Reconsidering Positional Supervision in Masked Diffusion Language Model Training

Mengyu Ye[*]
Tohoku University

Keito Kudo
Tohoku University & RIKEN

Ryosuke Takahashi
Tohoku University & RIKEN

Jun Suzuki
Tohoku University & RIKEN & NII LLMC.

Abstract

Masked diffusion language models (MDLMs) generate text by unmasking tokens in parallel and have recently emerged as alternatives to autoregressive language models. They can be viewed as parallel decoders trained with a position-wise cross-entropy (CE) loss, the same setup as non-autoregressive translation (NAT). In NAT, CE-trained parallel decoders have been argued to be sensitive to small positional shifts, since CE penalizes them harshly. We ask whether CE-trained MDLMs are similarly sensitive to such shifts under iterative decoding. To probe this, we apply a controlled intervention that introduces them during decoding. On LLaDA-8B-Instruct with Arena-Hard, displacing as little as 1% of generated tokens by one position substantially reduces win rates against the unintervened model, showing that MDLMs are sensitive to such small shifts under iterative parallel decoding. Motivated by this, we adapt connectionist temporal classification (CTC), an alignment-flexible objective known to mitigate it there, to MDLM supervised fine-tuning. By relaxing the strict position-wise match that CE imposes, CTC gives the loss room to absorb small positional shifts; concretely, we modified CTC objective to use a special <SLACK> token that absorbs positional uncertainty between target tokens and output positions, and a updated collapse map that preserves target surface forms. Across four open-ended generation benchmarks, the resulting model consistently improves over both the original model and a matched cross-entropy-trained baseline, with statistically significant gains on all four. These results identify training-side alignment flexibility as a useful design dimension for MDLM SFT, complementary to the inference-time approaches explored in prior work[2].

1 Introduction

Diffusion language models have recently emerged as promising alternatives for text generation [42, 71, 56, 28, 3]. Unlike autoregressive (AR) language models [57, 4] that generate text strictly left-to-right, diffusion language models decode multiple tokens in parallel. In particular, Masked Diffusion Language Models (MDLMs) have closed much of the perplexity gap to AR models at small scales [41, 44, 52, 55, 1], and at larger scales [42, 63, 71, 56, 3] match strong AR baselines on challenging math and reasoning tasks.

Standard MDLM training corrupts gold text by randomly masking some tokens, leaves the remaining gold tokens as conditioning context, and supervises the model with a cross-entropy (CE) loss to

[*]Corresponding to: <ye.mengyu.s1@dc.tohoku.ac.jp>; <is-failab-research@grp.tohoku.ac.jp>
[2]Code: https://github.com/muyo8692/mdlm-ctc

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

predict each masked position from the partially masked sequence. This choice simplifies and stabilizes training [52], but it builds positional rigidity into the objective at multiple points: each prediction is aligned to its target at a fixed output position, and the conditioning context surrounding each masked token is assumed to be gold text at gold positions. CE has no mechanism to absorb small displacements at either point. This combination of parallel decoding and position-wise CE is also the defining setup of non-autoregressive translation (NAT) [19], where CE-trained parallel decoders have been argued to be sensitive to small positional shifts [13, 10, 11]: CE penalizes small positional shifts. Whether MDLMs, which share the same parallel-prediction structure and position-wise CE objective, are similarly sensitive to small positional shifts under iterative decoding has not been examined.

We test this directly under open-ended generation, the setting aligned to NAT's task form, where positional shifts have the most room to accumulate over many denoising steps. We apply a minimal positional disturbance during decoding: we swap an unmasked token with an adjacent masked position, so token identities are preserved and only a one-position shift is introduced at the time of intervention. This controlled intervention is diagnostic: it isolates sensitivity to small positional shifts under irreversible decoding rather than modeling all naturally occurring errors. On LLaDA-8B-Instruct [42] with Arena-Hard [34], shifting as little as 1% of generated tokens by a single position drops the win rate against the unintervened model well below 50%. This suggests that MDLMs are sensitive to small shifts under iterative parallel decoding.

Motivated by this finding, we adapt a modified form of connectionist temporal classification (CTC) [16], an alignment-flexible training objective, to the MDLM SFT setting. CTC has already been applied to NAT [36, 51] and demonstrably mitigates this sensitivity there. The flexibility comes from a blank symbol: CTC lets the model emit blanks alongside content tokens, and treats output sequences that differ only in blank placement as equivalent. A content token assigned to a nearby position is then absorbed by an adjacent blank without penalty. We instantiate the blank as a special <SLACK> token and add the CTC objective as an auxiliary loss alongside the standard CE loss during supervised fine-tuning (SFT); <SLACK> tokens are stripped from outputs after decoding. To preserve target surface forms, we also modify standard CTC's collapse rule, which merges consecutive repeated content tokens and would otherwise distort outputs (collapsing "**" into "*" or "100" into "10"), so it removes only blank symbols.

Applied to LLaDA-8B-Instruct, our method improves over both the original model and a matched CE baseline on all four tested open-ended generation benchmarks (Arena-Hard, MT-Bench, WildBench and Creative Writing Bench), with gains that are statistically significant on all four. Our contribution is to bridge NAT and MDLM in the SFT setting: motivated by their shared parallel-CE structure, we empirically show that training-side alignment flexibility known to help NAT also helps MDLMs. In contrast to prior work on MDLM decoding failures, which has mostly operate at inference time [59, 58], these results identify alignment flexibility as a complementary training-side design dimension for MDLM SFT.

2 Masked Diffusion Language Models

MDLMs define a probability distribution over sequences through a forward noising process and a learned reverse denoising process. Let $x_0 = (x_0^1, \ldots, x_0^L)$ be a token sequence, whose length is L, and contain no masks. The forward process independently masks each token with a probability given by a strictly decreasing function $\alpha_t \in [0, 1]$ in t. Applying masks to the tokens with probability $1 - \alpha_t$ results in a partially masked sequence x_t. When $t = 1$, all tokens are masked; when $t = 0$, the sequence remains unchanged.

2.1 Training

Pre-training. The reverse process is parameterized by a mask predictor $P_\theta(\cdot \mid x_t)$, which takes the partially masked sequence x_t as input and predicts all masked tokens in parallel. Let the indicator $\mathbf{1}_{[x_t^i = \mathrm{M}]}$ be 1 if the i-th token in x_t, namely x_t^i, is masked (represented by $x_t^i = \mathrm{M}$), and 0 otherwise. Then, training proceeds by sampling t from a uniform distribution $\mathcal{U}[0, 1]$, i.e., $t \sim \mathcal{U}[0, 1]$, constructing x_t via the forward process, and minimizing a masked cross-entropy loss evaluated only

at masked positions; thus, the pre-training objective is:

$$\mathcal{L}_{\theta,t}^{\text{CE}} \triangleq -\mathbb{E}_{t,x_0,x_t}\left[\gamma_t \sum_{i=1}^{L} \mathbf{1}_{[x_t^i=\text{M}]} \log P_\theta(x_0^i \mid x_t)\right], \tag{1}$$

where $\gamma_t = -\frac{\alpha_t'}{1-\alpha_t}$, and α_t' represents the derivative of α_t with respect to t.[3]

Supervised fine-tuning. SFT adapts a pre-trained MDLM to instruction-following using paired data (q_0, r_0), where q_0 denotes the prompt and r_0 the corresponding response. The objective is to model the conditional distribution $P_\theta(r_0 \mid q_0)$. Similar to pre-training, each token in r_0 is independently masked with probability α_t, as explained above, a monotonically decreasing function of t, to obtain r_t, while q_0 is kept intact. The mask predictor conditions on the concatenation of q_0 and r_t to predict masked response tokens. We then redefine the pre-training objective shown in Eq. (1) as the SFT objective as follows:

$$\mathcal{L}_{\theta,t}^{\text{CE}} = -\mathbb{E}_{t,q_0,r_0,r_t}\left[\gamma_t \sum_{i=1}^{L_r} \mathbf{1}_{[r_t^i=\text{M}]} \psi_\theta(r_0^i, q_0, r_t)\right], \tag{2}$$

where L_r denotes the response length, and $\psi_\theta(r_0^i, q_0, r_t) = \log P_\theta(r_0^i \mid q_0, r_t)$

Relation between pre-training and SFT. Eq. (2) can also cover the pre-training objective if we view pre-training input token sequences as never including a prompt part q_0 but only a response part; namely, q_0 is always an empty sequence and $x_t = r_t$.

2.2 Inference

After training, text generation discretizes the reverse diffusion process to sample from the conditional distribution $P_\theta(r_0 \mid q_0)$. Following standard MDLM decoding (e.g., LLaDA), we fix the response length L_r as a hyperparameter and initialize generation of r_t at $t = 1$, yielding a fully masked response sequence, i.e.,

$$r_1 = (\texttt{<MASK>}, \ldots, \texttt{<MASK>}) \in (\mathcal{V} \cup \{\texttt{<MASK>}\})^{L_r}, \tag{3}$$

where $\mathcal{V}$ denotes the model's vocabulary. At an intermediate timestep t (from $t = 1$ to $t = 0$), all masked tokens are predicted simultaneously and independently. After prediction, a subset of the newly revealed tokens is re-masked at each step to ensure that the overall masking ratio follows the prescribed noise schedule. Following LLaDA, the low-confidence remasking [6] is adopted; tokens with lower prediction confidence are selected for re-masking, and tokens that remain unmasked are carried forward and not re-masked in subsequent steps, making this process irreversible.

MDLMs' generation are known to be sensitive to inference setting such as generation length, generation steps, denoising schedule, and masking strategy [42, 71], the focus of this work is to demonstrate the influence of misplaced token could cause and the effectiveness of relaxing it, not to optimize for the best absolute score, therefore we adopt a fixed inference configuration across all experiments: response length $L = 1024$, $N = 512$ diffusion steps, and the standard low-confidence remasking strategy used in LLaDA. This setup ensures that any observed differences are attributable to the applied training changes rather than to decoding hyperparameters.

