BEYOND EMBEDDING MIXING: VALUE-LEVEL CANDIDATE FUSION FOR LLM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Chain-of-thought reasoning selects a single token at each decoding step, limiting the direct use of alternative-candidate information. Soft Thinking mixes candidate embeddings to preserve multiple reasoning directions, but its trajectory often favors the top-1 candidate. Subsequent work introduces reweighting or regularization to mitigate this bias. However, our layerwise analysis shows that these adjustments can change the dominant candidate without preventing predictions from progressively approaching a single candidate's distribution across layers. We hypothesize that, when candidates are merged into a shared input representation before entering the network, layerwise processing amplifies higher-weighted candidates' influence while attenuating alternatives. We therefore examine the role of aggregation depth and find that delayed aggregation alleviates single-candidate dominance in model predictions. Building on this finding, we propose DIVER, a training-free decoding method that independently contextualizes candidate tokens under the same reasoning prefix before fusing their attention Values at selected internal layers. DIVER maintains a single discrete generation trajectory and, at these layers, replaces the selected token's Values with a probability-weighted mixture of the independently computed candidate Values. The fused Values are stored in the KV cache, making information from unselected candidates accessible to subsequent tokens through attention. We conduct Value-path patching experiments to identify fusion layers where candidate information can effectively influence subsequent reasoning. Entropy gating further concentrates fusion on uncertain steps, retaining standard CoT elsewhere to avoid unnecessary perturbations and computation. Experiments across reasoning benchmarks demonstrate average accuracy gains of up to **2.90%** and **4.14%** over CoT and Soft Thinking, respectively.

1 INTRODUCTION

Large language models (LLMs) generate intermediate reasoning steps autoregressively through chain-of-thought (CoT) (Wei et al., 2023), achieving strong performance across reasoning tasks. Each step becomes part of the context for subsequent reasoning. However, standard CoT selects a single token from the next-token distribution and uses it as the next input. At uncertain steps, other plausible candidates are not explicitly propagated to subsequent reasoning.

Recent research explores continuous representations as an alternative to discrete tokens for intermediate reasoning, aiming to improve reasoning performance by carrying richer information forward within a single generation trajectory. This line of work includes two representative directions. One, exemplified by Coconut (Hao et al., 2025), uses additional supervised training to enable reasoning with continuous hidden states, although subsequent analysis identifies reliance on dataset shortcuts (Zhang et al., 2025a). The other, exemplified by Soft Thinking (Zhang et al., 2025b), constructs the next input as a probability-weighted mixture of candidate token embeddings without updating model parameters. However, Wu et al. (2026) find that Soft Thinking predictions tend to become increasingly dominated by the top-1 candidate, forming a greedy feedback loop in subsequent generation that suppresses exploration of alternative candidate paths. Rizvi-Martel et al. (2026) further examine training-free continuous reasoning methods, showing that candidate superposition is often not effectively utilized. This suggests that Soft Thinking may not support multi-path reasoning, limiting its potential gains over standard CoT.

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

To improve candidate utilization, variants of Soft Thinking introduce input-level interventions. Stochastic Soft Thinking (Wu et al., 2026) adjusts candidate weights through Gumbel-Softmax, while SeLaR (Fu & Luo, 2026) uses contrastive regularization to amplify the offset of the mixed embedding relative to Top-1. We examine these methods by measuring the Jensen–Shannon (JS) divergence between layerwise Logit Lens (nostalgebraist, 2020) distributions induced by mixed inputs and discrete candidates. Although these adjustments can change which candidate dominates, the mixed-input distributions still collapse toward the distribution induced by a single discrete candidate. This collapse may arise as layerwise processing amplifies the input mixture's bias toward higher-weighted candidates, weakening the influence of alternatives. We therefore examine whether directly reintroducing independently computed candidate representations at different internal layers can restore this influence and mitigate single-candidate dominance. The results indicate that delaying candidate aggregation alleviates single-candidate dominance in model predictions.

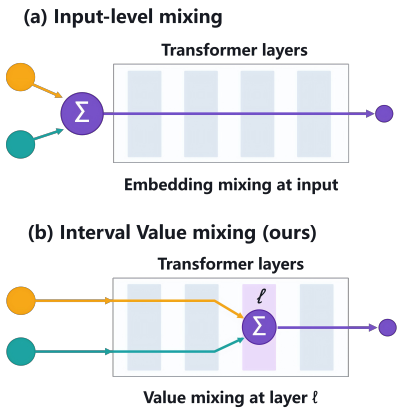


Figure 1: Soft Thinking mixes candidate embeddings at the input. DIVER processes candidates independently before fusing their Values at internal layers.

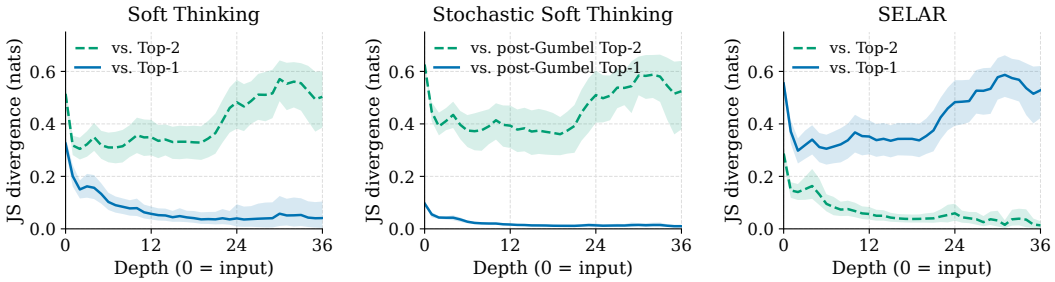
Building on this observation, we propose DIVER, a training-free method that independently contextualizes candidate tokens and fuses their attention Values at selected middle-to-late layers for continuous reasoning. DIVER first processes each candidate independently. To make these resulting information available for subsequent reasoning, we aggregate candidate information in attention Value space. Since Values carry the content read by self-attention, subsequent computations can directly access and utilize the aggregated information through attention. Compared with directly mixing the full hidden states of different candidates, Value aggregation focuses the intervention on introducing candidate information and may induce smaller representational perturbations. Recent research (Izawa et al., 2026; Mishra et al., 2026) also finds that directly intervening on the attention residual is more likely to push representations away from the model's natural activation distribution, impairing generation coherence, whereas localized interventions targeting attention heads can mitigate this issue. To select fusion layers, we use Value-path patching on controlled arithmetic tasks to identify layers where candidate-specific Values affect downstream answer preferences. During decoding, DIVER activates candidate Value fusion at these layers only when the model has low confidence in its next-token prediction, allowing multiple candidates to contribute to subsequent reasoning; at other steps, it follows standard CoT decoding to avoid unnecessary intervention.

Our main contributions are as follows:

- We propose DIVER, a training-free inference method that augments a single discrete reasoning trajectory with information from unselected candidates. Evaluations across model scales and reasoning benchmarks demonstrate improvements over CoT and Soft Thinking.
- Our layerwise analysis shows that predictions from input-level mixing remain dominated by a single candidate. We further observe that independent candidate computation followed by delayed aggregation at middle-to-late layers mitigates this dominance, strengthening the predictive influence of alternative candidates.
- We introduce Value-space fusion to enable multiple candidates to contribute to subsequent reasoning through attention. We conduct Value-path patching to identify layers where Value interventions most effectively influence subsequent predictions. Entropy gating concentrates fusion on uncertain steps, limiting unnecessary interventions and computation.

2 Background

CoT Reasoning. Given a query x, a language model generates intermediate reasoning steps followed by a final answer. At decoding step t, it predicts a distribution $p_t(v) = P_\theta(v \mid x, y_{<t})$ over the vocabulary $\mathcal{V}$. Standard CoT decoding selects a single token y_t through greedy decoding or sampling and uses its embedding $E(y_t)$ as the next input.



(a) (b) (c)

Figure 2: **Layerwise single-candidate dominance after input-level adjustments.** (a) Soft Thinking approaches Top-1 while remaining substantially different from Top-2. (b) Stochastic Soft Thinking approaches post-Gumbel Top-1 while remaining substantially different from post-Gumbel Top-2 (32 draws per position, $\tau = 0.5$). (c) SELAR approaches Top-2 while remaining substantially different from Top-1.

Soft Thinking. Soft Thinking replaces discrete token inputs during intermediate reasoning with probability-weighted mixtures of candidate embeddings, without updating model parameters. Given the top-k candidates $\{c_{t,i}\}_{i=1}^{k}$, it constructs a soft input:

$$e_t = \sum_{i=1}^{k} \widehat{p}_{t,i} E(c_{t,i}), \qquad \widehat{p}_{t,i} = \frac{p_t(c_{t,i})}{\sum_{j=1}^{k} p_t(c_{t,j})}. \tag{1}$$

Here, $\widehat{p}_{t,i}$ denotes the renormalized probability. The model processes this mixture as its next input, aiming to retain information from multiple candidates within a single reasoning trajectory.

3 Motivation

We examine whether the depth of candidate aggregation changes the final prediction. Layerwise Logit Lens (nostalgebraist, 2020) analysis first assesses input-level adjustments. We then mix independently computed candidate representations at different layers to test delayed aggregation. Section 4 develops the resulting Value-level intervention.

