

Time-Annealed Perturbation Sampling: Diverse Generation for Diffusion Language Models

Jingxuan Wu^{*1} Zhenglin Wan^{*2} Xingrui Yu^{†3}

Yuzhe Yang⁴ Yiqiao Huang⁵ Ivor Tsang³ Yang You²

¹  The University of North Carolina at Chapel Hill

²  National University of Singapore ³  CFAR, Agency for Science, Technology and Research

⁴  University of California, Santa Barbara ⁵  Harvard University

^{*}Equal contribution [†]Corresponding author

jingxwu@unc.edu, Yu_Xingrui@a-star.edu.sg

Abstract: Diffusion language models (Diffusion-LMs) introduce an explicit temporal dimension into text generation, yet how this structure can be leveraged to control generation diversity for exploring multiple valid semantic or reasoning paths remains underexplored. In this paper, we show that Diffusion-LMs, like diffusion models in image generation, exhibit a temporal division of labor: early denoising steps largely determine the global semantic structure, while later steps focus on local lexical refinement. Building on this insight, we propose **Time-Annealed Perturbation Sampling (TAPS)**, a training-free inference strategy that encourages semantic branching early in the diffusion process while progressively reducing perturbations to preserve fluency and instruction adherence. TAPS is compatible with both non-autoregressive and semi-autoregressive Diffusion backbones, demonstrated on LLaDA and TraDo in our paper, and consistently improves output diversity across creative writing and reasoning benchmarks without compromising generation quality.

 Project Page: <https://taps-dlm.github.io/>

1. Introduction

Diffusion language models (Diffusion-LMs) have recently emerged as a promising paradigm for natural language generation [1, 2, 3, 4, 5]. A growing body of work has proposed diverse architectures, ranging from models initialized from pretrained large language model weights to approaches trained entirely from scratch. Across a wide range of tasks, including creative writing, reasoning, and instruction following, these models have demonstrated competitive or even superior performance compared to autoregressive language models [6, 7]. Despite recent progress, Diffusion-LMs face a limitation shared with autoregressive models, frequently generating repetitive or conservative outputs under diversity-promoting instructions [8]. Diffusion-LMs use different inference dynamics from autoregressive language models, and existing diversity control methods do not transfer well. Output diversity in diffusion-based language generation, therefore, remains underexplored and insufficiently addressed in text

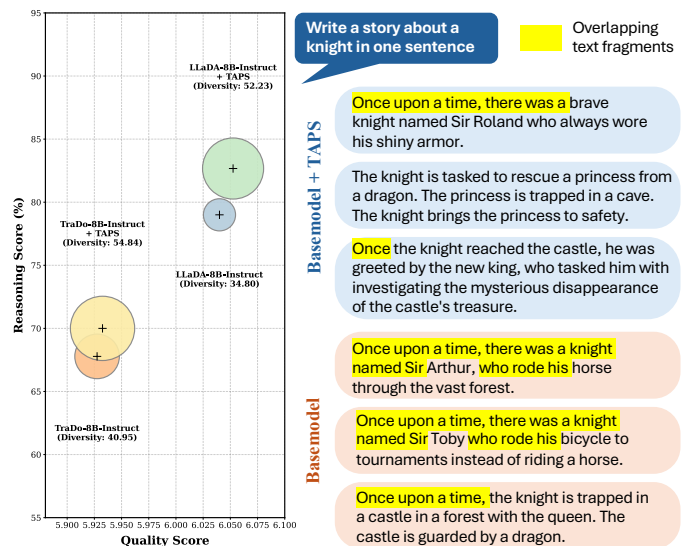


Figure 1: Comparison between TAPS and the base models across diversity, generation quality, and reasoning performance. Quality is assessed by GPT, and reasoning is measured via majority voting accuracy on GSM8K (Sec. 4).

generation.

Prior work on improving generation diversity in large language models can be broadly categorized into training-time and decoding-time approaches. Training-time methods typically modify the optimization objective during supervised fine-tuning or reinforcement learning [9, 10, 11]. While effective in certain settings, these approaches often require carefully curated datasets tailored to the modified objective and tend to possibly generalize poorly beyond the training distribution. Moreover, retraining or fine-tuning models for diversity can be costly and inflexible in practice. Alternatively, decoding-time approaches offer greater adaptability by operating directly at inference time, but they commonly face a fundamental trade-off between diversity and generation quality. Furthermore, prompt-based approaches encourage diversity by conditioning on prior generations or auxiliary instructions; however, their effectiveness depends heavily on the model’s inherent inference capabilities [12, 13]. Sampling-based methods increase diversity by modifying token sampling, such as temperature scaling or nucleus sampling [14, 15, 16]. However, aggressive tuning often harms coherence or fluency, making it difficult to achieve both high diversity and stable quality.

To address this limitation, we introduce a simple yet effective technique termed Time-Annealed Perturbation Sampling (TAPS). From a diffusion perspective, conditional generation can be viewed as sampling trajectories conditioned on a signal derived from the prompt. In standard Diffusion-LMs inference, this conditioning signal remains fixed throughout the denoising process, leading the model to repeatedly converge to similar semantic outcomes under multiple samplings. On the contrary, TAPS perturbs the conditioning signal during inference with a magnitude that decays over denoising steps. Intuitively, injecting stronger perturbations at early denoising steps reduces the model’s reliance on a single conditioning trajectory, enabling semantic branching when global structure is formed. As denoising progresses, the perturbation strength is gradually reduced, allowing the model to recover alignment with the original prompt and refine local lexical and stylistic details. This time-aware design allows TAPS to improve output diversity while preserving generation quality and instruction adherence. To illustrate the effect of TAPS, Figure 1 provides an overview comparison between TAPS and the base models across diversity, quality, and reasoning performance.

Our contributions are three-fold: (i) We empirically identify a temporal semantic structure in Diffusion-LMs, where early denoising steps shape high-level semantics while later steps refine lexical realization. (ii) Based on this insight, we propose Time-Annealed Perturbation Sampling (TAPS), a simple, training-free, and easily scalable inference strategy that leverages diffusion time semantics to improve output diversity without sacrificing generation quality. (iii) We validate TAPS across diverse tasks, including story generation, instruction following, open-ended generation, and mathematical reasoning. Experiments on multiple diffusion-based backbones show consistent gains in both semantic-level and token-level diversity with negligible overhead, where increased diversity translates into improved exploration of alternative semantic or reasoning paths.

2. Related work

Diversity is a fundamental property of conditional text generation, as a single input prompt often admits multiple valid and informative outputs in many tasks [17, 18, 19, 20]. Beyond improving surface-level variation, diversity directly affects the effectiveness of downstream usage, such as selecting high-quality candidates from multiple generations [21], supporting iterative refinement, and enabling exploration in both text-only and multimodal inference settings [22, 23, 24], which is crucial for broadening rollout trajectories in self-training or feedback-driven loops [25, 26, 27, 28]. As a result, improving generation diversity has become an important objective in language modeling research. As discussed above, existing efforts to improve diversity in language models can be broadly categorized into two lines of work: training-free methods and training-based methods.

Training-free Methods For Diversity. In training-free settings, efforts to improve LLM output diversity largely focus on decoding-time by modifying token selection to better navigate the quality–diversity trade-off. Deterministic decoding, such as greedy decoding and beam search, tends to follow the highest-probability trajectories, often yielding repetitive and generic outputs, while beam search additionally incurs non-trivial inference overhead [29]. In contrast, stochastic sampling injects randomness to broaden the space of continuations: temperature sampling rescales distribution sharpness to balance coherence and diversity [30], but overly high temperatures can degrade coherence; top- k sampling restricts candidates to a fixed set of the k most probable tokens [18], and may over-prune under high uncertainty since the truncation does not adapt to context-dependent confidence; top- p sampling preserves the probability “nucleus” by keeping the smallest set whose cumulative mass exceeds p [14], yet under high-temperature regimes it can still admit low-probability tail tokens and thus produce incoherent outputs. To better adapt across contexts with varying uncertainty, recent work explores uncertainty-aware dynamic schemes. One line leverages global statistics such as entropy—e.g., η -sampling and mirostat dynamically regulate the sampling pool/information rate to maintain a target uncertainty level [31, 32]. Another line adapts temperature online: Entropy-Driven Temperature (EDT) adjusts temperature as a function of model entropy [33], while KLD-based approaches tune temperature using KL divergence between two models [34]. Beyond entropy-based control, confidence-scaled truncation has also been studied, Li et al. [35] introduces an adaptive plausibility constraint within contrastive decoding, filtering tokens using a threshold proportional to the maximum probability. Building on this intuition, min- p sampling instantiates confidence-scaled truncation with $p_{\text{scaled}} = p_{\text{base}} \cdot p_{\text{max}}$, tightening the candidate set under high confidence and relaxing it under uncertainty to better balance coherence and diversity [36]. Complementarily, prompt-based conditioning on prior generations can steer later outputs without modifying truncation; for instance, Ruan et al. [13] uses guided prompting and selective intervention while controlling context growth via representative history selection, albeit with potential semantic drift and added inference cost [8].

Training Methods For Diversity. Neural language models often suffer from repetitiveness and output homogenization. Along the supervised-learning line, prior work modifies the maximum-likelihood objective to encourage diversity: maximum mutual information (MMI) reduces generic responses [17], while unlikelihood training explicitly penalizes repetitive continuations [37]. Beyond token-level penalties, target-shaping objectives such as the data-dependent Gaussian prior further regularize overconfident distributions [38]. More recently, preserving diversity has been studied directly in the SFT stage, aiming to mitigate diversity collapse during instruction tuning [9, 39]. A second line incorporates diversity into preference-based post-training. DivPO constructs preference pairs by selecting rare but high-quality responses under an explicit diversity signal, thereby improving diversity without sacrificing alignment quality [10]. Recent diverse preference learning methods analyze diversity collapse in KL-regularized preference optimization and recover diversity by decoupling entropy regularization from reward alignment within the KL penalty [40]. For creativity-oriented post-training, deviation-aware objectives can be applied on top of DPO/ORPO to learn from rare, high-quality instances and promote diverse creative writing [41]. Relatedly, Creative Preference Optimization injects modular creativity signals (including diversity) into the preference objective [42]. Finally, online RL offers a direct mechanism to shape exploration for multi-sample performance: unlikeliness reward corrects GRPO’s rank bias by up-weighting rare-but-correct trajectories, improving pass@N and sample diversity [43]. Li et al. [11] further couples a semantic-level diversity signal with a quality reward during online RL, explicitly optimizing for both diversity and quality rather than treating diversity as a side effect.

3. Methodology

3.1 Preliminary

Different language models adopt different generation paradigms, which fundamentally shape how randomness and diversity arise during inference. We briefly contrast autoregressive language models and diffusion language models, and discuss the implications of their generation mechanisms for diversity control.

Autoregressive Language Models. Autoregressive language models generate text by factorizing the conditional distribution as $p(y | x) = \prod_{t=1}^T p(y_t | x, y_{<t})$, and sample tokens sequentially from this distribution. Randomness is introduced at the token level through stochastic decoding strategies such as temperature scaling or nucleus sampling, leading to diversity across different decoding trajectories.

Diffusion Language Models. Diffusion-based language models generate text through an iterative denoising process that progressively refines a corrupted or masked sequence. Randomness in Diffusion-LMs arises not only from token sampling but also from the inference dynamics themselves, such as stochastic denoising updates and remasking decisions across steps [44]. Existing Diffusion-LMs can be broadly categorized into non-autoregressive and semi-autoregressive variants. Non-autoregressive Diffusion-LMs update all tokens in parallel at each denoising step by predicting masked tokens and selectively remasking them based on model confidence. In contrast, semi-autoregressive DLMs generate text in a block-wise manner: blocks are produced sequentially, while tokens within each block are refined using a non-autoregressive diffusion process.

3.2 Time-Annealed Perturbation Sampling

The core principle of Time-Annealed Perturbation Sampling (TAPS) is that, by perturbing the conditioning signal during inference in a time-dependent manner, the model is exposed to a slightly different input at each denoising step, leading to more diverse generations. In particular, we introduce stochastic perturbations to the context embeddings at early inference stages and gradually attenuate the perturbation strength as inference progresses. As illustrated in Figure 2, we contrast the inference trajectories of TAPS and the base model, highlighting the effect of time-annealed perturbations across denoising steps.