3 Motivation: Sensitivity to Positional Misalignment

Standard MDLM training builds positional rigidity into the objective at two interfaces. First, each prediction is aligned to its target at a fixed output position. Second, the context that surrounds each masked token is assumed to be gold text at gold positions. In NAT, single-pass decoding means only the first interface is in play, and at that interface CE-trained decoders have been argued to be sensitive to small positional shifts [13].

In contrast, MDLMs are exposed to both interfaces and decode iteratively. A small positional shift introduced at one step then enters the conditioning context for every later step, where it is treated as gold. We study two questions: whether such positional inconsistencies arise naturally during standard decoding, and whether introducing them causally degrades generation quality.

[3] See detailed explanation in [52].

3.1 Setup

We run experiments on LLaDA-8B-Instruct [42], the most widely used open MDLM trained from scratch under the standard masked diffusion objective described in § 2, where supervision at position i targets the token at position i (cf. Eq. (2)). This makes LLaDA a clean testbed for isolating the positional sensitivity that the masked diffusion objective itself induces[4]. We use the 750 prompts from Arena-Hard-v2.0 [35, 34] and generate one standard, unintervened response trajectory for each prompt. The two studies below share only this setup: the first reads these trajectories directly, and the second intervenes on them and compares against them.

3.2 Naturally Occurring Positional Regret

Metric. We record the model's top-1 prediction at every response position and decoding step in the unintervened trajectories. Let token a become permanently committed at position i at step c, and let $j \in \{i-1, i+1\}$. We count an adjacent later-regret event when two conditions hold: (i) at step c, a is the top-1 prediction at i but not at j; and (ii) at the earliest qualifying later step $u > c$, a is no longer top-1 at i but is top-1 at j. To rule out ordinary adjacent repetitions, we require the final token at j to differ from a, and we count each directed pair (i, j) only once.

Results. Among the 750 unintervened trajectories, 342 responses (45.6%) contain at least one adjacent later-regret event. On average, affected tokens are committed at 48.9% of response progress, while the matching regret signal appears at 72.2%, after much more surrounding context has been revealed. Thus, prediction patterns consistent with adjacent positional displacement arise naturally during standard MDLM decoding. This observational result, however, does not establish whether such displacement affects final generation quality.

3.3 Effect of Small Positional Shifts

Shift intervention. To test whether such positional displacements affect generation quality, we intervene at fixed time steps during decoding. At each such step, we uniformly sample K locations where the current token has already been unmasked and a neighboring position is still `<MASK>`, and we swap each such pair. Because the swap only exchanges a revealed token with an adjacent masked slot, it keeps the revealed token itself; the only change is a one-position displacement. We apply this procedure every 0.05 diffusion time units for $t \geq 0.5$, which yields 10 intervention events within the 512-step generation. Each event shifts exactly K tokens, so in total $10 \times K$ out of 1024 tokens are intervened. We call this the *shifted-token fraction*, and we sweep K to vary the intervention strength.

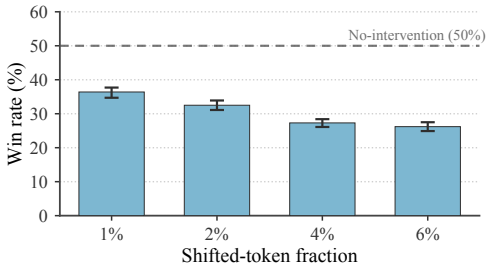

Figure 1: Win rate (%) on Arena-Hard of intervened LLaDA-8B-Instruct generations *against their non-intervened counterparts*. The dashed line at 50% marks the no-intervention reference. Error bars indicate 95% confidence intervals.

Evaluation. For each prompt, we compare the intervened response against its unintervened counterpart, following the official Arena-Hard protocol and reporting the overall win rate. This paired comparison isolates the effect of the positional intervention rather than the absolute generation quality. We use GPT-5.4 as the judge model, and we decode deterministically to eliminate the judge variance that comes from the default configuration.

Results. Figure 1 reports win rates under the controlled shift intervention. Even when only a small fraction of generated tokens (1%–6%) is displaced by one position, the win rate falls well below the

[4]We leave out Dream [63], another commonly used open MDLM, because it is initialized from an autoregressive model and uses a shift-operation training scheme that supervises the hidden state at position i to predict position $i+1$, carrying a left-to-right positional bias forward.

50% no-intervention reference. The effect appears at shifted-token fractions as low as 1%, showing that MDLMs are sensitive to small positional shifts under iterative parallel decoding[5].

Together, these results show that positional inconsistencies arise during standard decoding, and that controlled cases of such displacement measurably harm generation quality. This motivates us to explore relaxing CE's positional rigidity on the training side, as a design lever for MDLM SFT. In the NAT literature, this sensitivity has been addressed by replacing position-wise CE with an alignment-flexible objective such as CTC [51, 36]. We adapt this approach to the MDLM SFT setting in § 4, with modifications that fit the MDLM training pipeline and preserve target surface forms.

4 Method: Relaxing Positional Alignment

As motivated in § 3, we adapt a modified form of CTC [16] to the MDLM SFT setting as an auxiliary objective alongside the standard MDLM cross-entropy loss. Standard CTC allows the model to emit a longer latent alignment containing special blank symbols and repeated tokens, and recovers the target via a collapse map that removes blanks and merges adjacent duplicates. In our setting, however, merging adjacent duplicates can distort target surface forms, for example, collapsing "**" into "*" or "100" into "10". We therefore modify the collapse map to remove only blank symbols.

4.1 Vanilla CTC objective

Under the standard CTC conditional independence assumption, the alignment probability factorizes as:

$$P^{\text{CTC}}(a \mid q_0, \tilde{r}_t) = \prod_{i=1}^{\tilde{L}} P_\theta(a^i \mid q_0, \tilde{r}_t). \tag{4}$$

The CTC likelihood of the clean target response r_0 is obtained by marginalizing over all alignments that collapse to r_0:

$$P^{\text{CTC}}_\beta(r_0 \mid q_0, \tilde{r}_t) = \sum_{a \in \beta^{-1}(r_0)} P^{\text{CTC}}(a \mid q_0, \tilde{r}_t), \tag{5}$$

where β is a collapse map and $\beta^{-1}(r_0)$ denotes the set of alignments that collapse to r_0 under β. Standard CTC uses β_{std}, which first replaces each maximal run of identical symbols in a with a single symbol and then removes all <SLACK> symbols. The corresponding training objective is:

$$\mathcal{L}^{\text{CTC}}_{\theta,t} = -\mathbb{E}_{t,\, q_0,\, r_0,\, \tilde{r}_t}\left[\log P^{\text{CTC}}_{\beta_{\text{std}}}(r_0 \mid q_0, \tilde{r}_t)\right]. \tag{6}$$

4.2 Modified CTC Objective

Our method has two components: blank-augmented training data and a modified CTC objective.

Data augmentation. We construct the data using <SLACK> token as blank symbol needed in CTC object: $\tilde{r}_0 \in (\mathcal{V} \cup \{\text{<SLACK>}\})^{\tilde{L}}$ from the gold response $r_0 \in \mathcal{V}^L$ by inserting monotonic linear decreasing <SLACK> ratio with t, to match the monotonic increasing nature of <SLACK> tokens during the generation process. Specifically, we implement the <SLACK> ratio $s = s_{max}(1-t)$ with $s_{max} = 0.5$, so the slack ratio reaches at most 0.5 at $t = 0$, to prevent <SLACK> dominating the generated text.

CTC-S: <SLACK>-only collapse. As noted in Eq. (5), the duplicate-merging step in β_{std} can distort target surface forms (e.g., "**" $\rightarrow$ "*", "100" $\rightarrow$ "10"), which is problematic for text generation. We therefore define a modified collapse map β_{slack} that removes only <SLACK> symbols and preserves adjacent duplicates in a. Substituting β_{slack} for β_{std} in Eq. (5) yields our **CTC-S** objective:

$$\mathcal{L}^{\text{CTC-S}}_{\theta,t} = -\mathbb{E}_{t,\, q_0,\, r_0,\, \tilde{r}_t}\left[\log P^{\text{CTC}}_{\beta_{\text{slack}}}(r_0 \mid q_0, \tilde{r}_t)\right]. \tag{7}$$

CTC-S retains the alignment flexibility of standard CTC while preserving target surface forms, and is the auxiliary objective we use throughout the rest of the paper[6].

[5] Our trained model stays robust under the same intervention (§ 6), which indicates that the drop is not caused simply by linguistic corruption from the token swap.

[6] We provide algorithm on CTC-S objective in Appendix A

4.3 Combined Objective

We redefine the SFT objective to incorporate the CTC-S objective into SFT as follows:

$$\mathcal{L}_{\theta,t}^{\text{SFT}} = \mathcal{L}_{\theta,t}^{\text{CE}} + \lambda \mathcal{L}_{\theta,t}^{\text{CTC-S}}, \tag{8}$$

where λ controls the contribution of the CTC-S term. The CTC-S term is computed against the original slack-free target r_0, as explained in Eq. (6), whereas the CE term is computed over the slack-augmented target $\tilde{r}_0$. Moreover, both CTC-S and CE terms use the corrupted response sequence $\tilde{r}_t$, which is derived from $\tilde{r}_0$. Therefore, when computing the CE term, we compute Eq. (2) by substituting r_t and r_0 with $\tilde{r}_t$ and $\tilde{r}_0$, respectively.

This encourages the model to emit <SLACK> tokens in $\tilde{L}$ predictions when needed to accommodate positional uncertainty, while keeping the supervised output content equal to r_0 after collapse.