Single-Candidate Dominance. Following Wu et al. (2026), we examine the layerwise behavior of Soft Thinking using 240 branch points with entropy $H > 0.5$ extracted from Qwen3-8B's CoT trajectories on AIME 2024. At each point, we separately process the top-1 candidate c_1, top-2 candidate c_2, and a manually constructed soft token $e_{\text{soft}} = 0.6E(c_1) + 0.4E(c_2)$, using the same mixing weights as Wu et al. We measure the Jensen–Shannon divergence (Lin, 1991) between the Logit Lens distributions induced by the soft token and each discrete input.

We extend this analysis to Stochastic Soft Thinking (Wu et al., 2026) and SeLaR (Fu & Luo, 2026), which respectively use Gumbel-Softmax reweighting and contrastive adjustment of the input mixture. Figure 2 indicates that, as depth increases, the mixed-input distribution approaches Top-1 for Soft Thinking, post-perturbation Top-1 for Stochastic Soft Thinking, and Top-2 for SeLaR, while remaining distinct from the other candidate in each case. Thus, in this two-candidate setting, input-level adjustments change the dominant candidate without eliminating single-candidate dominance.

Despite these adjustments, these methods still merge candidate token embeddings into a shared input representation without first contextualizing each candidate independently. We hypothesize that higher-weighted candidates exert a stronger influence on this shared representation and that subsequent layerwise processing may further amplify this influence while attenuating the predictive influence of alternative candidates, leading to single-candidate dominance. Motivated by this hypothesis, we examine whether this dominance can be mitigated by first contextualizing candidates independently under the same reasoning prefix, then forming a weighted mixture of their representations at middle-to-late layers and injecting the resulting mixture into the main path.

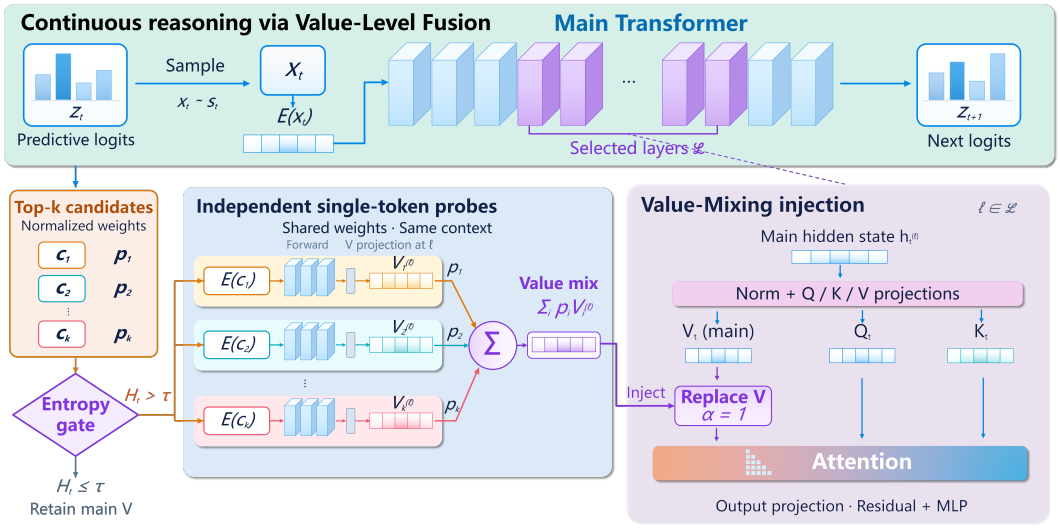


Figure 4: **Overview of DIVER.** At high-entropy steps, DIVER independently processes top candidates and replaces the selected token's Values at selected layers with their probability-weighted mixture. The fused Values are cached for subsequent decoding. Other steps follow standard CoT.

Delayed Candidate Aggregation. We process both candidates independently under the same reasoning prefix and aggregate their residual representations at a selected internal layer, rather than mixing their embeddings at the input. This allows each candidate, including the lower-weighted one, to form its own contextual representation before the mixing weights are applied.

Specifically, at each branch point, we use the Top-1 candidate as the main input and replace its residual representation at layer ℓ with $0.6h_\ell^{(1)} + 0.4h_\ell^{(2)}$, where $h_\ell^{(i)}$ is computed independently for candidate c_i under the same prefix. We vary the intervention layer and measure overlap between the resulting final top-10 next-token predictions and the top-10 predictions from each discrete-candidate reference run.

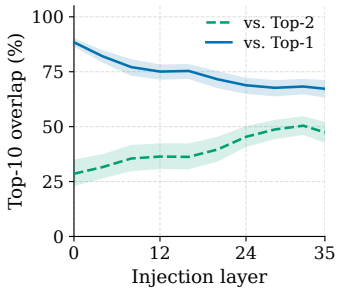


Figure 3: Overlap of residual-mixed top-10 predictions with the corresponding single-candidate reference predictions.

As shown in Figure 3, Top-2 overlap increases from approximately 29% at the earliest injection layer to around 50% with injection at middle-to-late layers. The final top-10 predictions retain tokens from the Top-1 reference set while including more tokens from the Top-2 reference set. These findings indicate that deferring candidate aggregation can mitigate single-candidate dominance in next-token prediction.

4 Method

We introduce DIVER, a training-free decoder guided by the delayed-aggregation observation in Section 3. As illustrated in Figure 4, it maintains one discrete sequence and independently contextualizes additional candidates at uncertain steps. Value-path patching selects the layers for candidate Value fusion. The fused Values enter the main path's attention computation and KV cache, making candidate content available to subsequent tokens.

4.1 Multi-Candidate Reasoning in Value Space

Independent candidate contextualization. At decoding step t, let $p_t(v)$ be the next-token distribution and $c_{t,1}, \ldots, c_{t,k}$ its top-k candidates. We fix the generated reasoning prefix and its historical KV cache. Each candidate $c_{t,i}$ undergoes an independent single-token forward pass from its own

embedding, without current-step fusion. Thus, each candidate acquires a contextual representation before the aggregation weights are applied.

Choice of aggregation space. After independent contextualization, we aggregate candidate information in attention Value space to support subsequent reasoning. In self-attention, Queries and Keys determine attention weights, while Values provide the content aggregated by those weights (Vaswani et al., 2023). DIVER fuses candidate Values while retaining Queries and Keys at each intervention site, introducing candidate information without directly changing the local attention weights. Unlike directly mixing full residual states, Value fusion confines the direct intervention to the attention module. This design is motivated by previous works that residual-stream interventions can shift hidden states away from their natural activation distribution (Mishra et al., 2026; Luo et al., 2026) and that fine-grained interventions can reduce unnecessary modifications while improving steering effectiveness (Feng et al., 2026; Torop et al., 2025).

For each candidate, we extract its Values at the selected fusion layers from the independent forward pass:

$$V_{t,i}^{(\ell)} = \Phi_{t,\ell}(E(c_{t,i})), \qquad \ell \in \mathcal{L}, \tag{2}$$

where E denotes the token embedding map, $\Phi_{t,\ell}$ maps a candidate embedding to its Value at layer ℓ under the fixed historical context, and $\mathcal{L}$ is the set of fusion layers.

Fusion layer selection and candidate aggregation. To determine $\mathcal{L}$, we design a Value-path patching experiment based on activation patching (Zhang & Nanda, 2024) to measure how candidate Value interventions at different layers affect model predictions. We first construct 1,000 simple arithmetic tasks, each with a fixed problem and reasoning prefix. We then individually insert two randomly sampled numbers, c_1 and c_2, as the intermediate answer and apply the same remaining calculation to obtain the corresponding final answers, y_1 and y_2.

For each task, we separately process c_1, c_2, and their soft mixture $0.6E(c_1) + 0.4E(c_2)$ at the intermediate-answer position. We group consecutive model layers into windows and replace the soft mixture's Values in each window W with those produced by c_1 or c_2 at the corresponding layers and position.

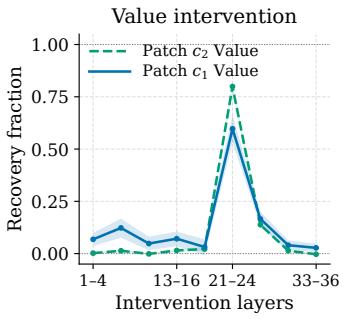


Figure 5: Recovery ratios for candidate Value replacement across layers on Qwen3-8B.

To quantify the effect of these Value replacements, we compute the model's preference $M = \log P(y_1) - \log P(y_2)$ using teacher forcing (Bengio et al., 2015). Let M_{soft} and M_i denote this score under the original soft mixture and discrete candidate c_i, respectively, and $M_{i,W}^{\text{patch}}$ the score after replacing the soft mixture's Values with those from c_i. We then define the recovery ratio is

$$R_i(W) = \frac{M_{i,W}^{\text{patch}} - M_{\text{soft}}}{M_i - M_{\text{soft}}}, \qquad i \in \{1, 2\}. \tag{3}$$

Here, $R_i(W) = 0$ indicates unchanged preference, while $R_i(W) = 1$ means that the patched preference score matches that of the discrete candidate.

We select the $\mathcal{L}$ with the highest recovery ratio as our fusion layers. For instance, Figure 5 shows that Value replacement in layers 21–24 has the strongest intervention effect on Qwen3-8B, so we use these layers for fusion. Results for the other models are reported in detail in Appendix A.3.