To preserve the essential correspondence between the conditioning context and the generated output, the injected noise is annealed towards zero and completely removed in the final stages of inference. As a result, the model retains strong conditioning fidelity while benefiting from increased exploration in the early semantic formation phase. The complete sampling procedure is summarized in Appendix B.

Formally, given a context embedding \mathbf{E} , we define the perturbed conditioning at inference step t as

$$\tilde{\mathbf{E}}^{(t)} = \begin{cases} \mathbf{E} + \sigma(t)\epsilon, & t \in [t_{\text{start}}, t_{\text{end}}], \\ \mathbf{E}, & \text{otherwise,} \end{cases} \quad (1)$$

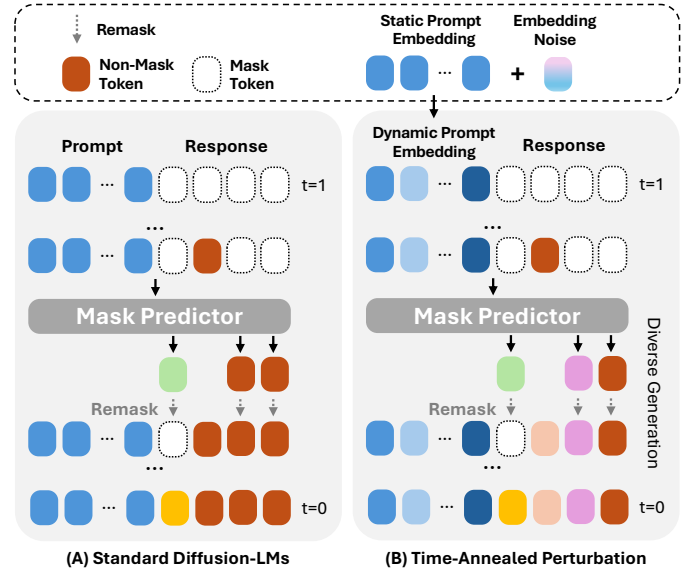


Figure 2: A conceptual comparison of the inference process between the base Diffusion-LM and our proposed method, TAPS, illustrating different context conditioning behaviors.

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and $\sigma(t)$ follows a monotonically decreasing annealing schedule. We note that an alternative form of conditioning perturbation can also be applied at the token level, where context tokens are randomly masked during inference to weaken conditioning strength. We discuss this variant and its empirical behavior in Appendix C.2.

Rescaling the Noise. Strong perturbations applied to conditioning embeddings can easily introduce excessive distributional drift, leading to degraded generation quality, especially in hard tasks. To stabilize the conditioning signal while still enabling effective stochastic exploration, we employ a distribution-aware rescaling mechanism that jointly aligns the statistical structure of the perturbed embeddings and controls their deviation magnitude.

Formally, Let $\tilde{\mathbf{E}}^{(t)}$ denote the perturbed conditioning defined in Eq. (1) at inference step t . We first rescale $\tilde{\mathbf{E}}$ to match the mean and standard deviation of the original embeddings:

$$\mathbf{E}' = \frac{\tilde{\mathbf{E}} - \mu(\tilde{\mathbf{E}})}{\sigma(\tilde{\mathbf{E}})} \cdot \sigma(\mathbf{E}) + \mu(\mathbf{E}),$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ are computed over both the token and embedding dimensions.

To further prevent overly aggressive deviations from the original conditioning signal, we interpolate between the rescaled embeddings \mathbf{E}' and the original embeddings \mathbf{E} using a mixing coefficient $\psi \in [0, 1]$:

$$\hat{\mathbf{E}} = \psi \mathbf{E}' + (1 - \psi) \mathbf{E}.$$

This interpolation provides explicit control over the effective perturbation strength, allowing strong stochastic variation while preserving semantic fidelity. Practically, we find this rescaling-and-mixing strategy crucial for maintaining generation quality under large conditioning noise scales.

Norm-Preserving Projection. While the rescaling and mixing operations preserve the global statistical structure of the conditioning embeddings, they do not explicitly constrain local geometric deviations at the token level. In particular, additive perturbations may significantly alter the embedding norms of individual tokens, which can disrupt the model’s learned representation geometry and lead to unstable generation.

To address this issue, we apply a norm-preserving projection that enforces per-token magnitude consistency between the perturbed and original embeddings. Specifically, let $\hat{\mathbf{E}} \in \mathbb{R}^{T \times d}$ denote the rescaled and mixed embeddings obtained from the previous step, and let \mathbf{E} be the original conditioning embeddings. We project $\hat{\mathbf{E}}$ onto the hypersphere defined by the original token-wise norms:

$$\mathbf{E}_i^* = \frac{\hat{\mathbf{E}}_i}{\|\hat{\mathbf{E}}_i\|_2 + \epsilon} \cdot \|\mathbf{E}_i\|_2, \quad (2)$$

where i indexes tokens in the context and ϵ is a small constant for numerical stability. This projection preserves the directional perturbation introduced by noise while maintaining the original embedding magnitudes, thereby constraining the perturbation to lie on a norm-preserving manifold.

4. Experiments

4.1 Experiments Setup

We briefly describe the experimental setup here. The detailed setup can be found in the Appendix A.

Datasets & Benchmarks. We evaluate on four benchmarks that jointly cover creative writing, real-world instruction following, and verifiable reasoning. Specifically, we use WritingPrompts as our main long-form story generation testbed [18], and complement it with NoveltyBench to stress-test diversity in the multi-sample setting [12]. Beyond open-ended generation, we include GSM8K as a controlled benchmark for multi-step mathematical reasoning [45], and utilize the creative-writing subset of Arena-Hard-Auto to evaluate writing quality and instruction-following under more challenging, preference-aligned prompts [46].

Backbones. We conduct experiments on two diffusion language model backbones LLaDA-8B-Instruct [2] and TraDo-8B-Instruct [3]. Both models are instruction-tuned variants. We initially experimented with their base; however, we found that these models often failed to follow instructions and frequently produced empty or degenerate outputs.

Baselines. We compare against five training-free baselines that encourage diversity through complementary mechanisms while keeping the backbone, prompts, and generation seed. At the decoding level, we include standard token-distribution truncation methods, top- k , top- p , and min- p [36], which control diversity by restricting the candidate set in different ways. To account for diffusion-specific decoding behavior, we additionally evaluate entropy-based dynamic temperature (EDT) [33], which modifies the denoising dynamics during generation. Finally, we include Diverse Prompt as an input-level baseline that perturbs the prompt to elicit varied outputs without changing the model itself [12].

Evaluation Metrics. We evaluate generation diversity using a comprehensive set of lexical, semantic, and embedding-level metrics. Specifically, we report IntraDistinct [17] and calculated Div-Blue as $1 - \text{Self-BLUE}$ [47] to measure surface-level lexical diversity, Sent-BERT [48] to capture semantic variation based on sentence embeddings, and Expectation-Adjusted Distinct Ngrams (EAD) [49] to quantify diversity in the embedding space (see definition details in Appendix A.4). We evaluate generation quality using multi-dimensional automatic evaluators, with the choice of evaluator adapted to each benchmark. On NoveltyBench, we employ Skywork-Reward-Gemma-2-27B-v0.2 [50] to score generated outputs from multiple quality aspects. On WritingPrompts and Arena-Hard-Auto, we follow standard practice and use GPT-4o-2024-08-06 [51] as the quality judge, which provides multi-aspect evaluations of generated responses. For GSM8K, we assess reasoning accuracy by sampling each prompt multiple times and reporting the final accuracy using a majority voting strategy over all generated answers.

4.2 Results

4.2.1. Diversity under Multi-Sample Generation

NoveltyBench consists of two complementary subsets: Curated, which contains carefully filtered prompts designed to elicit diverse yet valid responses, and WildChat, which comprises real-world user instructions collected from open-domain interactions. In our evaluation, we use the full Curated subset of 100 prompts. For WildChat, we randomly sample 500 prompts to ensure a diverse and representative set while keeping the evaluation computationally tractable. For each prompt in both subsets, we generate 10 independent samples under identical decoding conditions, following the standard multi-sample evaluation protocol of NoveltyBench. All diversity and quality metrics are computed over these generated sample sets. Additional results under different temperature settings are provided in the Appendix C.3.

Table 1 reports the diversity results on NoveltyBench, aggregated over both the Curated and WildChat subsets. Across all four diversity metrics, our method consistently achieves the highest scores on both LLaDA-

8B-Instruct and TraDo-8B-Instruct. Compared with baselines, our approach yields substantially larger gains in semantic and embedding-level diversity, as reflected by Sent-BERT and EAD, indicating more meaningful variation beyond surface-level lexical changes. To assess whether the increased diversity comes at the cost of generation quality, we further evaluate the same outputs using the Skywork-Reward-Gemma-2-27B-v0.2 model, which provides multi-aspect quality scores across six human preference dimensions (Figure 3). Our method achieves the strongest performance on creativity-related dimensions, suggesting that the induced diversity translates into genuinely more creative and engaging responses rather than superficial variation. Importantly, on other dimensions, our method performs comparably to or better than standard decoding baselines. Together, these results demonstrate that our approach simultaneously improves diversity and creative quality without compromising overall usefulness or coherence.

We further observe that the effectiveness of several baselines varies markedly across backbones. While Diverse Prompt yields noticeable gains on LLaDA, its performance on TraDo deteriorates substantially across both diversity and quality metrics. A similar backbone-dependent trend is observed for min- p decoding, which performs favorably on LLaDA but degrades on TraDo. In addition, we find that EDT fails to produce reasonable generations on NoveltyBench under our evaluation setting; despite extensive parameter tuning, EDT often leads to degenerate or incoherent outputs and is therefore omitted from the comparison. These observations suggest that prompt-based and token-level decoding strategies can be highly sensitive to the underlying model’s generative capacity and the task distribution. In contrast, our method exhibits more consistent improvements across backbones, indicating a more robust and model-agnostic mechanism for enhancing diversity.

Table 1: Diversity metrics comparison across two backbones on Novelty-Bench.

| Backbone / Method | Diversity metrics | | | |
|--------------------------|--------------------------|---------------------|----------------------|----------------|
| | IntraDistinct \uparrow | Div-Blue \uparrow | Sent-BERT \uparrow | EAD \uparrow |
| LLaDA-8B-Instruct | | | | |
| Base (Normal) | 72.47 | 48.44 | 25.80 | 50.22 |
| Top- p | 74.77 | 42.40 | 24.30 | 51.09 |
| Top- k | 75.64 | 45.16 | 23.38 | 50.50 |
| Min- p | 76.08 | 44.39 | 22.30 | 54.48 |
| Diverse Prompt | 71.50 | 24.32 | 19.84 | 48.58 |
| Ours Method | 78.32 | 66.26 | 36.04 | 63.35 |
| TraDo-8B-Instruct | | | | |
| Base (Normal) | 83.73 | 57.06 | 24.28 | 62.15 |
| Top- p | 82.67 | 53.08 | 22.44 | 60.94 |
| Top- k | 82.54 | 56.49 | 22.93 | 62.84 |
| Min- p | 81.86 | 50.06 | 20.69 | 58.56 |
| Diverse Prompt | 85.57 | 61.50 | 28.58 | 66.51 |
| Ours Method | 86.50 | 65.88 | 29.84 | 67.31 |

4.2.2. Open-Ended Story Generation

Table 2 presents results on WritingPrompts for long-form story generation. We randomly sample 250 prompts and generate 16 outputs per prompt. Due to the inherent instability of Diffusion-LMs, a small fraction of generations may degenerate into malformed or incoherent text; we therefore apply the same validity filtering across all methods and compute metrics on the top 12 valid outputs per prompt. Under this evaluation protocol, our method consistently achieves the highest diversity scores across lexical, semantic, and embedding-level metrics on both backbones. In particular, we observe clear improvements in Sent-BERT and EAD, indicating that the induced diversity extends beyond surface-level lexical variation and reflects more substantial semantic differences among generated stories. Compared to token-level truncation baselines such as top- k , top- p , and min- p , our approach yields stronger and more balanced gains across all diversity metrics, suggesting more effective exploration of the narrative space.