5 Experiments

To demonstrate the effectiveness of our method, we conduct experiments on four open-ended generation benchmarks.

5.1 Experimental Setup

Data preparation. We used the official filtered Magpie-Pro dataset [61], which contains approximately 300k high-quality instruction-following examples synthesized from Llama-3-70B-Instruct [14]. To balance computational cost and the effect of the CTC-S objective, we further filter the dataset by response length, retaining examples with responses between 512 and 1024 tokens under the LLaDA tokenizer. This results in approximately 291k training samples.

Models. We train three models on top of LLaDA-8B-Instruct (LLaDA hereafter) [42] to isolate the contributions of our method. The *CE-Only* model is trained with the standard MDLM cross-entropy objective on our Magpie subset, serving as a baseline that controls for the effect of additional SFT data. The *Jitter* model is trained with the same objective on the <SLACK>-augmented data, exposing the model to position-shifted contexts under hard position-wise supervision. The *CE + CTC-S* model (CTC-S hereafter) is trained with the combined CE + CTC-S objective on the same slack-augmented data; this is the only model that trained with CTC-S. All other training settings are identical.[7].

Benchmarks. We evaluate open-ended generation quality on four benchmarks: Arena-Hard-v2.0 [35, 34], MT-Bench [69], WildBench [38], and Creative Writing Bench v3 [45]. Note that for Arena-Hard-v2.0, it uses relative win rates against a predefined baseline. We replace the standard o3 baseline with CE-Only model, as LLaDA achieve near-zero win rates against o3, leaving no signal to discriminate among variants. Results are intended for controlled comparison in this work and are not leaderboard-comparable.

Metrics. For Arena-Hard, we include the bootstrapped 95% confidence interval following the official setting. Additionally for each benchmark, we performed paired-bootstrap comparisons between every pair of models on the evaluation instances scored for both models. We resampled the aligned instance-level score differences with replacement for 10,000 bootstrap iterations and computed the mean score difference and its 95% confidence interval. We regard a comparison as statistically significant when the confidence interval excludes zero[8].

5.2 Results

Overall. Table 1 reports results on all tested benchmarks. We read the table as a three-step ladder, where each step isolates one factor: (i) additional SFT data (CE-Only vs. LLaDA), (ii) data augmentation under hard alignment (Jitter vs. CE-Only), and (iii) flexible positional alignment (CTC-S vs. Jitter). At a high level, CE-Only consistently outperforms the original LLaDA model, confirming that the training pipeline and data are healthy; and CTC-S achieves the best score on every benchmark, with significant gains over both Jitter (‡) and CE-Only (†) on all four benchmarks.

[7]We describe the further implementation details in Appendix B

[8]Full pairwise confidence intervals are reported in Appendix C

Table 1: Performance comparison on open-ended text generation benchmarks. For Arena-Hard, we report bootstrapped 95% confidence intervals following the official setting. † and ‡ indicate that CE + CTC-S is significantly better than the CE-Only model and the Jitter model, respectively (paired bootstrap, $p < 0.05$). Bold indicates the best score in each column.

	Arena-Hard*	**Creative-Writing-Bench v3**	**MT-Bench**	**Wild-Bench**
LLaDA	43.7 $(-1.1/+1.4)$	23.2	2.84	-5.48
CE-Only	50.0 $(-0.0/+0.0)$	24.6	3.44	-4.53
Jitter	48.4 $(-1.4/+1.5)$	25.1	3.61	-4.46
CE + CTC-S	$\mathbf{55.1}^{\dagger\ddagger}$ $(-1.7/+1.3)$	$\mathbf{27.7}^{\dagger\ddagger}$	$\mathbf{4.35}^{\dagger\ddagger}$	$\mathbf{-4.23}^{\dagger\ddagger}$

Contribution of <SLACK> augmentation. Comparing the Jitter model and the CE-Only model isolates the effect of <SLACK> augmentation in our method. The Jitter model's results are mixed: on Arena-Hard, its confidence interval almost overlaps with that of the CE-Only model, while it shows moderate gains on the remaining benchmarks. One possible reading is that random <SLACK> tokens act primarily as a source of variation on the CE objective: the <SLACK> augmentation exposes the model to position-shifted contexts, but the objective itself still imposes hard position-wise alignment. We do not test this directly.

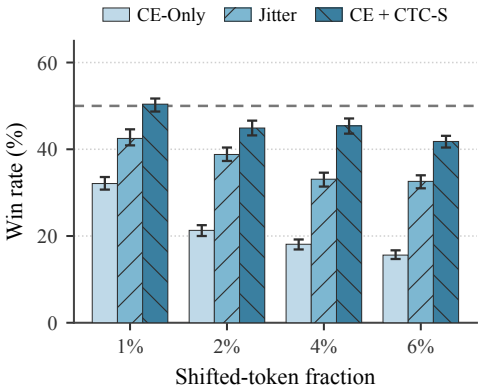

Figure 2: Robustness check under the controlled shift intervention. Win rate (%) on Arena-Hard of intervened model generations against their non-intervened counterparts. The dashed line at 50% marks the no-intervention reference. Error bars indicate 95% confidence intervals calculated following the standard Arena-Hard setting.

Contribution of CTC-S objective. Further comparing the CTC-S model and the Jitter model isolates the effect of alignment-flexible objective. The CTC-S model is significantly better than the Jitter model (‡) on all four tested benchmarks. This suggests that the alignment-flexible objective itself contributes to the improvement on open-ended generation, beyond what the slack-augmented data alone provides.

Takeaway. Across the three runs, the gain is concentrated at the CTC-S step, the only step that introduces alignment flexibility into the objective. This identifies training-side alignment flexibility as a useful design dimension for MDLM SFT[9], complementary to the inference-time interventions explored in prior work.

6 Discussion

Beyond the main results in § 5, several natural follow-up questions remain. (i) Since CTC-S targets the positional shift sensitivity, does it deliver the robustness property it was designed for? (ii) How does the CTC-S model use the <SLACK> token during generation? (iii) Does relaxing positional alignment come at the cost of general capabilities? (iv) How sensitive is the method to the maximum slack ratio $s_{\max}$? We address these in turn[10].

[9] Appendix D reports the same experiment on LLaDA-MoE.

[10] We also provide a transferability check on LLaDA-MoE and a comparison with inference-time block diffusion in Appendix D and Appendix E.

6.1 Robustness check under the controlled shift intervention

CTC-S targets robustness to the controlled shift intervention from §3, which we test here. We repeat the intervention for the CE-Only, Jitter, and CTC-S models; Figure 2 summarizes the results.

The CE-Only model shows nearly the same sensitivity as the original LLaDA (Figure 1). Jitter sits below the 50% no-intervention reference at all tested fractions; this indicates that slack augmentation closes the gap by a large margin, though not entirely. CTC-S substantially more robust than CE-Only and Jitter, therefore exhibits the robustness property it was designed for.

6.2 <SLACK> Token Behavior

We examine <SLACK> usage in the CTC-S model's MT-Bench generations along two axes: overall emission rate and positional distribution.

<SLACK> usage is selective across prompts. Although every training response contains <SLACK> tokens, only 83 of 160 MT-Bench samples (51.9%) emit any <SLACK> at inference. This suggests the model learned to treat <SLACK> as an optional resource rather than a required output feature.

<SLACK> emission rate is close to the training augmentation rate. Among samples that emit <SLACK>, it accounts for 27.5% of generated tokens on average, comparable to the 25% per-sample augmentation rate used during training. This suggests the model did not drift toward over- or under-emission relative to its training signal.

<SLACK> is mainly allocated to inter-word positions. <SLACK> is injected at random positions during training, including inside subword sequences. At inference, however, only 4.9% of emitted <SLACK> tokens fall at subword-internal positions[11]. Without explicit supervision, the model concentrates <SLACK> at inter-word boundaries rather than within subwords. This is consistent with the alignment-flexibility view: linguistically, positional uncertainty is most meaningful between words, where multiple lexicalizations and orderings are possible, but not within a word, since subword pieces are an artifact of tokenization rather than linguistic units and the boundaries between them carry no independent meaning.

6.3 General-Capability Benchmarks

To confirm that CTC-S does not harm capabilities beyond open-ended generation, we report results on GPQA [49], MMLU [21], MBPP [2], IFEval [70], GSM8K [7], and MATH500 [22, 37]. We use LLaDA's official OpenCompass evaluation and match its generation and evaluation settings.[12] The CTC-S vs. CE-Only comparison isolates the effect of our objective from the effect of additional SFT data.

Table 2: Performance on general capability benchmarks. Bold indicates the best score among models trained on our dataset (CE-Only, Jitter, CTC-S); the LLaDA column is included as a reference. * indicates scores taken from the official repository.

	Trained			Reference
	CE-Only	**Jitter**	**CTC-S**	**LLaDA**
GPQA	29.8	29.8	**31.3**	32.3*
MMLU	62.1	**63.8**	63.4	65.4*
MBPP	36.2	**37.2**	36.8	39.6*
IFEval	63.2	65.0	**66.9**	65.2*
GSM8K	66.1	**69.9**	68.9	68.8*
MATH500	26.6	**27.2**	26.0	31.4

Capability regression is data-driven, not objective-driven. All three trained models regress modestly relative to LLaDA on GPQA, MMLU, MBPP, and MATH500 (Table 2); the regression is shared across variants, indicating that it stems primarily from the SFT data rather than the CTC-S objective, consistent with the Magpie paper's observation [61] that Magpie-Pro underperforms on reasoning- and coding-heavy benchmarks.

CTC-S preserves base capabilities where the data permits. CTC-S matches or slightly exceeds the LLaDA reference on both (66.9 vs. 65.2 on IFEval; 68.9 vs. 68.8 on GSM8K). Relative to

[11] A <SLACK> position is classified as subword-internal when the following token does not begin with a leading space and is not punctuation; sequence-initial positions are excluded.