At each selected layer, we aggregate the independently computed candidate Values using their renormalized probabilities:

$$V_{t,\text{mix}}^{(\ell)} = \sum_{i=1}^{k} \widehat{p}_{t,i} V_{t,i}^{(\ell)}, \qquad \ell \in \mathcal{L}. \tag{4}$$

Value replacement and decoding. We select a token y_t using standard CoT decoding and feed it into the model at decoding step t. When the forward pass reaches a fusion layer ℓ, we update the

token's Value $V_{t,\text{main}}^{(\ell)}$ by blending it with the candidate mixture $V_{t,\text{mix}}^{(\ell)}$ computed at the same layer:

$$\widetilde{V}_t^{(\ell)} = (1-\alpha)V_{t,\text{main}}^{(\ell)} + \alpha V_{t,\text{mix}}^{(\ell)}, \qquad 0 \le \alpha \le 1, \quad \ell \in \mathcal{L}. \tag{5}$$

We use $\alpha = 1$ to replace the selected token's Value with the probability-weighted candidate mixture. . The fused Value enters the attention computation and is cached alongside the main path's Key, so later tokens can access the aggregated candidate content. Decoding then continues autoregressively along the selected token sequence.

4.2 Entropy-Gated Activation

Since the next-token distribution is concentrated on the top-1 candidate when the model is confident, other candidates contribute little to Value fusion. Value-level Fusion is therefore unlikely to provide substantial gains, while adding computational overhead and potentially introducing unnecessary perturbations. Following SeLaR (Fu & Luo, 2026), we enable Value fusion only at high-entropy decoding steps, where $H_t = -\sum_{i=1}^{k} \widehat{p}_{t,i} \log \widehat{p}_{t,i} > \eta$. Otherwise, decoding follows standard CoT. We set $\eta = 0.5$ in our experiments.

5 Experiments

5.1 Experimental Setup

Our main evaluation covers mathematical reasoning on AIME 2024 (Hugging Face H4, 2025), AIME 2025 (Zhang & Math-AI Team, 2025), GSM8K (Cobbe et al., 2021), and MATH500 (Lightman et al., 2023), as well as scientific reasoning on GPQA-Diamond (Rein et al., 2023). We evaluate DIVER on these benchmarks using Qwen3-4B-Thinking-2507, Qwen3-8B (Qwen Team, 2025a), and DeepSeek-R1-Distill-Qwen-14B (Guo et al., 2025). We additionally assess the performance of DIVER on three coding benchmarks, HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021), and LiveCodeBench (Jain et al., 2025), and four more models, Qwen3-1.7B, Qwen3-32B, QwQ-32B (Qwen Team, 2025b), and DeepSeek-R1-Distill-Qwen-32B.

Baselines include CoT with sampling and greedy decoding, together with four training-free latent reasoning methods. Soft Thinking (Zhang et al., 2025b) combines candidate embeddings at the input. Stochastic Soft Thinking (Wu et al., 2026) applies Gumbel–Softmax sampling to the mixture weights. SwiReasoning (Shi et al., 2026) switches between explicit and latent reasoning according to entropy trends, while SeLaR (Fu & Luo, 2026) combines selective activation with contrastive regularization of the input mixture. Detailed descriptions of these baselines are provided in Appendix A.2.

5.2 Main Results

DIVER achieves the highest average accuracy on all three models in Table 1, outperforming CoT by 2.90 points on Qwen-4B, 1.22 on Qwen3-8B, and 1.15 on DeepSeek-R1-Distill-Qwen-14B. Across the 15 model–benchmark comparisons, DIVER improves over CoT in 14 and ranks first in 11. Notable gains include 6.25 points on Qwen-4B AIME 2025 and 4.04 points on Qwen-4B GPQA-Diamond, while GSM8K accuracy remains close to CoT.

Compared with Soft Thinking, DIVER improves average accuracy by 2.15–4.14 points across the three backbones, with particularly large gains of 10.00 points on Qwen3-8B AIME 2024 and 11.25 points on DeepSeek-14B AIME 2025. On Qwen3-8B and DeepSeek-14B, Soft Thinking falls below CoT on average, whereas DIVER exceeds both. Among the latent reasoning baselines, SwiReasoning achieves the highest average accuracy on all three models and surpasses DIVER by 0.44 points on Qwen3-8B GPQA-Diamond. DIVER nevertheless maintains higher average accuracy than SwiReasoning, Stochastic Soft Thinking, and SeLaR across all three backbones.

These results support DIVER's strategy of independently processing candidates before fusing their Values at middle-to-late layers. Delaying fusion may reduce the attenuation of candidate-specific information associated with early mixing, and allow alternative candidates to continue influencing subsequent reasoning. The larger gains on AIME suggest that this approach is particularly useful

Table 1: Reasoning accuracy (%) across five benchmarks, reported as avg@8 except for CoT (greedy), which is evaluated in a single run.

Method	AIME 2024	AIME 2025	GPQA-Diamond	GSM8K	MATH500	Average
			Qwen3-4B-Thinking-2507			
CoT	81.67	72.50	64.08	94.90	97.63	82.16
CoT (greedy)	80.00	73.33	64.65	95.07	97.20	82.05
Soft Thinking	82.92	74.58	64.02	95.38	97.63	82.91
Stochastic Soft Thinking	82.92	72.92	64.58	**95.58**	97.65	82.73
SwiReasoning	82.08	74.17	65.34	95.35	98.13	83.01
SeLaR	79.58	76.25	64.14	94.62	97.80	82.48
DIVER	**84.58**	**78.75**	**68.12**	95.30	**98.53**	**85.06**
			Qwen3-8B			
CoT	77.08	68.33	57.64	95.77	96.88	79.14
CoT (greedy)	73.33	66.67	52.53	95.45	96.20	76.84
Soft Thinking	68.75	70.83	56.50	95.68	95.90	77.53
Stochastic Soft Thinking	75.83	64.17	58.08	**95.82**	96.75	78.13
SwiReasoning	75.83	63.75	**59.34**	95.56	96.80	78.26
SeLaR	72.92	69.58	57.58	95.15	95.60	78.17
DIVER	**78.75**	**71.67**	58.90	95.50	**97.00**	**80.36**
			DeepSeek-R1-Distill-Qwen-14B			
CoT	70.42	53.75	57.64	94.70	95.28	74.36
CoT (greedy)	70.00	33.00	52.53	95.00	93.20	68.75
Soft Thinking	66.67	44.17	57.70	94.88	93.43	71.37
Stochastic Soft Thinking	67.92	50.83	58.33	94.79	95.05	73.38
SwiReasoning	71.25	52.92	57.70	94.84	95.45	74.43
SeLaR	62.92	54.58	58.59	**95.38**	94.40	73.17
DIVER	**72.08**	**55.42**	**59.53**	94.92	**95.60**	**75.51**

for difficult problems, where finding a viable solution often requires more extensive exploration of reasoning directions across interdependent steps.

Runtime. Figure 6 compares accuracy and runtime averaged across AIME 2024 and 2025 on Qwen3-4B-Thinking-2507 and Qwen3-8B. For Self-Consistency (SC@8) (Wang et al., 2023), accuracy is computed by aggregating answers from eight CoT sampling runs, and runtime is summed across these runs. For the other methods, both accuracy and runtime are averaged over eight runs. Our method achieves the highest mean accuracy among the single-trajectory methods on both models. Its mean runtimes are 50.2 and 37.7 minutes, compared with 104.4 and 102.4 minutes for SC@8, corresponding to reductions of approximately 52% and 63%, respectively. DIVER therefore combines improved accuracy over the single-trajectory baselines with substantially lower runtime than Self-Consistency.

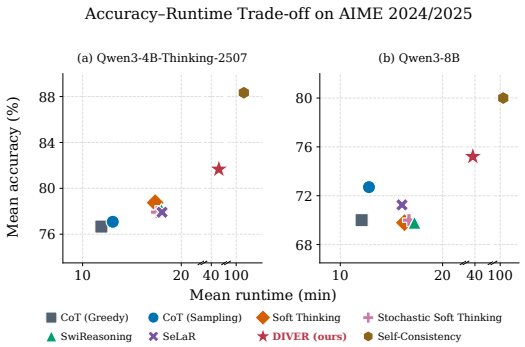


Figure 6: Mean accuracy and runtime of DIVER and baseline methods across AIME 2024 and AIME 2025.

5.3 Generalization

To assess DIVER across model scales and task domains, we extend the evaluation to Qwen-1.7B, Qwen-32B, QwQ-32B, and DeepSeek-32B on multiple datasets, and test Qwen3-8B on three coding benchmarks: LiveCodeBench, HumanEval, and MBPP.

Across models. We additionally evaluate DIVER on smaller and larger backbones at 1.7B and 32B parameters, with results reported in Table 2. DIVER achieves the highest average accuracy across the four backbones, reaching 74.17% on AIME 2024 and 60.00% on AIME 2025. Notably, we

Table 2: AIME pass@1 accuracy (%) on four additional models. Stochastic ST denotes Stochastic Soft Thinking. DeepSeek-32B denotes DeepSeek-R1-Distill-Qwen-32B. Full results on the other benchmarks are provided in Appendix C.

