SLICE: STRUCTURAL SELF-CONTRASTIVE GUIDANCE FOR DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion language models (DLMs) are emerging as an alternative to autoregressive language models, generating text by iterative decoding. This paradigm offers flexibility in generation order and inference-time compute, yet effectively exploiting this flexibility to improve generation quality remains challenging. Existing approaches typically improve generation through additional training, external reference models, confidence calibration, self-contrastive guidance or computationally expensive multi-trajectory search and aggregation, leaving the model's internal computational structure underexplored for inference-time guidance. To this end, we propose **S**elective **L**ayer **I**nterventions for **C**ontrastive **E**nhancement **(SLICE)**, the first method to guide DLM decoding through structural self-contrast within a single model. SLICE exploits depth-wise redundancy through prompt-adaptive layer skipping to construct a weakened branch without additional training. It converts the resulting storng–weak logit discrepancy into a bounded structural contrast to reweight the predictive distribution at each decoding step. We characterize the guided distribution as an exponential reweighting and the unique optimizer of a KL-regularized objective, and bound its single-step deviation from the full-model prediction. Across eight benchmarks, SLICE outperforms the low-confidence baseline on both LLaDA-Instruct and LLaDA-1.5 under the same decoding setting, with average gains of 5.91 and 3.67 percentage points, respectively.

1 INTRODUCTION

Diffusion language models offer a generation paradigm in which text is produced through iterative decoding over masked sequences. Representative DLMs such as LLaDA and Dream perform comparably to autoregressive models across language understanding, mathematical reasoning and code generation (Nie et al., 2025; Ye et al., 2025). By allowing flexible decoding orders and distributing computation across multiple decoding steps, DLMs offer greater flexibility in inference-time generation (Arriola & Kuleshov, 2026; Ni et al., 2026; Zhang et al., 2026; Zhong et al., 2026). Yet this flexibility makes generation quality sensitive to intermediate decoding decision: generation order influences error propagation (Zhang et al., 2026; Zhong et al., 2026), early decision shape subsequent decoding trajectories (Kim et al., 2026) and locally confident predictions can lead to globally suboptimal generations (Huang et al., 2026; Fang et al., 2026). Effectively exploiting this flexibility therefore requires reliable signals to guide predictions throughout iterative decoding.

Existing methods improve generation by incorporating informative signals through different mechanisms. Reinforcement learning and preference optimization integrate such signals through additional parameter updates (Zhao et al., 2025; Zhu et al., 2025a; Zhan, 2025; Zhu et al., 2025b). At inference time, guidance methods reshape token predictions using external signals or predictive contrast (Schiff et al., 2024; Chen et al., 2025a; Tejaswi et al., 2026). Other approaches generate and evaluate multiple decoding trajectories through refinement, search, voting or self-verification (Dang et al., 2025; Shen et al., 2026; Bai et al., 2026; Feng et al., 2026b; Song et al., 2026), or exploit signals available within the current decoding process, including calibrated confidence, temporal stability and self-contrast induced by input perturbations (Huang et al., 2026; Kim et al., 2025; Feng et al., 2026a). Despite these diverse sources of guidance, the model's internal computational structure remains underexplored as a source of structural self-contrast for DLM decoding.

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

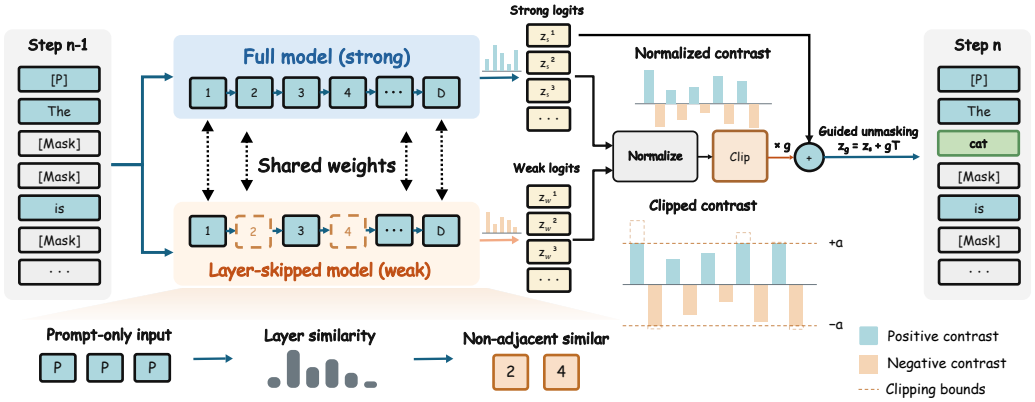

Figure 1: Overview of SLICE. Prompt-adaptive layer skipping constructs a weight-sharing weak branch, with the selected layer set fixed throughout decoding. At each step, both branches process the same sequence state. Their logits are independently standardized and the resulting difference is clipped and rescaled to the full branch's logit scale. This structural contrast provides an additive correction to the full-branch logits, while sequence updates follow the original decoding rule.

Recent studies reveal substantial depth-wise redundancy in DLMs, reflected in high inter-layer representation similarity and layer-collapse phenomena (Goel et al., 2026; Conzelmann et al., 2026). Prior work primarily exploits such redundancy to reduce inference cost through layer skipping (Goel et al., 2026). We instead leverage depth-wise redundancy to construct a structural contrast within the same model. By selectively bypassing layers with similar representations, we obtain a weak branch whose prediction discrepancy with the full branch arises from a change in computational depth. The weak branch therefore serves as a contrastive reference rather than a standalone decoder, allowing the model's internal structure to provide a self-contrastive signal for guiding generation.

Building on this idea, we propose **S**elective **L**ayer **I**nterventions for **C**ontrastive **E**nhancement (**SLICE**), a training-free decoding method that constructs structural self-contrast within a single DLM through prompt-adaptive layer interventions. SLICE selectively skips layers with high adjacent-layer representation similarity to construct a weight-sharing weak branch, while the selected layer set is reused throughout decoding. At each decoding step, the resulting strong–weak logit discrepancy is converted into a bounded structural contrast that corrects the full-branch prediction. We characterize the resulting guided distribution as an exponential reweighting of the full-model distribution and as the unique optimizer of a KL-regularized objective, while establishing explicit bounds on the deviation induced by a single guidance update.

We evaluate SLICE on LLaDA-8B-Instruct and LLaDA-8B-1.5 across eight benchmarks spanning language understanding, mathematical reasoning and code generation. Under the same decoding setting, SLICE improves over the low-confidence baseline on every benchmark for both models, with average gains of 5.91 and 3.67 percentage points, respectively. Among the evaluated decoding methods, SLICE achieves the highest average score on both models and the best result in 13 of the 16 comparisons. Across five decoding configurations, SLICE further improves over the baseline in 77 of 80 configuration comparisons, demonstrating that structural self-contrast remains effective across a broad range of decoding settings. We summarize our contributions as follows:

(i) We introduce SLICE, a training-free decoding method that turns depth-wise redundancy into structural self-contrast through prompt-adaptive selective layer interventions within a single DLM.

(ii) We theoretically characterize SLICE as an exponential reweighting and the unique optimizer of a KL-regularized objective, with explicit bounds on single-step guidance deviations.

(iii) We evaluate SLICE on two DLMs across eight benchmarks and five decoding configurations, showing improvements on all 16 comparisons under the main setting and on 77 of 80 comparisons across decoding configurations, demonstrating effectiveness across different models and tasks.

2 Related Work

Diffusion Language Models and Efficient Decoding. Large-scale DLMs such as LLaDA (Nie et al., 2025) and Dream (Ye et al., 2025) establish diffusion as a competitive alternative to autoregressive language modeling, while subsequent work further improves model capabilities through preference optimization and reinforcement learning (Zhao et al., 2025; Zhu et al., 2025a; Zhan, 2025; Zhu et al., 2025b). The iterative decoding process requires multiple model forward passes during generation, motivating extensive work on improving inference efficiency. Caching-based methods reuse intermediate computation across decoding steps (Liu et al., 2025; Ma et al., 2025; Wu et al., 2025; Jiang et al., 2025; Wu et al., 2026), while parallel and adaptive decoding methods reduce the number of decoding steps by selectively committing multiple reliable tokens at each iteration (Hong et al., 2025; Kim et al., 2025; Xu et al., 2025; Luo et al., 2026; Zoabi et al., 2026).

Inference-Time Enhancement and Contrastive Guidance. Recent works improve DLMs generation at inference time through trajectory search, candidate aggregation and distributional guidance. PG-DLM (Dang et al., 2025), OTS (Shen et al., 2026), Prism (Bai et al., 2026), DiffuReason (Song et al., 2026) and dVoting (Feng et al., 2026b) explore or aggregate multiple decoding trajectories, whereas RFG (Chen et al., 2025a) directly modifies token predictions through predictive contrast. Related contrastive approaches include Contrastive Decoding (Li et al., 2023), DoLa (Chuang et al., 2024), Autoguidance (Karras et al., 2024) and FoCore (Feng et al., 2026a), which obtain contrast from separate models, intermediate-layer predictions, weaker diffusion models or input remasking, respectively. Meanwhile, layer-wise analyses reveal substantial depth redundancy in DLMs (Goel et al., 2026; Conzelmann et al., 2026). SLICE differs by constructing structural self-contrast through selective layer interventions within the same DLM and guiding a single decoding trajectory.