[12] See `https://github.com/ML-GSAI/LLaDA/blob/main/EVAL.md` for details.

Table 3: Slack-ratio ablation results. For Arena-Hard, we report bootstrapped 95% confidence intervals following the official setting. †/§ indicate significantly better/worse than the CE-Only reference (paired bootstrap, $p < 0.05$). Bold indicates the best score in each column.

	Arena-Hard*	Creative-Writing-Bench v3	MT-Bench	Wild-Bench
CE-Only (reference)	50.0 $(-0.0/+0.0)$	24.6	3.44	-4.53
CE + CTC-S, $s_{\max} = 0.2$	$45.1^{\S}$ $(-1.3/+1.5)$	$27.31^{\dagger}$	$3.96^{\dagger}$	-4.50
CE + CTC-S, $s_{\max} = 0.5$	$\mathbf{55.1}^{\dagger}$ $(-1.7/+1.3)$	$\mathbf{27.70}^{\dagger}$	$4.35^{\dagger}$	$-4.23^{\dagger}$
CE + CTC-S, $s_{\max} = 0.8$	49.9 $(-1.5/+1.7)$	$27.68^{\dagger}$	$\mathbf{4.41}^{\dagger}$	$\mathbf{-4.15}^{\dagger}$

CE-Only, CTC-S matches or improves on five of the six benchmarks, with a marginal gap only on MATH500 (26.0 vs. 26.6). Relaxing positional alignment therefore does not come at the cost of broader capabilities: on the two benchmarks where the SFT data does not impose a ceiling, CTC-S preserves the base model's performance.

6.4 Slack-Ratio Ablation

To test how sensitive our method is to the slack ratio $s_{\max}$, we train two additional CE + CTC-S variants with $s_{\max} = 0.2$ and $s_{\max} = 0.8$ under the same training and evaluation setup as §5.

Table 3 shows that the method remains effective across this range: every setting improves significantly over CE-Only on Creative-Writing-Bench and MT-Bench. $s_{\max} = 0.5$ gives the best overall balance, while $s_{\max} = 0.8$ scores slightly higher on MT-Bench and WildBench. The only significant regression in the table is at the smallest setting, where $s_{\max} = 0.2$ falls below CE-Only on Arena-Hard; neither of the larger settings is significantly worse than CE-Only on any benchmark.

7 Related Work

7.1 Prior Approaches to MDLM Decoding Failures

Prior work has identified several failure modes in the fixed-length, position-wise decoding setup of MDLMs and proposed a range of remedies. One line enables variable-length generation through token-position adjustment, end-of-sequence prediction, or insertion/deletion operations, addressing failures arising from fixed canvases and uncertain span boundaries [66, 59, 62, 29, 20, 33]. A second line improves which tokens to commit at each step, through position-aware or guidance-based samplers and learned unmasking-order strategies [46, 25, 31, 27]. A third line targets early commitment by allowing already-revealed tokens to be revised, either at inference through re-masking and refinement [58, 47] or through training-side objectives that teach self-correction [26, 30, 67].

Our work differs in what it changes: rather than adjusting the canvas, the unmasking order, or the model's ability to revise committed tokens, we relax position-wise supervision during training, keeping the standard MDLM decoding setup unchanged.

7.2 Non-autoregressive Translation

Non-autoregressive translation [19] replaces left-to-right decoding with parallel prediction under a conditional independence assumption. Because a single source has many valid translations, a parallel decoder trained with position-wise cross-entropy receives gradient signal that mixes incompatible modes at the same position. Position-wise CE penalizes a correct token at the wrong position as harshly as a wrong token, a sensitivity to positional shifts that the NAT literature has repeatedly identified [13, 10, 11]. MDLMs inherit this setup, which motivates studying the same sensitivity.

NAT has developed a family of training-side responses to relax this constraint. CTC is one established line: first applied to NAT [36], paired with an iterative variant [5] to close the gap to AR baselines on WMT [51], and identified as an essential ingredient for a competitive fully-NAT system [18]. Other approaches reach similar gains through best-alignment objectives, marginalization over alignment paths, or curricula that progressively reveal target tokens [10, 11, 24, 48].

Our work transfers this training-side ingredient to MDLM SFT through CTC, whose blank-augmented alignment space and forward–backward marginalization let it absorb positional uncertainty within MDLMs' fixed decoding canvas.

7.3 Alignment-Flexible Objectives and Structured Supervision

CTC [16] originated as a training criterion for sequence labeling with unobserved input–output alignment, most prominently in speech recognition [15, 17]. More broadly, recent work has highlighted the importance of task-specific supervision and representation design [54, 65, 64, 39, 32]. Other alignment-flexible objectives also includes stochastic edit-distance models [50, 43], semi-Markov conditional random fields [53], and differentiable dynamic time warping [9]. Within this class, CTC is distinctive in marginalizing over alignments via a blank-augmented label space and a deterministic collapse map, admitting an efficient forward–backward dynamic program. Later work combines the marginalized objective with iterative refinement [23] and extends CTC to non-speech sequence transduction, including non-autoregressive speech translation [60] and latent-variable extensions [12].

Our method adapts the canonical CTC objective to text generation.

8 Conclusion

This work studies MDLMs through the lens of non-autoregressive translation (NAT): both train parallel decoders with position-wise cross-entropy, and CE-trained parallel decoders have been argued to be sensitive to small positional shifts in the NAT setting. Through a controlled shift intervention on LLaDA-8B-Instruct, we find that displacing as little as 1% of generated tokens by a single position substantially reduces win rates on Arena-Hard, indicating that MDLMs share this sensitivity under iterative parallel decoding. Motivated by this, we adapt connectionist temporal classification (CTC), an alignment-flexible objective known to mitigate this sensitivity in NAT, to MDLM supervised fine-tuning. We introduce CTC-S, a modified form of CTC that uses a special <SLACK> token to absorb positional uncertainty between target tokens and output positions and an adjusted collapse map that preserves target surface forms. The resulting model consistently improves over both the original model and a matched cross-entropy baseline across four open-ended generation benchmarks, while preserving performance on standard capability benchmarks.

These results identify training-side alignment flexibility as a useful design dimension for MDLM supervised fine-tuning, complementary to the inference-time interventions explored in prior work. We hope this NAT–MDLM bridge encourages further work on flexible alignment objectives in MDLM training, including extension to pre-training and to other instantiations of alignment flexibility.

Limitations. The <SLACK> mechanism is one instantiation of alignment flexibility; other formulations, including non-CTC objectives and learned slack budgets, may offer further gains. Our experiments are limited to SFT stage, leaving the effect of relaxing positional alignment during pre-training unexplored. Our method operates at the training-objective level and is model-agnostic, verifying it on MDLMs trained under different schemes is left to future work. We also fix the CTC-S loss weight λ without tuning; sweeping it may yield further gains.

Impact statement. This work is methodological: we modify the SFT objective for masked diffusion language models and evaluate on standard open-ended generation benchmarks. To the extent that our approach narrows the open-ended generation gap between MDLMs and autoregressive models, it may contribute to MDLMs becoming a more practical alternative for general-purpose text generation, and inherits the broader societal considerations that apply to instruction-tuned LLMs. Our method does not introduce new capabilities beyond those of the base LLaDA-8B-Instruct model.

Ethics statement. Our experiments use only publicly released artifacts: the LLaDA-8B-Instruct model and the Magpie-Pro dataset, the latter synthesized from Llama-3-70B-Instruct. We did not collect data from human subjects, and our training data and base model inherit whatever biases are present in these upstream resources. To promote transparency and reproducibility, we will release all code used in our experiments. Comprehensive details of our experimental setup are provided in each section and the appendix to ensure reproducibility.

Acknowledgments.

We want to express our gratitude to the members of the Tohoku NLP Group for their insightful comments. We used ABCI 3.0 provided by AIST and AIST Solutions with support from "ABCI 3.0 Development Acceleration Use" for our main experiments. A part of the computation was carried out using the computer resource offered under the category of General Projects by Research Institute for Information Technology, Kyushu University. This work was supported by the "R&D Hub Aimed at Ensuring Transparency and Reliability of Generative AI Models" project of the Ministry of Education, Culture, Sports, Science and Technology; JST Moonshot R&D Grant Number JPMJMS2011-35 (fundamental research); JSPS KAKENHI Grant Numbers JP25KJ0615 and JP26H02511; JST BOOST, Japan Grant Number JPMJBS2421.

Competing interests. We declare no competing interests.

References

[1] M. Arriola, S. S. Sahoo, A. Gokaslan, Z. Yang, Z. Qi, J. Han, J. T. Chiu, and V. Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tyEyYT267x.

[2] J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, E. Jiang, C. Cai, M. Terry, Q. Le, and C. Sutton. Program synthesis with large language models, 2021. URL https://arxiv.org/abs/2108.07732.

[3] T. Bie, M. Cao, K. Chen, L. Du, M. Gong, Z. Gong, Y. Gu, J. Hu, Z. Huang, Z. Lan, C. Li, C. Li, J. Li, Z. Li, H. Liu, L. Liu, G. Lu, X. Lu, Y. Ma, J. Tan, L. Wei, J.-R. Wen, Y. Xing, X. Zhang, J. Zhao, D. Zheng, J. Zhou, J. Zhou, Z. Zhou, L. Zhu, and Y. Zhuang. Llada2.0: Scaling up diffusion language models to 100b, 2025. URL https://arxiv.org/abs/2512.15745.

[4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *The Thirty-fourth Annual Conference on Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[5] W. Chan, C. Saharia, G. Hinton, M. Norouzi, and N. Jaitly. Imputer: Sequence modelling via imputation and dynamic programming. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1403–1413. PMLR, 2020. URL https://proceedings.mlr.press/v119/chan20b.html.