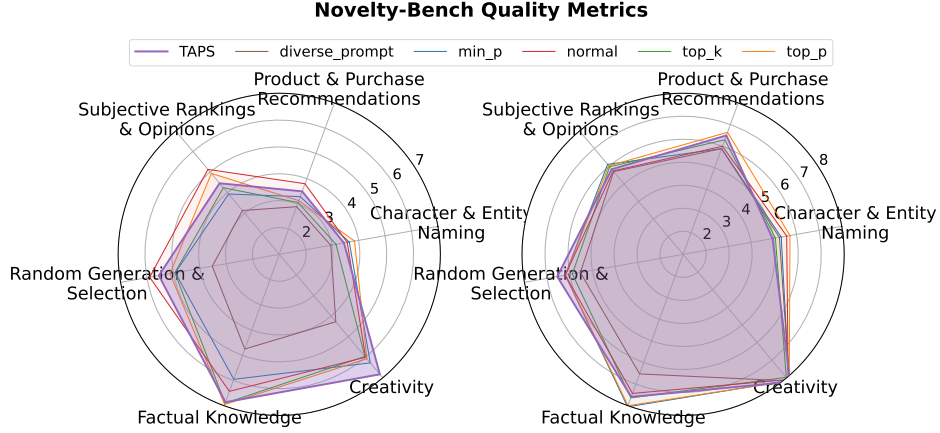


Figure 3: Quality comparison on Novelty-Bench across multiple human preference dimensions. The radar plots compare our method with decoding baselines, showing consistent improvements on creativity-related evaluation dimensions while maintaining comparable overall quality on other dimensions. Results for LLaDA are shown on the left and for TraDo on the right.

To assess generation quality, we use GPT-4o as an automatic judge and report scores along four dimensions: creativity, coherence, writing quality, and relevance. As shown in Table 2, our method maintains quality scores that are comparable to, and in some cases slightly better than, baseline decoding strategies across all evaluated dimensions. Notably, the substantial gains in diversity do not lead to degraded coherence or relevance, indicating that the generated stories remain well-structured and faithful to the input prompts. In contrast, min- p decoding exhibits consistent drops across diversity metrics, while EDT leads to noticeable degradation in quality scores, highlighting the difficulty of preserving narrative quality under entropy-based temperature modulation in long-form generation.

4.2.3. Preference-Aligned Creative Writing

We further evaluate our method on the creative-writing subset of Arena-Hard-Auto, which consists of 250 preference-aligned prompts derived from real user interactions. For each prompt, we generate 8 outputs under identical decoding conditions. As in the previous experiment, we apply the same filtering across all methods and compute metrics on the top 6 valid outputs per prompt. As shown in Table 2, our method consistently achieves the highest diversity scores across diversity metrics on both backbones, closely mirroring the trends observed on WritingPrompts. Notably, the improvements are particularly pronounced on semantic and embedding-based metrics, indicating that the induced diversity reflects meaningful narrative variation rather than superficial lexical changes. Following the same evaluation protocol as in Section 4.2.2, we use GPT-4o to assess quality along four aspects. On Arena-Hard-Auto, our method maintains quality scores that are comparable to or slightly better than other baselines. In contrast, Diverse Prompt exhibits a substantial drop in diversity under this setting, while EDT leads to pronounced degradation in quality scores; min- p decoding consistently underperforms the base model across both diversity and quality dimensions. Besides, we further conduct a preference-based evaluation by directly comparing generations from different methods using GPT-4o as a pairwise judge. The resulting win-rate statistics, reported in Table 10, show that our method is consistently preferred over baseline decoding strategies, providing additional evidence that the increased diversity aligns well with human-aligned preferences rather than introducing undesirable randomness. Additional results under different temperature settings are provided in Appendix C.3.

Table 2: Diversity and multi-aspect quality comparison across two backbones on two benchmarks.

| Backbone / Method | Diversity metrics | | | | Quality (GPT-4o) | | | |
|---|--------------------------|---------------------|----------------------|----------------|------------------|----------------|---------------|----------------|
| | IntraDistinct \uparrow | Div-Blue \uparrow | Sent-BERT \uparrow | EAD \uparrow | Crea \uparrow | Coh \uparrow | WQ \uparrow | Rel \uparrow |
| <i>WritingPrompt</i> | | | | | | | | |
| LLaDA-8B-Instruct | 58.39 | 57.44 | 34.80 | 51.91 | 6.26 | 6.05 | 6.33 | 5.07 |
| Top- p | 58.51 | 57.16 | 34.76 | 51.96 | 6.34 | 6.07 | 6.31 | 4.93 |
| Top- k | 56.95 | 55.01 | 32.82 | 49.45 | 5.99 | 5.61 | 6.01 | 4.66 |
| Min- p | 55.90 | 53.86 | 32.20 | 47.62 | 6.18 | 5.92 | 6.27 | 4.74 |
| Diverse Prompt | 57.01 | 42.40 | 22.50 | 42.15 | 6.16 | 5.42 | 5.91 | 4.37 |
| EDT | 58.85 | 58.90 | 35.02 | 52.72 | 5.95 | 5.51 | 5.98 | 4.58 |
| Ours Method | 60.51 | 60.82 | 40.95 | 55.33 | 6.21 | 5.98 | 6.23 | 5.31 |
| TraDo-8B-Instruct | 67.70 | 81.71 | 52.23 | 66.46 | 5.31 | 6.09 | 5.84 | 6.93 |
| Top- p | 61.61 | 78.50 | 53.54 | 59.76 | 5.28 | 6.21 | 5.92 | 7.05 |
| Top- k | 58.47 | 76.20 | 51.57 | 55.83 | 5.16 | 6.11 | 5.75 | 7.00 |
| Min- p | 55.54 | 75.08 | 51.37 | 51.80 | 5.06 | 6.11 | 5.78 | 6.97 |
| Diverse Prompt | 61.64 | 77.36 | 39.06 | 65.35 | 5.23 | 6.17 | 5.87 | 6.94 |
| EDT | 67.80 | 81.90 | 52.43 | 66.61 | 5.15 | 5.85 | 5.60 | 6.79 |
| Ours Method | 68.44 | 82.37 | 53.84 | 67.06 | 5.32 | 6.10 | 5.83 | 6.96 |
| <i>Arena-Hard-Auto (Creative-Writing)</i> | | | | | | | | |
| LLaDA-8B-Instruct | 54.02 | 67.79 | 24.80 | 44.72 | 4.49 | 4.46 | 4.48 | 5.07 |
| Top- p | 53.75 | 66.80 | 24.18 | 44.27 | 4.56 | 4.53 | 4.60 | 5.17 |
| Top- k | 52.80 | 63.48 | 22.78 | 43.28 | 4.44 | 4.51 | 4.61 | 5.08 |
| Min- p | 51.26 | 60.81 | 22.53 | 41.05 | 4.40 | 4.48 | 4.57 | 5.01 |
| Diverse Prompt | 50.48 | 40.51 | 17.72 | 29.64 | 4.47 | 4.43 | 4.45 | 4.90 |
| EDT | 54.25 | 67.90 | 25.32 | 45.61 | 4.31 | 4.20 | 4.24 | 4.71 |
| Ours Method | 57.70 | 69.53 | 27.35 | 48.12 | 4.66 | 4.59 | 4.43 | 4.95 |
| TraDo-8B-Instruct | 77.65 | 81.17 | 34.57 | 73.76 | 5.56 | 5.08 | 4.84 | 6.05 |
| Top- p | 76.90 | 77.81 | 32.36 | 70.78 | 5.52 | 5.39 | 5.16 | 6.17 |
| Top- k | 75.14 | 76.73 | 32.24 | 69.93 | 5.42 | 5.29 | 5.05 | 6.07 |
| Min- p | 73.25 | 74.41 | 29.90 | 65.55 | 5.47 | 5.22 | 5.14 | 6.16 |
| Diverse Prompt | 63.85 | 47.36 | 17.17 | 48.34 | 5.42 | 5.21 | 5.07 | 6.00 |
| EDT | 78.20 | 81.92 | 34.80 | 73.91 | 5.20 | 4.33 | 4.27 | 5.56 |
| Ours Method | 80.90 | 82.93 | 35.65 | 74.66 | 5.39 | 5.24 | 5.08 | 6.12 |

4.2.4. Reasoning Robustness

We evaluate the robustness of our method on GSM8K to examine whether injecting context-level noise adversely affects mathematical reasoning performance. We randomly sample 300 questions from GSM8K and generate 10 independent reasoning trajectories per question under identical decoding conditions, varying the decoding temperature. We report accuracy under two evaluation protocols: (i) single-sample accuracy computed from the first generated solution (Pass@1), and (ii) majority-vote accuracy obtained by aggregating the final answers from all 10 samples.

As shown in Figure 4, our observations can be summarized from two complementary perspectives. First, compared to other baselines, our method incurs only a mild drop in Pass@1 accuracy while consistently

outperforming all baselines under majority-vote evaluation. Second, as temperature increases, Pass@1 accuracy inevitably decreases for all methods, which would typically also lead to degraded majority-vote performance. However, we observe the opposite trend for our method and Top- k , whose majority-vote accuracy improves at higher temperatures.

These two observations together suggest that our method benefits from structured exploration of diverse reasoning paths. Although stronger exploration slightly affects single-sample reliability, it produces sets of solutions with less correlated errors, allowing aggregation to recover correct answers more effectively. In contrast, the improvement of Top- k arises from a different mechanism: by aggressively truncating low-probability tokens, it enforces a stricter sampling space and yields more reliable answers across repeated samples, even under increased stochasticity. To further illustrate this behavior, Appendix E.1 provides qualitative examples comparing multiple reasoning trajectories generated under different decoding strategies, illustrating how our method explores distinct reasoning paths and how majority voting benefits from their complementarity.

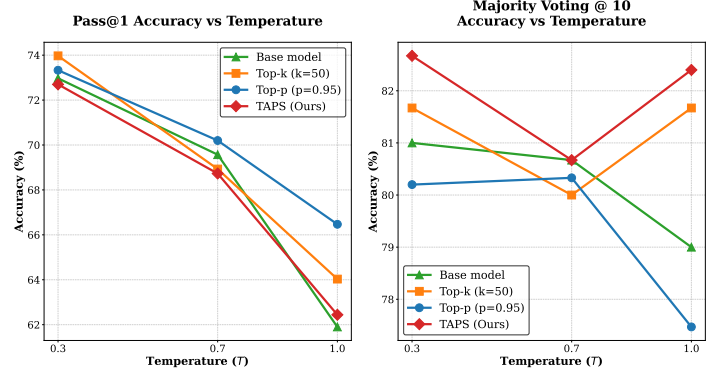


Figure 4: GSM8K accuracy on 300 questions with 10 samples per question. We report single-sample and majority-vote accuracy under three temperatures.

4.3 Ablation Study

This section explores the role of the most important parameters in our method on the final quality and diversity of generated samples. Additional ablations on the role of other parameters are provided in Appendix D.

Noise scale σ . As shown in Table 3, the effect of noise scale exhibits distinct behaviors across the two backbones. For LLaDA-8B-Instruct, increasing the noise scale consistently improves diversity-related metrics, indicating that stronger perturbations encourage broader exploration of the semantic space at the cost of degrading quality. In contrast, for TraDo-8B-Instruct, increasing the noise scale primarily improves the token-level diversity metric, whereas the semantic-level diversity measured by Sent-BERT shows a degradation trend. This suggests that stronger noise in TraDo primarily affects surface-level lexical variation rather than high-level semantic branching. We attribute this difference to the semi-autoregressive diffusion design of TraDo, whose denoising dynamics are less tolerant to strong perturbations compared to non-autoregressive diffusion backbones. As a result, excessive noise is more likely to disrupt semantic coherence in TraDo, reflecting its greater proximity to autoregressive generation and a weaker separation between semantic formation and

Table 3: Robustness under different noise scales σ and injection windows t .

| Backbone / Setting | Div-Blue \uparrow | Sent-BERT \uparrow | EAD \uparrow | Quality \uparrow |
|--------------------------------|---------------------|----------------------|----------------|--------------------|
| LLaDA-8B-Instruct | | | | |
| $\sigma=0.1, t \in [0.9, 0.3]$ | 66.90 | 34.27 | 59.97 | 7.08 |
| $\sigma=0.3, t \in [0.9, 0.3]$ | 68.64 | 37.28 | 60.25 | 6.96 |
| $\sigma=0.2, t \in [0.9, 0.3]$ | 67.80 | 35.51 | 59.58 | 7.05 |
| $\sigma=0.2, t \in [0.9, 0.5]$ | 67.55 | 35.40 | 59.44 | 7.09 |
| $\sigma=0.2, t \in [0.5, 0.1]$ | 67.10 | 34.22 | 59.90 | 6.99 |
| TraDo-8B-Instruct | | | | |
| $\sigma=0.1, t \in [0.9, 0.5]$ | 85.87 | 54.14 | 70.00 | 5.52 |
| $\sigma=0.3, t \in [0.9, 0.5]$ | 85.64 | 54.04 | 70.15 | 5.50 |
| $\sigma=0.2, t \in [0.9, 0.5]$ | 85.83 | 54.54 | 69.49 | 5.54 |
| $\sigma=0.2, t \in [0.9, 0.1]$ | 85.98 | 53.84 | 70.38 | 5.49 |
| $\sigma=0.2, t \in [0.5, 0.1]$ | 85.71 | 53.62 | 69.59 | 5.46 |

surface refinement.

