DENOISING IS NOT THE END: DISCRETE DIFFUSION LANGUAGE MODELS WITH SELF-CORRECTION

Jinwei Zhang, Dimitri von Rütte, Yuhui Ding, Thomas Hofmann

ABSTRACT

Discrete Diffusion Models have emerged as an effective approach to text generation, providing an alternative to autoregressive models. However, the generated texts often suffer from quality issues, including grammatical errors, lack of fluency and factual inaccuracies. This paper systematically evaluates different self-correction strategies for hybrid-noise (masking and uniform) discrete diffusion language models. The outputs of the base model are compared with self-corrected outputs under different strategies, with improvements measured in terms of quality, fluency, and fidelity. Our experiments show significant quality gains from iterative self-correction, with improvements in grammar, factuality, clarity, and creativity reaching double-digit percentages. We also observe a tradeoff between text quality and content preservation and identify optimal configurations that achieve significant quality enhancements while maintaining high fidelity to the original output. Finally, we find that it is optimal to allocate inference compute to denoising and self-correction in approximately equal proportions. Our code and evaluation scripts will be open-sourced upon acceptance. The code and evaluation script are open source: `https://github.com/timosoy/gidd`

1 INTRODUCTION

Autoregressive Language Models (ALMs) have experienced great success in generative AI using the next token prediction mechanism (Brown et al., 2020; OpenAI et al., 2024). Since the generated tokens are assumed to be correct and therefore fixed, autoregressive models suffer from error accumulation, where early mistakes propagate and degrade subsequent generation quality (Bengio et al., 2015; Holtzman et al., 2020). As a non-autoregressive approach, Discrete Diffusion Models (DDMs) generate text by denoising all tokens in parallel from a fully corrupted state. However, existing DDM frameworks (Austin et al., 2021; Sahoo et al., 2024), and particularly masked diffusion models (MDMs), still face a critical limitation: once a token is unmasked from noise, it is treated as a fixed decision for the generation process. This problem of irreversibility reintroduces a form of error propagation, where local inconsistencies cannot be fixed. Therefore, prior work has proposed to add a small fraction of uniform noise to the masked diffusion process to solve this (von Rütte et al., 2025): some tokens are replaced with random tokens in the forward process, so the model learns to decide whether an already generated token should be corrected in the generative process, in addition to filling in missing tokens.

While the potential for self-correction is obvious, the optimal mechanism for implementing it remains relatively unexplored. In the domain of masked diffusion, previous works have introduced iterative refinement techniques such as adaptive sampling (Kim et al., 2025b; Ben-Hamu et al., 2025) and remasking strategies (Wang et al., 2025; Huang et al., 2025; Kim et al., 2025a), which enable the model to identify and regenerate the low-confidence tokens during inference at each step. However, one key practical gap remains: adaptive inference adjusts the update schedule but still cannot resample decided tokens. On the other hand, remasking enables targeted regeneration, but may increase inference cost and require additional post-training to be effective. In this work, we address this gap using the framework of GIDD (von Rütte et al., 2025), enabling inference-time resampling. We explore and compare iterative self-correction algorithms based on *replacement confidence* and *Negative Log-Likelihood (NLL) uncertainty*. We also introduce dynamic thresholding to improve the efficiency of correction. We then conduct a comprehensive evaluation of the proposed methods to determine their impact on the quality of self-corrected samples. The impact of each strategy is analyzed using a multi-dimensional framework that includes standard fluency metrics (e.g., self-

Table 1: Qualitative self-correction examples across different error types. The model identifies erroneous tokens and generates contextually appropriate replacements.

Original Generation	**Self-Corrected Output**
Type 1: Artifact Removal & Idiom Repair	
" TRANS worry of young Football League managers is that the odds of the Championship making another had over the coming campaigns are already flattening dramatically."	" The worry of young Football League managers is that the odds of the Championship making another run in the coming season are already slimming dramatically."
Type 2: Factuality & Entity Correction	
"...according to Russ Doler, a spokeswoman for Councilman Patrick Gallego and a spokeswoman for Mayor Marty Hors . Gru Service officers took over that training..."	"...according to Jennifer Dye, a spokeswoman for Councilman Brian Gallego and a spokeswoman for Mayor Michael Hancock . Emergency response officers took over that training..."
Type 3: Contextual Refinement	
"...but the additional risk and 'overconfident' aspect of this equation is starting with the business models... What interests me is it is "iranfinance" going mainstream."	"...but the additional risk and 'overconfident' aspect of this equation is seen with the business models... What interests me is it is "cyberfinance" going mainstream."

and generative perplexity), as well as LLM-based quality metrics (clarity, factuality, grammaticality, etc.) and fidelity metrics measuring similarity to the original text.