	Qwen3-1.7B		Qwen3-32B		QwQ-32B		DeepSeek-32B		Average	
Method	AIME24	AIME25	AIME24	AIME25	AIME24	AIME25	AIME24	AIME25	AIME24	AIME25
CoT	**50.00**	33.00	83.33	66.67	80.00	66.67	66.67	53.33	70.00	54.92
CoT (greedy)	40.00	36.67	83.33	66.67	76.67	63.33	63.33	43.33	65.83	52.50
Soft Thinking	40.00	33.00	80.00	**76.67**	83.33	**70.00**	63.33	50.00	66.67	57.42
Stochastic ST	46.67	36.67	**86.67**	66.67	**86.67**	56.67	66.67	50.00	71.67	52.50
SwiReasoning	**50.00**	36.67	**86.67**	66.67	76.67	66.67	73.33	53.33	71.67	55.84
SeLaR	40.00	**43.33**	83.33	63.33	76.67	63.33	70.00	53.33	67.50	55.83
DIVER	46.67	**43.33**	**86.67**	73.33	**86.67**	66.67	**76.67**	**56.67**	**74.17**	**60.00**

observe that the effectiveness of input-level mixing varies across model scales, as shown in Tables 1 and 2. Soft Thinking is more competitive on some 32B backbones but underperforms CoT on Qwen-1.7B and Qwen3-8B, possibly reflecting differences in tolerance to mixed embeddings outside the training distribution. Our approach achieves more consistent improvements over CoT across model scales, which may be attributed to its selective use of internal fusion. Restricting modifications to high-entropy steps and a small set of key middle-to-late layers may limit distribution shifts, while independently computed candidate Values preserve alternative information for subsequent reasoning, enabling greater exploration.

Code generation. Beyond mathematical and scientific reasoning, we also evaluate DIVER on code generation benchmarks using Qwen3-8B. As shown in Table 3, DIVER achieves 62.01% accuracy on LiveCodeBench, outperforming CoT and the strongest baseline, SwiReasoning, by 2.88 and 2.51 points, respectively. DIVER also achieves competitive results on HumanEval and MBPP, with accuracies of 95.12% and 95.33%, respectively. The smaller differences relative to CoT on HumanEval and MBPP may reflect their relatively lower difficulty for Qwen3-8B, as CoT already exceeds 94% accuracy on both datasets, leaving limited room for further improvement. In contrast, DIVER's substantial gain on the more challenging LiveCodeBench benchmark suggests that exploring alternative reasoning directions is effective for solving complex coding problems.

Table 3: Code generation pass@1 accuracy on Qwen3-8B. LCB denotes LiveCodeBench. Stochastic ST denotes Stochastic Soft Thinking

Method	LCB	HumanEval	MBPP
CoT	59.13	94.51	95.72
CoT (greedy)	53.05	85.98	88.72
Soft Thinking	52.33	**95.12**	91.44
Stochastic ST	58.06	93.90	**96.11**
SwiReasoning	59.50	93.29	95.33
SeLaR	58.42	92.07	95.72
DIVER	**62.01**	**95.12**	95.33

5.4 Ablation Studies

We vary fusion depth and candidate count on Qwen3-8B to examine where candidate information is most useful and how broadly it should be retained. Additional ablations on the mixing strength α and the activation threshold η are provided in appendix D.

Table 4: Effect of fusion layers and candidate count on Qwen3-8B.

Layers	K	AIME24	AIME25	GPQA-D	GSM8K	MATH500	Avg.
1–4	3	76.67	70.00	58.08	95.75	97.20	79.54
	10	76.67	**76.67**	57.07	95.45	96.60	80.49
	15	73.33	63.33	58.59	95.68	96.60	77.51
21–24	3	80.00	73.33	59.09	**95.91**	97.00	81.07
	10	**83.33**	73.33	**60.10**	95.53	**97.40**	**81.94**
	15	80.00	70.00	58.59	95.30	96.80	80.14
All layers	3	70.00	53.33	59.09	95.38	**97.40**	75.04
	10	66.67	70.00	56.06	95.83	96.40	76.99
	15	66.67	63.33	55.05	95.83	96.40	75.46

Fusion layers and candidate count. Fusion at layers 21–24 achieves the highest average accuracy across all evaluated candidate counts (K), outperforming fusion at layers 1–4 (Table 4). Consistent with our analysis, early fusion may allow candidate-specific information to weaken during subsequent layerwise processing, driving predictions toward a single candidate. Delayed fusion instead allows candidates to develop contextual representations independently through earlier layers before aggregation. This may reduce information attenuation after fusion and preserve alternative candidates' influence on subsequent reasoning, supporting implicit broader exploration. Extending fusion to all layers, however, reduces average accuracy. Repeated interventions throughout the network may amplify deviations from the model's training distribution, offsetting the benefits of alternative candidate information.

Across most fusion depths, $K = 3$ and $K = 10$ yield higher average accuracy than $K = 15$, with $K = 10$ performing best at all three depths. These results indicate that increasing the candidate count does not necessarily improve reasoning performance. Including more low-probability candidates may introduce information less relevant to the current reasoning while diluting the contributions of more useful candidates, thereby reducing reasoning accuracy.

Value versus residual fusion. We next compare Value and residual fusion on AIME across three Qwen models of different sizes (Table 5).[1] Value fusion achieves better overall performance than residual fusion. This difference may reflect the broader perturbations introduced by directly mixing residual states, which can shift hidden representations away from the model's natural activation distribution and disrupt subsequent reasoning. By modifying attention Values rather than the full residual state, Value fusion may introduce less disruption while allowing alternative-candidate information to contribute to subsequent reasoning. We further examine this issue in the following subsection.

Table 5: Value and residual fusion on AIME.

Model	Value fusion		Residual fusion	
	AIME24	AIME25	AIME24	AIME25
Qwen-4B	**86.67**	**80.00**	76.67	73.33
Qwen3-8B	**83.30**	**73.30**	76.67	70.00
Qwen-32B	**86.67**	**73.33**	**86.67**	66.67

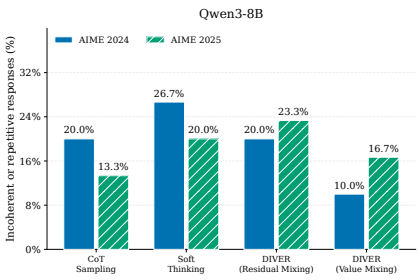


Figure 7: Incoherent or repetitive responses on Qwen3-8B.

OOD Analysis. Soft Thinking attributes repetitive generation collapse to OOD inputs introduced by continuous token embeddings (Zhang et al., 2025b). Motivated by this observation, we examine generation degradation potentially associated with OOD representations. We use GPT-4o (OpenAI, 2024) to inspect the thinking traces of all 30 responses generated by Qwen3-8B per method on each AIME benchmark and measure the proportion of incoherent or repetitive responses. As shown in Figure 7, Soft Thinking and residual fusion exhibit higher rates of such responses on AIME 2024 and AIME 2025, while Value fusion and CoT sampling exhibit lower rates. These results suggest that restricting fusion to Value space reduces generation disruption relative to embedding and residual mixing.

6 Conclusion

We presented DIVER, a training-free decoding method that defers candidate aggregation to selected internal layers after independent candidate processing. It is motivated by our analysis showing that input-level adjustments do not eliminate single-candidate dominance, while delayed aggregation strengthens the predictive influence of alternative candidates. DIVER performs fusion in attention Value space to make candidate information available to subsequent reasoning through interventions focused on the content transmitted by attention. Meanwhile, Value-path patching identifies effective fusion layers. Experiments across model scales and reasoning benchmarks demonstrate DIVER's strong performance. Despite its effectiveness, DIVER incurs computational overhead from independent candidate processing and relies on next-token probabilities for fusion, which do not necessarily reflect candidates' contributions to subsequent reasoning. Future work will explore more efficient candidate computation and fusion weights that better reflect reasoning utility.

[1] Details of layer selection for residual fusion are provided in appendix A.3.

Ethics Statement

This work develops a training-free method for improving LLM reasoning using publicly available models and benchmarks. Although DIVER improves performance in our experiments, its outputs may still contain factual errors, flawed reasoning, or biases inherited from the underlying models. Generated solutions should therefore be independently verified before use in consequential applications, with appropriate human oversight. We also acknowledge the additional computational costs associated with independent candidate processing.

Reproducibility Statement

To facilitate reproducibility, we provide implementation details, decoding settings, and model-specific fusion configurations in Appendix A.3. The benchmarks and models are described in Section 5.1. All evaluation datasets and pretrained models used in this study are publicly available.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models, 2021. URL https://arxiv.org/abs/2108.07732.

Ali Backour. The dynamics of continuous mixture collapse in language models, 2026. URL https://arxiv.org/abs/2609.02049.

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens, 2025. URL https://arxiv.org/abs/2303.08112.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks, 2015. URL https://arxiv.org/abs/1506.03099.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michał Podstawski, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17682–17690, Mar 2024. doi: 10.1609/aaai.v38i16.29720. URL https://ojs.aaai.org/index.php/AAAI/article/view/29720.

Eden Biran, Daniela Gottesman, Sohee Yang, Mor Geva, and Amir Globerson. Hopping too late: Exploring the limitations of large language models on multi-hop queries, 2024. URL https://arxiv.org/abs/2406.12775.

Natasha Butt, Ariel Kwiatkowski, Ismail Labiad, Julia Kempe, and Yann Ollivier. Soft tokens, hard truths. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=9JjKTp8Jmy.

Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. Dola: Decoding by contrasting layers improves factuality in large language models, 2024. URL https://arxiv.org/abs/2309.03883.

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

Zijian Feng, Tianjiao Li, Zixiao Zhu, Hanzhang Zhou, Junlang Qian, Li Zhang, Jia Jim Deryl Chua, Lee Onn Mak, Gee Wah Ng, and Kezhi Mao. Fine-grained activation steering: Steering less, achieving more, 2026. URL https://arxiv.org/abs/2602.04428.