Noise injection time windows t . As shown in Table 3, the timing of noise injection plays a more critical role in shaping semantic diversity. Injecting noise primarily during the early denoising stage consistently yields higher Sent-BERT and EAD scores, whereas configurations that shift noise injection toward later stages exhibit a noticeable decline in Sent-BERT performance. This indicates that late-stage noise has a limited influence on high-level semantic formation and mainly affects local lexical variation, leading to weaker semantic diversification. Moreover, injecting noise at later stages can interfere with the final refinement process, preventing the denoising dynamics from fully absorbing the perturbations and resulting in mild degradation in generation quality. These results support the intuition that early-stage perturbations are more effective for guiding semantic branching in Diffusion-LMs.

Noise rescale coefficient ψ . As shown in Table 4, the noise rescale coefficient ψ effectively safeguards generation quality against embedding-level noise. When $\psi=1.0$, the psi-mix mechanism is disabled and the model fully adopts perturbed embeddings, resulting in the lowest quality scores across both backbones. Meanwhile, varying ψ has only a limited effect on diversity metrics: although $\psi=1.0$ yields slightly higher diversity, the differences remain modest, and even strong quality protection causes only minor diversity degradation. Overall, these results demonstrate that ψ provides a smooth and robust trade-off between diversity and quality. By partially pulling perturbed embeddings back toward the clean conditioning signal, psi-mix effectively stabilizes generation quality without severely compromising diversity, even when ψ takes extreme values.

Table 4: Ablation study of the quality-protection coefficient ψ .

| ψ | IntraDistinct \uparrow | Div-Blue \uparrow | Sent-BERT \uparrow | EAD \uparrow | Quality \uparrow |
|--------------------------|--------------------------|---------------------|----------------------|----------------|--------------------|
| LLaDA-8B-Instruct | | | | | |
| $\psi=1.0$ | 71.94 | 75.43 | 35.90 | 72.12 | 7.18 |
| $\psi=0.9$ | 71.73 | 75.62 | 35.44 | 71.82 | 7.30 |
| $\psi=0.5$ | 71.43 | 75.59 | 36.04 | 71.43 | 7.28 |
| $\psi=0.2$ | 71.54 | 75.66 | 35.93 | 72.11 | 7.23 |
| TraDo-8B-Instruct | | | | | |
| $\psi=1.0$ | 69.74 | 85.81 | 54.91 | 70.19 | 5.45 |
| $\psi=0.9$ | 69.63 | 85.72 | 55.15 | 70.06 | 5.50 |
| $\psi=0.5$ | 69.63 | 85.53 | 54.96 | 69.90 | 5.49 |
| $\psi=0.2$ | 69.55 | 85.63 | 54.70 | 69.96 | 5.48 |

5. Conclusion

We propose Time-Annealed Perturbation Sampling (TAPS), a simple yet effective, training-free decoding strategy for Diffusion Language Models that enhances generation diversity through time-dependent noise injection. Extensive experiments show that TAPS consistently improves output diversity while maintaining a favorable balance between diversity and quality, and generalizes well across different tasks. Inspired by diffusion-based image generation, our results further indicate that Diffusion-LMs inherit key characteristics of diffusion models—namely, the separation between early semantic formation and late-stage refinement—highlighting Diffusion-LMs as a promising direction for future research on controlled and diverse text generation.

References

- [1] Shuang Cheng, Yihan Bian, Dawei Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, Kai Chen, Biqing Qi, et al. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. *arXiv preprint arXiv:2510.06303*, 2025.
- [2] 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.

- [3] Yinjie Wang, Ling Yang, Bowen Li, Ye Tian, Ke Shen, and Mengdi Wang. Revolutionizing reinforcement learning framework for diffusion large language models. *arXiv preprint arXiv:2509.06949*, 2025.
- [4] 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.
- [5] 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.
- [6] Tianyi Li, Mingda Chen, Bowei Guo, and Zhiqiang Shen. A survey on diffusion language models. *arXiv preprint arXiv:2508.10875*, 2025.
- [7] Jaesung Tae, Hamish Ivison, Sachin Kumar, and Arman Cohan. Tess 2: A large-scale generalist diffusion language model. *arXiv preprint arXiv:2502.13917*, 2025.
- [8] Yiming Zhang, Avi Schwarzschild, Nicholas Carlini, Zico Kolter, and Daphne Ippolito. Forcing diffuse distributions out of language models. *arXiv preprint arXiv:2404.10859*, 2024.
- [9] Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Zhi-Quan Luo, and Ruoyu Sun. Preserving diversity in supervised fine-tuning of large language models. *arXiv preprint arXiv:2408.16673*, 2024.
- [10] Jack Lanchantin, Angelica Chen, Shehzaad Dhuliawala, Ping Yu, Jason Weston, Sainbayar Sukhbaatar, and Ilia Kulikov. Diverse preference optimization. *arXiv preprint arXiv:2501.18101*, 2025.
- [11] Tianjian Li, Yiming Zhang, Ping Yu, Swarnadeep Saha, Daniel Khashabi, Jason Weston, Jack Lanchantin, and Tianlu Wang. Jointly reinforcing diversity and quality in language model generations. *arXiv preprint arXiv:2509.02534*, 2025.
- [12] Yiming Zhang, Harshita Diddee, Susan Holm, Hanchen Liu, Xinyue Liu, Vinay Samuel, Barry Wang, and Daphne Ippolito. Noveltybench: Evaluating language models for humanlike diversity. *arXiv preprint arXiv:2504.05228*, 2025.
- [13] Zhiwen Ruan, Yixia Li, Yefeng Liu, Yun Chen, Weihua Luo, Peng Li, Yang Liu, and Guanhua Chen. G2: Guided generation for enhanced output diversity in llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14127–14145, 2025.
- [14] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.
- [15] Max Peeperkorn, Tom Kouwenhoven, Dan Brown, and Anna Jordanous. Is temperature the creativity parameter of large language models? *arXiv preprint arXiv:2405.00492*, 2024.
- [16] Yuqi Zhu, Jia Li, Ge Li, YunFei Zhao, Zhi Jin, and Hong Mei. Hot or cold? adaptive temperature sampling for code generation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 437–445, 2024.
- [17] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 110–119, 2016.

- [18] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*, 2018.
- [19] Nate Gruver, Samuel Stanton, Nathan Frey, Tim GJ Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew G Wilson. Protein design with guided discrete diffusion. *Advances in neural information processing systems*, 36:12489–12517, 2023.
- [20] Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers, 2024. URL <https://arxiv.org/abs/2409.04109>.
- [21] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.
- [22] Gabriele Corso, Yilun Xu, Valentin de Bortoli, Regina Barzilay, and Tommi Jaakkola. Particle guidance: non-i.i.d. diverse sampling with diffusion models, 2023. URL <https://arxiv.org/abs/2310.13102>.
- [23] Seyedmorteza Sadat, Jakob Buhmann, Derek Bradley, Otmar Hilliges, and Romann M Weber. Cads: Unleashing the diversity of diffusion models through condition-annealed sampling. *arXiv preprint arXiv:2310.17347*, 2023.
- [24] Jingxuan Wu, Zhenglin Wan, Xingrui Yu, Yuzhe Yang, Bo An, and Ivor Tsang. Oscar: Orthogonal stochastic control for alignment-respecting diversity in flow matching, 2025. URL <https://arxiv.org/abs/2510.09060>.
- [25] Yihan Du, Anna Winnicki, Gal Dalal, Shie Mannor, and R Srikant. Exploration-driven policy optimization in rlhf: Theoretical insights on efficient data utilization. *arXiv preprint arXiv:2402.10342*, 2024.
- [26] Minghan Chen, Guikun Chen, Wenguan Wang, and Yi Yang. Seed-grpo: Semantic entropy enhanced grpo for uncertainty-aware policy optimization. *arXiv preprint arXiv:2505.12346*, 2025.
- [27] Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*, 2025.
- [28] Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.
- [29] Markus Freitag and Yaser Al-Onaizan. Beam search strategies for neural machine translation. *arXiv preprint arXiv:1702.01806*, 2017.
- [30] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.
- [31] Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R Varshney. Miostat: A neural text decoding algorithm that directly controls perplexity. *arXiv preprint arXiv:2007.14966*, 2020.
- [32] John Hewitt, Christopher D Manning, and Percy Liang. Truncation sampling as language model desmoothing. *arXiv preprint arXiv:2210.15191*, 2022.

- [33] Shimao Zhang, Yu Bao, and Shujian Huang. Edt: Improving large language models’ generation by entropy-based dynamic temperature sampling. *arXiv preprint arXiv:2403.14541*, 2024.
- [34] Chung-Ching Chang, David Reitter, Renat Aksitov, and Yun-Hsuan Sung. Kl-divergence guided temperature sampling. *arXiv preprint arXiv:2306.01286*, 2023.
- [35] Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori B 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)*, pages 12286–12312, 2023.
- [36] Minh Nhat Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Schwartz-Ziv. Turning up the heat: Min-p sampling for creative and coherent llm outputs. *arXiv preprint arXiv:2407.01082*, 2024.
- [37] Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*, 2019.
- [38] Zuchao Li, Rui Wang, Kehai Chen, Masso Utiyama, Eiichiro Sumita, Zhuosheng Zhang, and Hai Zhao. Data-dependent gaussian prior objective for language generation. In *International Conference on Learning Representations*, 2020.
- [39] Long Mai and Julie Carson-Berndsen. Improving linguistic diversity of large language models with possibility exploration fine-tuning. *arXiv preprint arXiv:2412.03343*, 2024.
- [40] Stewart Slocum, Asher Parker-Sartori, and Dylan Hadfield-Menell. Diverse preference learning for capabilities and alignment. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [41] John Joon Young Chung, Vishakh Padmakumar, Melissa Roemmele, Yuqian Sun, and Max Kreminski. Modifying large language model post-training for diverse creative writing. *arXiv preprint arXiv:2503.17126*, 2025.
- [42] Mete Ismayilzada, Antonio Laverghetta, Simone A Luchini, RN Patel, Antoine Bosselut, Lonneke Van Der Plas, and Roger E Beaty. Creative preference optimization. *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 9580–9609, 2025.
- [43] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, 2024.
- [44] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343, 2022.
- [45] 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.

- [46] Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline, 2024. URL <https://arxiv.org/abs/2406.11939>.
- [47] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100, 2018.
- [48] Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of rlhf on llm generalisation and diversity. *arXiv preprint arXiv:2310.06452*, 2023.
- [49] Siyang Liu, Sahand Sabour, Yinhe Zheng, Pei Ke, Xiaoyan Zhu, and Minlie Huang. Rethinking and refining the distinct metric. *arXiv preprint arXiv:2202.13587*, 2022.
- [50] Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*, 2024.
- [51] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.
- [52] Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025.