Through our experiments, we observe that there is an inherent tradeoff between quality improvement and content preservation: fixing mistakes and improving fluency necessarily entails making semantically meaningful changes. We quantify this tradeoff through fidelity metrics such as BLEU score, token difference and cosine similarity of semantic sentence embeddings. However, this tradeoff is not uniform. We observe different tradeoff profiles across different token selection strategies, that is, the rules that decide which tokens are selected and updated at each correction step. For example, aggressive strategies can achieve larger quality gains but may introduce stronger semantic drift. This indicates that practitioners can choose appropriate configurations based on their requirements. Our contributions are as follows:

1. Building upon the GIDD (von Rütte et al., 2025) framework, we propose and examine four distinct token selection mechanisms: **confidence-based** (top-1 and top-8 tokens) and **uncertainty-based** (NLL; top-K and dynamic thresholding). Each strategy is evaluated with a broad variation of hyperparameter settings, including sampling temperature, the number of correction steps, and the number of replaced tokens per step, to ensure that each strategy is evaluated at its peak performance.

2. Extending the evaluation protocol of von Rütte et al. (2025), we develop an comprehensive analysis framework across three dimensions: **text quality** based on LLM judgment of high-level writing properties (e.g., grammaticality, factuality, style); **fidelity**, measuring how much of the original content is preserved after correction (BLEU score, token change count, semantic cosine similarity); and **fluency**, measuring naturalness of the text based on self- and generative perplexity (PPL).

3. We quantify the tradeoff between correction aggressiveness and content preservation. While we find that unconstrained correction results in semantic drift, carefully tuned strategies can maximize quality improvements while keeping high similarity to the original text.

4. We also investigate the optimal allocation of inference compute: for a given budget, how many steps should be allocated to denoising and generating high-quality samples, and how many should be spent on self-correction? We find that equal allocation is often ideal, with text quality improving by up to 40% compared to pure denoising without self-correction.

2 PRELIMINARIES

Generalized Interpolating Discrete Diffusion (GIDD; von Rütte et al., 2025) is a generative framework for discrete data that extends standard Masked Diffusion Models (MDMs) to flexible noise distributions. Let $\mathbf{x} \in \{0,1\}^{L\times|\mathcal{V}|}$ denote a one-hot encoded text sequence of length L from a vocabulary $\mathcal{V}$. The diffusion process is defined over a continuous time interval $t \in [0,1]$, where $t=0$ corresponds to the clean data distribution and $t=1$ represents a tractable prior distribution. Unlike standard MDMs, which rely solely on the absorbing masking process, GIDD models the forward process as a general interpolation between clean data and some noise distribution $\boldsymbol{\pi}_t$. The marginal distribution of a noisy token z_t given the clean token $\mathbf{x}$ is defined as a categorical distribution:

$$q_t(z_t|\mathbf{x}) = \text{Cat}(z_t; \alpha_t\mathbf{x} + \beta_t\boldsymbol{\pi}_t) \tag{1}$$

Here, $\alpha_t \in [0,1]$ controls the signal strength, decreasing from 1 to 0, while $\beta_t = 1-\alpha_t$. The term $\boldsymbol{\pi}_t$ represents the *mixing distribution*, which defines the type of noise added to the sequence at a given time t.

Following von Rütte et al. (2025), we use a *hybrid-noise strategy* that interpolates between the data, the mask token m and the uniform noise u over the vocabulary excluding the mask token, i.e., $u \sim \mathcal{U}(\mathcal{V}\setminus\{m\})$. The amount of uniform noise is parameterized by a uniform noise level p_{unif}. The noise distribution $\boldsymbol{\pi}_t$ is defined such that the model encounters masked tokens and incorrect random tokens during training. Formally, the mixing schedule ensures that at any intermediate time step, a token is either kept as data, replaced by m, or replaced by a random token sampled from $\mathcal{U}(\mathcal{V})$. The exposure to random noise enables the model to not only fill in missing information, but also identify and correct wrong tokens, which is precisely what enables *self-correction* at inference time. See Appendix A for further details on the GIDD framework.

3 SELF-CORRECTION VIA ITERATIVE REFINEMENT

Since DDMs generate text in a non-autoregressive manner, the generated sequence often contains local inconsistencies, such as grammatical errors, repetitions, or logical errors. While the diffusion process terminates once the sequence is fully denoised (i.e., at $t \approx 0$), a *hybrid-noise* diffusion model has an additional ability: to detect and correct errors in fully observed data. This property can be utilized to perform post-hoc self-correction on a sequence generated by the model, denoted as $\hat{\mathbf{x}} = \{\hat{x}_1, \ldots, \hat{x}_L\}$. Our goal is to design a post-hoc refinement function $f(\hat{\mathbf{x}}) \to \hat{\mathbf{x}}'$ that improves perceptual quality while preserving the semantic meaning as much as possible. We formulate this as an iterative optimization problem: given the generated sequence $\hat{\mathbf{x}}$, we identify a set of low-confidence tokens $\mathcal{M}$ based on some confidence/uncertainty criterion $\mathcal{C}$ and resample them using the trained diffusion model $\mathbf{x}_\theta$.

3.1 ITERATIVE REFINEMENT PROCESS

Let $z^{(0)} = \hat{\mathbf{x}}$ be the initial sequence generated by the diffusion model. The self-correction process iteratively updates this sequence until convergence or for a maximum of S_{corr} steps. Each iteration consists of three phases outlined below. The procedure is also provided as pseudocode in Alg. 1.