Renyu Fu and Guibo Luo. Selar: Selective latent reasoning in large language models, 2026. URL https://arxiv.org/abs/2604.08299.

Halil Alperen Gozeten, Muhammed Emrullah Ildiz, Xuechen Zhang, Hrayr Harutyunyan, Ankit Singh Rawat, and Samet Oymak. Continuous chain of thought enables parallel exploration and reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=sTPKDKn5ig.

Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://www.nature.com/articles/s41586-025-09422-z.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason E Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Itxz7S4Ip3.

Hugging Face H4. AIME 2024. Hugging Face Datasets, 2025. URL https://huggingface.co/datasets/HuggingFaceH4/aime_2024. 2024 AIME I and II; dataset released in 2025.

Yoshihiro Izawa, Gouki Minegishi, Koshi Eguchi, Sosuke Hosokawa, and Kenjiro Taura. Steering at the source: Style modulation heads for robust persona control. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=r6EuKr1hq4.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=chfJJYC3iL.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=aLLuYpn83y.

Zhaoyi Li, Gangwei Jiang, Hong Xie, Linqi Song, Defu Lian, and Ying Wei. Understanding and patching compositional reasoning in LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics ACL 2024*, pp. 9668–9688, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-acl.576.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991. doi: 10.1109/18.61115.

Grace Luo, Jiahai Feng, Trevor Darrell, Alec Radford, and Jacob Steinhardt. Learning a generative meta-model of LLM activations. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=lXX3Bi2Adq.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt, 2023. URL https://arxiv.org/abs/2202.05262.

Aayush Mishra, Daniel Khashabi, and Anqi Liu. Steered llm activations are non-surjective, 2026. URL https://arxiv.org/abs/2604.09839.

Hosein Mohebbi, Willem Zuidema, Grzegorz Chrupała, and Afra Alishahi. Quantifying context mixing in transformers. In Andreas Vlachos and Isabelle Augenstein (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 3378–3400, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.245. URL https://aclanthology.org/2023.eacl-main.245/.

nostalgebraist. Interpreting gpt: The logit lens. LessWrong Blog Post, August 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

OpenAI. GPT-4o system card. OpenAI, August 2024. URL https://openai.com/index/gpt-4o-system-card/.

Qwen Team. Qwen3 technical report, 2025a. URL https://arxiv.org/abs/2505.09388.

Qwen Team. QwQ-32B: Embracing the power of reinforcement learning. Qwen Blog, March 2025b. URL https://qwenlm.github.io/blog/qwq-32b/.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Michael Rizvi-Martel, Guillaume Rabusseau, and Marius Mosbach. The illusion of superposition? a principled analysis of latent thinking in language models, 2026. URL https://arxiv.org/abs/2604.06374.

Yuval Shalev, Amir Feder, and Ariel Goldstein. Distributional reasoning in llms: Parallel reasoning processes in multi-hop reasoning, 2024. URL https://arxiv.org/abs/2406.13858.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. CODI: Compressing chain-of-thought into continuous space via self-distillation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 677–693, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.36. URL https://aclanthology.org/2025.emnlp-main.36/.

Dachuan Shi, Abedelkadir Asi, Keying Li, Xiangchi Yuan, Leyan Pan, Wenke Lee, and Wen Xiao. Swireasoning: Switch-thinking in latent and explicit for pareto-superior reasoning LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=t33kMzEAg8.

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=aB3Hwh4UzP.

Wenhui Tan, Fiorenzo Parascandolo, Enver Sangineto, Jianzhong Ju, Zhenbo Luo, Qian Cao, Rita Cucchiara, Ruihua Song, and Jian Luan. Restoring exploration after post-training: Latent exploration decoding for large reasoning models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=2jXDMGaNZk.

Yao Tang, Li Dong, Yaru Hao, Qingxiu Dong, Furu Wei, and Jiatao Gu. Multiplex thinking: Reasoning via token-wise branch-and-merge, 2026. URL https://arxiv.org/abs/2601.08808.

Max Torop, Aria Masoomi, Masih Eskandar, and Jennifer Dy. Disco: Disentangled communication steering for large language models, 2025. URL https://arxiv.org/abs/2509.16820.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Kang Wang, Xiangyu Duan, and Tianyi Du. Improving latent reasoning in llms via soft concept mixing, 2025. URL https://arxiv.org/abs/2511.16885.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023. URL https://arxiv.org/abs/2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Junhong Wu, Jinliang Lu, Zixuan Ren, Gangqiang Hu, Zhi Wu, Dai Dai, and Hua Wu. LLMs are single-threaded reasoners: Demystifying the working mechanism of soft thinking. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ASLuOoP78o.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. Softcot: Soft chain-of-thought for efficient reasoning with llms, 2025a. URL https://arxiv.org/abs/2502.12134.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. Softcot++: Test-time scaling with soft chain-of-thought reasoning, 2025b. URL https://arxiv.org/abs/2505.11484.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=5Xc1ecxO1h.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods, 2024. URL https://arxiv.org/abs/2309.16042.

Yifan Zhang and Math-AI Team. American invitational mathematics examination (AIME) 2025, 2025. URL https://huggingface.co/datasets/math-ai/aime25.

Yuyi Zhang, Boyu Tang, Tianjie Ju, Sufeng Duan, and Gongshen Liu. Do latent tokens think? a causal and adversarial analysis of chain-of-continuous-thought, 2025a. URL https://arxiv.org/abs/2512.21711.

Zhen Zhang, Xuehai He, Weixiang Yan, Ao Shen, Chenyang Zhao, Shuohang Wang, Yelong Shen, and Xin Eric Wang. Soft thinking: Unlocking the reasoning potential of llms in continuous concept space. *arXiv preprint arXiv:2505.15778*, 2025b.

Zhi Zheng, Yu Gu, Wei Liu, Yee Whye Teh, and Wee Sun Lee. Soft-grpo: Surpassing discrete-token llm reinforcement learning via gumbel-reparameterized soft-thinking policy optimization, 2026. URL https://arxiv.org/abs/2511.06411.

Hanlin Zhu, Shibo Hao, Zhiting Hu, Jiantao Jiao, Stuart Russell, and Yuandong Tian. Reasoning by superposition: A theoretical perspective on chain of continuous thought. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=UdOEZgWJLc.

Yufan Zhuang, Liyuan Liu, Chandan Singh, Jingbo Shang, and Jianfeng Gao. Text generation beyond discrete token sampling, 2025. URL https://arxiv.org/abs/2505.14827.

A Appendix

A.1 Related Work

Latent Reasoning. Chain-of-thought (CoT) prompting expresses intermediate reasoning through discrete token sequences (Wei et al., 2023). Self-consistency aggregates sampled traces (Wang et al., 2023), while Tree of Thoughts and Graph of Thoughts organize reasoning into branching or recombinable structures (Yao et al., 2023; Besta et al., 2024). Latent reasoning carries intermediate computation in continuous representations. Coconut recurrently feeds hidden states back as inputs (Hao et al., 2025), CODI distills explicit reasoning into continuous states (Shen et al., 2025), and SoftCoT learns to map assistant-generated soft thoughts into a frozen language model (Xu et al., 2025a). SoftCoT++ further promotes diverse latent thoughts (Xu et al., 2025b). Theoretical studies show that continuous states can support parallel exploration under specific constructions and supervision schemes (Zhu et al., 2025; Gozeten et al., 2026).

Training-based approaches additionally optimize how models use continuous mixtures. Soft Tokens, Hard Truths introduces embedding-space noise during reinforcement learning (Butt et al., 2026), while SofT-GRPO uses Gumbel-reparameterized policy optimization (Zheng et al., 2026). Multiplex Thinking independently samples candidate tokens and merges their embeddings into one input before model computation (Tang et al., 2026). Soft Concept Mixing adds a probability-weighted embedding mixture to hidden states and optimizes the policy with GRPO (Wang et al., 2025).

Among training-free methods, Soft Thinking feeds probability-weighted token embeddings back into the model (Zhang et al., 2025b), while Mixture of Inputs combines a sampled token with its generating distribution through Bayesian estimation (Zhuang et al., 2025). Wu et al. (2026) identify dominant-token behavior in Soft Thinking and introduce stochastic reweighting, including Gumbel–Softmax, to improve exploration. SeLaR combines entropy-gated activation with a contrastive adjustment away from the dominant embedding (Fu & Luo, 2026), and SwiReasoning switches between latent and explicit reasoning using block-level entropy trends (Shi et al., 2026). Related analyses report superposition collapse or limited use in pretrained models (Rizvi-Martel et al., 2026) and distinguish mixture distortion from collapse driven by softmax readout and autoregressive feedback (Backour, 2026). Our layerwise analysis examines candidate dominance after input-level adjustments. DIVER postpones fusion until candidate representations have undergone separate contextual computation, then aggregates their attention Values at selected layers without additional training.

Representation Interventions. Layerwise probing reveals evolving predictions and intermediate answer representations (Belrose et al., 2025; Shalev et al., 2024). Causal tracing and mediation analysis identify computations involved in factual recall and arithmetic (Meng et al., 2023; Stolfo et al., 2023), while studies of compositional reasoning connect intermediate representations and their layerwise utilization to downstream success (Biran et al., 2024; Li et al., 2024). Value Zeroing measures context mixing by ablating token-specific attention Values (Mohebbi et al., 2023). Inference-time methods steer activations toward behavioral directions (Li et al., 2023; Rimsky et al., 2024), with DISCO directly separating query- and Value-space interventions (Torop et al., 2025). DoLa and Latent Exploration Decoding exploit intermediate-layer predictions to modify output distributions (Chuang et al., 2024; Tan et al., 2026). DIVER uses Value-path patching to identify layers where independently computed candidate information can recover its influence on later predictions. It aggregates candidate-specific Values using their probabilities and restricts fusion to high-entropy steps. The intervention therefore depends on current candidates and their shared reasoning history.