Content of Appendix

| | | |
|----------|---|-----------|
| A | Implementation Details | 17 |
| A.1 | Hyperparameter Setting | 17 |
| A.2 | Framework and Baseline Implementation Details | 17 |
| A.3 | Backbone-Specific Generation Settings | 17 |
| A.4 | Evaluation Details | 18 |
| B | Algorithm | 19 |
| C | Additional Experiment Results | 19 |
| C.1 | A toy Experiment | 19 |
| C.2 | Token-level Mask-based Noise Injection | 20 |
| C.3 | Results under Different Temperatures | 21 |
| D | Additional Ablation Study | 23 |
| D.1 | Studied Parameters | 23 |
| D.2 | Robustness w.r.t. Annealing Strategy | 24 |
| E | Additional Analysis | 25 |
| E.1 | Examples on GSM8K Reasoning | 25 |
| F | Limitations | 30 |

A. Implementation Details

A.1 Hyperparameter Setting

We use backbone-specific hyperparameter configurations for TRaDO-8B-Instruct and LLaDA-8B-Instruct, following their respective default decoding setups. Differences in denoising steps, block length, and remasking strategies reflect architectural and training variations between the two diffusion language models. Unless otherwise specified, these hyperparameters are fixed across all experiments for each backbone. In addition, the hyperparameters of our method are kept identical across all tasks and temperature settings, providing further evidence of the robustness and generality of the proposed approach.

Table 5: Hyperparameter settings for TraDo-8B-Instruct.

| Hyperparameter | Value |
|-------------------------------------|------------------------|
| TraDo-8B-Instruct (Backbone) | |
| Mask ID | 151669 |
| Generation Length | 200 |
| Block Length | 4 |
| Denoising Steps | 4 |
| Remasking Strategy | low_confidence_dynamic |
| Confidence Threshold | 0.9 |
| TAPS Settings (Ours) | |
| Noise Scale σ | 0.20 |
| Noise Injection Window t | [0.90, 0.50] |
| Noise Rescale Coefficient | 0.90 |
| Annealing Strategy | cosine |

Table 6: Hyperparameter settings for LLaDA-8B-Instruct.

| Hyperparameter | Value |
|-------------------------------------|----------------|
| LLaDA-8B-Instruct (Backbone) | |
| Mask ID | 126336 |
| Generation Length | 256 |
| Block Length | 128 |
| Denoising Steps | 256 |
| CFG Scale | 0.0 |
| Remasking Strategy | low_confidence |
| TAPS Settings (Ours) | |
| Noise Scale σ | 0.20 |
| Noise Injection Window t | [0.90, 0.30] |
| Noise Rescale Coefficient | 0.90 |
| Annealing Strategy | cosine |

A.2 Framework and Baseline Implementation Details

For **TraDo-8B-Instruct**¹ and **LLaDA-8B-Instruct**², we follow the official Hugging Face implementations and usage guidelines released by the model authors. For the **Diverse Prompt** baseline, we build upon the method proposed in Ruan et al. [13], with minor adaptations including a sliding-window mechanism to better support long-form generation. For the **EDT** baseline, we find that the hyperparameter settings recommended in the original work can lead to degenerate outputs when directly applied to diffusion language models; therefore, we re-tune the hyperparameters based on empirical validation to ensure stable and fluent generation. For the **Min- p** baseline, we strictly follow the recommended configurations from the original work across all tasks.

A.3 Backbone-Specific Generation Settings

Although our method is model-agnostic in principle, we adopt different generation configurations for LLaDA and TraDo due to architectural and practical constraints.

¹<https://huggingface.co/Gen-Verse/TraDo-8B-Instruct>

²<https://huggingface.co/GSAI-ML/LLaDA-8B-Instruct>

TraDo is trained on SDAR weights and follows a block-wise masked generation paradigm [1], which adapts an autoregressive Transformer language model. In practice, we find that the block length in TraDo must be set to a small value, specifically, 4 tokens in the official GitHub. Increasing the block length leads to severe generation instability, including garbled outputs or empty generations. Consequently, generating a sequence of 256 tokens requires 64 blocks. Under this setting, diffusion steps within each block are necessarily limited, and semantic construction and surface-level refinement are tightly coupled within a very short horizon. To accommodate this constraint, we inject context noise across blocks with a gradually decaying schedule, allowing early blocks to introduce diversity while later blocks stabilize the output.

In contrast, LLaDA supports substantially larger block lengths. In our experiments, we set the block length to 128 tokens and use a larger number of denoising steps within each block. This configuration more closely aligns with the intended behavior of diffusion language models, where early denoising steps primarily determine high-level semantic structure, and later steps focus on lexical choice and local fluency. As a result, context perturbations injected early in the diffusion process can more effectively induce semantic diversity without significantly harming generation quality. Overall, the greater flexibility in block length and denoising depth allows our method to better exploit the temporal structure of diffusion in LLaDA, which partly explains its stronger empirical performance compared to TraDo.

A.4 Evaluation Details

We evaluate diversity using lexical- and semantic-level metrics computed from multiple generations per prompt. For each prompt, we first apply light text cleaning by removing template/special tokens (e.g., patterns like `<|...|>` and short XML-like tags) and normalizing whitespace. We then discard generations shorter than a minimum character threshold and keep at most the top N longest remaining samples per prompt; prompts with fewer than two valid samples are excluded from evaluation. Tokenization for lexical metrics follows a simple regex-based tokenizer that splits text into word tokens and punctuation marks.

IntraDistinct is computed as the average distinct- n ratio over samples, with $n \in \{1, 2, 3\}$:

$$\text{Distinct}_n(x) = \frac{|\text{uniq}(\text{ngram}_n(x))|}{|\text{ngram}_n(x)|}, \quad \text{IntraDistinct} = \frac{1}{3} \sum_{n=1}^3 \mathbb{E}_x[\text{Distinct}_n(x)],$$

where $\text{ngram}_n(x)$ denotes sets of n -grams in generation x . We also report EAD following an occupancy-style normalization: for a set of generations \mathcal{X} of a prompt, let C_n be the total number of extracted n -grams across \mathcal{X} and N_n be the number of unique n -grams; with global vocabulary size V_n computed from all evaluated generations, we define

$$\text{EAD}_n(\mathcal{X}) = \frac{N_n}{V_n \left(1 - \left(\frac{V_n-1}{V_n}\right)^{C_n}\right)}.$$

We average EAD_n over $n = 1, \dots, 5$ to obtain a single EAD score scaled by 100 for presentation.

For semantic diversity, we use Sentence-BERT embeddings from the all-MiniLM-L6-v2 checkpoint.³ We encode all generations for the same prompt, L2-normalize the embeddings, and compute the mean pairwise cosine distance over all upper-triangular pairs; the resulting SBERT diversity score is scaled by 100. All metrics are first computed per prompt and then averaged across prompts to obtain dataset-level results.

³<https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

B. Algorithm

This section presents the complete TAPS Algorithm 1, including embedding-level conditioning perturbation, noise annealing, and the associated quality preservation mechanisms.

Algorithm 1: TAPS

Require: Diffusion LM f_θ ; prompt token ids \mathbf{p} ; mask token id m ; total steps S ; generation length L ; block length B ; noise window $[t_{\text{start}}, t_{\text{end}}]$; annealing schedule $\sigma(t)$ with maximum scale σ_{max} ; mixing coefficient $\psi \in [0, 1]$.

Output: Completed sequence $\mathbf{x} \in \mathbb{N}^{|\mathbf{p}|+L}$.

$\mathbf{x} \leftarrow [\mathbf{p}, \underbrace{m, \dots, m}_L]$ $\mathbf{E} \leftarrow \text{Embed}(\mathbf{p}) \in \mathbb{R}^{T \times d}$ $N \leftarrow L/B$ $S_b \leftarrow \lfloor S/N \rfloor$ $S' \leftarrow N \cdot S_b$

for $b \leftarrow 0$ **to** $N - 1$ **do**

Let \mathcal{I}_b be the token indices of the current block;

Precompute transfer counts $\{k_i\}_{i=1}^{S_b}$ from the block mask pattern

for $i \leftarrow 1$ **to** S_b **do**

$g \leftarrow b \cdot S_b + i$ $t \leftarrow g/S'$

if $t \in [t_{\text{start}}, t_{\text{end}}]$ **then**

$\tilde{\mathbf{E}} \leftarrow \mathbf{E} + \sigma(t)\sigma_{\text{max}}\epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I});$ // Inject embedding noise

Normalize $\tilde{\mathbf{E}}$ to match the mean and variance of \mathbf{E}

$\mathbf{E}' \leftarrow \psi \tilde{\mathbf{E}} + (1 - \psi) \mathbf{E};$ // ψ -mix for quality preservation

for $j \leftarrow 1$ **to** T **do**

$\mathbf{E}'_j \leftarrow \frac{\mathbf{E}'_j}{\|\mathbf{E}'_j\|_2 + \epsilon} \cdot \|\mathbf{E}_j\|_2;$ // Norm-preserving projection

else

$\mathbf{E}' \leftarrow \mathbf{E}$

$\mathbf{z} \leftarrow f_\theta(\mathbf{x}; \mathbf{E}')$ $\mathbf{x}_0 \leftarrow \arg \max(\mathbf{z} + \text{Gumbel}(\tau))$ Compute confidence scores \mathbf{c} from \mathbf{z} Select k_i masked positions in \mathcal{I}_b with lowest confidence Update selected positions in \mathbf{x} using \mathbf{x}_0

return \mathbf{x}

C. Additional Experiment Results

C.1 A toy Experiment

We design a toy experiment to provide an intuitive illustration of semantic branching and diversity evolution in diffusion language models. Using TraDo-8B as the backbone, we construct a semantically ambiguous prompt—“*The mysterious prisoner looked at the guard and suddenly*”—which naturally admits multiple coherent but distinct continuations. We compare Standard DLM decoding with TAPS, where perturbations are injected during the early denoising stage (the first 30% of steps) to encourage exploration of diverse semantic trajectories.

To visualize the evolution of semantic diversity, we embed generated samples using Sentence-BERT and project them into a shared two-dimensional space via t-SNE. We capture snapshots at three representative stages (early, middle, and final) and visualize semantic coverage using convex hulls, as shown in Figure 5. For the intermediate stage, since DLMs predict all tokens at each denoising step followed by re-masking, we retain all predicted tokens when computing semantic representations. The results reveal a clear contrast: the

Standard DLM exhibits progressive semantic contraction and mode collapse toward a single dominant region, whereas TAPS consistently maintains broader and more multimodal semantic coverage across all stages. This visualization highlights the importance of early-stage intervention in preserving semantic diversity and mitigating repetitive generation in diffusion language models.

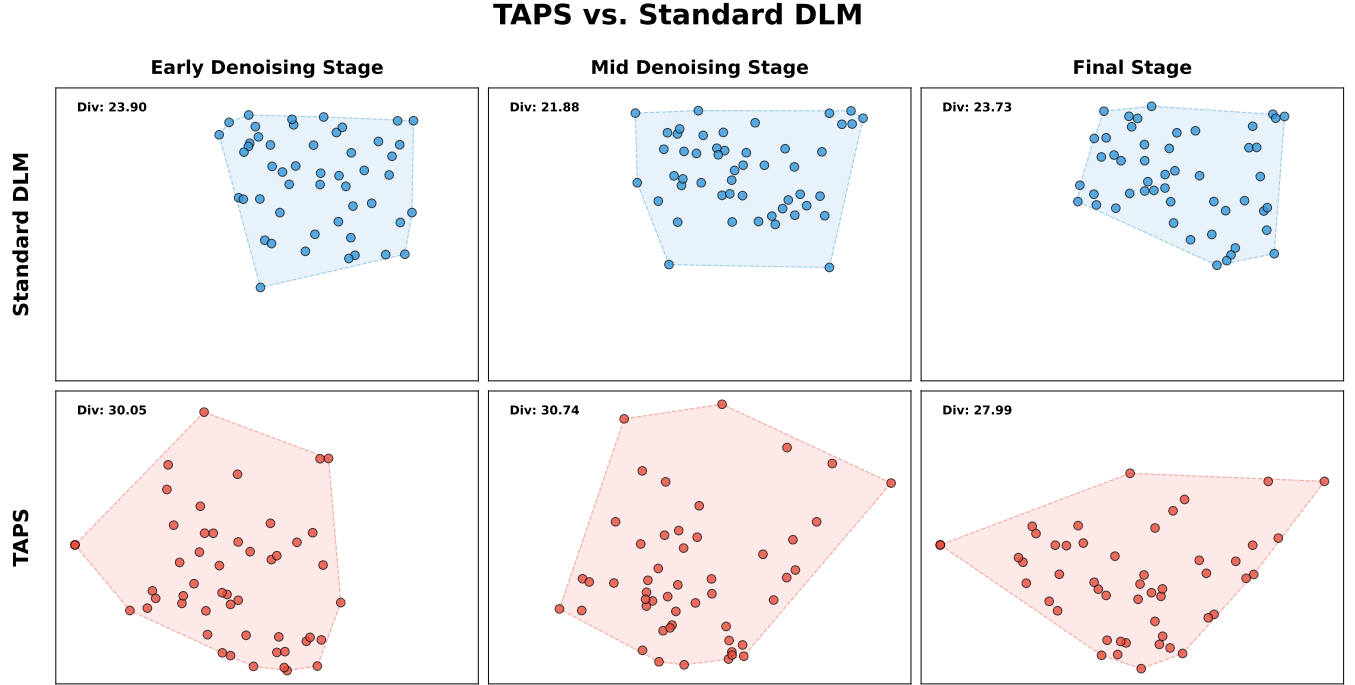


Figure 5: Toy experiment on semantic branching with TraDo-8B. We compare Standard DLM and TAPS by projecting SBERT embeddings of generated samples into a shared 2D space via t-SNE at three denoising stages (early, mid, final). Convex hulls illustrate semantic coverage. TAPS maintains broader and more multimodal coverage across stages, while Standard DLM exhibits progressive semantic contraction. For the mid-stage snapshot, we retain all predicted tokens before re-masking when computing semantic representations.