1. **Inference:** The model $\mathbf{x}_\theta$ takes the current sequence $z^{(k)}$ as input and predicts the categorical distribution, where τ is a temperature parameter that controls the stochasticity.[1]
$$\mathbf{p}^{(k)} = \text{softmax}\big(\mathbf{x}_\theta(z^{(k)}, t \approx 0)/\tau\big), \tag{2}$$
2. **Candidate sampling:** Once a candidate sequence $\tilde{z}^{(k)}$ is sampled from $\mathbf{p}^{(k)}$, the set of *candidate correction indices* $\mathcal{S}$ is given by all positions where the candidate differs from the current sequence:
$$\mathcal{S} = \{i \in \{1,\ldots,L\} \mid \tilde{z}_i^{(k)} \neq z_i^{(k)}\} \tag{3}$$
3. **Select and update:** From the set of candidate positions, a subset $\mathcal{M} \subseteq \mathcal{S}$ is chosen according to some selection heuristic (e.g., select candidate tokens with high confidence), and the candidate tokens are committed at those positions, i.e.
$$z_i^{(k+1)} = \begin{cases} \tilde{z}_i^{(k)} & \text{if } i \in \mathcal{M} \\ z_i^{(k)} & \text{otherwise.} \end{cases} \tag{4}$$

[1] We set $t = 0.01$ in all our experiments and force the model to only predict unmasked tokens by setting the probability of the mask token m to 0.

Algorithm 1 Self-Correction

Require: Generated sequence z, model $\mathbf{x}_\theta$, temperature τ, max. steps S_{corr}
for $k = 1, \ldots, S_{\text{corr}}$ **do**
  Predict: $\mathbf{p} \leftarrow \text{softmax}(\mathbf{x}_\theta(z, t = 0.01)/\tau)$
  Sample: $\tilde{z} \sim \text{Cat}(\mathbf{p})$
  Identify Candidates: $\mathcal{S} \leftarrow \{i \mid \tilde{z}_i \neq z_i\}$
  if $\mathcal{S} = \emptyset$ **then**
    break (Convergence)
  end if
  Select: $\mathcal{M} \subseteq \mathcal{S}$ {e.g., select highest confidence change(s)}
  Update: $z_j \leftarrow \tilde{z}_j$ for all $j \in \mathcal{M}$
end for
return Refined sequence z

3.2 Token Selection Strategies

The core challenge of an effective self-correction strategy is to identify problematic, low-quality tokens. We hypothesize that tokens with high model uncertainty are correlated with errors in the generation process or random noise, which motivates the following token selection strategies.

Baseline. In the original GIDD framework (von Rütte et al., 2025), a conservative strategy is introduced as a baseline. Among all potential changes in $\mathcal{S}$, the algorithm selects only the single token with the highest model confidence to update. Formally, the update set $\mathcal{M}$ consists of the index j that maximizes confidence $\mathcal{C}$, i.e.

$$\mathcal{M} = \big\{ \arg\max_{i \in \mathcal{S}} \mathcal{C}_i \big\}, \quad \mathcal{C}_i = p_i^{(k)}(\tilde{z}_i^{(k)}). \tag{5}$$

This conservative approach ensures that only the most confident error is corrected at each step, preventing the model from changing the sequence structure too quickly. This is the starting point of our research, and we will propose improved selection strategies to address key limitations of this baseline in the following.

Confidence-based top-K selection. A simple extension of the baseline is to not only consider the single most confident token, but instead to facilitate parallel refinements by updating the top-K most confident tokens. Consequently, this procedure is identical to the baseline with the exception of selecting $\mathcal{M} \subseteq \mathcal{S}$. We now have

$$\mathcal{M} = \arg\operatorname{topk}_{i \in \mathcal{S}} \mathcal{C}_i, \quad \mathcal{C}_i = p_i^{(k)}(\tilde{z}_i^{(k)}). \tag{6}$$

This allows the model to correct multiple high-confidence errors simultaneously, significantly accelerating convergence compared to the single-token update.

NLL-based top-K selection. Selection strategies based on negative log-likelihood (NLL) focus on replacing tokens with low likelihood (i.e., high uncertainty) regardless of whether a suitable alternative is available. This is based on the principle that high uncertainty tokens are considered to be questionable and need to be reconsidered and is in contrast to confidence-based strategies, where token replacements are confined to positions where the model is able to propose a good alternative.

To this end, we compute the per-token NLL of the current sequence $z_i^{(k)}$ under the model's predicted distribution $\mathbf{p}^{(i)}$. In top-K mode, we order all positions according to their NLL scores in descending order and take the K positions with the highest scores (lowest probabilities):

$$\mathcal{M} = \arg\operatorname{topk}_{i \in \mathcal{S}} \mathcal{C}_i, \quad \mathcal{C}_i = -\log p_i^{(k)}(\tilde{z}_i^{(k)}). \tag{7}$$

By selecting these positions, we let the model regenerate the most unexpected or implausible part of the sequence.