[6] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman. Maskgit: Masked generative image transformer. *arXiv preprint arXiv:2202.04200*, 2022. URL https://arxiv.org/abs/2202.04200.

[7] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

[8] O. Contributors. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass, 2023.

[9] M. Cuturi and M. Blondel. Soft-dtw: a differentiable loss function for time-series. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 894–903. JMLR.org, 2017.

[10] C. Du, Z. Tu, and J. Jiang. Order-agnostic cross entropy for non-autoregressive machine translation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2849–2859. PMLR, 2021.

[11] C. Du, Z. Tu, L. Wang, and J. Jiang. ngram-OAXE: Phrase-based order-agnostic cross entropy for non-autoregressive machine translation. In *Proceedings of the 29th International Conference on Computational Linguistics*, 2022. URL https://aclanthology.org/2022.coling-1.446/.

[12] Y. Fujita, S. Watanabe, X. Chang, and T. Maekaku. Lv-ctc: Non-autoregressive asr with ctc and latent variable models. *arXiv preprint arXiv:2403.19207*, 2024. URL https://arxiv.org/abs/2403.19207.

[13] M. Ghazvininejad, V. Karpukhin, L. Zettlemoyer, and O. Levy. Aligned cross entropy for non-autoregressive machine translation. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3515–3523. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/ghazvininejad20a.html.

[14] A. Grattafiori, A. Dubey, and A. J. et.al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[15] A. Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012. URL https://arxiv.org/abs/1211.3711.

[16] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 369–376, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143891. URL https://doi.org/10.1145/1143844.1143891.

[17] A. Graves, A. rahman Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. *arXiv preprint arXiv:1303.5778*, 2013. URL https://arxiv.org/abs/1303.5778.

[18] J. Gu and X. Kong. Fully non-autoregressive neural machine translation: Tricks of the trade. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 120–133, 2021. URL https://aclanthology.org/2021.findings-acl.11/.

[19] J. Gu, J. Bradbury, C. Xiong, V. O. Li, and R. Socher. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1l8BtlCb.

[20] M. Havasi, B. Karrer, I. Gat, and R. T. Q. Chen. Edit flows: Flow matching with edit operations. *arXiv preprint arXiv:2506.09018*, 2025. URL https://arxiv.org/abs/2506.09018.

[21] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. In *The Ninth International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

[22] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

[23] Y. Higuchi, S. Watanabe, N. Chen, T. Ogawa, and T. Kobayashi. Mask ctc: Non-autoregressive end-to-end asr with ctc and mask predict. *arXiv preprint arXiv:2005.08700*, 2020. URL https://arxiv.org/abs/2005.08700.

[24] F. Huang, H. Zhou, Y. Liu, H. Li, and M. Huang. Directed acyclic transformer for non-autoregressive machine translation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*. PMLR, 2022.

[25] P. Huang, S. Liu, Z. Liu, Y. Yan, S. Wang, Z. Chen, and T. Xiao. Pc-sampler: Position-aware calibration of decoding bias in masked diffusion models, 2025. URL https://arxiv.org/abs/2508.13021.

[26] Z. Huang, Y. Wang, Z. Chen, and G.-J. Qi. Don't settle too early: Self-reflective remasking for diffusion language models. *arXiv preprint arXiv:2509.23653*, 2025. URL https://arxiv.org/abs/2509.23653.

[27] M. Jazbec, T. X. Olausson, L. Béthune, P. Ablin, M. Kirchhof, J. Monteiro, V. Turrisi, J. Ramapuram, and M. Cuturi. Learning unmasking policies for diffusion language models. *arXiv preprint arXiv:2512.09106*, 2025. URL https://arxiv.org/abs/2512.09106.

[28] S. Khanna, S. Kharbanda, S. Li, H. Varma, E. Wang, S. Birnbaum, Z. Luo, Y. Miraoui, A. Palrecha, S. Ermon, A. Grover, and V. Kuleshov. Mercury: Ultra-fast language models based on diffusion, 2025. URL https://arxiv.org/abs/2506.17298.

[29] J. Kim, L. Cheuk-Kit, C. Domingo-Enrich, Y. Du, S. Kakade, T. Ngotiaoco, S. Chen, and M. Albergo. Any-order flexible length masked diffusion. *arXiv preprint arXiv:2509.01025*, 2025. URL https://arxiv.org/abs/2509.01025.

[30] J. Kim, S. Kim, T. Lee, D. Z. Pan, H. Kim, S. Kakade, and S. Chen. Fine-tuning masked diffusion for provable self-correction. *arXiv preprint arXiv:2510.01384*, 2025. URL https://arxiv.org/abs/2510.01384.

[31] S. Lee, S. Kim, J. Park, and D. Park. Lookahead unmasking elicits accurate decoding in diffusion language models. *arXiv preprint arXiv:2511.05563*, 2025. URL https://arxiv.org/abs/2511.05563.

[32] B. Li, Y. Shen, Y. Liu, Y. Xu, J. Liu, X. Li, Z. Li, J. Zhu, Y. Zhong, F. Lan, et al. Toward cognitive supersensing in multimodal large language model. *arXiv preprint arXiv:2602.01541*, 2026.

[33] J. Li, X. Dong, Y. Zang, Y. Cao, J. Wang, and D. Lin. Beyond fixed: Variable-length denoising for diffusion large language models. *arXiv preprint arXiv:2508.00819*, 2025. URL https://arxiv.org/abs/2508.00819.

[34] T. Li, W.-L. Chiang, E. Frick, L. Dunlap, B. Zhu, J. E. Gonzalez, and I. Stoica. From live data to high-quality benchmarks: The arena-hard pipeline, April 2024. URL https://lmsys.org/blog/2024-04-19-arena-hard/.

[35] T. Li, W.-L. Chiang, E. Frick, L. Dunlap, T. Wu, B. Zhu, J. E. Gonzalez, and I. Stoica. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=KfTf9vFvSn.

[36] J. Libovický and J. Helcl. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3016–3021, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1336. URL https://aclanthology.org/D18-1336/.

[37] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

[38] B. Y. Lin, Y. Deng, K. Chandu, A. Ravichander, V. Pyatkin, N. Dziri, R. L. Bras, and Y. Choi. Wildbench: Benchmarking LLMs with challenging tasks from real users in the wild. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=MKEHCx25xp.

[39] Y. Liu, J. Zhu, Y. Mo, G. Li, X. Cao, J. Jin, Y. Shen, Z. Li, T. Yu, W. Yuan, et al. PALM: Progress-aware policy learning via affordance reasoning for long-horizon robotic manipulation. *arXiv preprint arXiv:2601.07060*, 2026.

[40] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[41] A. Lou, C. Meng, and S. Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=CNicRIVIPA.

[42] S. Nie, F. Zhu, Z. You, X. Zhang, J. Ou, J. Hu, J. Zhou, Y. Lin, J.-R. Wen, and C. Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025. URL https://arxiv.org/abs/2502.09992.

[43] J. Oncina and M. Sebban. Learning stochastic edit distance: Application in handwritten character recognition. *Pattern Recognition*, 39(9):1575–1587, 2006. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2006.03.011. URL https://www.sciencedirect.com/science/article/pii/S0031320306001245.

[44] J. Ou, S. Nie, K. Xue, F. Zhu, J. Sun, Z. Li, and C. Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sMyXP8Tanm.

[45] S. J. Paech. Eq-bench creative writing benchmark v3. https://github.com/EQ-bench/creative-writing-bench, 2025.

[46] D. Patel, T. Naseem, G. Pandey, M. A. Sultan, A. McCallum, and R. F. Astudillo. Improved sampling from masked diffusion models with position contrastive guidance. In *NeurIPS 2025 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2025. URL https://openreview.net/forum?id=e0WmOrWbtc.

[47] F. Z. Peng, Z. Bezemek, S. Patel, J. Rector-Brooks, S. Yao, A. J. Bose, A. Tong, and P. Chatterjee. Path planning for masked diffusion model sampling. *arXiv preprint arXiv:2502.03540*, 2025. URL https://arxiv.org/abs/2502.03540.

[48] L. Qian, H. Zhou, Y. Bao, M. Wang, L. Qiu, W. Zhang, Y. Yu, and L. Li. Glancing transformer for non-autoregressive neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1993–2003, 2021. URL https://aclanthology.org/2021.acl-long.155/.

[49] D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

[50] E. Ristad and P. Yianilos. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522–532, 1998. doi: 10.1109/34.682181.

[51] C. Saharia, W. Chan, S. Saxena, and M. Norouzi. Non-autoregressive machine translation with latent alignments. In B. Webber, T. Cohn, Y. He, and Y. Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1098–1108, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.83. URL https://aclanthology.org/2020.emnlp-main.83/.

[52] S. S. Sahoo, M. Arriola, A. Gokaslan, E. M. Marroquin, A. M. Rush, Y. Schiff, J. T. Chiu, and V. Kuleshov. Simple and effective masked diffusion language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=L4uaAR4ArM.

[53] S. Sarawagi and W. W. Cohen. Semi-markov conditional random fields for information extraction. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004. URL https://proceedings.neurips.cc/paper_files/paper/2004/file/eb06b9db06012a7a4179b8f3cb5384d3-Paper.pdf.

[54] Y. Shen, Y. Liu, J. Zhu, X. Cao, X. Zhang, Y. He, W. Ye, J. Rehg, and I. Lourentzou. Fine-grained preference optimization improves spatial reasoning in vlms. In *Advances in Neural Information Processing Systems*, volume 38, pages 17929–17960, 2025.

[55] J. Shi, K. Han, Z. Wang, A. Doucet, and M. Titsias. Simplified and generalized masked diffusion for discrete data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=xcqSOfHt4g.