3 Method

3.1 Preliminaries

Masked Diffusion Language Models. Conditioned on a prompt c, a masked diffusion language model (DLM) generates sequences through iterative decoding (Nie et al., 2025; Ye et al., 2025). Let $x_t \in (\mathcal{V} \cup \{\textsc{mask}\})^L$ denote the partially masked sequence at decoding step t, where $\mathcal{V}$ is the token vocabulary. At each position i, the model f_θ parameterizes the conditional predictive distribution as

$$z_{\theta,t,i} = f_\theta(x_t, c)_i, \qquad p_\theta(\cdot \mid x_t, c, i) = \mathrm{softmax}(z_{\theta,t,i}). \tag{1}$$

Here, $p_\theta(\cdot \mid x_t, c, i)$ denotes a token-level predictive distribution at a fixed decoding state, whereas the distribution over complete responses is induced by the full decoding process. We denote the generation length, block length and total number of decoding iterations by L, B and N, respectively.

3.2 Selective Layer Interventions

At a decoding state x_t and masked position i, the full branch uses all layers and produces logits

$$\boldsymbol{z}_{\mathrm{s}} = f_\theta(x_t, c)_i. \tag{2}$$

To construct a structural contrast, SLICE introduces a prompt-dependent set of skipped layers $\mathcal{S}(c)$. Let F_ℓ denote the transformation of layer ℓ. For layers selected by $\mathcal{S}(c)$, the weak branch replaces the original transformation with an identity mapping:

$$\widetilde{F}_\ell(h) = \begin{cases} h, & \ell \in \mathcal{S}(c), \\ F_\ell(h), & \ell \notin \mathcal{S}(c). \end{cases} \tag{3}$$

Applying these selective layer interventions yields a structurally weakened branch that shares the pretrained weights and output head with the full branch. Given the same input (x_t, c), it produces weak-branch logits $\boldsymbol{z}_{\mathrm{w}}$. To determine $\mathcal{S}(c)$, we perform a single forward pass of the full model on prompt c and measure the mean representation similarity between adjacent layers:

$$\rho_\ell(c) = \frac{1}{|\mathcal{P}(c)|} \sum_{j \in \mathcal{P}(c)} \frac{\langle h_{\ell,j}, h_{\ell+1,j} \rangle}{\|h_{\ell,j}\|_2 \|h_{\ell+1,j}\|_2}, \tag{4}$$

where $\mathcal{P}(c)$ denotes the set of prompt-token positions. High adjacent-layer similarity is used as a proxy for representational redundancy. Among candidates satisfying $\rho_\ell(c) \geq \tau$, we greedily select at most k layers in descending order of similarity, excluding any layer adjacent to a previously selected one. The resulting set $\mathcal{S}(c)$ is selected once from the prompt and reused throughout decoding.

3.3 Structural Self-Contrastive Guidance

At a fixed decoding state and masked position, we independently standardize the full and weak-branch logits over the vocabulary to account for branch specific differences in logit scale before constructing the token-wise structural contrast. For $j \in \{\mathrm{s}, \mathrm{w}\}$, define

$$\mu_j = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} z_{j,v}, \qquad \sigma_j = \sqrt{\frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} (z_{j,v} - \mu_j)^2 + \varepsilon}, \qquad u_{j,v} = \frac{z_{j,v} - \mu_j}{\sigma_j}, \tag{5}$$

where $\varepsilon > 0$ is a numerical stabilizer.

To limit extreme discrepancies, we clip the standardized difference coordinate-wise and rescale it to the full branch's logit scale, yielding a bounded structural contrast:

$$T_v = \sigma_{\mathrm{s}} \, \mathrm{clip}\big(u_{\mathrm{s},v} - u_{\mathrm{w},v}, -a, a\big), \tag{6}$$

where $a > 0$ is the clipping threshold. This construction bounds the structural contrast relative to the full branch's logit dispersion, with $|T_v| \leq \sigma_{\mathrm{s}} a$.

SLICE applies this contrast as an additive correction to the original full-branch logits:

$$\boldsymbol{z}_g = \boldsymbol{z}_{\mathrm{s}} + g\boldsymbol{T}, \qquad q_g = \mathrm{softmax}(\boldsymbol{z}_g), \tag{7}$$

where $g \in \mathbb{R}$ controls the strength and direction of guidance. Positive g favors tokens with larger T_v relative to those with smaller values, whereas negative g reverses this preference and $g = 0$ recovers the full-branch distribution. All normalization statistics and contrast signals are recomputed position-wise at each decoding iteration from the current full and weak-branch logits.

Algorithm 1 summarizes the complete SLICE decoding procedure, combining prompt-dependent layer selection with iterative structural self-contrastive guidance. The layer set $\mathcal{S}(c)$ is selected once and reused throughout decoding, with both branches evaluated on the same current sequence state at each iteration. The resulting guided distributions are passed to the decoder to update the sequence under its original decoding rule, including the block-wise schedule and token-commitment criterion.

Algorithm 1 SLICE Decoding

Require: $f_\theta, c, x_0, \{\mathcal{D}_n\}_{n=1}^N; k, \tau, g, a, \varepsilon$
Ensure: decoded response x_N

Prompt-time layer selection

$\rho_\ell(c) \leftarrow \dfrac{1}{|\mathcal{P}(c)|} \displaystyle\sum_{i \in \mathcal{P}(c)} \frac{\langle \boldsymbol{h}_{\ell,i}, \boldsymbol{h}_{\ell+1,i} \rangle}{\|\boldsymbol{h}_{\ell,i}\|_2 \|\boldsymbol{h}_{\ell+1,i}\|_2}, \quad \ell = 1{:}D-1$

$\pi \leftarrow \mathrm{argsort}^{\downarrow}_{\ell:\rho_\ell(c) \geq \tau} \rho_\ell(c)$

Scan $\ell \in \pi$ and add it to $\mathcal{S}(c)$ iff $|\mathcal{S}(c)| < k$ and $|\ell - r| > 1$ for all $r \in \mathcal{S}(c)$

Denoising-time structural self-contrast

for $n = 1, \ldots, N$ **do**
  $\boldsymbol{z}_{\mathrm{s}} \leftarrow f_\theta(x_{n-1}, c)$; $\boldsymbol{z}_{\mathrm{w}} \leftarrow f_{\theta,-\mathcal{S}(c)}(x_{n-1}, c)$
  for $i \in \mathcal{M}_{n-1} := \{r : x_{n-1,r} = \mathrm{MASK}\}$ **do**
    for $j \in \{\mathrm{s}, \mathrm{w}\}$ **do**
      $\mu_{j,i} \leftarrow \dfrac{1}{|\mathcal{V}|} \displaystyle\sum_v z_{j,i,v}$; $\sigma_{j,i} \leftarrow \sqrt{\dfrac{1}{|\mathcal{V}|} \displaystyle\sum_v (z_{j,i,v} - \mu_{j,i})^2 + \varepsilon}$
      $\nu_{j,i,v} \leftarrow (z_{j,i,v} - \mu_{j,i})/\sigma_{j,i}, \quad v \in \mathcal{V}$
    end for
    $T_{i,v} \leftarrow \sigma_{\mathrm{s},i} \, \mathrm{clip}(\nu_{\mathrm{s},i,v} - \nu_{\mathrm{w},i,v}, -a, a)$; $q_{g,i} \leftarrow \mathrm{softmax}(\boldsymbol{z}_{\mathrm{s},i} + g\boldsymbol{T}_i)$
  end for
  $x_n \leftarrow \mathcal{D}_n\big(x_{n-1}, \{q_{g,i}\}_{i \in \mathcal{M}_{n-1}}\big)$
end for
return x_N

3.4 THEORETICAL CHARACTERIZATION

In this section, we characterize the SLICE-guided distribution and its relation to the original full-model prediction and further derive bounds on the deviation induced by a single guided update.

Theorem 1 (Distributional characterization)**.** *At a fixed decoding state and token position, let*

$$q_g = \mathrm{softmax}(\boldsymbol{z}_\mathrm{s} + g\boldsymbol{T}), \qquad p_\mathrm{s} = \mathrm{softmax}(\boldsymbol{z}_\mathrm{s}).$$

Then

$$q_g(v) = \frac{p_\mathrm{s}(v)e^{gT_v}}{\sum_{u\in\mathcal{V}} p_\mathrm{s}(u)e^{gT_u}}, \tag{8}$$

and q_g is the unique optimizer of

$$\max_{q\in\Delta(\mathcal{V})} \left\{ g\,\mathbb{E}_q[T] - D_\mathrm{KL}(q\|p_\mathrm{s}) \right\}. \tag{9}$$

The proof is given in Appendix B.1. Theorem 1 shows that SLICE balances preference for the signed contrast gT against KL divergence from p_s. Equivalently, for any $u, v \in \mathcal{V}$,

$$\log \frac{q_g(v)}{q_g(u)} = \log \frac{p_\mathrm{s}(v)}{p_\mathrm{s}(u)} + g(T_v - T_u), \tag{10}$$

so the contrast difference directly determines the shift in pairwise token preferences. For positive g, a larger contrast increases a token's odds relative to its competitor and negative g reverses this effect. The full-model distribution remains the reference, rather than being replaced by the weak branch.

Theorem 2 (Local boundedness of SLICE guidance)**.** *At a fixed decoding state and token position, let $M = |g|\sigma_\mathrm{s}a$. Then, for any tokens $u, v \in \mathcal{V}$,*

$$|(z_{g,v} - z_{g,u}) - (z_{\mathrm{s},v} - z_{\mathrm{s},u})| \le 2M, \tag{11}$$

$$\left|\log \frac{q_g(v)}{p_\mathrm{s}(v)}\right| \le 2M. \tag{12}$$

The proof is given in Appendix B.2. The first bound controls changes in token ranking margins, while the second controls changes in individual token probabilities on a logarithmic scale. Both adapt to the full branch's logit dispersion through σ_s. In particular, the full-model top-1 token is preserved whenever its logit margin over every competing token exceeds $2M$.