NLL-based dynamic threshold selection. Top-K strategies with fixed K tend to be quite inflexible. This inflexibility may lead to over-correction of valid tokens in high-quality sequences, or under-correction in low-quality sequences. Therefore, we propose a dynamic selection mechanism in which the number of corrections is dependent on the uncertainty distribution of the sequence at hand. A dynamic threshold δ will be defined based on the maximum observed uncertainty in the current sequence. Let $\text{NLL}_{\text{max}} = \max_j\{-\log p_j^{(k)}(z_j^{(k)})\}$ be the peak uncertainty. The threshold δ is then given by

$$\delta = \alpha \cdot \text{NLL}_{\text{max}}, \tag{8}$$

where $\alpha \in [0, 1]$ is a hyperparameter (empirically set to 0.9 in our experiments). The positions selected for correction are defined as:

$$\mathcal{M} = \{i \in \{1, \ldots, L\} \mid -\log(p^{(i)}(z_t^{(i)})) \geq \delta\} \tag{9}$$

This mechanism makes sure that only the most extreme outliers are subjected to correction. As the quality of the sequence improves and NLL_{max} decreases, the self-correction mechanism becomes more conservative and stable without external modifications.

4 Experimental Setup

4.1 Base Model and Data Generation

To evaluate the effectiveness of self-correction methods, we use pre-trained GIDD models (Sec. 2) as the sample generator. In particular, we used the **GIDD-base** checkpoint with a uniform noise probability of $p_{\text{unif}} = 0.2$. See Appendix D for further details. A fixed set of **1,000** samples was generated for the evaluation dataset. The generation process was performed using standard 128 inference steps. For consistency, all subsequent experiments are conducted on the same set of the 1,000 samples.

4.2 Key Hyperparameters

In total, there are four important hyperparameters that control the behavior of the proposed self-correction mechanism: **tokens per step** (K), **sampling temperature** (τ), **inference steps** (S_{corr}) and **dynamic threshold ratio** (α). These settings are chosen to capture the tradeoff between stability, diversity, and computational cost. See Appendix B for further details.

4.3 Evaluation Metrics

To systematically assess the performance of the self-correction framework, a combination of metrics is employed. We use self-PPL, generative PPL, and sequence entropy to measure the fluency and naturalness of generated samples, and BLEU score, token difference, and semantic cosine similarity to measure fidelity to the original text. See Appendix C for further details.

Additionally, we rely on LLM-based evaulation to measure qualitative aspects of generated samples. We use the **Gemma 3 12B** checkpoint (Gemma Team et al., 2025) to score the generations on a scale from 1–10 across five distinct dimensions: **grammaticality**, **factuality**, **style**, **clarity**, **creativity**. The detailed prompt and system instructions used for this evaluation are provided in Appendix H.

5 Results and Analysis

5.1 Sanity check

We first determine whether self-correction functions as intended: as a Monte Carlo fixed-point iteration, we expect a hill-climbing effect on the model's likelihood, leading to a decrease in self-PPL and, hopefully, also generative PPL as judged by a reference model. Figure 1 illustrates the changes of key metrics across different inference steps. Indeed, we observe a consistent monotonic decrease in both self-PPL and gen.-PPL as the number inference steps increases, which stands for improvement in fluency and model confidence. Figure 1 (middle) shows that the gen.-PPL drops from an initial baseline of **130.0** to **37.1** after 32 inference steps (using the NLL-based top-K strategy with $K = 8, \tau = 0.1$). Extending the inference steps to 128 steps yields further improvements. The Gen-PPL decreases to **28.1**. This shows that self-correction is effective and can bring the text closer to the natural language distribution. We observe a similar trend on the model's intrinsic metrics. The self-PPL drops by orders of magnitude, decreasing from **3,256** in the original generated text to

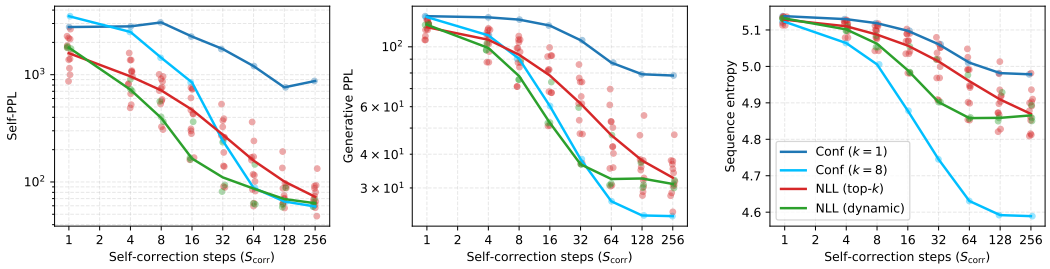

Figure 1: Fluency metrics (self- and gen.-PPL) improve monotonically in the number of correction steps, suggesting that all self-correction procedures successfully hill-climb the likelihood landscape of the model. The mode-seeking nature of this fixed-point iteration is further confirmed by the stable decrease in sequence entropy, which indicates a reduction, but not complete collapse, of diversity.

