			
	Level 1	Level 2	Level 3	Level 4	Level 1	Level 2	Level 3	Level 4
Baseline (Last Layer)	98.2	57.1	30.2	2.1	97.0	55.8	30.8	1.1
Confident Decoding (Ours)	97.8	69.8	56.8	24.6	92.7	59.1	35.3	23.5
<i>Absolute Δ</i>	-0.4	+12.7	+26.6	+22.5	-4.3	+3.3	+4.5	+22.4

Table 12: Performance comparison (Accuracy %) stratified by task difficulty on MATH and Omni-MATH. **Experiments conducted using gpt-oss-20b.** Difficulty levels are dynamically determined by the baseline model’s Pass@1 rate.

Decoding Strategy	MATH (Difficulty Level)				Omni-MATH (Difficulty Level)			
	Level 1	Level 2	Level 3	Level 4	Level 1	Level 2	Level 3	Level 4
Baseline (Last Layer)	97.1	57.9	33.0	2.7	96.0	56.1	31.7	0.3
Confident Decoding (Ours)	97.0	79.8	50.0	11.9	94.6	58.2	35.3	7.5
<i>Absolute Δ</i>	-0.1	+21.9	+17.0	+9.2	-1.4	+2.1	+3.6	+7.2

Table 13: Performance comparison (Accuracy %) stratified by task difficulty on MATH and Omni-MATH. **Experiments conducted using Qwen3.5-35B-A3B.**

(Base: 77% content; Instruct: 60% content). This confirms that the alignment tax preferentially suppresses semantically informative tokens in favor of safe, high-frequency alternatives.

L Complete Model Results

Table 9 reports results across all evaluated model backbones, including smaller models not included in the main text.

M Computational Overhead: Detailed Benchmarks

Table 11 provides detailed FLOPs and wall-clock benchmarks for Confident Decoding on Qwen3.5-35B-A3B.

N Future Directions

Our findings expose a critical architectural conflict between pre-trained reasoning and post-training pragmatics that warrants deeper mechanistic exploration. Future research should investigate training paradigms that inherently decouple these objectives, such as applying alignment penalties exclusively to designated routing heads rather than the core residual stream. Additionally, exploring the persistence of the *Guess-Refine-Perturbation* dynamics in multimodal foundation models, and leveraging layer-wise entropy metrics to design more geometrically precise reward functions for reinforcement learning, represent promising frontiers for developing natively robust reasoning models.

O LLM Usage

In the course of this research, we primarily used large language models (LLMs) to assist in tasks such as text refinement and image generation.

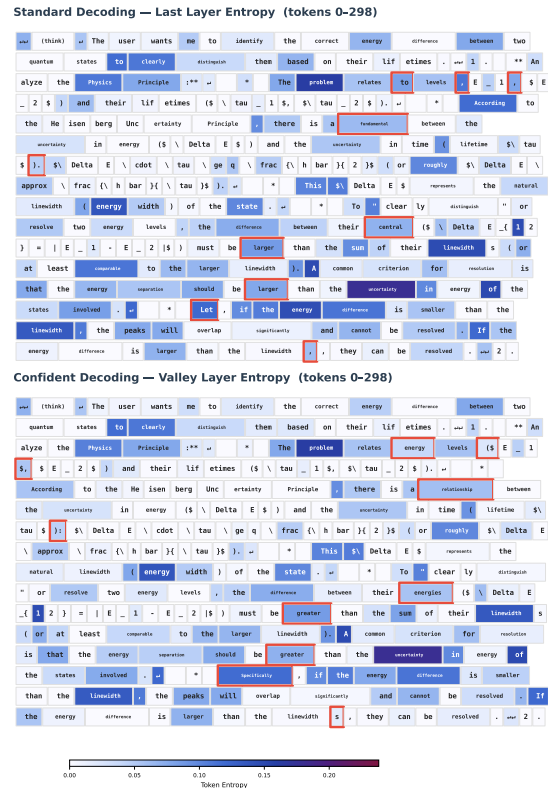


Figure 5: Visualization of token entropy (part 1).

Base Model				Instruct Model				
Standard (Last Layer)		Confident (Trough)		Standard (Last Layer)		Confident (Trough)		
1	.	4.8%	Actually	4.1%	.	7.7%	(\$	6.7%
2	the	4.4%	< box_end >	2.5%	\{\}n	4.2%	carb	2.1%
3	,	3.1%	careful	2.3%	the	4.2%	ring	1.8%
4	is	2.9%	perpendicular	1.9%	\$	3.5%	\$,	1.7%
5	(space)	2.7%	carb	1.8%	is	3.5%	\$\{\}	1.6%
6	k	2.5%	phenotype	1.7%	\$	3.0%	Structure	1.6%
7	g	2.0%	fusion	1.7%	(space)	2.1%	\$.	1.4%
8	(	1.9%	maybe	1.5%	(	2.1%	\$(	1.3%
9	:	1.9%	fused	1.5%	,	2.0%	option	1.1%
10	a	1.8%	indeed	1.4%	\$\{\}	1.7%	Sometimes	1.0%
11	So	1.8%	actually	1.3%	).	1.5%	methyl	0.9%
12	to	1.7%	each	1.3%	?	1.5%	***	0.9%
13	with	1.4%	neither	1.2%	\{\}	1.4%	\{\}	0.9%
14	The	1.3%	an	1.0%	The	1.3%	normalization	0.7%
15	that	1.3%	mutating	1.0%	a	1.3%	definitely	0.7%
16	it	1.2%	exactly	0.8%	=	1.2%	Corey	0.6%
17	G	1.2%	oxygen	0.8%	So	1.2%	an	0.6%
18	)	1.0%	sqrt	0.8%	:	1.1%	?).	0.6%
19	But	1.0%	acting	0.8%	to	1.0%	+	0.6%
20	are	1.0%	sometimes	0.7%	Carb	1.0%	's	0.6%
Category breakdown of all substituted tokens (%):								
	Content	29%		77%		26%		60%
	Function	39%		10%		27%		6%
	Punctuation	22%		6%		43%		26%
	Other	10%		7%		4%		8%

Table 14: Top-20 most frequently substituted tokens when Confident Decoding disagrees with Standard Decoding, comparing Base and Instruct variants of Qwen3.5-35B-A3B on GPQA-Diamond. *Standard* columns show the tokens that the final layer (layer 40) would emit; *Confident* columns show what the entropy-trough layer selects instead. Percentages denote the fraction of all substitution events in each model. Category breakdowns (Content / Function / Punctuation) are summarized at the bottom. Base: 956 substitutions out of 40,526 tokens (2.4%); Instruct: 939 out of 36,111 (2.6%).



Figure 6: Visualization of token entropy (part 2).



Figure 7: Visualization of token entropy (part 3).



Figure 8: Visualization of token entropy (Part 4). For brevity, we omit the CoT following Figure 7 and preceding Figure 8.



Figure 9: Visualization of token entropy (part 5).

Standard Decoding — Last Layer Entropy (tokens 1839-2162)



Confident Decoding — Valley Layer Entropy (tokens 1839-2162)

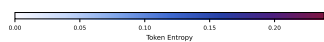


Figure 10: Visualization of token entropy (part 6).