Conditional improvement. Boundedness alone does not imply improvement. At a fixed decoding state and position, define a target token's centered contrast as $\alpha_y = T_y - \mathbb{E}_{p_\mathrm{s}} T$. The centering is essential: a positive raw contrast T_y alone is insufficient because guidance also reweights competing tokens. Useful alignment must be assessed relative to the full-model expectation, rather than inferred from weak-branch accuracy alone. If $\alpha_y > 0$, then $q_g(y) > p_\mathrm{s}(y)$ for sufficiently small $g > 0$. Appendix B.5 gives an explicit sufficient range and connects alignment to target-specific degradation under the scale-aligned structural intervention, accounting for clipping. This is a token-probability guarantee, not a guarantee of sequence-level accuracy.

3.5 SLICE DECODING PROCEDURE

SLICE combines prompt-level layer selection with iterative self-contrastive guidance in a unified decoding procedure. The layer set $\mathcal{S}(c)$ is selected once from the prompt and reused throughout generation. At each decoding step, the full and weak branches are evaluated on the same sequence state and their structural contrast is used to construct the guided distribution q_g, which is then passed to the original decoder for the next state update without altering its underlying decoding rule.

For N decoding iterations, SLICE requires one prompt-time probing pass, together with N full-branch and N weak-branch evaluations. Each weak-branch evaluation executes $D - |\mathcal{S}(c)|$ Transformer layers, while layer selection is performed only once before iterative decoding. Thus, matching (B, L, N) fixes the decoding schedule, while the total inference cost additionally depends on the structural guidance evaluations and the realized number of skipped layers.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models and Benchmarks. We evaluate SLICE on LLaDA-8B-Instruct (Nie et al., 2025) and LLaDA-8B-1.5 (Zhu et al., 2025a), two representative masked diffusion language models. Our evaluation covers eight benchmarks spanning language understanding, reasoning and code generation: ARC-Challenge (ARC-C) and ARC-Easy (ARC-E) (Clark et al., 2018), GSM8K (Cobbe et al., 2021), MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024), HumanEval (Chen et al., 2021), Countdown (Pan et al., 2025), Sudoku (Black-Phoenix, 2019) and GPQA (Rein et al., 2024).

Baselines. We compare SLICE with low-confidence and random decoding (Nie et al., 2025), Best-of-N (BoN), entropy-based Best-of-N (E-BoN) (Chen et al., 2025b), UnCode (Huang et al., 2026), Order-Token Search (OTS) (Shen et al., 2026), Prism (Bai et al., 2026), FoCore (Feng et al., 2026a) and dVoting (Feng et al., 2026b). All baseline results are reproduced by us, using official code where available and independent implementations for OTS and FoCore.

Evaluation Protocol. Unless otherwise specified, single-trajectory methods use $B = 32$, $L = 256$, greedy decoding and one completion per example. SLICE modifies only the predictive distribution while retaining the token-commitment rule. Multi-trajectory methods use method-specific sampling protocols: BoN and E-BoN use four candidates, while dVoting uses its released adaptive remasking configuration. Layer-selection and guidance configurations vary by model, task and decoding setting, while shared implementation settings remain fixed (Appendix A). We additionally evaluate different block and lengths in Section 4.2. Because inference-time methods use different numbers and types of model evaluations, the main comparison is not compute-matched and the wall-clock efficiency is reported in Table 2.

4.2 MAIN RESULTS

Overall Performance. As shown in Table 1, SLICE improves over the low-confidence baseline on all eight benchmarks for both models at $B = 32$ and $L = 256$. The average score increases from 45.33 to 51.24 on LLaDA-8B-Instruct and from 48.07 to 51.74 on LLaDA-8B-1.5, corresponding to gains of 5.91 and 3.67 points. SLICE achieves the largest gains on ARC-C and ARC-E with LLaDA-8B-Instruct, improving over the low-confidence baseline by 8.53 and 9.64 points. With LLaDA-8B-1.5, SLICE achieves the largest gains on Sudoku and Countdown, with improvements of 8.40 and 6.64 points. Among the evaluated decoding methods, SLICE achieves the highest average score on both models and the best results in 13 of the 16 settings.

Computational Cost. SLICE requires a single prompt-time probe and one additional weak-branch evaluation per decoding step. The weak branch bypasses selected layers and the layer set is reused throughout generation. Table 2 reports generation latency on the fixded examples using H800 GPUs, averaged over four runs and excluding model loading and scoring. SLICE takes 5.508 and 5.891 seconds per completion on LLaDA-8B-Instruct and LLaDA-8B-1.5, corresponding to relative latencies of $1.68\times$ and $1.81\times$ against the low-confidence. In comparison, BoN, E-BoN and OTS require $5.03\times$–$5.33\times$ the baseline latency, while dVoting and Prism require $2.79\times$–$3.10\times$. FoCore has lower relative latencies of $1.61\times$ and $1.63\times$, but achieves lower average benchmark scores. Together, these results show that SLICE attains higher average performance than the evaluated search, multi-candidate selection and voting methods at lower measured latency.

Extended Evaluation across Decoding Settings. Beyond the main setting, we evaluate SLICE across five combinations of block length B and generation length L, with $N = L/2$, on both models and all eight benchmarks (Table 3). SLICE improves over the corresponding unguided baseline in 77 of 80 model–benchmark combinations. SLICE achieves particularly large gains on ARC with larger blocks: at $B = 64$ and $L = 512$, it improves ARC-C and ARC-E by 24.91 and 25.93 points on LLaDA-8B-Instruct. The effect of generation length is task-dependent, and increasing L does not consistently improve performance. Overall, these results demonstrate improvements across a broad range of decoding configurations, with the magnitude of the gains varying across tasks and settings.

Table 1: Performance comparison of decoding methods across eight benchmarks on LLaDA-8B-Instruct and LLaDA-8B-1.5 at the same setting $B = 32$, $L = 256$. Avg. denotes the mean score across eight benchmarks. Bold and underlined values indicate the best and second-best results.

Method	ARC-C	ARC-E	GSM8K	MATH-500	HumanEval	Countdown	Sudoku	GPQA	Avg.
LLaDA-8B-Instruct									
Low-confidence	75.51	83.67	79.15	33.00	37.80	20.31	10.40	22.77	45.33
Random	78.24	88.09	65.73	22.00	15.24	20.31	10.40	22.77	40.35
E-BoN ($K = 4$)	76.62	85.65	77.10	33.80	32.32	27.73	5.80	21.65	45.08
BoN ($K = 4$)	77.22	86.66	79.83	32.20	31.10	24.22	5.60	19.87	44.59
FoCore	75.77	83.38	79.68	31.80	37.20	25.39	8.20	21.65	45.38
dVoting	76.19	82.83	76.95	**38.60**	31.71	23.44	<u>11.20</u>	22.54	45.43
OTS	79.01	85.86	<u>80.52</u>	34.00	<u>40.24</u>	<u>27.73</u>	9.60	23.88	<u>47.61</u>
Prism	<u>81.91</u>	<u>93.10</u>	75.97	31.40	25.00	19.14	3.40	<u>24.78</u>	44.34
UnCode	80.38	90.95	77.79	30.20	32.93	21.48	3.60	18.30	44.45
SLICE (Ours)	**84.04**	**93.31**	**81.35**	<u>37.40</u>	**40.24**	**28.52**	**15.60**	**29.46**	**51.24**
LLaDA-8B-1.5									
Low-confidence	83.70	91.67	80.44	35.20	38.41	22.66	9.00	23.44	48.07
Random	80.38	91.58	69.90	25.60	20.73	22.66	9.00	23.21	42.88
E-BoN ($K = 4$)	83.28	92.26	79.83	34.40	35.37	<u>28.91</u>	7.40	25.45	48.36
BoN ($K = 4$)	84.04	92.51	81.65	35.40	35.98	25.78	9.00	25.00	48.67
FoCore	83.53	91.25	79.83	34.40	36.59	23.44	8.60	23.66	47.66
dVoting	78.07	85.06	79.00	**39.80**	37.80	26.17	<u>13.20</u>	<u>25.89</u>	48.12
OTS	83.96	92.38	**83.09**	34.60	<u>38.41</u>	25.78	11.60	24.78	<u>49.33</u>
Prism	<u>84.13</u>	92.05	78.47	31.20	22.56	20.70	5.40	25.22	44.97
UnCode	82.76	<u>92.72</u>	80.14	34.00	33.54	21.48	4.60	22.99	46.53
SLICE (Ours)	**84.56**	**93.73**	<u>82.56</u>	<u>36.00</u>	**42.68**	**29.30**	**17.40**	**27.68**	**51.74**

Table 2: Generation latency comparison on fixed examples using H800 GPUs with $B = 32$, $L = 256$. Latency is averaged over four runs and reported in seconds per selected completion, including all candidate-generation costs. Relative latency is normalized to the low-confidence decoder.

Method	LLaDA-8B-Instruct		LLaDA-8B-1.5	
	Seconds	Relative Latency	Seconds	Relative Latency
Low-confidence	3.287	1.00×	3.263	1.00×
Random	3.267	0.99×	3.270	1.00×
E-BoN ($K = 4$)	16.756	5.10×	16.786	5.14×
BoN ($K = 4$)	16.523	5.03×	16.548	5.07×
FoCore	5.298	1.61×	5.326	1.63×
dVoting	10.184	3.10×	9.099	2.79×
OTS	17.407	5.30×	17.400	5.33×
Prism	10.204	3.10×	9.742	2.99×
UnCode	3.274	1.00×	3.260	1.00×
SLICE (Ours)	5.508	1.68×	5.891	1.81×

4.3 Ablation Studies and Discussion

We investigate whether SLICE's gains arise from layer skipping alone or from using the weakened branch as a contrastive reference. We then examine how layer selection and skip budget affect weak-branch proximity to the full model and whether SLICE remains effective when combined with inference acceleration. Experimental settings are provided in the corresponding tables and figures.