A.2 Details of the Baselines

Soft Thinking (Zhang et al., 2025b) constructs continuous reasoning inputs through probability-weighted mixtures of candidate token embeddings. These soft inputs carry information from multiple candidates into subsequent computation. The method also introduces an entropy-based Cold Stop mechanism to terminate reasoning when predictive uncertainty remains low, after which the model returns to discrete decoding for answer generation.

Stochastic Soft Thinking (Wu et al., 2026) introduces randomness into the weights used to construct soft inputs. Motivated by the tendency of Soft Thinking to favor the highest-probability can-

didate, it uses stochastic reweighting to increase the influence of alternative candidates. Its Gumbel–Softmax variant perturbs candidate weights while retaining a continuous mixture of token embeddings.

SwiReasoning (Shi et al., 2026) dynamically alternates between explicit token decoding and latent embedding mixing. Transitions are guided by predictive entropy relative to a reference within each reasoning block, using latent reasoning to explore alternatives and explicit reasoning to consolidate progress. A switch-count controller regulates the transition to answer generation to limit overthinking.

SeLaR (Fu & Luo, 2026) selectively activates latent reasoning according to the normalized entropy of the candidate distribution. High-entropy steps use mixed embeddings, while low-entropy steps retain discrete decoding. It further applies entropy-aware contrastive regularization to move the mixed embedding away from the highest-probability candidate's embedding, encouraging alternative candidates to influence reasoning.

A.3 Implementation Details

We conduct all experiments on two NVIDIA A800 GPUs and set the maximum generation length to 32,768 tokens for all methods. Unless otherwise specified, we use temperature 0.6, top-$p = 0.95$, sampling top-$k = 30$, and relative min-$p = 0.001$. Here, sampling top-k controls the truncation of the discrete sampling distribution, while K denotes the number of candidates retained for embedding mixing or Value fusion. For baseline-specific hyperparameters, we adopt the settings recommended in the respective papers.

CoT baselines. CoT uses the default sampling configuration. CoT Greedy uses temperature 0, top-$p = 1$, sampling top-$k = 1$, and min-$p = 0$ throughout generation.

Soft Thinking. Soft Thinking retains $K = 15$ candidates and uses their probability weights without stochastic perturbations. Its entropy-based early-stopping mechanism ends the reasoning phase when entropy remains below 0.01 for 256 consecutive steps. Stochastic Soft Thinking uses the same candidate count and stopping criterion, with Gumbel–Softmax temperature $\tau_{\mathrm{G}} = 1.0$.

SwiReasoning. We retain $K = 15$ candidates and use an explicit-mode window of 512 steps before an explicit-to-latent transition can occur. The initial coefficients for thinking-related signal mixing are $\alpha_0 = 1.0$ and $\beta_0 = 0.7$.

SeLaR. We use sampling top-$k = 20$, min-$p = 0$, and $K = 3$ candidates. Selective activation uses a normalized entropy threshold of 0.5, and the contrastive regularization weight is 1.0.

DIVER (ours). We activate candidate computation and Value fusion when the candidate entropy exceeds $\eta = 0.5$ and use a mixing strength of $\alpha = 1.0$. We use $K = 10$ candidates for Qwen3-4B-Thinking-2507 and Qwen3-8B with fusion in layers 21–24, layers 52–54 for Qwen3-32B, and for DeepSeek-R1-Distill-Qwen-14B with fusion in layers 33–36. We use $K = 3$ with fusion in layers 50–52 for QwQ-32B, layer 21 for Qwen3-1.7B, and layers 49–52 for DeepSeek-R1-Distill-Qwen-32B. Figure 10 presents the Value-path patching results supporting fusion-layer selection for the six additional models. For residual mixing, we select the intervention layers based on the results in Figure 3.

B Further Analysis of DIVER

B.1 Length of Correct Responses

We examine the token length of correct responses relative to CoT across three backbones and five benchmarks (Figure 8). On Qwen3-4B-Thinking-2507, DIVER increases correct-response length by 0.6%–5.9%. Larger increases occur on DeepSeek-R1-Distill-Qwen-14B, reaching 23.1% on AIME 2024, 36.0% on AIME 2025, and 13.9% on MATH500, while lengths decrease on GPQA-Diamond and GSM8K.

The longer responses on mathematical benchmarks may reflect more extensive exploration during reasoning. By retaining alternative candidate information at high-entropy steps, DIVER may encourage the model to examine additional intermediate possibilities or verify tentative conclusions

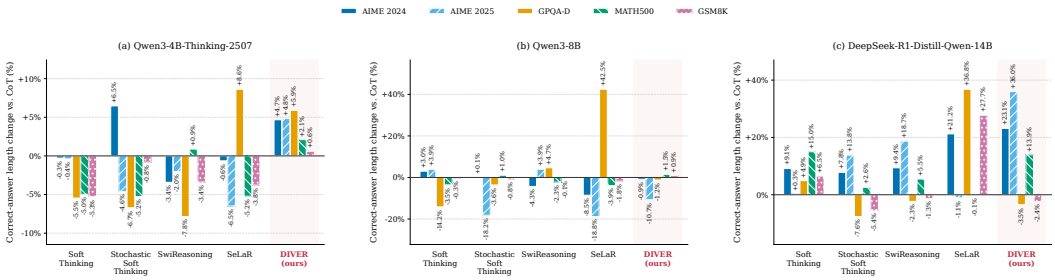


Figure 8: Relative change in the token length of correct responses compared with CoT across five benchmarks on Qwen3-4B-Thinking-2507, Qwen3-8B, and DeepSeek-R1-Distill-Qwen-14B. Positive values indicate longer responses, while negative values indicate shorter responses.

before completing a solution. Such additional derivation and verification can increase the number of generated tokens.

B.2 Collapse Analysis

We compare Soft-token mixing and Value mixing on 100 controlled arithmetic problems using Qwen3-8B. Soft-token mixing combines the embeddings of the top-1 and top-2 candidates with weights of 0.6 and 0.4, respectively. Value mixing instead retains the top-1 input and mixes the candidates' Value vectors at layers 21–24. We compare the overlap between their final-layer top-20 predictions and those of the top-1 and top-2 reference paths. As shown in Table 6, although both methods remain biased toward the Top-1 reference, Value mixing narrows the prediction gap between the Top-1 and Top-2 references.

Table 6: Overlap (%) between the top-20 predictions of each method and those of the Top-1 and Top-2 reference paths on Qwen3-8B.

Method	Top-1 reference	Top-2 reference
Soft-Thinking	74.42	11.29
DIVER	64.49	21.41

C Full Results on Additional Models

Table 7 presents complete results on five reasoning benchmarks for four additional models spanning 1.7B to 32B parameters, extending the AIME evaluation in the main text to GPQA-Diamond, GSM8K, and MATH500.

DIVER achieves the highest average accuracy on all four models, reaching 62.43% on Qwen3-1.7B, 84.29% on Qwen3-32B, 81.94% on QwQ-32B, and 77.47% on DeepSeek-32B. The average improvements range from 1.04 to 2.80 points over CoT and from 0.21 to 5.32 points over Soft Thinking. The gains are most pronounced on AIME, where DIVER outperforms CoT in six of the eight comparisons and matches it in the remaining case.

Results on GPQA-Diamond are more dependent on the backbone. DIVER improves over CoT by 2.53 points on Qwen3-32B and 0.50 points on DeepSeek-32B, but falls below it by 1.52 points on Qwen3-1.7B and 2.02 points on QwQ-32B. These declines may partly reflect the benchmark's dependence on specialized domain knowledge. SeLaR (Fu & Luo, 2026) similarly notes that additional latent exploration can offer limited benefit when performance is constrained by access to relevant knowledge. High entropy in this setting may reflect uncertainty about prerequisite facts rather than a choice among useful reasoning continuations. On QwQ-32B, DIVER's activation rate reaches 30.10% on GPQA-Diamond, compared with the typical range of 10%–20% observed in our experiments. Frequent fusion under such uncertainty may introduce alternatives with insufficient factual support, reducing the reliability of subsequent reasoning.

Table 7: Reasoning pass@1 accuracy (%) on four additional models. DeepSeek-32B denotes DeepSeek-R1-Distill-Qwen-32B. Stochastic Soft denotes Stochastic Soft Thinking.