C.2 Token-level Mask-based Noise Injection

In addition to embedding-level perturbation, we explore a token-level variant that injects noise by masking a subset of tokens during decoding. Due to architectural differences, this variant is evaluated only on LLaDA-8B-Instruct. Specifically, TraDo-8B-Instruct is adapted from autoregressive model weights and does not naturally support token-level masking within the diffusion decoding process.

For token-level perturbation, we randomly mask 5% of the context tokens and apply the same time-annealed schedule as in TAPS, where the masking effect gradually decays within a predefined early denoising window. We conduct experiments at temperature $T = 1.0$ on a writing prompt benchmark consisting of 100 samples, and compare token-level masking with embedding-level perturbation as well as the base decoding strategy.

As shown in Table 7, both perturbation variants improve diversity-related metrics over the base model, indicating that TAPS can be instantiated using either discrete or continuous perturbation mechanisms. However, while token-level masking achieves comparable or higher gains in certain diversity metrics, it exhibits a larger degradation in quality-related scores. In contrast, embedding-level perturbation consistently

maintains a more favorable balance between diversity and generation quality, suggesting that continuous perturbations offer better stability for diffusion language models in practice.

Table 7: Comparison between token-level mask-based perturbation and embedding-level perturbation on LLaDA-8B-Instruct at temperature $T = 1.0$. Results are averaged over 100 writing prompts.

| Method | Diversity Metrics | | | | Quality (GPT-4o) | | | |
|--------------------------|--------------------------|---------------------|----------------------|----------------|------------------|----------------|---------------|----------------|
| | IntraDistinct \uparrow | Div-BLEU \uparrow | Sent-BERT \uparrow | EAD \uparrow | Crea \uparrow | Coh \uparrow | WQ \uparrow | Rel \uparrow |
| LLaDA-8B-Instruct | | | | | | | | |
| Base Model | 72.44 | 75.49 | 35.95 | 71.14 | 6.86 | 7.32 | 7.18 | 7.66 |
| TAPS (Token-level Mask) | 71.14 | 76.69 | 36.12 | 72.43 | 6.81 | 7.18 | 6.95 | 7.60 |
| TAPS (Embedding-level) | 72.68 | 76.60 | 36.51 | 72.68 | 6.90 | 7.23 | 7.25 | 7.73 |

C.3 Results under Different Temperatures

Due to space constraints, we report results at temperature $T = 1.0$ in the main paper. In this appendix, we further present results under additional temperature settings, $T = 0.8$ and $T = 1.5$, on NoveltyBench and Arena-Hard-Auto. We limit our evaluation to these three values, as diffusion language models tend to produce unstable or degenerate outputs when the temperature falls outside this range. Experimental results demonstrate that, across different temperatures, our method consistently improves diversity-related metrics while maintaining comparable generation quality.

C.3.1. Arena-Hard-Auto

For Arena-Hard-Auto, the main paper reports results on the full creative writing subset. In this appendix, we additionally evaluate on a randomly sampled subset of 50 creative writing prompts, where each prompt is generated 8 times to facilitate diversity evaluation. Results under different temperatures are reported in Table 8 ($T = 0.8$) and Table 9 ($T = 1.5$).

At lower temperature ($T = 0.8$), our method consistently outperforms baseline decoding strategies on most diversity metrics across both backbones. Notably, TAPS also achieves the best performance on multiple quality-related metrics, indicating that time-annealed perturbation can effectively enhance diversity without sacrificing generation quality in conservative decoding regimes. At a higher temperature ($T = 1.5$), we observe distinct behaviors between the two backbones. For LLaDA-8B-Instruct, a fully non-autoregressive diffusion model, TAPS remains robust under increased sampling stochasticity and continues to achieve strong diversity gains while maintaining competitive quality. In contrast, for TraDo-8B-Instruct, which retains certain semi-autoregressive characteristics, high temperature amplifies the inherent randomness of autoregressive decoding, leading to noticeable quality degradation for several baseline methods. Under this setting, Top- k sampling achieves the strongest overall quality performance on TraDo-8B-Instruct. This behavior is expected, as Top- k explicitly filters out low-probability tokens and thus provides a stronger constraint on local token selection. Given TraDo’s partial autoregressive nature, such token-level filtering is particularly effective at mitigating error accumulation under high-temperature decoding.

Overall, these results suggest that TAPS provides stable and effective diversity control across temperature regimes, with its robustness being particularly pronounced for diffusion-style backbones. The observed differences further highlight how the degree of autoregressive structure influences a model’s tolerance to high-temperature stochasticity.

Table 8: Diversity and multi-aspect quality comparison across two backbones in Arena-Hard-Auto ($T=0.8$).

| Backbone / Method | Diversity metrics | | | | Quality (GPT-4o) | | | |
|--------------------------|--------------------------|---------------------|----------------------|----------------|------------------|----------------|---------------|----------------|
| | IntraDistinct \uparrow | Div-Blue \uparrow | Sent-BERT \uparrow | EAD \uparrow | Crea \uparrow | Coh \uparrow | WQ \uparrow | Rel \uparrow |
| LLaDA-8B-Instruct | | | | | | | | |
| Base (Normal) | 75.35 | 40.81 | 20.66 | 48.22 | 4.85 | 5.02 | 5.20 | 5.08 |
| Top- p | 76.82 | 38.85 | 19.16 | 45.48 | 4.70 | 4.82 | 4.87 | 4.97 |
| Top- k | 75.07 | 44.98 | 23.27 | 50.23 | 4.94 | 5.04 | 4.97 | 5.25 |
| Min- p | 77.15 | 33.09 | 14.95 | 40.90 | 4.12 | 4.36 | 4.43 | 4.40 |
| EDT | 50.50 | 27.44 | 10.08 | 21.80 | 4.20 | 4.23 | 4.31 | 4.25 |
| Diverse Prompt | 69.98 | 10.87 | 10.31 | 27.97 | 4.75 | 4.72 | 4.77 | 4.81 |
| Ours Method | 78.47 | 43.57 | 24.63 | 49.35 | 4.96 | 5.07 | 5.23 | 5.21 |
| TraDo-8B-Instruct | | | | | | | | |
| Base (Normal) | 81.80 | 40.45 | 20.29 | 49.72 | 5.83 | 5.12 | 5.07 | 6.02 |
| Top- p | 81.63 | 38.22 | 19.90 | 48.40 | 5.80 | 5.19 | 5.16 | 6.03 |
| Top- k | 81.64 | 40.36 | 20.25 | 49.54 | 5.87 | 5.20 | 5.12 | 6.02 |
| Min- p | 80.51 | 37.39 | 19.65 | 47.45 | 5.65 | 5.05 | 5.06 | 5.90 |
| EDT | 80.50 | 27.44 | 10.08 | 21.80 | 3.40 | 2.09 | 2.28 | 2.31 |
| Diverse Prompt | 69.83 | 17.06 | 6.30 | 19.37 | 3.55 | 2.62 | 2.79 | 2.77 |
| Ours Method | 82.47 | 40.77 | 20.36 | 50.30 | 5.90 | 5.19 | 5.09 | 6.07 |

Preference-Based Evaluation on Arena-Hard-Auto Arena-Hard-Auto is an automatic preference-based evaluation benchmark that adopts an LLM-as-a-Judge paradigm to approximate human judgments in Chatbot Arena. It performs pairwise comparisons between a candidate method and a baseline model (Normal decoding in our setting), where a strong judge model (GPT-4o) determines which response is preferred. To mitigate positional bias, each comparison is conducted twice with swapped response orders. The final score represents the win rate against the baseline, with 50% indicating parity, and 95% confidence intervals are estimated via bootstrap resampling.

As shown in Table 10, most decoding strategies—including Min- p , Top- p , Top- k , the Base Model, and TAPS—achieve comparable preference scores clustered around 50%, indicating similar overall quality under this evaluation protocol. In contrast, EDT and Diverse Prompt perform significantly worse, suggesting that aggressively enforcing diversity can substantially harm human-aligned preference. Notably, TAPS achieves the second-highest overall score among all methods, while remaining within the confidence range of the top-performing baselines. These results suggest that TAP is able to enhance diversity without introducing preference-level degradation, maintaining competitiveness under LLM-based preference evaluation.

C.3.2. NoveltyBench

For Novelty-Bench, we additionally evaluate on both the curated and wildchat subsets by randomly sampling 50 prompts from each subset. For each prompt, we generate 10 samples and report the average results. Table 11 summarizes the diversity metrics under different temperature settings.

At lower temperature ($T = 0.8$), our method consistently achieves the best or near-best performance across almost all diversity-related metrics on both backbones, demonstrating its effectiveness in promoting diverse generation under conservative decoding regimes. Importantly, these diversity gains do not come at the

Table 9: Diversity and multi-aspect quality comparison across two backbones in Arena-Hard-Auto ($T=1.5$).

| Backbone / Method | Diversity metrics | | | | Quality (GPT-4o) | | | |
|--------------------------|--------------------------|---------------------|----------------------|----------------|------------------|----------------|---------------|----------------|
| | IntraDistinct \uparrow | Div-Blue \uparrow | Sent-BERT \uparrow | EAD \uparrow | Crea \uparrow | Coh \uparrow | WQ \uparrow | Rel \uparrow |
| LLaDA-8B-Instruct | | | | | | | | |
| Base (Normal) | 83.33 | 54.02 | 32.44 | 64.50 | 4.08 | 3.75 | 3.73 | 4.14 |
| Top- p | 79.76 | 49.30 | 27.31 | 54.40 | 4.60 | 4.61 | 4.66 | 4.78 |
| Top- k | 75.98 | 47.34 | 28.60 | 49.14 | 4.37 | 4.65 | 4.79 | 4.84 |
| Min- p | 80.27 | 37.47 | 21.35 | 44.06 | 4.21 | 4.35 | 4.63 | 4.42 |
| EDT | 79.12 | 42.44 | 22.79 | 41.10 | 4.13 | 4.21 | 4.48 | 4.29 |
| Diverse Prompt | 75.43 | 29.38 | 18.16 | 41.21 | 4.48 | 3.75 | 3.76 | 4.48 |
| Ours Method | 84.29 | 55.58 | 32.14 | 65.80 | 6.21 | 5.52 | 6.23 | 5.31 |
| TraDo-8B-Instruct | | | | | | | | |
| Base (Normal) | 87.51 | 58.32 | 23.18 | 60.47 | 3.27 | 1.85 | 1.77 | 2.57 |
| Top- p | 87.33 | 57.38 | 23.21 | 61.21 | 3.52 | 2.08 | 2.23 | 3.04 |
| Top- k | 83.05 | 44.26 | 12.25 | 52.21 | 5.00 | 4.31 | 4.28 | 4.97 |
| Min- p | 78.45 | 37.78 | 9.30 | 46.31 | 4.82 | 4.23 | 4.19 | 4.80 |
| EDT | 82.12 | 29.20 | 12.09 | 24.75 | 2.51 | 1.48 | 1.66 | 1.61 |
| Diverse Prompt | 70.93 | 18.11 | 9.38 | 22.78 | 2.62 | 1.69 | 1.83 | 1.83 |
| Ours Method | 88.30 | 59.58 | 24.04 | 61.80 | 4.26 | 3.53 | 3.20 | 3.72 |

expense of generation quality: as shown in Figure 6, TAP maintains competitive or superior quality scores across multiple categories, including creativity and subjective preference, compared to baseline decoding strategies. In contrast, the Diverse Prompt baseline exhibits noticeable quality degradation across several aspects, suggesting that aggressively encouraging diversity through prompt manipulation can negatively impact overall generation quality.