Layer-Skipping Ablations. Table 4 compares Base, weak-only decoding, random-skip guidance and SLICE to distinguish the effect of layer skipping alone from its use in contrastive guidance. Weak-only decoding underperforms Base in all 12 model–benchmark–setting combinations, suggesting that skipping alone does not explain SLICE's gains. Random-skip guidance improves ARC-C but yields inconsistent gains on GSM8K and HumanEval, highlighting the importance of layer selection. SLICE performs best in all 12 comparisons, with average gains of 4.06–11.04 percentage points over Base and 4.13–8.09 over random-skip guidance.

Table 3: Performance of SLICE across different block lengths B and generation lengths L. Each block reports the SLICE score, with changes from the matched $g = 0$ baseline shown in blue. Bold indicates the highest score across decoding configurations for each benchmark within each model.

Benchmark	$B = 32$			$B = 64$		Avg.
	$L = 128$	$L = 256$	$L = 512$	$L = 256$	$L = 512$	
LLaDA-8B-Instruct						
ARC-C	84.98 (+4.18)	84.04 (+8.53)	85.15 (+10.32)	85.41 (+20.31)	**85.92** (+24.91)	85.10 (+13.65)
ARC-E	93.94 (+5.64)	93.31 (+9.64)	93.06 (+10.52)	93.86 (+24.58)	**93.98** (+25.93)	93.63 (+15.26)
GSM8K	73.39 (+2.65)	81.35 (+2.20)	79.30 (-1.21)	79.53 (+0.76)	**81.88** (+1.52)	79.09 (+1.18)
HumanEval	39.02 (+4.26)	40.24 (+2.44)	41.46 (+4.26)	41.46 (+3.66)	**43.29** (+6.70)	41.09 (+4.26)
MATH-500	32.40 (+5.00)	**37.40** (+4.40)	36.60 (+1.20)	36.00 (+3.40)	36.40 (+0.60)	35.76 (+2.92)
Countdown	**37.50** (+13.28)	28.52 (+8.20)	29.30 (+10.94)	30.47 (+12.50)	34.77 (+12.11)	32.11 (+11.41)
Sudoku	18.40 (+13.00)	15.60 (+5.20)	8.20 (+4.40)	**25.40** (+16.20)	8.60 (+5.00)	15.24 (+8.76)
GPQA	28.13 (+0.00)	**29.46** (+6.70)	25.22 (+2.90)	**29.46** (+7.37)	28.79 (+7.14)	28.21 (+4.82)
LLaDA-8B-1.5						
ARC-C	86.60 (+3.41)	84.56 (+0.85)	85.07 (+1.02)	**87.20** (+9.47)	86.35 (+10.15)	85.96 (+4.98)
ARC-E	**94.57** (+1.47)	93.73 (+2.06)	94.28 (+3.03)	92.97 (+8.67)	93.64 (+8.38)	93.84 (+4.72)
GSM8K	73.31 (+1.21)	**82.56** (+2.12)	82.26 (+1.44)	81.58 (+1.36)	81.50 (+1.44)	80.24 (+1.51)
HumanEval	41.46 (+4.26)	42.68 (+4.27)	41.46 (-0.61)	**43.29** (+3.05)	42.07 (+0.61)	42.19 (+2.32)
MATH-500	29.80 (+0.20)	36.00 (+0.80)	38.20 (+0.60)	36.60 (+1.60)	**38.60** (+0.40)	35.84 (+0.72)
Countdown	36.33 (+5.86)	29.30 (+6.64)	34.77 (+6.64)	36.33 (+12.11)	**36.72** (+10.94)	34.69 (+8.44)
Sudoku	22.20 (+14.00)	17.40 (+8.40)	9.40 (+5.20)	**31.80** (+20.60)	12.40 (+6.40)	18.64 (+10.92)
GPQA	28.79 (+5.80)	27.68 (+4.24)	28.35 (+4.47)	**29.24** (+2.46)	**29.24** (+3.57)	28.66 (+4.11)

Table 4: Component ablation of SLICE on three benchmarks with $B = 64$, $L \in \{256, 512\}$ and $N = L/2$. We compare the base decoding, weak-only decoding, random-skip guidance and the SLICE. Weak-only decoding uses the weak branch's predictions directly for decoding without structural self-contrastive guidance. Random-skip results are averaged over three runs.

Method	$L = 256$				$L = 512$			
	ARC-C	GSM8K	HumanEval	Avg.	ARC-C	GSM8K	HumanEval	Avg.
LLaDA-8B-Instruct								
Base	65.10	78.77	37.80	60.56	61.01	80.36	36.59	59.32
Weak only	20.05	77.63	0.00	32.56	15.61	7.20	1.22	8.01
Random skip	70.85	77.58	36.59	61.67	71.13	79.08	36.59	62.27
SLICE (Ours)	**85.41**	**79.53**	**41.46**	**68.80**	**85.92**	**81.88**	**43.29**	**70.36**
LLaDA-8B-1.5								
Base	77.73	80.21	40.24	66.06	76.19	80.06	41.46	65.91
Weak only	28.58	63.76	0.00	30.78	22.01	75.82	34.76	44.20
Random skip	81.46	80.52	37.60	66.52	79.41	80.31	37.80	65.84
SLICE (Ours)	**87.20**	**81.58**	**43.29**	**70.69**	**86.35**	**81.50**	**42.07**	**69.97**

Structural Proximity. Figure 2 compares similarity-selected and randomly selected weak branches with the same number of skip layers. Similarity-based selection yields lower KL divergence and logit distance and higher top-1 agreement with the full model. Deviations increase with the number of skipped layers but similarity-selected branches consistently remain closer to the full model than random branches. Table 8 reports the complete numerical results for both models.

Compatibility with Inference Acceleration. Table 5 evaluates SLICE combined with KLASS and Fast-dLLM on LLaDA-8B-Instruct across six benchmarks. Adding SLICE to KLASS improves performance on five benchmarks and maintains the HumanEval score. Similarly, Fast-dLLM benefits from SLICE on all six benchmarks. Both combinations reduce the measured latency relative to SLICE alone. Relative latency decreases from $1.68\times$ for SLICE to $1.20\times$ with KLASS and $0.99\times$ with Fast-dLLM, normalized to the base decoder. Notably, Fast-dLLM combined with SLICE outperforms the base on all six benchmarks at approximately the same measured latency.

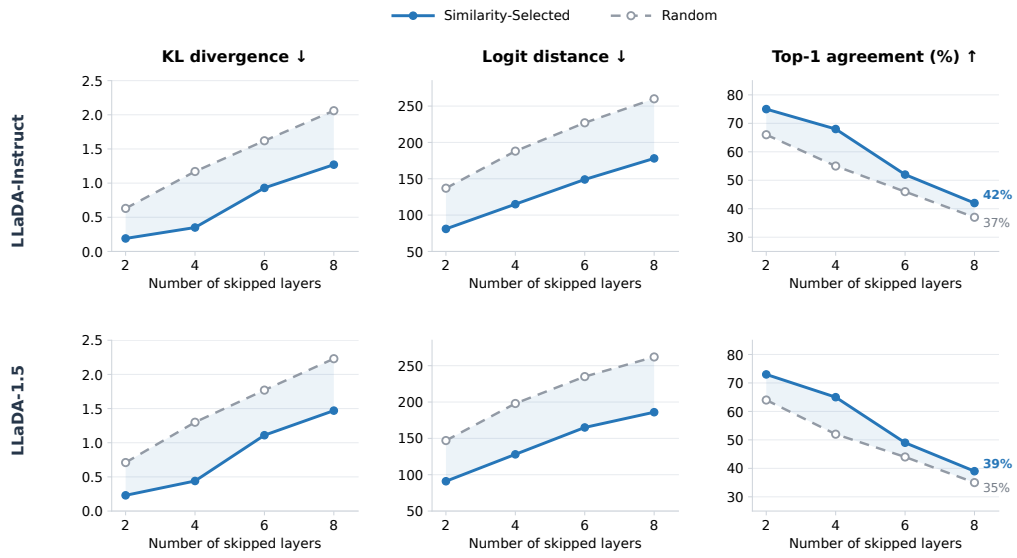

Figure 2: Structural proximity of similarity-selected and random weak branches to the full model at matched skip layers. Similarity-selected branches exhibit lower divergence and logit distance, and higher top-1 agreement across all evaluated skip layers.

Table 5: Compatibility of SLICE with inference acceleration methods on LLaDA-Instruct across six benchmarks with $B = 32$, $L = 256$ and $N = 128$. Each method is evaluated with and without SLICE. Avg. is the mean score across six benchmarks. Latency is measured on fixed examples using H800 GPUs, averaged over four runs and normalized to Base.

Method	MATH-500	HumanEval	GSM8K	Countdown	Sudoku	GPQA	Avg.	Latency ↓
Base	33.00	37.80	79.15	20.31	10.40	22.77	33.91	1.00×
SLICE	37.40	40.24	81.35	28.52	15.60	29.46	38.76	1.68×
KLASS	34.80	37.80	78.32	21.48	9.60	24.33	34.39	0.71×
KLASS + SLICE	35.20	37.80	80.67	27.73	13.60	24.55	36.59	1.20×
Fast-dLLM	34.00	40.12	81.20	31.25	11.00	22.32	36.65	0.70×
Fast-dLLM + SLICE	35.20	40.85	81.58	33.98	14.60	22.99	38.20	0.99×

Limitations and Future Directions. Our evaluation covers two 8B LLaDA models, while generalization to other architectures and model scales remains unverified. SLICE incurs an additional weak-branch evaluation per step, with measured latencies of 1.68× and 1.81× relative to Base, and fixes the prompt-selected layer set throughout decoding. Future work could explore intermittent guidance and branch-aware caching to reduce overhead, alongside low-cost layer reselection to adapt to changing layer redundancy.