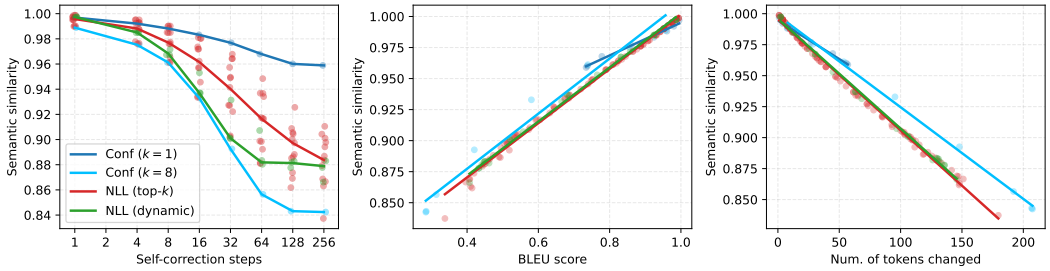

Figure 2: Semantic similarity to the original sequence decreases monotonically in the number of self-correction steps increase (left), with semantic drift correlating linearly with BLEU scores (middle). The number of changed tokens also correlates with semantic drift, although confidence-based strategies are able to replace more tokens while maintaining the same semantic drift (right).

93.8 at step 32, and further to **61.6** at step 128 (NLL-based top-K strategy with $K = 8, \tau = 0.1$). Meanwhile, the self-accuracy of the model also increases from initially 56% to around 80% after 128 inference steps. The reduction in self-PPL shows that the uncertainty of the model about the corrected text has been lowered.

Preservation of lexical diversity. The risk of mode collapse can occur when the model tries to minimize loss by simplifying text into repetitive or generic patterns (e.g., short, safe sentences). Therefore, the Shannon entropy of the token distribution is tracked. The entropy shown in Figure 1 (right) remains relatively stable throughout the correction process. Starting from a baseline of **5.14** and then reducing to **4.96** at the first 32 steps, the entropy still remains stable with **4.87** at 128 steps. Thus, the self-correction functions as a precise editor and not a simplifier, where it tends to make a correction to the content while maintaining the same level of lexical variety and sentence structure.

5.2 Fidelity Analysis

As the section 5.1 shows improvement in fluency, it is essential to demonstrate the magnitude of the modifications and the impact on semantic preservation.

Linearity between BLEU and semantic similarity. An important step is to validate that the model's edit are meaningful and that the evaluation metrics are consistent. Figure 2 (Left) illustrates the correlation between BLEU scores and Semantic Cosine Similarity across all experimental configurations. A strong linear correlation is observed between the two metrics($R^2 > 0.99$ for the NLL-based approach and $R^2 = 0.97$ for the confidence-based ones). This linear correlation indicates that the self-correction mechanism applies word-level changes and changes in meaning simultaneously. Due to this redundancy, **Semantic Cosine Similarity** will be utilized to represent fidelity metrics in the subsequent chapters, as it aligns better with human perception of meaning preservation.

Table 2: Largest observed quality gains for each strategy. Quality improvements are always tied to semantic drift, suggesting an inherent tradeoff between improving quality and preserving semantics.

	Gramm.	Fact.	Style	Clar.	Creat.	Avg.	Sim. (%)
Generated (no correction)	1.66	1.71	1.50	1.59	2.30	1.75	100
Conf ($K = 1$; baseline)	1.89	1.92	1.67	1.72	2.42	1.92	95.9
Conf ($K = 8$)	2.25	2.43	1.91	2.10	2.36	2.21	84.2
NLL (top-K)	**3.04**	**2.84**	**2.48**	**2.54**	2.61	**2.70**	**83.7**
NLL (dynamic)	2.89	2.63	2.34	2.38	**2.64**	2.58	86.6
Gains of best (%)	+83.1	+66.0	+65.3	+59.8	+14.7	+54.2	-16.3

Fidelity changes over inference steps. Figure 2 (right) illustrates tracks the Semantic Cosine Similarity as the correction steps increases. A tradeoff between aggressiveness of the correction and fidelity to the original text sample is observed. For the NLL-based approach ($K = 4, \tau = 0.1$), the Cosine Similarity remains relatively high for the first 32 steps ($> 92\%$). As the self-correction process extends to 64 steps, similarity further decreases to **89.8%**. At 256 steps, the similarity stays at **87.9%**, which could indicate that the model begin to rewrite the sentence extensively.

In contrast, the confidence-based strategy demonstrates a more conservative decay profile. Using the default confidence-based approach ($K = 1, \tau = 0.1$), the Cosine Similarity still remains **96.0%** at 128 inference steps. As explained in Section 3.2, the Confidence-based approach only applies safe changes where the alternative is strictly dominant. However, the NLL-based approach performs more aggressively, correcting any position with high uncertainty. This fundamental difference makes the NLL- strategy more effective in resolving deep-seated fluency issues (as seen in Section 5.1), but at the cost of greater semantic drift.