At higher temperature ($T = 1.5$), we observe a different behavior. For the base model, this temperature already approaches the upper limit of stable generation, leading to a higher likelihood of degenerate or incoherent outputs. Under this setting, several methods exhibit a sharp increase in diversity metrics, including TAP and Diverse Prompt, but this is accompanied by substantial degradation in generation quality. In contrast, Top- p and Top- k sampling retain relatively strong quality performance at high temperature, which aligns with their intrinsic filtering mechanisms that suppress low-probability or implausible tokens. These results suggest that while high temperature can artificially inflate diversity scores, it may do so by sacrificing output quality, and highlight the importance of balanced diversity control under realistic decoding regimes.

D. Additional Ablation Study

D.1 Studied Parameters

In the robustness study, we vary the following hyperparameters of TAPS while keeping all others fixed to their default values. We consider (i) the noise scale (σ), (ii) noise injection time windows t defined by the start and end denoising steps, which are treated jointly as they together determine the duration of perturbation, and (iii) the noise annealing strategy, including cosine and linear decay. Unless otherwise specified, all results are reported using the same decoding configuration as in the main experiments.

Table 10: Arena-Hard-v2.0 Preference Results (Category: Creative Writing)

| Method | Scores (%) | CI (%) |
|----------------|------------|----------------|
| Min-P | 52.6 | (-10.7 / +7.9) |
| TAPS | 51.7 | (-7.2 / +9.1) |
| Top-P | 50.5 | (-8.6 / +6.9) |
| Base Model | 50.0 | (-0.0 / +0.0) |
| Top-K | 49.3 | (-9.8 / +8.8) |
| EDT | 22.3 | (-7.1 / +8.2) |
| diverse prompt | 19.2 | (-6.7 / +9.5) |

Table 11: Diversity comparison across two backbones on Novelty-Bench under different temperatures. Left: $T = 0.8$. Right: $T = 1.5$.

| Diversity metrics | | | | | Diversity metrics | | | | |
|--------------------------|--------------------------|---------------------|----------------------|----------------|--------------------------|--------------------------|---------------------|----------------------|----------------|
| Backbone / Method | IntraDistinct \uparrow | Div-Blue \uparrow | Sent-BERT \uparrow | EAD \uparrow | Backbone / Method | IntraDistinct \uparrow | Div-Blue \uparrow | Sent-BERT \uparrow | EAD \uparrow |
| LLaDA-8B-Instruct | | | | | LLaDA-8B-Instruct | | | | |
| Base (Normal) | 80.13 | 46.64 | 34.57 | 54.05 | Base (Normal) | 79.01 | 62.33 | 30.89 | 61.18 |
| Top- p | 77.91 | 47.24 | 36.75 | 54.24 | Top- p | 84.76 | 77.99 | 41.65 | 77.14 |
| Top- k | 81.52 | 46.53 | 34.49 | 53.97 | Top- k | 82.70 | 72.36 | 36.76 | 72.49 |
| Min- p | 81.39 | 40.82 | 32.88 | 52.18 | Min- p | 82.03 | 65.81 | 33.02 | 65.61 |
| EDT | – | – | – | – | EDT | – | – | – | – |
| Diverse Prompt | 77.86 | 36.09 | 27.07 | 45.64 | Diverse Prompt | 86.10 | 80.45 | 41.52 | 79.23 |
| Ours Method | 82.27 | 47.61 | 36.64 | 56.14 | Ours Method | 88.33 | 88.84 | 49.12 | 84.11 |
| TraDo-8B-Instruct | | | | | TraDo-8B-Instruct | | | | |
| Base (Normal) | 84.79 | 72.60 | 29.33 | 74.88 | Base (Normal) | 84.79 | 72.60 | 29.33 | 74.88 |
| Top- p | 84.17 | 72.82 | 28.81 | 74.88 | Top- p | 87.53 | 85.58 | 36.70 | 82.77 |
| Top- k | 84.33 | 72.23 | 28.96 | 74.75 | Top- k | 87.30 | 89.43 | 38.88 | 83.81 |
| Min- p | 84.28 | 72.12 | 29.18 | 73.95 | Min- p | 84.97 | 81.67 | 32.69 | 79.06 |
| EDT | – | – | – | – | EDT | – | – | – | – |
| Diverse Prompt | 84.02 | 68.28 | 30.21 | 72.84 | Diverse Prompt | 98.02 | 96.90 | 55.21 | 104.57 |
| Ours Method | 85.60 | 73.91 | 31.20 | 75.72 | Ours Method | 94.45 | 98.69 | 56.08 | 94.15 |

D.2 Robustness w.r.t. Annealing Strategy

Overall, as shown in Table 12, we observe that TAPS exhibits similar performance under both cosine and linear annealing schedules. This suggests that the method does not critically rely on a specific annealing function, provided that the injected noise decays over time. More specifically, for both LLaDA-8B-Instruct and TraDo-8B-Instruct, introducing TAPS with either linear or cosine annealing consistently improves diversity-related metrics over the base decoding strategy. While cosine annealing achieves slightly higher scores in most diversity metrics, the overall performance gap between the two schedules remains small, and both variants exhibit comparable quality scores across creativity, coherence, writing quality, and relevance. These results indicate that the effectiveness of TAPS is robust to the choice of annealing strategy, and that monotonic noise decay—rather than a specific functional form—is the key factor in promoting diverse generation.

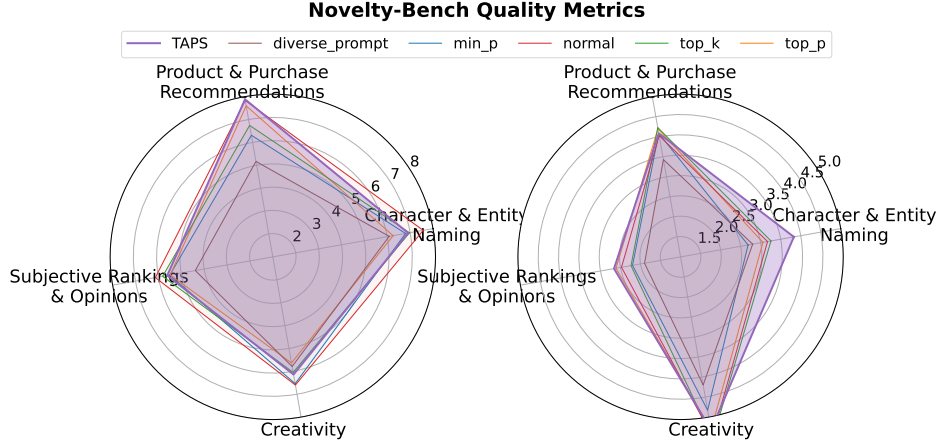


Figure 6: Multi-aspect quality comparison on Novelty-Bench at temperature $T = 0.8$. We evaluate generation quality across four dimensions: Product & Purchase Recommendations, Character & Entity Naming, Subjective Rankings & Opinions, and Creativity. TAPS maintains strong and balanced performance across all aspects, while some baseline methods, such as Diverse Prompt, exhibit noticeable quality degradation despite increased diversity.

Table 12: Robustness study w.r.t. annealing strategy. We compare cosine versus linear noise decay on two backbones while keeping other hyperparameters fixed.

| Backbone / Method | Diversity metrics | | | | Quality (GPT-4o) | | | |
|--------------------------|--------------------------|---------------------|----------------------|----------------|------------------|----------------|---------------|----------------|
| | IntraDistinct \uparrow | Div-BLEU \uparrow | Sent-BERT \uparrow | EAD \uparrow | Crea \uparrow | Coh \uparrow | WQ \uparrow | Rel \uparrow |
| LLaDA-8B-Instruct | | | | | | | | |
| Base (Normal) | 57.82 | 65.11 | 33.53 | 57.60 | 6.25 | 5.66 | 6.04 | 6.76 |
| TAPS (Linear Anneal) | 59.06 | 66.32 | 34.82 | 58.44 | 6.30 | 5.74 | 6.03 | 6.73 |
| TAPS (Cosine Anneal) | 59.73 | 66.38 | 34.41 | 59.20 | 6.31 | 5.79 | 6.11 | 6.65 |
| TraDo-8B-Instruct | | | | | | | | |
| Base (Normal) | 76.02 | 89.60 | 56.63 | 77.65 | 5.03 | 5.17 | 5.06 | 5.92 |
| TAPS (Linear Anneal) | 76.88 | 89.87 | 57.09 | 78.04 | 5.08 | 5.17 | 5.10 | 5.81 |
| TAPS (Cosine Anneal) | 77.07 | 89.93 | 57.40 | 78.22 | 5.07 | 5.19 | 5.11 | 5.83 |

E. Additional Analysis

E.1 Examples on GSM8K Reasoning

In this section, we present qualitative comparisons between our method and the base model on GSM8K problems with different levels of difficulty. We focus on two representative cases to illustrate how diversity-enhancing perturbations interact with mathematical reasoning under repeated sampling.

For an easier problem, we show that our method substantially increases the diversity of generated solutions while preserving correct reasoning, producing multiple valid reasoning paths that differ in their intermediate steps and descriptions. For a more challenging problem, where the solution space contains multiple uncertain or misleading paths, we demonstrate that injecting noise encourages broader exploration and helps prevent

repeated sampling from collapsing into the same erroneous reasoning pattern. Together, these examples provide concrete insight into how controlled exploration enables diverse reasoning paths and benefits aggregation-based evaluation.

Example 1. Lloyd’s chickens produce 252 eggs per day. He sells eggs at \$2 per dozen. How much money does he make in one week?

TAPS 10 reasoning trajectories (accuracy 100%)

- (1) Lloyd’s chickens produce 252 eggs per day. Since one dozen equals 12 eggs, $252 / 12 = 21$ dozen eggs per day. He sells each dozen for \$2, so he makes $21 \times \$2 = \42 per day. In a week (7 days), he makes $\$42 \times 7 = \294 . The answer is \$294.
- (2) 252 eggs per day equals $252 / 12 = 21$ dozens of eggs. If Lloyd sells each dozen for \$2, then he makes $21 \times \$2 = \42 per day. In a week, Lloyd makes $\$42 \times 7 = \294 . The answer is \$294.
- (3) Lloyd’s chickens produce 252 eggs per day. Since there are 12 eggs in a dozen, $252 / 12 = 21$ dozens per day. He sells them for \$2 per dozen, so he makes \$42 per day. Over 7 days, he earns \$294. The answer is \$294.
- (4) 252 eggs per day means $252 \times 7 = 1764$ eggs per week. Dividing by 12 eggs per dozen gives 147 dozens per week. At \$2 per dozen, Lloyd earns \$294 per week. The answer is \$294.
- (5) Lloyd produces 252 eggs per day, which is 21 dozens per day. Selling each dozen for \$2 gives \$42 per day. Over a week, this totals \$294. The answer is \$294.
- (6) Lloyd produces 21 dozens of eggs per day. Multiplying by 7 days gives 147 dozens per week. At \$2 per dozen, his weekly income is \$294. The answer is \$294.
- (7) 252 eggs per day divided by 12 gives 21 dozens. If Lloyd sells 21 dozens per day at \$2 per dozen, he earns \$42 per day. Over 7 days, this equals \$294. The answer is \$294.
- (8) 252 eggs per day equals 21 dozens per day. Selling at \$2 per dozen yields \$42 per day. In one week, Lloyd earns \$294. The answer is \$294.
- (9) 252 eggs per day gives 21 dozens per day. Over 7 days, that is 147 dozens. At \$2 per dozen, Lloyd earns \$294 per week. The answer is \$294.
- (10) 252 eggs per day yields 21 dozens. Selling each dozen for \$2 gives \$42 per day. Over a week, Lloyd earns \$294. The answer is \$294.