5 Conclusion

We introduced SLICE, a training-free method that turns prompt-adaptive selective layer skipping into structural self-contrast for diffusion language model decoding. By constructing a weight-sharing weak branch, SLICE derives a bounded and position-wise correction to the full-model predictions without external reward models or a separate weak model. Experiments on two LLaDA models across eight benchmarks demonstrate improvements over standard decoding across a range of block lengths and generation lengths. Layer-skipping ablations support using the weak branch as a contrastive reference rather than a standalone decoder, while experiments with KLASS and Fast-dLLM demonstrate compatibility with existing inference acceleration methods. Together, these findings highlight internal model structure as a useful source of inference-time guidance, extending the role of layer skipping beyond computational savings to improving generation quality.

AI USE STATEMENT

We used ChatGPT to assist with manuscript drafting and language polishing, literature retrieval and discovery, and the development and checking of mathematical derivations and proofs. All references, mathematical arguments and final scientific claims were independently reviewed and verified by the authors, who retain full responsibility for the content of the paper.

REPRODUCIBILITY STATEMENT

Section 3 and Algorithm 1 provide the complete SLICE decoding procedure, including prompt-dependent layer selection, weak-branch construction and structural self-contrastive guidance. Section 4.1 specifies the evaluated models, benchmarks, baseline methods and evaluation protocol. Appendix A reports the search and decoding settings, fixed hyperparameters, baseline implementation details, structural-proximity protocol and timing methodology. The main benchmark results, decoding-configuration analysis, component ablations and inference-acceleration experiments are reported in Section 4, with full benchmark sizes provided in Appendix A. Proofs of the main theoretical results and additional analyses of standardization, clipping, distributional deviation, and layer similarity are provided in Appendix B.

REFERENCES

Marianne Arriola and Volodymyr Kuleshov. Set Diffusion: Interpolating Token Orderings Between Autoregression and Diffusion for Fast and Flexible Decoding. *arXiv preprint arXiv:2607.01775*, 2026. URL https://arxiv.org/abs/2607.01775.

Jinbin Bai, Yixuan Li, Yuchen Zhu, Yi Xin, Qingyu Shi, Aosong Feng, Xiaohong Liu, Molei Tao, Jianru Xue, Xiangtai Li, and Ming-Hsuan Yang. Prism: Efficient Test-Time Scaling via Hierarchical Search and Self-Verification for Discrete Diffusion Language Models. *arXiv preprint arXiv:2602.01842*, 2026. URL https://arxiv.org/abs/2602.01842.

Black-Phoenix. 4x4-sudoku-dataset: 1 million 4x4 sudoku boards. https://github.com/Black-Phoenix/4x4-Sudoku-Dataset, 2019.

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

Tianlang Chen, Minkai Xu, Jure Leskovec, and Stefano Ermon. RFG: Test-Time Scaling for Diffusion Large Language Model Reasoning with Reward-Free Guidance. *arXiv preprint arXiv:2509.25604*, 2025a. URL https://arxiv.org/abs/2509.25604.

Ziyu Chen, Xinbei Jiang, Peng Sun, and Tao Lin. Optimizing Decoding Paths in Masked Diffusion Models by Quantifying Uncertainty. *arXiv preprint arXiv:2512.21336*, 2025b. URL https://arxiv.org/abs/2512.21336.

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. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/edc36117f795ca52a0cbf6a7b3882859-Abstract-Conference.html.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

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. URL https://arxiv.org/abs/2110.14168.

Alexander Conzelmann, Albert Catalan-Tatjer, and Shiwei Liu. Layer Collapse in Diffusion Language Models. *arXiv preprint arXiv:2605.06366*, 2026. URL https://arxiv.org/abs/2605.06366.

Meihua Dang, Jiaqi Han, Minkai Xu, Kai Xu, Akash Srivastava, and Stefano Ermon. Inference-Time Scaling of Diffusion Language Models via Trajectory Refinement. *arXiv preprint arXiv:2507.08390*, 2025. URL https://arxiv.org/abs/2507.08390.

Liancheng Fang, Aiwei Liu, Henry Peng Zou, Yankai Chen, Enze Ma, Leyi Pan, Chunyu Miao, Wei-Chieh Huang, Xue Liu, and Philip S. Yu. Locally Confident, Globally Stuck: The Quality-Exploration Dilemma in Diffusion Language Models. *arXiv preprint arXiv:2604.00375*, 2026. URL https://arxiv.org/abs/2604.00375.

Jinyuan Feng, Xin Yu, Yiqun Chen, Xiaochi Wei, Yan Gao, Yi Wu, Yao Hu, and Zhiqiang Pu. Focus on the Core: Empowering Diffusion Large Language Models by Self-Contrast. *arXiv preprint arXiv:2605.01373*, 2026a. URL https://arxiv.org/abs/2605.01373.

Sicheng Feng, Zigeng Chen, Xinyin Ma, Gongfan Fang, and Xinchao Wang. dVoting: Fast Voting for dLLMs. *arXiv preprint arXiv:2602.12153*, 2026b. URL https://arxiv.org/abs/2602.12153.

Raghavv Goel, Risheek Garrepalli, Sudhanshu Agrawal, Chris Lott, Mingu Lee, and Fatih Porikli. A Comparative analysis of Layer-wise Representational Capacity in AR and Diffusion LLMs. *arXiv preprint arXiv:2603.07475*, 2026. URL https://arxiv.org/abs/2603.07475.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. URL https://arxiv.org/abs/2103.03874.

Feng Hong, Geng Yu, Yushi Ye, Haicheng Huang, Huangjie Zheng, Ya Zhang, Yanfeng Wang, and Jiangchao Yao. Wide-In, Narrow-Out: Revokable Decoding for Efficient and Effective DLLMs. *arXiv preprint arXiv:2507.18578*, 2025. URL https://arxiv.org/abs/2507.18578.

Pengcheng Huang, Tianming Liu, Zhenghao Liu, Yukun Yan, Shuo Wang, Tong Xiao, Zulong Chen, and Maosong Sun. Empirical analysis of decoding biases in masked diffusion models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6853–6876, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.311. URL https://aclanthology.org/2026.acl-long.311/.

Yuchu Jiang, Yue Cai, Xiangzhong Luo, Jiale Fu, Jiarui Wang, Chonghan Liu, and Xu Yang. d^2Cache: Accelerating Diffusion-Based LLMs via Dual Adaptive Caching. *arXiv preprint arXiv:2509.23094*, 2025. URL https://arxiv.org/abs/2509.23094.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a Diffusion Model with a Bad Version of Itself. *arXiv preprint arXiv:2406.02507*, 2024. URL https://arxiv.org/abs/2406.02507.

Jiyeon Kim, Sungik Choi, Yongrae Jo, Moontae Lee, and Minjoon Seo. Early Decisions Matter: Proximity Bias and Initial Trajectory Shaping in Non-Autoregressive Diffusion Language Models. *arXiv preprint arXiv:2604.10567*, 2026. URL https://arxiv.org/abs/2604.10567.

Seo Hyun Kim, Sunwoo Hong, Hojung Jung, Youngrok Park, and Se-Young Yun. KLASS: KL-Guided Fast Inference in Masked Diffusion Models. *arXiv preprint arXiv:2511.05664*, 2025. URL https://arxiv.org/abs/2511.05664.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 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)*, pp. 12286–12312, 2023. doi: 10.18653/v1/2023.acl-long.687. URL https://aclanthology.org/2023.acl-long.687/.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Zhiyuan Liu, Yicun Yang, Yaojie Zhang, Junjie Chen, Chang Zou, Qingyan Wei, Shaobo Wang, Yichen Zhu, and Linfeng Zhang. dLLM-Cache: Accelerating Diffusion Large Language Models with Adaptive Caching. *arXiv preprint arXiv:2506.06295*, 2025. URL https://arxiv.org/abs/2506.06295.

Lizhuo Luo, Zhuoran Shi, Jiajun Luo, Zhi Wang, Shen Ren, Wenya Wang, and Tianwei Zhang. DAWN: Dependency-Aware Fast Inference for Diffusion LLMs. *arXiv preprint arXiv:2602.06953*, 2026. URL https://arxiv.org/abs/2602.06953.

Xinyin Ma, Runpeng Yu, Gongfan Fang, and Xinchao Wang. dKV-Cache: The Cache for Diffusion Language Models. *arXiv preprint arXiv:2505.15781*, 2025. URL https://arxiv.org/abs/2505.15781.

Zanlin Ni, Shenzhi Wang, Yang Yue, Tianyu Yu, Weilin Zhao, Yeguo Hua, Tianyi Chen, Jun Song, Cheng Yu, Bo Zheng, and Gao Huang. The Flexibility Trap: Rethinking the Value of Arbitrary Order in Diffusion Language Models. *arXiv preprint arXiv:2601.15165*, 2026. URL https://arxiv.org/abs/2601.15165.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large Language Diffusion Models. *arXiv preprint arXiv:2502.09992*, 2025. URL https://arxiv.org/abs/2502.09992.

Jiayi Pan, Junjie Zhang, Xingyao Wang, Lifan Yuan, Hao Peng, and Alane Suhr. Tinyzero. https://github.com/Jiayi-Pan/TinyZero, 2025. Countdown dataset and evaluation utilities.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-torre, Bernardo P. de Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple Guidance Mechanisms for Discrete Diffusion Models. *arXiv preprint arXiv:2412.10193*, 2024. URL https://arxiv.org/abs/2412.10193.