[56] Y. Song, Z. Zhang, C. Luo, P. Gao, F. Xia, H. Luo, Z. Li, Y. Yang, H. Yu, X. Qu, Y. Fu, J. Su, G. Zhang, W. Huang, M. Wang, L. Yan, X. Jia, J. Liu, W.-Y. Ma, Y.-Q. Zhang, Y. Wu, and H. Zhou. Seed diffusion: A large-scale diffusion language model with high-speed inference, 2025. URL https://arxiv.org/abs/2508.02193.

[57] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[58] G. Wang, Y. Schiff, S. Sahoo, and V. Kuleshov. Remasking discrete diffusion models with inference-time scaling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=IJryQAOy0p.

[59] Z. Wu, L. Zheng, Z. Xie, J. Ye, J. Gao, Y. Feng, Z. Li, V. W., G. Zhou, and L. Kong. Dreamon: Diffusion language models for code infilling beyond fixed-size canvas, 2025. URL https://hkunlp.github.io/blog/2025/dreamon.

[60] C. Xu, X. Liu, X. Liu, Q. Sun, Y. Zhang, M. Yang, Q. Dong, T. Ko, M. Wang, T. Xiao, A. Ma, and J. Zhu. CTC-based non-autoregressive speech translation. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13321–13339, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.744. URL https://aclanthology.org/2023.acl-long.744/.

[61] Z. Xu, F. Jiang, L. Niu, Y. Deng, R. Poovendran, Y. Choi, and B. Y. Lin. Magpie: Alignment data synthesis from scratch by prompting aligned LLMs with nothing. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Pnk7vMbznK.

[62] Y. Yang, C. Wang, S. Wang, Z. Wen, B. Qi, H. Xu, and L. Zhang. Diffusion llm with native variable generation lengths: Let [eos] lead the way. *arXiv preprint arXiv:2510.24605*, 2025. URL https://arxiv.org/abs/2510.24605.

[63] J. Ye, Z. Xie, L. Zheng, J. Gao, Z. Wu, X. Jiang, Z. Li, and L. Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025. URL https://arxiv.org/abs/2508.15487.

[64] T. Yu, X. Li, Y. Shen, Y. Liu, and I. Lourentzou. CoRe3D: Collaborative reasoning as a foundation for 3d intelligence. *arXiv preprint arXiv:2512.12768*, 2025.

[65] T. Yu, X. Li, Y. Shen, O. Susladkar, Y. Liu, X. Zhou, and I. Lourentzou. ELSA3D: Elastic semantic anchoring for unified 3d understanding and generation. *arXiv preprint arXiv:2607.06565*, 2026.

[66] A. Zhang, A. Sivakumar, C.-W. Tang, and C. Thomas. Flexible-length text infilling for discrete diffusion models. In C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 31344–31359, Suzhou, China, Nov. 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1597. URL https://aclanthology.org/2025.emnlp-main.1597/.

[67] S. Zhang, F. Z. Peng, Y. Zhang, J. Pan, and G. G. Chrysos. Corrective diffusion language models. *arXiv preprint arXiv:2512.15596*, 2025. URL https://arxiv.org/abs/2512.15596.

[68] S. Zhao, D. Gupta, Q. Zheng, and A. Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=7ZVRlBFuEv.

[69] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=uccHPGDlao.

[70] J. Zhou, T. Lu, S. Mishra, S. Brahma, S. Basu, Y. Luan, D. Zhou, and L. Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

[71] F. Zhu, R. Wang, S. Nie, X. Zhang, C. Wu, J. Hu, J. ZHOU, J. Chen, Y. Lin, J.-R. Wen, and C. Li. LLaDA 1.5: Variance-reduced preference optimization for large language diffusion models. *arXiv preprint arXiv:2505.19223*, 2025. URL https://arxiv.org/abs/2505.19223.

[72] F. Zhu, Z. You, Y. Xing, Z. Huang, L. Liu, Y. Zhuang, G. Lu, K. Wang, X. Wang, L. Wei, H. Guo, J. Hu, W. Ye, T. Chen, C. Li, C. Tang, H. Feng, J. Hu, J. Zhou, X. Zhang, Z. Lan, J. Zhao, D. Zheng, C. Li, J. Li, and J.-R. Wen. Llada-moe: A sparse moe diffusion language model, 2025. URL https://arxiv.org/abs/2509.24389.

Algorithm 1 CTC-S forward pass per example.

Require: Log-probabilities $\log P \in \mathbb{R}^{\tilde{L}\times|\mathcal{V}|}$ at response positions, where $\log P[t, v] = \log P_\theta(v \mid q_0, \tilde{r}_t)$; clean target $r_0 \in \mathcal{V}^L$.
Ensure: CTC-S loss $\mathcal{L}^{\text{CTC-S}}$.

$\ell \leftarrow$ (<SLACK>, r_0^1, <SLACK>, r_0^2, …, r_0^L, <SLACK>) ▷ Extended label sequence of length $2L{+}1$
$\alpha[1, 0] \leftarrow \log P[1, \text{<SLACK>}]$ ▷ Initialization at $t = 1$
$\alpha[1, 1] \leftarrow \log P[1, r_0^1]$ if $L > 0$, else $-\infty$
$\alpha[1, s] \leftarrow -\infty$ for $s \in \{2, \ldots, 2L\}$
for $t = 2$ to $\tilde{L}$ **do** ▷ Forward recurrence
  for $s = 0$ to $2L$ **do**
    $a_{\text{self}} \leftarrow \alpha[t{-}1, s]$ if s even, else $-\infty$ ▷ M1
    $a_{\text{adv}} \leftarrow \alpha[t{-}1, s{-}1]$ if $s \geq 1$, else $-\infty$
    $a_{\text{skip}} \leftarrow \alpha[t{-}1, s{-}2]$ if s odd and $s \geq 2$, else $-\infty$ ▷ M2
    $\alpha[t, s] \leftarrow \text{LOGSUMEXP}(a_{\text{self}}, a_{\text{adv}}, a_{\text{skip}}) + \log P[t, \ell_s]$
  end for
end for
$\mathcal{L}^{\text{CTC-S}} \leftarrow -\text{LOGSUMEXP}\big(\alpha[\tilde{L}, 2L], \alpha[\tilde{L}, 2L{-}1]\big)$ ▷ Termination
return $\mathcal{L}^{\text{CTC-S}}$

A CTC-S Forward Pass

This section describes how we compute the CTC-S loss $\mathcal{L}^{\text{CTC-S}}_{\theta,t}$ defined in Eq. (7) via a modified forward recurrence. The procedure parallels the standard CTC forward–backward of [16]; we describe the forward pass in Algorithm 1, and the backward pass is symmetric.

Given a clean target $r_0 \in \mathcal{V}^L$, we construct an extended label sequence of length $2L + 1$ by inserting a <SLACK> symbol before, between, and after every target token. States at even indices $s \in \{0, 2, \ldots, 2L\}$ correspond to <SLACK> positions; states at odd indices $s \in \{1, 3, \ldots, 2L{-}1\}$ correspond to the $\lceil s/2 \rceil$-th target token. The model emits log-probabilities $\log P[t, \cdot]$ over the vocabulary at each response position $t \in \{1, \ldots, \tilde{L}\}$, and the forward variable $\alpha[t, s]$ accumulates the log-likelihood of all prefixes of length t that end in state s.

The algorithm differs from the standard CTC forward pass in exactly two transition rules, marked **M1** and **M2**:

M1 (self-loop restricted to <SLACK> states). Standard CTC permits a self-loop $\alpha[t{-}1, s] \to \alpha[t, s]$ at every state, which combined with the duplicate-merging collapse lets a single target token be emitted over multiple consecutive positions. Under β_{slack}, duplicates are not merged, so a self-loop at a label state would let the alignment emit the same target token at multiple adjacent positions. We therefore restrict self-loops to even (<SLACK>) states; label states must advance at every step.

M2 (unconditional skip at label states). Standard CTC permits the skip transition $\alpha[t{-}1, s{-}2] \to \alpha[t, s]$ only when the label at state s differs from the label at state $s{-}2$, precisely to prevent two identical target tokens from being merged into one via the duplicate-collapse rule. Since β_{slack} preserves duplicates, this guard is unnecessary: skipping the intervening <SLACK> state is always safe. We therefore allow the skip transition unconditionally at odd (label) states with $s \geq 2$.

Together, M1 and M2 ensure that the alignments reachable by the recurrence are exactly those in $\beta^{-1}_{\text{slack}}(r_0)$. The forward direction is immediate from the transition rules. For the reverse direction, any alignment a with $\beta_{\text{slack}}(a) = r_0$ admits a unique decomposition into the three transition types (self-loop at <SLACK> states, advance, skip-at-label), since at each position the token a_t and the current label index together determine the transition. The termination step marginalizes over the two valid final states ($s = 2L$, ending in <SLACK>, and $s = 2L - 1$, ending in the final target token), yielding $\log P^{\text{CTC}}_{\beta_{\text{slack}}}(r_0 \mid q_0, \tilde{r}_t)$ from Eq. (5).

B Implementation Details

B.1 Overall Framework

We perform SFT using a modified version of the training code from [68]. The CE-Only baseline is trained with the standard diffusion cross-entropy objective on prompt/response pairs from our Magpie-Pro subset. The Jitter and CE + CTC-S models share an identical data pipeline that augments responses with <SLACK> tokens at training time and differ only in the loss function: Jitter applies the standard cross-entropy objective to the slack-augmented inputs, while CE + CTC-S adds the auxiliary CTC-S term defined in Eq. (7).

Data preparation. We use Magpie-Pro-300K-Filtered tokenized with the LLaDA tokenizer, discard examples whose response length falls outside $[512, 1024]$, and pad each example to a fixed length of 2048 tokens. For each example we retain the original response as the CTC-S target, before any <SLACK> insertion.