Method	AIME24	AIME25	GPQA-D	GSM8K	MATH500	Avg.
			Qwen3-1.7B			
CoT	50.00	33.00	**40.91**	**91.13**	91.60	61.33
CoT (greedy)	40.00	36.67	36.36	90.22	90.20	58.69
Soft Thinking	40.00	33.00	36.87	90.83	90.40	58.22
Stochastic Soft	46.67	36.67	38.38	90.52	**92.40**	60.93
SwiReasoning	50.00	36.67	36.36	90.98	91.60	61.12
SeLaR	40.00	**43.33**	37.37	90.60	91.60	60.58
DIVER	46.67	**43.33**	39.39	90.75	92.00	**62.43**
			Qwen3-32B			
CoT	83.33	66.67	65.15	96.13	**97.80**	81.82
CoT (greedy)	83.33	66.67	67.17	**96.29**	97.00	82.09
Soft Thinking	80.00	**76.67**	66.16	95.38	97.40	83.12
Stochastic Soft	**86.67**	66.67	66.67	**96.29**	97.60	82.78
SwiReasoning	**86.67**	66.67	66.67	96.13	**97.80**	82.79
SeLaR	83.33	63.33	**68.18**	95.91	97.00	81.55
DIVER	**86.67**	73.33	67.68	95.98	**97.80**	**84.29**
			QwQ-32B			
CoT	80.00	66.67	64.14	96.29	97.40	80.90
CoT (greedy)	76.67	63.33	63.64	96.29	96.40	79.27
Soft Thinking	83.33	**70.00**	61.62	96.29	97.40	81.73
Stochastic Soft	**86.67**	56.67	65.66	96.29	97.40	80.54
SwiReasoning	76.67	66.67	65.66	**96.44**	96.60	80.41
SeLaR	76.67	63.33	**66.16**	96.29	96.60	79.81
DIVER	**86.67**	66.67	62.12	**96.44**	**97.80**	**81.94**
			DeepSeek-32B			
CoT	66.67	53.33	62.63	95.53	95.20	74.67
CoT (greedy)	63.33	43.33	57.07	94.92	93.20	70.37
Soft Thinking	63.33	50.00	60.10	95.22	93.60	72.45
Stochastic Soft	66.67	50.00	61.62	95.45	95.40	73.83
SwiReasoning	73.33	53.33	62.12	95.38	**95.80**	75.99
SeLaR	70.00	53.33	62.12	**96.06**	94.40	75.18
DIVER	**76.67**	**56.67**	**63.13**	95.68	95.20	**77.47**

Changes on MATH500 and GSM8K are comparatively small. On the three 32B models, CoT already exceeds 95% accuracy on both benchmarks, leaving limited room for further improvement.

D Additional Ablation Studies

D.1 Mixing Strength

We vary the mixing strength α in $V_{\text{new}} = (1-\alpha)V_{\text{main}} + \alpha V_{\text{mix}}$ on Qwen3-8B. Table 8 reports average accuracies of 80.37%, 77.43%, and 81.94% for $\alpha = 0.1$, 0.5, and 1, respectively. This pattern may reflect competing effects of preserving the chosen continuation and incorporating alternatives. Since V_{mix} already includes the main candidate, its effective weight is $1 - \alpha + \alpha\widehat{p}_{\text{main}}$, where $\widehat{p}_{\text{main}}$ denotes its normalized candidate probability. At $\alpha = 0.1$, the main candidate retains at least 90% of the mixture weight, leaving its original contribution largely intact. At $\alpha = 0.5$, alternative information introduces a larger change, but the main candidate still receives more than half of the total weight, even when the original candidate distribution is diffuse. This intermediate setting may

weaken the original continuation while giving alternatives insufficient influence to establish a more useful one.

Table 8: Effect of mixing strength on Qwen3-8B. Scores are accuracy (%).

α	AIME24	AIME25	GPQA-D	GSM8K	MATH500	Avg.
0.1	80.00	70.00	60.10	95.53	96.20	80.37
0.5	70.00	63.33	**61.11**	**95.91**	96.80	77.43
1.0	**83.33**	**73.33**	60.10	95.53	**97.40**	**81.94**

D.2 Entropy Threshold and Candidate Count

Entropy threshold on Qwen3-8B. The entropy threshold η determines which decoding steps activate candidate computation and Value fusion. An intermediate threshold of $\eta = 0.5$ yields the highest average accuracy of 81.94%, compared with 80.31% at $\eta = 0.4$ and 79.54% at $\eta = 0.6$ (Table 9). Lowering the threshold may introduce unnecessary interventions at steps where the model already favors a clear continuation. Raising it may exclude moderately uncertain steps at which alternative candidate information remains useful. The best threshold also varies across benchmarks, with $\eta = 0.6$ yielding the highest GPQA-Diamond accuracy and $\eta = 0.4$ the highest GSM8K accuracy.

Table 9: Effect of the entropy threshold on Qwen3-8B. Activation rate is the percentage of decoding steps using candidate computation and Value fusion. Avg. reports the unweighted mean across five benchmarks for each metric.

η	AIME24	AIME25	GPQA-D	GSM8K	MATH500	Avg.
Accuracy (%)						
0.4	80.00	**73.33**	55.05	**95.98**	97.20	80.31
0.5	**83.33**	**73.33**	60.10	95.53	**97.40**	**81.94**
0.6	73.33	70.00	**62.12**	95.45	96.80	79.54
Activation rate (%)						
0.4	13.27	15.02	22.49	14.18	13.96	15.78
0.5	12.92	12.46	19.50	11.82	11.96	13.73
0.6	9.97	9.85	15.51	9.32	9.19	10.77

Additional threshold ablations. We further examine how the threshold affects AIME 2024 and AIME 2025 accuracy on Qwen3-4B and DeepSeek-14B (Table 10). Averaged across the two benchmarks, $\eta = 0.5$ yields the highest accuracy for both models.

Additional candidate-count ablations. To complement the candidate-count ablation in the main text, we vary k on Qwen3-4B and DeepSeek-14B with $\eta = 0.5$ (Table 11). Both $k = 3$ and $k = 10$ perform well across the two models, whereas increasing k to 15 does not improve accuracy in these settings.

Table 10: Accuracy (%) with different entropy thresholds η, with $K = 10$. Qwen-4B denotes Qwen3-4B-Thinking-2507, DeepSeek-14B denotes DeepSeek-R1-Distill-Qwen-14B

	Qwen3-4B		DeepSeek-14B	
η	AIME 2024	AIME 2025	AIME 2024	AIME 2025
0.4	80.00	73.33	76.67	46.67
0.5	86.67	80.00	73.33	60.00
0.6	86.67	76.67	73.33	50.00

Table 11: Accuracy (%) with different candidate counts k, with $\eta = 0.5$. Qwen-4B denotes Qwen3-4B-Thinking-2507, DeepSeek-14B denotes DeepSeek-R1-Distill-Qwen-14B

	Qwen-4B		DeepSeek-14B	
k	AIME 2024	AIME 2025	AIME 2024	AIME 2025
3	86.67	76.67	70.00	56.67
10	86.67	80.00	73.33	60.00
15	83.33	73.33	66.67	50.00

Table 12: Accuracy (%) of Value-only and joint Query–Value fusion on AIME.

Method	Qwen3-4B		Qwen3-8B	
	AIME 2024	AIME 2025	AIME 2024	AIME 2025
V-Mixing	**86.67**	**80.00**	83.33	**73.33**
QV-Mixing	83.33	**80.00**	**86.67**	70.00

Value versus Query–Value fusion. We compare Value-only fusion (V-Mixing) with joint Query–Value fusion (QV-Mixing) on Qwen3-4B and Qwen3-8B using AIME 2024 and AIME 2025 (Table 12). V-Mixing matches or exceeds QV-Mixing in three of the four settings, indicating that additionally mixing Queries does not consistently improve reasoning accuracy.

D.3 Case Studies

We compare CoT sampling and DIVER on three examples from AIME 2025, GPQA-Diamond, and LiveCodeBench. These cases illustrate how differences in intermediate reasoning affect the final solution across mathematical reasoning, scientific reasoning, and code generation.

Identifying the feasible region. The AIME 2025 example requires finding the area of a bounded region on the plane $x + y + z = 75$ under a chain of inequalities (Figure 9). CoT introduces the additional constraint $x + y < 37$, excluding valid points, and ultimately returns the incorrect answer 42. DIVER identifies the complete bounded region, whose projection satisfies $-1 < x < 25$ and $x < y < (75 - x)/2$. The projected area is 507, yielding a surface area of $507\sqrt{3}$ and the correct answer 510. The crucial distinction is the formulation of the feasible region, which determines the domain of the subsequent area calculation.

Selecting the governing physical law. In the GPQA-Diamond example, the task asks how the Lorentz factor must change to increase the fraction of particles reaching a detector wall from one third to two thirds (Figure 9). CoT assumes that survival probability scales linearly with the Lorentz factor and doubles it from 20 to 40. DIVER instead uses exponential decay with time dilation. Taking the particle speed to be approximately c, the survival probability is $P \simeq \exp[-R/(c\gamma\tau_0)]$, giving $\gamma' = 20\ln(3)/\ln(3/2) \approx 54.19$ and the correct answer 54. Here, the improvement follows from using the appropriate relationship between the physical quantities, rather than extrapolating their apparent proportionality.

Preserving constraints in the implementation. The LiveCodeBench example requires finding a grid path along which health remains strictly positive (Figure 9). CoT omits the starting-cell cost, initializing the accumulated cost to zero without deducting that cell's value. Its program therefore incorrectly returns `True` for `grid = [[1, 0]]` with initial health 1. DIVER deducts the starting-cell cost before traversal and rejects the input when the remaining health is nonpositive. Its solution returns `False` on this counterexample and passes all 43 recorded official tests. This case highlights the importance of carrying a global constraint into the initialization of an algorithm, where an omission can invalidate otherwise plausible traversal logic.

Across these examples, an incorrect intermediate assumption propagates into the final answer or implementation. DIVER's solutions avoid these errors at the corresponding decision points, illus-

trating where preserving alternative reasoning directions may help prevent premature commitment to an incomplete or incorrect formulation.