Yangyi Shen, Tianjian Feng, Jiaqi Han, Wen Wang, Tianlang Chen, Chunhua Shen, Jure Leskovec, and Stefano Ermon. Improving Diffusion Language Model Decoding through Joint Search in Generation Order and Token Space. *arXiv preprint arXiv:2601.20339*, 2026. URL https://arxiv.org/abs/2601.20339.

Yiping Song, Jinyu You, Zhiliang Tian, Jinsong Su, Minlie Huang, and Chenping Hou. DiffuReason: Enhancing Reasoning Ability for Diffusion Language Models via Monte Carlo Tree Search. In *Proceedings of the International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/64457.

Atula Tejaswi, Litu Rout, Constantine Caramanis, Sanjay Shakkottai, and Sujay Sanghavi. Entropy Aware Reward Guidance for Diffusion Language Model Alignment. *arXiv preprint arXiv:2602.05000*, 2026. URL https://arxiv.org/abs/2602.05000.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dLLM: Training-free Acceleration of Diffusion LLM by Enabling KV Cache and Parallel Decoding. *arXiv preprint arXiv:2505.22618*, 2025. URL https://arxiv.org/abs/2505.22618.

Tianyi Wu, Xiaoxi Sun, Yanhua Jiao, Yulin Li, Yixin Chen, YunHao Cao, YiQi Hu, and Zhuotao Tian. Dynamic-dLLM: Dynamic Cache-Budget and Adaptive Parallel Decoding for Training-Free Acceleration of Diffusion LLM. *arXiv preprint arXiv:2606.26120*, 2026. URL https://arxiv.org/abs/2606.26120.

Chenkai Xu, Yijie Jin, Jiajun Li, Yi Tu, Guoping Long, Dandan Tu, Mingcong Song, Hongjie Si, Tianqi Hou, Junchi Yan, and Zhijie Deng. LoPA: Scaling dLLM Inference via Lookahead Parallel Decoding. *arXiv preprint arXiv:2512.16229*, 2025. URL https://arxiv.org/abs/2512.16229.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7B: Diffusion Large Language Models. *arXiv preprint arXiv:2508.15487*, 2025. URL https://arxiv.org/abs/2508.15487.

Anthony Zhan. Simple Policy Gradients for Reasoning with Diffusion Language Models. *arXiv preprint arXiv:2510.04019*, 2025. URL https://arxiv.org/abs/2510.04019.

Shaorong Zhang, Longxuan Yu, Rob Brekelmans, Luhan Tang, Salman Asif, and Greg Ver Steeg. Generation Order and Parallel Decoding in Masked Diffusion Models: An Information-Theoretic Perspective. *arXiv preprint arXiv:2602.00286*, 2026. URL https://arxiv.org/abs/2602.00286.

Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling Reasoning in Diffusion Large Language Models via Reinforcement Learning. *arXiv preprint arXiv:2504.12216*, 2025. URL https://arxiv.org/abs/2504.12216.

Yangyang Zhong, Yanmei Gu, Zhengqing Zang, Xiaomeng Li, Yuqi Ding, Xibei Jia, Yuting Shen, Zhenzhong Lan, Liwang Zhu, Weiping Liu, Junlin Zhou, Haisheng Liu, Zhong Xin Yu, Pengxin Luo, Donglian Qi, Yunfeng Yan, and Junbo Zhao. Parallelism and Generation Order in Masked Diffusion Language Models: Limits Today, Potential Tomorrow. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026. URL https://aclanthology.org/2026.findings-acl.357/.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. LLaDA 1.5: Variance-Reduced Preference Optimization for Large Language Diffusion Models. *arXiv preprint arXiv:2505.19223*, 2025a. URL https://arxiv.org/abs/2505.19223.

Ying Zhu, Jiaxin Wan, Xiaoran Liu, Siyang He, Qiqi Wang, Xu Guo, Tianyi Liang, Zengfeng Huang, Ziwei He, and Xipeng Qiu. DiRL: An Efficient Post-Training Framework for Diffusion Language Models. *arXiv preprint arXiv:2512.22234*, 2025b. URL https://arxiv.org/abs/2512.22234.

Tamim Zoabi, Ameen Ali, Liran Ringel, and Lior Wolf. Mean-Field Parallel Decoding for Discrete Diffusion Language Models. *arXiv preprint arXiv:2606.15805*, 2026. URL https://arxiv.org/abs/2606.15805.

A Implementation and Evaluation Details

Decoding configurations. Table 6 reports the configurations associated with the main comparison at $B = 32$, $L = 256$ and $N = 128$. Layer-selection and guidance configurations are selected separately for each model, task and decoding setting; the shared implementation settings below remain fixed. The skip budget k is an upper bound, and the realized number of skipped layers may be smaller because of the selection constraints.

Table 6: SLICE configurations for the main comparison ($B = 32$, $L = 256$, $N = 128$). k denotes the skip budget and g the guidance scale. Scores (%) correspond to Table 1.

	LLaDA-8B-Instruct			LLaDA-8B-1.5		
Benchmark	k	g	Score	k	g	Score
ARC-C	6	0.40	84.04	4	0.50	84.56
ARC-E	6	0.40	93.31	8	0.55	93.73
GSM8K	8	0.25	81.35	8	0.30	82.56
MATH-500	4	0.20	37.40	4	0.75	36.00
HumanEval	2	0.25	40.24	2	0.10	42.68
Countdown	4	0.35	28.52	4	0.25	29.30
Sudoku	6	0.25	15.60	2	0.10	17.40
GPQA	8	1.00	29.46	2	0.50	27.68

Shared settings. Branch-wise `zscore_clip` standardization with `strong_std` rescaling, $a = 5$ and $\varepsilon = 10^{-6}$; prompt-only selection with nonconsecutive skips; temperature zero, one completion per problem, no best-of-N and no prompt truncation. The similarity threshold is set to $\tau = 0.95$.

A.1 Datasets and Evaluation

Table 7 summarizes the evaluation sets and metrics. Countdown and Sudoku are evaluated on all examples in our fixed sets. The preliminary 256-example GPQA and Sudoku selection subsets are contained in their respective final evaluation sets.

Table 7: Evaluation sets and metrics across eight benchmarks.

Benchmark	Examples	Evaluation set	Metric
ARC-C	1,172	Challenge test split	Accuracy
ARC-E	2,376	Easy test split	Accuracy
GSM8K	1,319	Test split	Answer accuracy
MATH-500	500	MATH evaluation subset	Answer accuracy
HumanEval	164	Original main tests	Pass@1
Countdown	256	Three-number arithmetic	Solution accuracy
Sudoku	500	4×4 puzzles	Solution accuracy
GPQA	448	Main subset	Accuracy

Task instances. Countdown uses three-number arithmetic problems, and Sudoku uses 4×4 puzzles. GPQA is evaluated on its Main subset, with answer options deterministically shuffled for each question. All methods receive the same evaluation examples and for GPQA, the same option order.

HumanEval execution. We report correctness against the original HumanEval main tests for the single completion returned by each method. For multi-candidate methods, candidate selection precedes execution and does not use hidden-test outcomes. The evaluator uses a default execution timeout of 10 seconds.

A.2 Integration with Inference Accelerators

SLICE modifies the predictive logits before token commitment. Both branches process the same sequence state, and the guided distribution supplies KLASS confidence and cross-step stability as

well as Fast-dLLM commitment confidence. Without SLICE, these decisions use the full-model distribution. The prompt-selected skip set remains fixed throughout decoding.

We use Fast-dLLM's non-cache parallel sampler, not its complete KV-cache pipeline. Full- and weak-branch calls and latency are measured separately from the nominal decoding budget.

A.3 Structural Proximity and Skip-Budget Sensitivity

For structural-proximity analysis, auto-selected and size-matched random weak branches are evaluated on the same recorded decoding states so that trajectory divergence does not confound the intervention. States are sampled from early, middle and late decoding; random skipping uses three nonconsecutive layer-sets matched by realized skip count. Component ablations keep prompts, evaluator and (B, L, N) fixed.

Table 8 reports lower KL divergence and logit distance and higher top-1 agreement, for similarity-selected branches at every evaluated skip budget. Increasing the number of skipped layers reduces agreement for both strategies. These measurements characterize predictive proximity and they do not by themselves establish task-level improvement.

Table 8: Structural proximity and skip-budget diagnostics. Auto denotes similarity-selected skipping; Random uses the same number of skipped layers. Top-1 agreement is a fraction. $\Delta_{\max}$ reports the best score change across guidance values, in percentage points.

$\lvert S\rvert$	KL ↓		Top-1 agreement ↑		Logit ↓		$\Delta_{\max}$
	Auto	Random	Auto	Random	Auto	Random	
LLaDA-8B-Instruct							
2	0.19	0.63	0.75	0.66	81	137	+6.89
4	0.35	1.17	0.68	0.55	115	188	+7.93
6	0.93	1.62	0.52	0.46	149	227	+11.32
8	1.27	2.06	0.42	0.37	178	260	+13.43
LLaDA-8B-1.5							
2	0.23	0.71	0.73	0.64	91	147	+5.98
4	0.44	1.30	0.65	0.52	128	198	+6.15
6	1.11	1.77	0.49	0.44	165	235	+5.99
8	1.47	2.23	0.39	0.35	186	262	+6.38

A.4 Latency Measurement

Table 2 reports generation latency on the fixed samples, with batch size one and four repeats per model and method. Each run includes a full-example warm-up and CUDA synchronization around the timed generation. Model loading and scoring are excluded, while multi-candidate methods include all candidate generation. We report mean seconds per input, normalized to the corresponding base decoder.