<SLACK> insertion. At training time, given a per-example timestep $t \sim \mathcal{U}[0, 1]$, we insert $n = \min\big(\lfloor s_{\max}(1-t)\,R \rfloor,\, A\big)$ <SLACK> tokens at uniformly sampled positions in the response, where R is the response length, $s_{\max} = 0.5$, and A is the available capacity within the 2048-token budget. The trailing <EOT> and <EOS> are excluded from <SLACK> insertion.

Loss computation. The cross-entropy term follows the standard MDLM formulation. The CTC-S term is computed on the response slice between the response start index and the end of the augmented response, against the slack-free clean response retained before augmentation, via the recurrence in Algorithm 1. The final training objective is the sum of the two terms with the CTC-S weight set to $\lambda = 0.1$, as in Eq. (8).

B.2 Training Hyperparameters

All three models are trained under the same optimization settings. We use LoRA adapters with rank $r = 128$, scaling factor $\alpha = 32$, and dropout 0.05, without bias adaptation. For LLaDA-8B-Instruct, adapters are applied to the transformer projection layers $\{\texttt{q_proj}, \texttt{k_proj}, \texttt{v_proj}\}$, the token embedding layer (`wte`) and the output projection of the feed-forward block (`ff_out`).

Optimization uses AdamW [40] with a learning rate of 2×10^{-4}, weight decay 0.1, and gradient clipping at 1.0, under bfloat16 precision. Training is conducted for three epochs with a per-device batch size of 3, no gradient accumulation, and a maximum sequence length of 2048. After length filtering, the dataset contains approximately 291k training examples. With distributed batching and batch-processing details, three epochs correspond to approximately 36k optimizer update steps. We use a linear learning rate schedule without warmup and fix the random seed to 42. Checkpoints are evaluated and saved every 10% of the training steps, and the checkpoint with the lowest evaluation loss is selected. We do not perform any hyperparameter search in this work.

LoRA targets for LLaDA-MoE. The transferability experiment in Appendix D repeats the same three-model comparison on LLaDA-MoE. The LoRA configuration mirrors the LLaDA setting (rank, scaling factor, dropout, optimizer, schedule, and seed are all unchanged), but the target-module names are remapped to the LLaDA-MoE module naming: $\{\texttt{q_proj}, \texttt{k_proj}, \texttt{v_proj}\}$, the token embedding (`embed_tokens`), and the language-model head (`lm_head`). The router and expert MLPs are intentionally excluded so that the comparison isolates positional supervision rather than the MoE routing dynamics.

B.3 Assets Used

Table 10 lists all existing assets used in this work, together with their sources, licenses, and citations. We use the LLaDA-8B-Instruct and LLaDA-MoE-7B-A1B-Instruct checkpoints as base models, Magpie-Pro-300K-Filtered [61] as the SFT dataset, and a modified version of the d1 codebase as our training framework. For evaluation, we use the four open-ended generation benchmarks reported in § 5 (Arena-Hard-Auto v2.0, Creative-Writing-Bench v3, MT-Bench, and WildBench), and the six capability benchmarks reported in §6.3 (GPQA, MMLU, MBPP, IFEval, GSM8K, and MATH500),

Table 4: Performance comparison on open-ended text generation benchmarks on LLaDA-MoE. For Arena-Hard, we report bootstrapped 95% confidence intervals following the official setting. † and ‡ indicate that CE + CTC-S is significantly better than the CE-Only model and the Jitter model, respectively (paired bootstrap, $p < 0.05$). Bold indicates the best score in each column.

	Arena-Hard*	**Creative-Writing-Bench v3**	**MT-Bench**	**Wild-Bench**
LLaDA-MoE	$43.7\ (-1.6/+1.5)$	22.7	2.46	-5.48
CE-Only-MoE	$50.0\ (-0.0/+0.0)$	22.4	2.84	-5.42
Jitter-MoE	$48.9\ (-1.7/+1.6)$	22.2	**3.34**	-5.22
CE + CTC-S (MoE)	$\mathbf{63.7}^{\dagger\ddagger}\ (-1.2/+1.3)$	$\mathbf{25.1}^{\dagger\ddagger}$	$3.21^{\dagger}$	$\mathbf{-5.03}^{\dagger\ddagger}$

the latter run through the LLaDA-forked OpenCompass evaluation pipeline. All assets are used in accordance with their published licenses.

C Pairwise Significance Test Results

To support the significance markers in Tables 1, 4, and 5, we report the full pairwise 95% paired-bootstrap confidence intervals for all relevant model pairs on aligned evaluation instances. For each benchmark and model pair, we compute instance-level score differences on the subset of examples scored for both models, resample those aligned differences with replacement for 10,000 bootstrap iterations, and report the empirical 2.5th and 97.5th percentiles of the bootstrap mean difference (row minus column for the dense and MoE matrices; CE+CTC-S minus the column baseline at its best block size for the block-diffusion table). We regard a comparison as statistically significant when the confidence interval excludes zero.

Table 7 corresponds to Table 1 (LLaDA-8B-Instruct), Table 8 to Table 4 (LLaDA-MoE), and Table 9 to Table 5 (block-diffusion sweep ablation).

D Transferability to LLaDA-MoE

We report the open-ended generation comparison from §5 on a second model, LLaDA-MoE, as a transferability check[13]. Table 4 reports the results. CE + CTC-S achieves the best score on three of four benchmarks and significantly improves over CE-Only (†) on all four. The improvement over Jitter (‡) is significant on Arena-Hard, Creative-Writing-Bench v3, and WildBench. On MT-Bench, CE + CTC-S is significantly better than CE-Only; the difference from Jitter is not significant, and Jitter has the highest score on this benchmark.

The overall pattern observed on LLaDA-8B-Instruct carries over: relaxing positional alignment via CTC-S consistently improves open-ended generation on a second MDLM trained under the same standard objective. Full pairwise confidence intervals are reported in Table 8.

E Comparison to Block Diffusion

Block diffusion is an inference-time lever that limits the decoding canvas, and a smaller canvas could suppress the occurrence of positional misplacement. A natural question is whether the gains we observe with CTC-S can be recovered by tuning this lever, i.e., whether CTC-S provides anything beyond what an appropriately tuned block-diffusion baseline already delivers.

To address this, we sweep block sizes $b \in \{32, 64, 128, 256, 512\}$ for LLaDA-8B-Instruct and the CE-Only baseline at inference and report the best per-benchmark score across the sweep. CE + CTC-S is run as a single configuration without block-diffusion sweeping. This setup gives the baselines the most favourable inference configuration available under this strategy and asks whether CTC-S

[13]The training setup matches §5; the LoRA target-module remapping for LLaDA-MoE is described in Appendix B.

Table 5: Block-diffusion sweep ablation. We sweep block sizes $b \in \{32, 64, 128, 256, 512\}$ for LLaDA and CE-Only and report the best score per column (winning b in parentheses); CE + CTC-S is a single non-swept run. $\circ$ and $\diamond$ indicate that CE + CTC-S is significantly better than the best-swept CE-Only and LLaDA variants, respectively (paired bootstrap on aligned sample IDs, 10,000 iterations, 95% CI excludes zero). Bold indicates the best score in each column. The lone non-significant comparison is MT-Bench vs. LLaDA at b=32 (mean Δ=+0.05, CI_{95}=$[-0.29, +0.38]$).

	Creative-Writing-Bench v3	**MT-Bench**	**Wild-Bench**
LLaDA	23.2	2.84	−5.48
+ Block Diffusion (best b)	26.19 (b=512)	4.30 (b=32)	−4.63 (b=64)
CE-Only	24.6	3.44	−4.53
+ Block Diffusion (best b)	26.44 (b=512)	3.53 (b=512)	−4.51 (b=512)
CE + CTC-S	**27.7**$^{\circ\diamond}$	**4.35**$^{\circ}$	**−4.23**$^{\circ\diamond}$

still provides additional value on top of it. We restrict the sweep to Creative-Writing-Bench v3, MT-Bench, and WildBench for computational tractability.[14]

We deliberately do not include a CE + CTC-S + block-diffusion configuration in this comparison. The question this section is designed to answer is whether the training-side lever (CTC-S) provides value beyond what an inference-side lever (block diffusion) can already recover. Answering it cleanly requires varying one factor at a time: a training-side lever (CTC-S, applied to a single inference configuration) against an inference-side lever (block-size sweep applied to models without training-side lever). Combining the two would test a different question, namely whether the training-side and inference-side interventions compose, which we leave to future work.

Table 5 reports the results. Even after sweeping, CE + CTC-S is significantly better than the best-swept CE-Only on all three benchmarks and significantly better than the best-swept LLaDA on two of three. The one exception is MT-Bench against LLaDA at b=32, where the comparison is a statistical tie (mean $\Delta = +0.05$, $\mathrm{CI}_{95} = [-0.29, +0.38]$). Across the six paired comparisons, training-time alignment flexibility outperforms a strictly better-tuned block-diffusion baseline on five and ties on one. This indicates that the contribution of CTC-S is not subsumed by inference-time canvas restriction: relaxing positional supervision during SFT provides gains that reducing the parallelism of decoding does not recover. Full pairwise confidence intervals are reported in Table 9

F Robustness to a Shorter Response Length

All experiments in §5 use the fixed inference configuration of §2.2 (response length $L = 1024$, $N = 512$ diffusion steps). To check that the gains of CTC-S are not tied to this canvas, we repeat the open-ended generation comparison on LLaDA-8B-Instruct with $L = 512$ and $N = 256$ steps, keeping the ratio of steps to response length, the low-confidence remasking strategy, the trained checkpoints, and the evaluation protocol unchanged[15]. Table 6 reports the results. CE + CTC-S achieves the best score on three of four benchmarks. It is significantly better than CE-Only (†) on Creative-Writing-Bench v3 and MT-Bench, and significantly better than Jitter (‡) on Arena-Hard and MT-Bench. On Arena-Hard, CE + CTC-S does not surpass the CE-Only reference at this length (-1.1 pp, 95% CI $[-4.2, +1.9]$); on WildBench, the three trained models are within each other's confidence intervals.