5.3 LLM-Based Quality Evaluation

While measuring intrinsic quantitative metrics focuses on quantifying statistical likelihood, it is insufficient to evaluate high-level linguistic attributes such as logical coherence. Therefore, we use Gemma-3-12B as an external qualitative judge to conduct a comprehensive evaluation. The samples are scored on a scale of 1–10 across five distinct dimensions. The self-correction mechanism demonstrates the capability to refine low-quality generations into coherent text. Figure 3 (left) illustrates the progression of the Average Quality Score (mean of all five dimensions) throughout the inference steps. The baseline score for the quality evaluation is 1.75 for the generated samples.

Confidence-based approaches. As shown in Figure 3 (left), the confidence-based strategy yields the most conservative improvements. With an average score of 1.92, the naive approach ($K = 1$) has a **9.4%** overall improvement in quality. The multitoken approach ($K = 8$), in the meantime, gains an improvement of **25.2%** with the mean score of 2.19 at 128 steps.

NLL-based approaches. In terms of quality improvement, the NLL (dynamic threshold) strategy has the best performance among all the strategies. With the configuration $\tau = 0.1$, the average score outperforms by reaching 2.46 at 64 inference steps, with a relative improvement of **40.7%**. The NLL (top-K) strategy also illustrates significant quality gains. With the optimal configuration ($K = 2, \tau = 1.0$), the scores increases monotonically to **2.58** at 128 steps, which is a relative improvement of **+47.2%**.

Table 2 illustrates the qualitative impact of the self-correction mechanism. The largest observed gains in **grammaticality (+83.1%)** and **factuality (+66.0%)** confirm that the NLL-based strategy effectively resolves structural fractures and non-word hallucinations. Importantly, the concurrent rise in **creativity (+14.7%)** shows that over-smoothing does not apply, suggesting that self-correction successfully repairs errors without losing the unique narrative identity of the original generation.

5.4 Tradeoff Between Sample Fidelity and Quality Improvement

One of the fundamental challenges of self-correction mechanism is to find the optimal balance between keeping semantic similarity and improving quality. Aggressive strategies that modify many

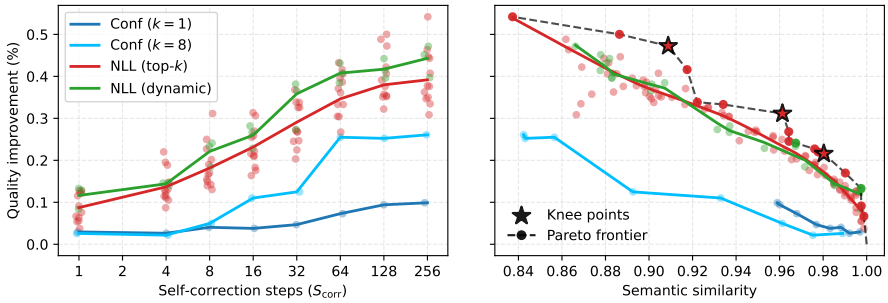

Figure 3: *Left:* The quality score consistently improves as the model iteratively refines the text. *Right:* The dashed line represents the Pareto frontier of NLL-based strategies. The *knee points* (marked with a star) identify optimal tradeoff configurations balancing substantial quality gains (22%, 31%, and 47%) with high semantic retention (98%, 96%, and 91%).

tokens often yield greater improvements, but could result in higher semantic drift. On the other hand, conservative strategies can stay closer to the original text, but may fail to resolve important quality issues. Investigating this tradeoff is essential for empirical applications. Instead of treating quality improvement and semantic similarity as isolated metrics, we investigate them together to identify the sweet spots. To this end, we use **Pareto frontier analysis** to eliminate inefficient strategies and **knee point detection** to find the configuration that offers the best quality while maintaining low semantic drift.

We apply Pareto Dominance Analysis (Lin et al., 2022)to identify the configurations representing optimal tradeoffs: a configuration is considered Pareto optimal if there is no other configuration increasing both the quality improvement and semantic similarity. The *Pareto frontier* consists of all such dominating configurations and is illustrated in Figure 3 (right). The analysis identifies 17 configurations on the frontier, all of which use NLL-based token selection strategies. Details on the pareto-optimal configurations are provided in Appendix B.2. This shows that NLL-based approaches strictly outperform the confidence-based ones, yielding better quality improvements at any given semantic preservation level while also taking fewer self-correction steps to do so.

Among the Pareto optimal configurations, we look for the *knee point*, where the marginal tradeoff between quality and fidelity exhibits the most significant change. Geometrically, this is the point of greatest curvature on the Pareto frontier. As illustrated in Figure 3 (right), there are three points with particularly high curvature. We pick the point with highest quality gain as the preferred configuration, but for settings with strict semantic preservation requirements, a different knee point may be preferable. The preferred configuration is given in Table 3 and achieves a quality improvement of **47.2%**, while maintaining **90.9%** semantic cosine similarity.

Table 3: Preferred configuration identified via knee point analysis.