B Proofs and Additional Analysis

Scope and notation. We consider a finite vocabulary and finite logits, so $p_{\mathrm{s}}(v) > 0$ for every token. All logarithms are natural. Unless stated otherwise, the prompt, decoding state and token position are fixed; in particular, $\boldsymbol{T}$ and σ_{s} are held fixed when differentiating with respect to g. Sections B.1 and B.2 prove the two main-text theorems; the remaining sections give supplementary properties and conditional interpretations. All statements concern a fixed-state predictive distribution, not a guarantee of sequence-level accuracy.

B.1 Proof of the Distributional Characterization

Proof of Theorem 1. Dividing the numerator and denominator of the guided softmax by $\sum_u e^{z_{\mathrm{s},u}}$ gives

$$q_g(v) = \frac{e^{z_{\mathrm{s},v}+gT_v}}{\sum_u e^{z_{\mathrm{s},u}+gT_u}} = \frac{p_{\mathrm{s}}(v)e^{gT_v}}{Z(g)}, \qquad Z(g) = \mathbb{E}_{p_{\mathrm{s}}}[e^{gT}].$$

Since $Z(g) > 0$, this is a well-defined distribution with full support. For any $q \in \Delta(\mathcal{V})$, using $0 \log 0 = 0$,

$$\begin{aligned} D_{\mathrm{KL}}(q\|q_g) &= \sum_v q(v)\left[\log\frac{q(v)}{p_{\mathrm{s}}(v)} - gT_v + \log Z(g)\right] \\ &= D_{\mathrm{KL}}(q\|p_{\mathrm{s}}) - g\mathbb{E}_q[T] + \log Z(g). \end{aligned}$$

Rearranging yields the stated objective as $\log Z(g) - D_{\mathrm{KL}}(q\|q_g)$. The divergence is nonnegative and vanishes exactly at $q = q_g$, proving existence and uniqueness for every real g, including $g = 0$. □

B.2 Proof of Local Boundedness and Top-1 Preservation

Proof of Theorem 2. By construction, $|T_v| \le \sigma_{\mathrm{s}} a$ for every $v \in \mathcal{V}$. Therefore,

$$|g(T_v - T_u)| \le 2|g|\sigma_{\mathrm{s}} a = 2M,$$

which proves the pairwise-margin bound. From Theorem 1,

$$\log\frac{q_g(v)}{p_{\mathrm{s}}(v)} = gT_v - \log \mathbb{E}_{p_{\mathrm{s}}}\left[e^{gT}\right]. \tag{13}$$

Since $gT_v \in [-M, M]$ for every v,

$$e^{-M} \le \mathbb{E}_{p_{\mathrm{s}}}\left[e^{gT}\right] \le e^{M},$$

and hence the log-normalizer lies in $[-M, M]$. The stated log-probability bound follows. □

Top-1 consequence. Let v^* be the unique full-model maximizer, with margin $m = \min_{u \ne v^*}(z_{\mathrm{s},v^*} - z_{\mathrm{s},u})$. For every competitor,

$$z_{g,v^*} - z_{g,u} \ge z_{\mathrm{s},v^*} - z_{\mathrm{s},u} - 2M \ge m - 2M.$$

Thus $m > 2M$ preserves the unique top-1 token, but not necessarily the commitment order: guidance can still change confidence rankings across positions.

B.3 Standardization and Clipping Properties

Proposition 1 (Offset invariance and scale dependence)**.** *For $z \in \mathbb{R}^V$, write $\bar{z} = V^{-1}\sum_v z_v$, $s^2 = V^{-1}\sum_v (z_v - \bar{z})^2$ and $U_\varepsilon(z) = (z - \bar{z}\mathbf{1})/\sqrt{s^2 + \varepsilon}$. For $\varepsilon > 0$, $\alpha > 0$ and any $b \in \mathbb{R}$,*

$$U_\varepsilon(z + b\mathbf{1}) = U_\varepsilon(z), \tag{14}$$

$$U_\varepsilon(\alpha z + b\mathbf{1}) = \frac{z - \bar{z}\mathbf{1}}{\sqrt{s^2 + \varepsilon/\alpha^2}}. \tag{15}$$

If $s > 0$, the relative deviation from $U_0(z)$ satisfies

$$\frac{\|U_\varepsilon(z) - U_0(z)\|_2}{\|U_0(z)\|_2} = 1 - \left(1 + \frac{\varepsilon}{s^2}\right)^{-1/2} \le \frac{\varepsilon}{2s^2}. \tag{16}$$

Proof. The transformed mean is $\alpha\bar{z} + b$ and its variance is $\alpha^2 s^2$. Substitution and division by $\alpha > 0$ prove the first two identities. For $s > 0$, $U_\varepsilon(z) = (1 + \varepsilon/s^2)^{-1/2} U_0(z)$. The norm ratio follows, and the bound uses $1 - (1+t)^{-1/2} = \int_0^t \frac{1}{2}(1+r)^{-3/2}\, dr \le t/2$ for $t \ge 0$. □

Independent additive offsets leave $\boldsymbol{T}$ and q_g unchanged. Standardized logits are invariant to positive scaling when $\varepsilon = 0$ and variance is nonzero, or when ε is scaled by α^2. For fixed $\varepsilon > 0$, this invariance is only approximate. The guided distribution itself is not scale-invariant, because scaling the strong logits changes both its base distribution and correction magnitude.

For fixed σ_{s}, clipping is also nonexpansive. If $C_a(d)$ clips each coordinate to $[-a, a]$, then

$$\|\sigma_{\mathrm{s}} C_a(d) - \sigma_{\mathrm{s}} C_a(d')\|_\infty \le \sigma_{\mathrm{s}} \|d - d'\|_\infty.$$

Indeed, the scalar clipping map is monotone with slope either zero or one away from its two breakpoints, and is continuous at both; its coordinate-wise difference is therefore at most the input difference. This property bounds amplification by clipping itself, not by the preceding standardization when branch variances are small.

B.4 Distributional Deviation and Local Expansion

Proposition 2 (Quadratic divergence control). *Let $R = \max_v T_v - \min_v T_v \le 2\sigma_{\mathrm{s}} a$ and $M = |g|\sigma_{\mathrm{s}} a$. For every real g,*

$$D_{\mathrm{KL}}(q_g \| p_{\mathrm{s}}),\ D_{\mathrm{KL}}(p_{\mathrm{s}} \| q_g) \le \frac{g^2 R^2}{8} \le \frac{M^2}{2}, \tag{17}$$

$$\mathrm{TV}(q_g, p_{\mathrm{s}}) \le \min\{1, |g|R/4\} \le \min\{1, M/2\}, \tag{18}$$

where $\mathrm{TV}(p, q) = \frac{1}{2}\sum_v |p(v) - q(v)|$.

Proof. Define $A(g) = \log \mathbb{E}_{p_{\mathrm{s}}} e^{gT}$. Finite support allows differentiation under the sum, giving

$$A(0) = 0, \qquad A'(g) = \mathbb{E}_{q_g} T, \qquad A''(g) = \mathrm{Var}_{q_g}(T) \le R^2/4.$$

For the last inequality, use the midpoint of the range of T: the variance is no greater than the mean squared distance to that midpoint, which is at most $R^2/4$. The exponential-family identity gives

$$D_{\mathrm{KL}}(q_g \| p_{\mathrm{s}}) = gA'(g) - A(g) = g^2 \int_0^1 tA''(tg)\, dt,$$

$$D_{\mathrm{KL}}(p_{\mathrm{s}} \| q_g) = A(g) - gA'(0) = g^2 \int_0^1 (1-t)A''(tg)\, dt.$$

These integral identities hold also for negative g. Bounding A'' proves both KL inequalities. For the TV bound, take $E = \{v : p(v) \ge q(v)\}$. The log-sum inequality gives $D_{\mathrm{KL}}(p\|q) \ge d(p(E)\|q(E))$, where d is binary KL. For fixed $y \in (0,1)$, $d(y\|y) = \partial_x d(y\|y) = 0$ and $\partial_x^2 d(x\|y) = 1/[x(1-x)] \ge 4$. Hence $d(x\|y) \ge 2(x-y)^2$, proving $\mathrm{TV}(p,q) \le \sqrt{D_{\mathrm{KL}}(p\|q)/2}$; identical distributions are immediate. Applying the KL bound completes the proof. □

A Taylor expansion at the fixed state further yields

$$q_g(v) = p_{\mathrm{s}}(v)\big[1 + g(T_v - \mathbb{E}_{p_{\mathrm{s}}} T)\big] + O(g^2), \tag{19}$$

$$D_{\mathrm{KL}}(q_g \| p_{\mathrm{s}}) = \tfrac{1}{2} g^2 \,\mathrm{Var}_{p_{\mathrm{s}}}(T) + O(|g|^3). \tag{20}$$

The reverse KL has the same quadratic term. These follow by expanding $A(g)$ at zero and substituting into the identities above. They describe a local expansion of the guided distribution, not approximate linearity of the neural network or of the full decoding trajectory. Since σ_{s} varies across states, the bound is state-dependent rather than a universal constant.

B.5 When the Contrast Direction Can Help

Proposition 3 (Local utility alignment). *For any fixed token utility $r : \mathcal{V} \to \mathbb{R}$,*

$$\frac{d}{dg}\mathbb{E}_{q_g}[r] = \mathrm{Cov}_{q_g}(r, T). \tag{21}$$

Thus $\mathbb{E}_{q_g} r - \mathbb{E}_{p_{\mathrm{s}}} r = g\,\mathrm{Cov}_{p_{\mathrm{s}}}(r, T) + O(g^2)$. A nonzero covariance implies improved expected utility for sufficiently small g of the same sign.