The pattern observed at $L = 1024$ therefore carries over to a shorter canvas: the improvement remains concentrated at the CTC-S step, while the advantage on Arena-Hard is smaller than in Table 1.

[14] Arena-Hard requires pairwise judgments against a baseline at every b, which inflates the cost.

[15] No model is retrained; the three checkpoints of §5 are decoded with the shorter canvas, and generation and judging otherwise follow §5.

Table 6: Performance comparison on open-ended text generation benchmarks on LLaDA-8B-Instruct with response length $L = 512$ and $N = 256$ diffusion steps. For Arena-Hard, we report bootstrapped 95% confidence intervals following the official setting. † and ‡ indicate that CE + CTC-S is significantly better than the CE-Only model and the Jitter model, respectively (paired bootstrap, $p < 0.05$). Bold indicates the best score in each column.

	Arena-Hard*	**Creative-Writing-Bench v3**	**MT-Bench**	**Wild-Bench**
LLaDA	$44.5\ (-1.2/+1.7)$	27.5	3.80	-4.64
CE-Only	$\mathbf{50.0}\ (-0.0/+0.0)$	28.7	3.93	-4.12
Jitter	$43.8\ (-1.3/+1.4)$	29.2	4.06	-4.04
CE + CTC-S	$48.3^{\ddagger}\ (-1.7/+1.5)$	$\mathbf{30.4}^{\dagger}$	$\mathbf{4.52}^{\dagger\ddagger}$	$\mathbf{-4.03}$

G Compute Statement

Most experiments presented in this paper were run on two clusters, both consisting of NVIDIA H100 GPUs with 80 GB of memory. Model training runs on eight H100 GPUs and completes within 10 hours. Generation runs on a single H100 GPU with 80 GB of memory. Evaluation time varies across benchmarks: Arena-Hard takes around 6 hours, MT-Bench typically under 2 hours, Creative-Writing-Bench v3 around 1.5 hours, and WildBench around 10 hours.

On the same dataset and hardware (eight NVIDIA H100 80GB GPUs), standard CE fine-tuning takes approximately 6.5 hours, while CE + CTC-S fine-tuning takes approximately 10 hours (1.54× wall-clock time). CTC-S also requires more training memory, primarily due to the longer sequences introduced by the additional <SLACK> tokens rather than the CTC dynamic programming itself. Our method introduces no architectural changes or additional inference-time overhead.

Table 7: Pairwise significance matrix for Table 1. Each populated cell reports the 95% paired-bootstrap confidence interval for the mean score difference (row minus column); * denotes intervals that exclude zero.

Arena-Hard				
	LLaDA	**CE-Only**	**Jitter**	**CE+CTC-S**
LLaDA	–			
CE-Only	[0.027, 0.098]*	–		
Jitter	[0.004, 0.091]*	[-0.051, 0.022]	–	
CE+CTC-S	[0.068, 0.157]*	[0.014, 0.087]*	[0.024, 0.103]*	–

Creative-Writing-Bench v3				
	LLaDA	**CE-Only**	**Jitter**	**CE+CTC-S**
LLaDA	–			
CE-Only	[0.310, 2.434]*	–		
Jitter	[0.763, 3.162]*	[-0.429, 1.585]	–	
CE+CTC-S	[3.245, 5.869]*	[2.075, 4.316]*	[1.499, 3.735]*	–

MTBench				
	LLaDA	**CE-Only**	**Jitter**	**CE+CTC-S**
LLaDA	–			
CE-Only	[0.125, 1.031]*	–		
Jitter	[0.344, 1.169]*	[-0.212, 0.550]	–	
CE+CTC-S	[1.075, 1.938]*	[0.594, 1.244]*	[0.412, 1.081]*	–

WildBench				
	LLaDA	**CE-Only**	**Jitter**	**CE+CTC-S**
LLaDA	–			
CE-Only	[0.764, 1.133]*	–		
Jitter	[0.838, 1.201]*	[-0.078, 0.223]	–	
CE+CTC-S	[1.049, 1.439]*	[0.150, 0.441]*	[0.080, 0.367]*	–

Table 8: Pairwise significance matrix for Table 4 (LLaDA-MoE). Each populated cell reports the 95% paired-bootstrap confidence interval for the mean score difference (row minus column); * denotes intervals that exclude zero.

Arena-Hard				
	LLaDA	**CE-Only**	**Jitter**	**CE+CTC-S**
LLaDA	–			
CE-Only	[0.027, 0.099]*	–		
Jitter	[0.012, 0.093]*	[-0.046, 0.026]	–	
CE+CTC-S	[0.160, 0.240]*	[0.102, 0.171]*	[0.107, 0.189]*	–

Creative-Writing-Bench v3				
	LLaDA	**CE-Only**	**Jitter**	**CE+CTC-S**
LLaDA	–			
CE-Only	[-1.462, 0.955]	–		
Jitter	[-1.543, 0.768]	[-1.272, 0.841]	–	
CE+CTC-S	[1.031, 3.472]*	[1.657, 3.701]*	[1.843, 3.881]*	–

MTBench				
	LLaDA	**CE-Only**	**Jitter**	**CE+CTC-S**
LLaDA	–			
CE-Only	[-0.037, 0.787]	–		
Jitter	[0.481, 1.262]*	[0.181, 0.825]*	–	
CE+CTC-S	[0.294, 1.188]*	[0.044, 0.688]*	[-0.469, 0.194]	–

WildBench				
	LLaDA	**CE-Only**	**Jitter**	**CE+CTC-S**
LLaDA	–			
CE-Only	[-0.119, 0.234]	–		
Jitter	[0.090, 0.432]*	[0.074, 0.330]*	–	
CE+CTC-S	[0.260, 0.635]*	[0.252, 0.523]*	[0.064, 0.307]*	–

Table 9: Pairwise significance matrix for Table 5 (block-diffusion sweep ablation). Each populated cell reports the 95% paired-bootstrap confidence interval, on aligned evaluation instances, for the mean score difference between CE+CTC-S (single, non-swept run) and the column baseline at its best block size; * denotes intervals that exclude zero.

	LLaDA + Block Diffusion (best b)	**CE-Only + Block Diffusion (best b)**
Creative-Writing-Bench v3	[0.526, 2.535]*	[0.395, 2.237]*
MT-Bench	[-0.294, 0.381]	[0.481, 1.169]*
WildBench	[0.232, 0.559]*	[0.139, 0.422]*

Table 10: Existing assets used in this work, with their sources, licenses, and citations. Models, the SFT dataset, and the training framework are listed first, followed by open-ended generation benchmarks (§ 5) and capability benchmarks (§ 6).

Asset Type	Asset Name	Link	License	Citation
Model	LLaDA-8B-Instruct	HuggingFace	MIT License	[42]
Model	LLaDA-MoE-7B-A1B-Instruct	HuggingFace	Apache 2.0	[72]
Dataset	Magpie-Pro-300K-Filtered	HuggingFace	Llama 3 Community License	[61]
Code	d1 (SFT framework)	Github	Apache 2.0	[68]
Code	OpenCompass (LLaDA fork)	Github	Apache 2.0	[8, 42]
Benchmark	Arena-Hard-Auto (v2.0)	Github	Apache 2.0	[34, 35]
Benchmark	Creative-Writing-Bench v3	Github	Not specified	[45]
Benchmark	MT-Bench (FastChat)	Github	Apache 2.0	[69]
Benchmark	WildBench	Github	Apache 2.0	[38]
Benchmark	GPQA	Github	MIT License	[49]
Benchmark	MMLU	Github	MIT License	[21]
Benchmark	MBPP	Github	Apache 2.0	[2]
Benchmark	IFEval	Github	Apache 2.0	[70]
Benchmark	GSM8K	Github	MIT License	[7]
Benchmark	MATH500	Github	MIT License	[22, 37]

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state both contributions delivered in the paper: the controlled shift intervention shows positional sensitivity (§ 3) and the CTC-S auxiliary objective achieves consistent gains across four open-ended generation benchmarks (§ 5).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A dedicated Limitations paragraph at the end of § 8 discusses the SFT-only scope, the two MDLMs from the LLaDA family, both trained from scratch under the standard masked diffusion objective.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include formal theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: § 5 specifies data, models, benchmarks, and metrics; Appendix B details the training framework, <SLACK> insertion, loss computation, and hyperparameters; Appendix A specifies the CTC-S forward pass.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We attach an anonymized version of the code as supplementary material and will release it publicly upon acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training and evaluation details (data filtering, model variants, benchmarks, metrics) are reported in § 5; full hyperparameters, optimizer, LoRA configuration, and schedule are in Appendix B.2. We also provide anonymized code as supplementary material.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Arena-Hard reports the official bootstrapped 95% confidence intervals; for all benchmarks we additionally report paired-bootstrap 95% confidence intervals over 10,000 iterations on aligned instance-level differences (§ 5, Appendix C).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We describe the compute used in Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses only publicly released artifacts, involves no human subjects, and conforms to the NeurIPS Code of Ethics; see the Ethics statement.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See the Impact statement.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our modifications to the SFT objective do not introduce capabilities beyond those of the base LLaDA-8B-Instruct model, which is already publicly released.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets are credited and listed in Table 10 with available license information. For Creative-Writing-Bench v3, we were unable to identify a specified license and therefore mark it as "Not specified"; all other assets are used under their published licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new datasets or model checkpoints. Code will be released for reproducibility but is not itself a contributed asset.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are used as judges in the evaluation protocol for open-ended generation benchmarks, which is standard for these benchmarks and not part of the proposed training method. Beyond evaluation, any LLM use was limited to writing and editing.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.