E LLM USAGE

We employed large language models (LLMs) to assist with manuscript refinement and literature retrieval. Specifically, LLMs were used to correct grammar, enhance clarity, improve phrasing, and identify relevant prior work. GPT-4o (OpenAI, 2024) was also used to assess incoherence and repetition in generated reasoning traces, as described in the paper. We take full responsibility for the scientific contributions, interpretation of results, and final manuscript.

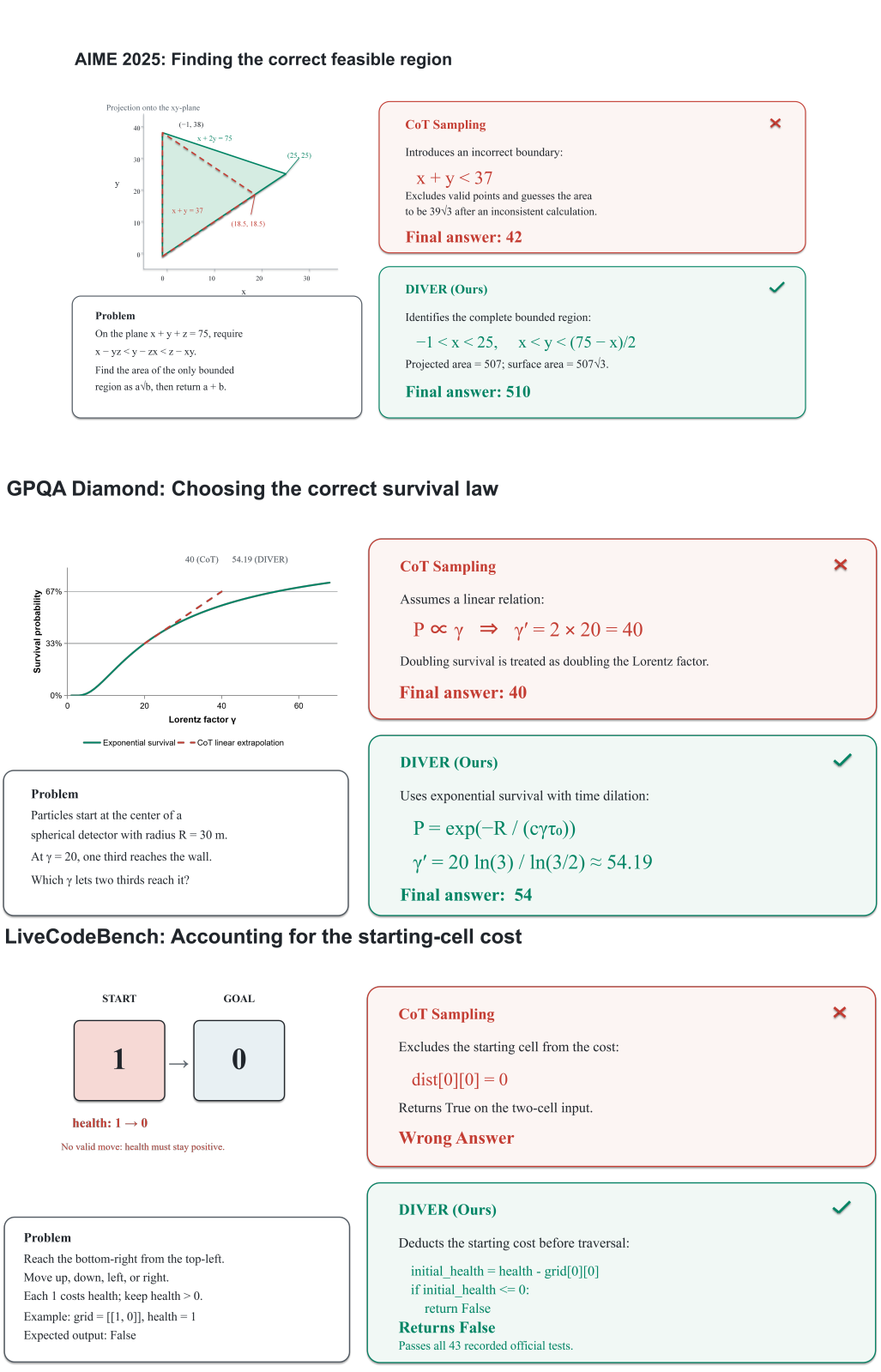


Figure 9: Case studies comparing CoT sampling and DIVER on AIME 2025 (top), GPQA-Diamond (middle), and LiveCodeBench (bottom). DIVER identifies the complete feasible region, applies the exponential survival law, and accounts for the starting-cell cost, respectively.

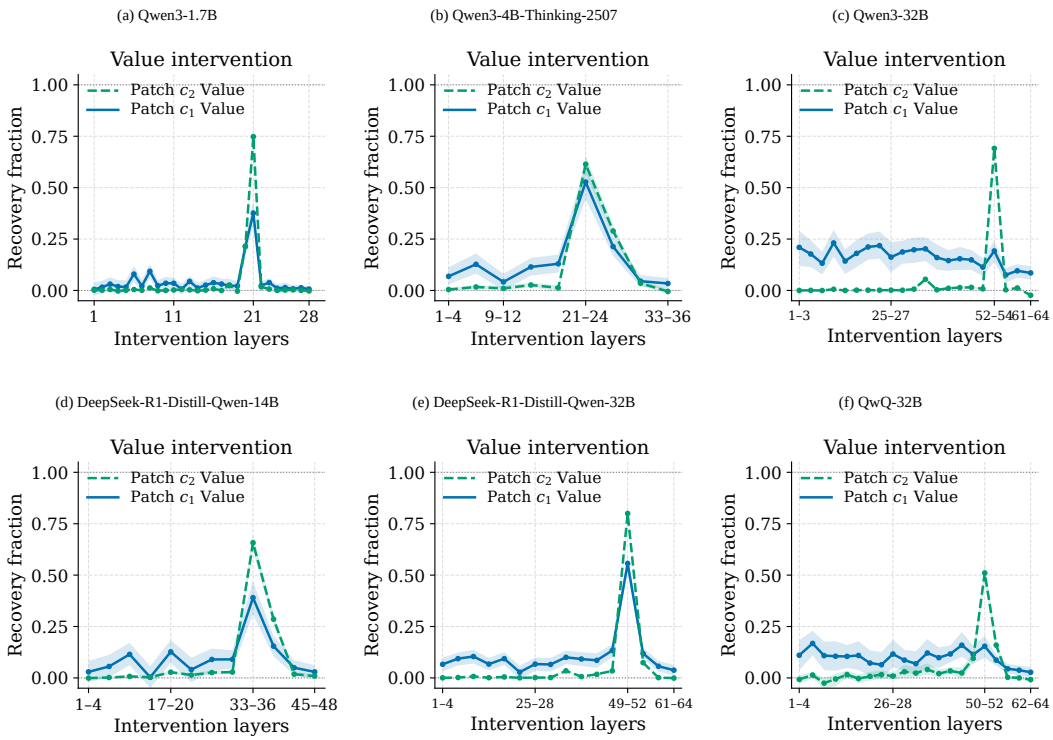


Figure 10: Value-path patching results across six additional models.

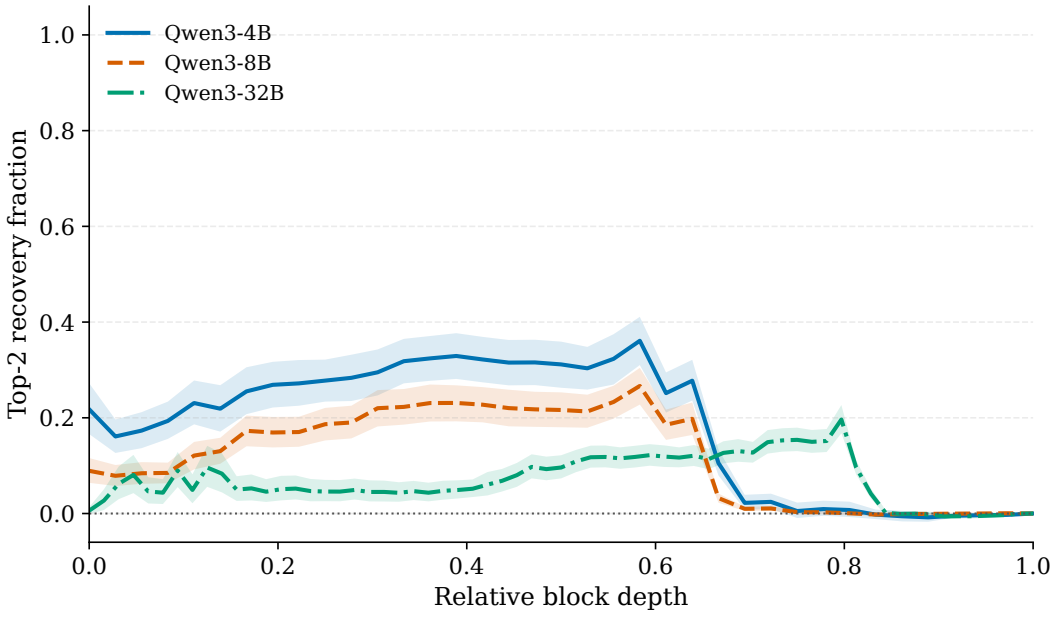


Figure 11: We replace the Top-1 candidate's residual representation with $0.6h_1 + 0.4h_2$ at different layers and measure recovery of Top-2's influence on subsequent predictions. We select the layers with the highest recovery: layer 20 for Qwen3-4B and Qwen3-8B, and layer 50 for Qwen3-32B.