Parameter	**Selected val.**
Selection strat.	NLL (top-K)
Sampling temp.	$\tau = 1.0$
Tokens per step	$K = 2$
Inference steps	$S_{\text{corr}} = 128$
Semantic sim.	90.88%
Quality improv.	+47.24%

This configuration exhibits some interesting characteristics. It combines a **high temperature** ($\tau = 1.0$), enabling diverse candidate sampling and exploration of alternative token choices, with **limited tokens per step** ($K = 2$), restraining the modification budget in each step and preventing overly aggressive modifications. With 128 correction step, it also features **extended refinement**. While not being excessively expensive, it provides sufficient steps for the necessary corrections to be performed. Qualitative examples produced with this configuration are provided in Appendix 1. Additionally, our experiments show that extending inference further (e.g., to 256 steps) yields **diminishing returns**, where doubling the computational cost for negligible quality gains does not pay off. Based on our analysis, we recommend the following settings:

- **High quality:** NLL (top-K; $K = 2, \tau = 1.0, S_{\text{corr}} = 128$) achieves a significant improvement in quality (+47.2%) while still preserving semantics (90.9%).
- **Balanced:** NLL (top-K; $K = 8, \tau = 0.1, S_{\text{corr}} = 8$) achieves a good balance between quality (+31.1%) and semantic similarity (96.1%).

- **High fidelity:** For tasks requiring minimal edits, the more conservative NLL (top-K; $K = 1$, $\tau = 0.1$, $S_{\text{corr}} = 16$) yields a +21.6% gain while keeping 98.0% semantic similarity.
- **Creative rewriting:** For scenarios where semantic drift is acceptable, higher temperatures and additional steps can further increase quality, with NLL (top-K; $K = 8$, $\tau = 1.0$, $S_{\text{corr}} = 256$) achieving a +54.2% improvement.

5.5 Optimal Allocation of Inference Compute

We further investigate the allocation of fixed inference budget between initial generation and correction. Let S_{tot} be the total number of steps. This is split into S_{gen} (denoising/generation steps) and S_{corr} (correction steps), where $S_{\text{tot}} = S_{\text{gen}} + S_{\text{corr}}$. We define the denoising ratio $r = S_{\text{gen}}/S_{\text{tot}}$. Using the same preferred configuration (Tab. 3) and the LLM-based Average Quality Score, we evaluate the tradeoff between generation steps and self-correction steps with different ratios r under three budgets $S_{\text{tot}} \in \{64, 128, 256\}$.

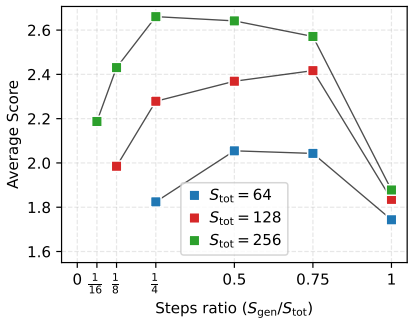

Figure 4: Balancing denoising and self-correction with a fixed inference budget.

Figure 4 shows a non-linear relationship between step ratio and average quality score. Spending the entire budget on generation ($r = 1.0$) produces the worst average quality score, since no budget is used for self-correction. In contrast, an excessively low ratio (e.g., $\leq 1/8$) limits the base generation phase, which leads to a drop in overall quality. This suggests that the self-correction mechanism struggles to fix overly-noisy initial sequences even with more inference budget. Empirically, we find that a balanced ratio of $r \approx 0.5$ (equal steps for generation and correction) achieves the best performance overall. For each budget, allocating the steps optimally between generation and self-correction results in a **+17.9%**, **+31.7%**, and **+41.7%** improvement for 64, 128, and 256 steps, respectively, compared to allocating all steps to generation. This indicates that it is always ideal to reserve at least some inference budget for post-generation self-correction steps, but also that self-correction requires a consistent semantic base to work well.

6 Conclusion

This paper conducted a systematic and comprehensive assessment of the self-correction mechanisms under the framework of GIDD, leveraging LLM judgment. Our results identify self-correction as an effective and promising way to improve the quality of text generation. We observed a tradeoff between quality gain and semantic retention, and demonstrated that one of our proposed strategies could strike a satisfactory balance. We also found that balancing the number of steps for denoising and self-correction improved the overall performance with a fixed inference budget. Based on the findings of this paper, future work could explore how to unify denoising and self-correction into a single process. It would also be interesting to combine self-correction with insertion/deletion operations to address variable-length text generation.

References

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces, 2021. URL https://arxiv.org/abs/2107.03006.

Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas Nolte, and Brian Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking, 2025. URL https://arxiv.org/abs/2505.24857.

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

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Aishwarya Kamath Gemma Team, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2020. URL https://arxiv.org/abs/1904.09751.

Zemin Huang, Yuhang Wang, Zhiyang Chen, and Guo-Jun Qi. Don't settle too early: Self-reflective remasking for diffusion language models, 2025. URL https://arxiv.org/abs/2509.23653.

Jaeyeon Kim, Seunggeun Kim, Taekyun Lee, David Z. Pan, Hyeji Kim, Sham Kakade, and Sitan Chen. Fine-tuning masked diffusion for provable self-correction, 2025a. URL https://arxiv.org/abs/2510.01384.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions, 2025b. URL https://arxiv.org/abs/2502.06768.