Majority answer: \$294 ✓

Top- k 10 reasoning trajectories (accuracy 80%)

- (1) 252 eggs per day are sold at \$2 per dozen. Since $252/12 = 21$ dozen eggs, Lloyd earns $21 \times \$2 = \42 per day. Over a week, he earns $42 \times 7 = \$294$. The answer is \$294.
- (2) 252 eggs per day is equal to $252 / 12 = 21$ dozen eggs per day. In sells them for 2 per dozen, so he makes $21 \times 2 = \$42$ per day. In a week, 7 days, he makes $42 \times 7 = \$294$. The answer is 294.
- (3) 252 eggs per day equals $252/12 = 21$ dozen eggs per day. Selling each dozen for \$2 yields \$42 per day. Over 7 days, Lloyd earns \$294. The answer is \$294.
- (4) 252 eggs per day corresponds to $252/12 = 21$ dozen eggs per day. Over a week, this is $21 \times 7 = 147$ dozens. At \$2 per dozen, Lloyd earns \$294 per week. The answer is \$294.
- (5) 252 eggs per day is equal to $252 / 12 = 21$ dozen eggs per day. Lloyd sells each dozen for \$2, so he makes $21 \times \$2 = \42 per day. In a week, he makes $42 \times 7 = \$294$. The answer is \$294.
- (6) 252 eggs per day equals 21 dozens per day. At \$2 per dozen, Lloyd earns \$42 per day. Over 7 days, he earns \$294. The answer is \$294.
- (7) 252 eggs per day gives $252/12 = 21$ dozen eggs per day. Over a week, that is $21 \times 7 = 147$ dozens. At \$2 per dozen, Lloyd earns \$294 per week. The answer is \$294.
- (8) Lloyd produces $252 \times 7 = 1664$ eggs per week. Dividing by 12 gives $1664/12 = 132$ dozens. At \$2 per dozen, he earns \$264 per week. The answer is \$264.
- (9) 252 eggs per day equals $252/12 = 21$ dozen eggs per day. Selling at \$2 per dozen gives \$42 per day. Over a week, Lloyd earns \$294. The answer is \$294.
- (10) 252 eggs per day equals 21 dozen eggs per day. Selling each dozen for \$2 yields \$42 per day. Over a week, Lloyd earns \$294. The answer is \$294.

Majority answer: \$294 ✓

Base Model 10 reasoning trajectories (accuracy 60%)

- (1) 252 eggs per day equals $252 / 12 = 21$ dozens per day. Over 7 days, this is 147 dozens per week. At \$2 per dozen, Lloyd makes \$294 per week. The answer is \$294.
- (2) 252 eggs per day equals $252 / 12 = 21$ dozen eggs per day. Selling at \$2 per dozen yields \$42 per day. Over 7 days, Lloyd makes \$294 per week. The answer is \$294.
- (3) 252 eggs per day over 7 days gives 1764 eggs per week. The calculation is incorrect and concludes Lloyd makes \$254 per week.
- (4) 252 eggs per day equals 21 dozen eggs per day. The reasoning incorrectly computes daily income as \$84 and weekly income as \$588. The answer is \$588.
- (5) $252 / 12 = 21$ dozens per day. Selling at \$2 per dozen yields \$42 per day. Over 7 days, Lloyd makes \$294. The answer is \$294.
- (6) 252 eggs per day equals 21 dozens per day. Over 7 days, this is 147 dozens per week. At \$2 per dozen, Lloyd makes \$294. The answer is \$294.
- (7) 252 eggs per day over 7 days yields 1764 eggs per week. Dividing by 12 gives 147 dozens per week. At \$2 per dozen, Lloyd makes \$294 per week. The answer is \$294.
- (8) The reasoning contains multiple arithmetic errors and concludes Lloyd makes \$358 per week.
- (9) The calculation incorrectly derives extremely large numbers and concludes Lloyd makes \$2968 per week.
- (10) The reasoning incorrectly doubles the egg count and concludes Lloyd makes \$588 per week.

Majority answer: \$294 ✓

Example 2. Steve and Tim decide to see who can get home from school the fastest. Steve lives further away than Tim, so he is allowed to ride his bike. Steve lives 3 miles from the school and can bike at 440 feet per minute. Tim lives 2 miles away from the school. If Tim can ride his skateboard at 264 feet per minute, how long will the winner be waiting at their house before the loser finishes the race?

TAPS 10 reasoning trajectories (accuracy 30%)

- (1) Steve lives 3 miles from school. Converting miles to feet, the total distance is 5400 feet. Tim lives 2 miles from school, which is 3600 feet. Time for Steve = $5400 / 440 = 12.5$ minutes. Time for Tim = $3600 / 264 = 9.5$ minutes. The difference is calculated as 13 minutes. The answer is 13 minutes.
- (2) 2 miles equals 78720 feet. Tim's time is computed as $78720 / 264 = 30$ minutes. Steve travels 3 miles (118080 feet) and is computed as $118080 / 440 = 262$ minutes. The difference is 232 minutes. The answer is 232.
- (3) Steve: 3 miles = 15,840 feet; $15,840 / 440 = 36$ minutes. Tim: 2 miles = 10,560 feet; $10,560 / 264 = 40$ minutes. The difference is 4 minutes. The answer is 4 minutes.
- (4) 3 miles = 52,800 feet; $52,800 / 440 = 120$ minutes. 2 miles = 24,000 feet; $24,000 / 264 = 90$ minutes. The difference is 30 minutes. The answer is 30 minutes.
- (5) Steve travels 5280 feet at 440 feet per minute, and the time is computed as 110 minutes. Tim travels 3520 feet at 264 feet per minute. The answer is 11000.
- (6) Steve: $15,840 / 440 = 36$ minutes. Tim: $10,560 / 264 = 40$ minutes. The difference is 4 minutes. The answer is 4.
- (7) Steve bikes 3 miles (15,840 feet) at 440 feet per minute, taking 36 minutes. Tim walks 2 miles (10,560 feet) at 264 feet per minute, taking 40 minutes. The difference is 4 minutes. The answer is 4.
- (8) Steve travels 14,520 feet at 440 feet per minute, taking 33 minutes. Tim travels 9,680 feet at 264 feet per minute, taking 25 minutes. The winner waits 8 minutes. The answer is 8.
- (9) Steve: $15,840 / 440 = 36$ minutes. Tim: $10,560 / 264 = 40$ minutes. The difference is 4 minutes. The answer is 4 minutes.
- (10) Steve travels 17,640 feet at 440 feet per minute, taking 40 minutes. Tim travels 11,760 feet at 264 feet per minute, taking 45 minutes. The difference is 5 minutes. The answer is 5.

Majority answer: 4 minutes ✓

Base Model 10 reasoning trajectories (accuracy 10%)

- (1) Steve's speed is 440 feet per minute and Tim's speed is 264 feet per minute. Several intermediate calculations are performed with inconsistent reasoning. The solution concludes that the winner will be waiting for 50 minutes. The answer is 50.
- (2) Steve lives 3 miles away, converted to 5440 feet. At 440 feet per minute, Steve takes 13 minutes. Tim lives 2 miles away, converted to 2200 feet. At 264 feet per minute, Tim takes 8.5 minutes. The difference is 4.5 minutes. The winner waits 4.5 minutes.
- (3) Steve lives 3 miles from school, converted to 10,400 feet. At 440 feet per minute, it takes him 24.8 minutes. Tim lives 2 miles away, converted to 5280 feet. At 264 feet per minute, it takes him 20 minutes. The difference is 4.8 minutes. The answer is 4.8 minutes.
- (4) Steve lives 3 miles from school, which equals 15,840 feet. At 440 feet per minute, Steve takes 36 minutes. Tim lives 2 miles away, which equals 10,560 feet. At 264 feet per minute, Tim takes 30 minutes. The solution concludes that the winner waits 6 minutes. The answer is 6 minutes.
- (5) Steve bikes 15,840 feet at 440 feet per minute, taking 36 minutes. Tim travels approximately 10,762 feet at 264 feet per minute, taking about 38 minutes. The reasoning becomes inconsistent and concludes that the winner waits about 8 minutes.
- (6) Steve travels 20,800 feet at 440 feet per minute and is estimated to take several minutes due to incorrect calculations. Tim travels 10,560 feet at 264 feet per minute, taking 40 minutes. The difference is reported as 5 minutes. The answer is 5 minutes.
- (7) Steve travels 4280 feet at 440 feet per minute, taking about 10 minutes. Tim travels 6280 feet at 264 feet per minute, taking about 23 minutes. The difference is incorrectly computed as 13 minutes. The answer is 13 minutes.
- (8) Steve travels 15,840 feet at 440 feet per minute, taking 36 minutes. Tim travels 10,560 feet at 264 feet per minute, taking 40 minutes. The winner waits 4 minutes. The answer is 4 minutes.
- (9) Distances are converted inconsistently and mixed with unrelated arithmetic. Steve is estimated to finish in under 1 minute, while Tim takes about 50 minutes. The reasoning is incoherent and no consistent conclusion is reached.
- (10) The problem statement is partially restated with incorrect unit conversions and arithmetic. The final calculation is incomplete and no valid waiting time is obtained.

Majority answer: 50 minutes ×

Top- k 10 reasoning trajectories (accuracy 30%)

- (1) Steve lives 3 miles from school, which is $3 \times 5280 = 15840$ feet. Tim lives 2 miles away, or $2 \times 5280 = 10560$ feet. Steve's time is $15840/440 = 36$ minutes, while Tim's time is $10560/264 = 40$ minutes. The waiting difference is $40 - 36 = 4$ minutes. The answer is 4 minutes.
- (2) Steve takes 36 minutes to get home, while Tim takes 30 minutes. The difference is $36 - 30 = 6$ minutes. The answer is 6 minutes.
- (3) Steve takes 36 minutes and Tim takes 40 minutes. The waiting time is $44 - 36 = 8$ minutes. The answer is 8 minutes.
- (4) Steve bikes 3 miles and Tim travels 2 miles, yielding a time difference of $15 - 12 = 3$ minutes. The answer is 3 minutes.
- (5) After converting the distances and speeds, the time difference is computed as $1/140$ minutes, which equals 3 seconds. The answer is 3 seconds.
- (6) Steve takes 12 minutes and Tim takes 15 minutes to get home. The difference is $15 - 12 = 3$ minutes. The answer is 3 minutes.
- (7) Steve's travel time is 36 minutes and Tim's is 40 minutes. The difference is $40 - 36 = 4$ minutes. The answer is 4 minutes.
- (8) Steve takes 36 minutes and Tim takes 33 minutes. The difference is $36 - 33 = 3$ minutes. The answer is 3 minutes.
- (9) Steve arrives in 36 minutes, while Tim arrives in 40 minutes. The winner waits $40 - 36 = 4$ minutes. The answer is 4 minutes.
- (10) Steve finishes in 36 minutes and Tim in 37 minutes. The waiting time is 1 minute. The answer is 1 minute.

Majority answer: 3 minutes ×

F. Limitations

While TAPS provides a simple and effective mechanism for enhancing diversity in diffusion language models, it also has several limitations. First, TAPS applies perturbations uniformly at the representation level, without distinguishing the relative importance of different tokens within a sequence. Prior work suggests that certain tokens can play a decisive role in shaping the overall semantic trajectory of a sentence [52]. Incorporating token-level importance or saliency into the perturbation process could further improve the effectiveness of early-stage interventions.

Second, although TAPS generally maintains a favorable balance between diversity and quality, we observe mild degradation in certain quality metrics compared to the base model, particularly under higher temperature settings where stochasticity is already amplified. This suggests that additional mechanisms for quality preservation, such as adaptive perturbation strength or quality-aware scheduling, may be beneficial. We leave the exploration of more fine-grained and quality-aware perturbation strategies to future work.