Proof. Differentiating $q_g(v) = p_{\mathrm{s}}(v)e^{gT_v - A(g)}$ gives $q_g'(v) = q_g(v)(T_v - \mathbb{E}_{q_g}T)$. Multiplying by $r(v)$ and summing proves the covariance identity. Taylor expansion at zero gives the second statement. When its linear coefficient is nonzero, it determines the sign for sufficiently small $|g|$. □

For a nonempty proper target set C and $r = \mathbf{1}_C$, the derivative at zero equals

$$p_{\mathrm{s}}(C)\big(1 - p_{\mathrm{s}}(C)\big)\left(\mathbb{E}_{p_{\mathrm{s}}}[T \mid C] - \mathbb{E}_{p_{\mathrm{s}}}[T \mid C^c]\right).$$

Improvement therefore requires target alignment, which neither proximity nor clipping guarantees. Zero covariance gives no first-order conclusion. The result concerns expected token utility, not greedy-decoding accuracy or subsequent state-dependent updates.

Connecting alignment to the structural intervention. For a designated target token y at the fixed state, define its centered contrast

$$\alpha_y = T_y - \mathbb{E}_{p_{\mathrm{s}}}T, \qquad T_v^0 = \sigma_{\mathrm{s}}(u_{\mathrm{s},v} - u_{\mathrm{w},v}), \qquad \widetilde{p}_{\mathrm{w}} = \mathrm{softmax}(\sigma_{\mathrm{s}}\boldsymbol{u}_{\mathrm{w}}). \tag{22}$$

Here T^0 is the contrast before clipping. The scale-aligned reference $\widetilde{p}_{\mathrm{w}}$ is generally different from the native weak distribution $\mathrm{softmax}(\boldsymbol{z}_{\mathrm{w}})$: this distinction is required by SLICE's independent branch standardization and strong-scale rescaling. The target y is used only for analysis, not by the decoding algorithm.

Proposition 4 (Target-specific degradation and clipping)**.** *Let*

$$d_y = \log \frac{p_{\mathrm{s}}(y)}{\widetilde{p}_{\mathrm{w}}(y)}, \qquad \kappa = D_{\mathrm{KL}}(p_{\mathrm{s}} \| \widetilde{p}_{\mathrm{w}}), \qquad \epsilon_{\mathrm{clip}} = \|\boldsymbol{T} - \boldsymbol{T}^0\|_\infty.$$

Then the unclipped centered contrast satisfies the exact identity

$$T_y^0 - \mathbb{E}_{p_{\mathrm{s}}}T^0 = d_y - \kappa, \tag{23}$$

and the implemented, clipped contrast obeys

$$|\alpha_y - (d_y - \kappa)| \le 2\big(1 - p_{\mathrm{s}}(y)\big)\epsilon_{\mathrm{clip}} \le 2\epsilon_{\mathrm{clip}}. \tag{24}$$

In particular, $d_y > \kappa + 2\epsilon_{\mathrm{clip}}$ is sufficient for $\alpha_y > 0$.

Proof. Since $\sigma_{\mathrm{s}}\boldsymbol{u}_{\mathrm{s}} = \boldsymbol{z}_{\mathrm{s}} - \mu_{\mathrm{s}}\mathbf{1}$, softmax offset invariance gives $\log[p_{\mathrm{s}}(v)/\widetilde{p}_{\mathrm{w}}(v)] = T_v^0 + b$ for a constant b independent of v. Taking the p_{s} expectation and subtracting it from the identity at y proves Eq. (23). Write $e_v = T_v - T_v^0$. The remaining difference is

$$e_y - \mathbb{E}_{p_{\mathrm{s}}}e = \sum_{v \ne y} p_{\mathrm{s}}(v)(e_y - e_v).$$

Each summand has $|e_y - e_v| \le 2\epsilon_{\mathrm{clip}}$, proving Eq. (24) and the sufficient condition. □

Equation (23) separates target-specific loss of support under the scale-aligned intervention from its average distributional disruption. Positive unclipped alignment requires the former to exceed the latter; merely making the weak branch less accurate or close to the full branch does not establish this inequality. Clipping preserves this sign when the alignment margin exceeds its perturbation bound. Thus the proposition supplies a checkable condition for a useful structural contrast, not a claim that similarity-based layer selection necessarily satisfies it. In particular, the proximity measurements in Table 8 cannot be substituted for κ unless their definition matches the scale-aligned reference above.

From alignment to a finite guidance update. The local derivative can be strengthened to a finite-step guarantee for target probability, using the same range R as Proposition 2.

Proposition 5 (Finite-step target-probability improvement)**.** *For every real g,*

$$\log \frac{q_g(y)}{p_{\mathrm{s}}(y)} \ge g\alpha_y - \frac{g^2R^2}{8}. \tag{25}$$

If $R > 0$ and $\alpha_y > 0$, then $q_g(y) > p_{\mathrm{s}}(y)$ for every $0 < g < 8\alpha_y/R^2$. More conservatively, if $m_y := d_y - \kappa - 2\epsilon_{\mathrm{clip}} > 0$, then $0 < g < 8m_y/R^2$ suffices.

Proof. With $A(g) = \log \mathbb{E}_{p_s} e^{gT}$ as in Appendix B.4, the exponential reweighting identity gives

$$\log \frac{q_g(y)}{p_s(y)} = g\alpha_y - (A(g) - gA'(0)) = g\alpha_y - D_{\mathrm{KL}}(p_s \| q_g).$$

Proposition 2 bounds the last term by $g^2R^2/8$, proving Eq. (25). The stated intervals make its right-hand side strictly positive; Proposition 4 gives $\alpha_y \geq m_y$. If $R = 0$, T is constant, $\alpha_y = 0$ and $q_g = p_s$, so no strict improvement occurs. For negative guidance, the same bound applies with the signed condition $g\alpha_y > g^2R^2/8$. □

Scope and empirical implications. These results concern the probability of a designated token at a shared decoding state, not the probability of every valid completion. A probability increase need not change the greedy prediction: y becomes the unique maximizer only if

$$z_{s,y} - z_{s,v} + g(T_y - T_v) > 0 \quad \text{for every } v \neq y.$$

Guidance can also change which positions are committed and hence later states, so the fixed-state result does not imply end-to-end accuracy improvement. Testing the proposed mechanism requires paired full- and weak-branch logits at shared states and a well-defined target token or target set. Such data permit measuring α_y, $d_y - \kappa$, clipping error and the realized log-probability change, with problem-level aggregation and matched realized skip counts when comparing selection strategies. Final generated answers alone cannot recover these quantities. This is a validation requirement, not an additional experimental result reported here.

B.6 A Conditional Link from Layer Similarity to Logit Proximity

Proposition 6 (Propagation of skipped-layer perturbations)**.** *Let $H_\ell = F_\ell(H_{\ell-1})$ denote full hidden states for the entire input sequence and $\widetilde{H}_\ell$ the states obtained by replacing layers in S with identity maps, with $H_0 = \widetilde{H}_0$. Assume each retained layer is L_ℓ-Lipschitz in Frobenius norm on a region containing the relevant full and weak states. Define*

$$\delta_\ell = \|F_\ell(H_{\ell-1}) - H_{\ell-1}\|_F, \qquad c_\ell = \begin{cases} 1, & \ell \in S, \\ L_\ell, & \ell \notin S. \end{cases}$$

If the output map G, including final normalization, the head and extraction of the token position, is K-Lipschitz from Frobenius norm to logit ℓ_∞ norm, then

$$\|z_s - z_w\|_\infty \leq K \sum_{\ell \in S} \delta_\ell \prod_{j=\ell+1}^{D} c_j. \tag{26}$$

Proof. Write $e_\ell = \|H_\ell - \widetilde{H}_\ell\|_F$ and note $e_0 = 0$. At a retained layer, Lipschitz continuity yields $e_\ell \leq L_\ell e_{\ell-1}$. At a skipped layer, $\widetilde{H}_\ell = \widetilde{H}_{\ell-1}$, and the triangle inequality gives

$$e_\ell \leq \|F_\ell(H_{\ell-1}) - H_{\ell-1}\|_F + \|H_{\ell-1} - \widetilde{H}_{\ell-1}\|_F = \delta_\ell + e_{\ell-1}.$$

Unrolling these two recurrences gives $e_D \leq \sum_{\ell \in S} \delta_\ell \prod_{j=\ell+1}^{D} c_j$ (an empty product is one). Applying the Lipschitz assumption on G proves the claim. □

What cosine similarity controls. For nonzero vectors a, b, an exact identity is

$$\|b - a\|_2^2 = (\|b\|_2 - \|a\|_2)^2 + 2\|a\|_2\|b\|_2\big(1 - \cos(a, b)\big). \tag{27}$$

It follows by expanding the squared distance and substituting the definition of cosine similarity. Suppose, at the same decoding state, all m token pairs before and after layer ℓ have norms at most R_ℓ, norm differences at most d_ℓ, and mean cosine similarity at least $\bar{\rho}_\ell$. Summing Eq. (27) gives

$$\delta_\ell^2 \leq m\left[d_\ell^2 + 2R_\ell^2(1 - \bar{\rho}_\ell)\right]. \tag{28}$$

With controlled downstream amplification, Eq. (26) converts this residual bound into a logit-proximity bound.

Cosine similarity alone is insufficient: $b = \alpha a$ has cosine one for $\alpha > 0$, but residual $|\alpha - 1|\|a\|_2$. SLICE's prompt-only scores do not establish the all-position, decoding-state assumptions above. The bound is conditional; prompt-to-decoding transfer, selection optimality and accuracy gains require empirical validation.