Xi Lin, Zhiyuan Yang, Xiaoyuan Zhang, and Qingfu Zhang. Pareto set learning for expensive multi-objective optimization, 2022. URL https://arxiv.org/abs/2210.08495.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin (eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language

models, 2024. URL https://arxiv.org/abs/2406.07524.

Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Nan Wang, and Han Xiao. jina-embeddings-v3: Multilingual embeddings with task LoRA, 2024. URL https://arxiv.org/abs/2409.10173.

Dimitri von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion, 2025. URL https://arxiv.org/abs/2503.04482.

Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling, 2025. URL https://arxiv.org/abs/2503.00307.

Table 4: Hyperparameter search space for each self-correction algorithm.

Method	Hyperparameters ranges		
Confidence-based	$\tau \in \{0.1\}$	$K \in \{1, 8\}$	$S_{\text{corr}} \in \{1, 4, 8, 16, 32, 64, 128, 256\}$
NLL (top-K)	$\tau \in \{0.01, 0.1, 1.0\}$	$K \in \{1, 2, 4, 8\}$	$S_{\text{corr}} \in \{1, 4, 8, 16, 32, 64, 128, 256\}$
NLL (dynamic)	$\tau \in \{0.01, 0.1, 1.0\}$	$\alpha \in \{0.9\}$	$S_{\text{corr}} \in \{1, 4, 8, 16, 32, 64, 128, 256\}$

A GIDD Details

Training objective (ELBO). A neural network $\mathbf{x}_\theta(z_t, t)$ is trained to predict clean data $\mathbf{x}$ from noisy input z_t. The training objective is derived from the Evidence Lower Bound (ELBO) of the continuous-time diffusion process. For a sequence $\mathbf{x}$, the negative ELBO can be simplified to a weighted reconstruction loss:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{t, z_t} \left[w_t(z_t, \mathbf{x}) \cdot \mathcal{L}_{\text{KL+IS}}(\mathbf{x}_\theta(z_t, t), \mathbf{x}) \right] \tag{10}$$

where $t \sim \mathcal{U}(0, 1)$, $z_t \sim q_t(z_t|\mathbf{x})$, and $\mathcal{L}_{\text{KL+IS}}$ is a reconstruction loss consisting of two divergence terms (Kullback-Leibler divergence and point-wise Itakura-Saito divergence). The weighting term $w_t(z_t, \mathbf{x})$ is essential in GIDD, as it assigns different weights to tokens depending on whether they are masked, corrupted by uniform noise, or remain as clean data. This enables the model to focus on the most informative parts of the sequence during denoising. See the original paper by von Rütte et al. (2025) for additional details on the loss function and training procedure.

Sampling and denoising. Generation is performed by ancestral sampling. Starting from a pure noise sequence z_1 (sampled from the prior $\boldsymbol{\pi}_1$), we iteratively denoise the sequence over a discrete time grid $1 = t_T > t_{T-1} > \cdots > t_0 = 0$. At each step $s \to t$ (where $s > t$), the next state is sampled according to the reverse transition kernel:

$$z_t \sim p_\theta(z_t|z_s) \propto q_{s|t}(z_s|z_t) q_t(z_t|\mathbf{x}_\theta(z_s, s)) \tag{11}$$

This process gradually removes noise—both masks and random tokens—to recover a coherent text sequence z_0. See the original paper for additional details on the sampling procedure.

B Key Hyperparameters

Tokens per step (K): This parameter determines the maximum number of tokens modified in one iteration (i.e., $|\mathcal{M}| \leq K$). A smaller K results in relative local and conservative changes. While a larger K enables the model to address multiple independent errors at the same time and converge faster.

Sampling temperature (τ): The sampling temperature is used to control how logits are sampled at the resampling phase. It effects the stochasticity of candidate generation. A low temperature results in a sharpened probability distribution, which makes the model more conservative and favors the high-probability corrections. While a higher temperature enables the model to explore more creative changes, but could potentially perform riskier rewrites.

Correction steps (S_{corr}): This defines the total budget for the refinement loop. Since self-correction happens iteratively, more inference steps allows for sequential improvements. However, it is important to determine the point of diminishing returns where additional computational effort does not yield significant quality gains.

Dynamic threshold ratio (α): This is the NLL-based dynamic threshold strategy in 3.2, where this ratio defines the sensitivity of error detection. It sets the cutoff threshold relative to the maximum uncertainty in the current sequence so that the corrections are restricted to the most relevant problematic tokens in the present context.

B.1 Hyperparameter ranges for self-correction

Table 4 summarizes the hyperparameter ranges explored for each inference-time self-correction algorithm. We report the full grid used in our runs (16 configurations for confidence-based selection, 96 for NLL (top-K), and 24 for NLL dynamic thresholding).

Table 5: All Pareto-optimal configurations and their semantic similarity and quality improvement.

Method	τ	K	S_{corr}	α	Sim. (%)	Qual. gain (%)
NLL (top-K)	0.01	1	1	-	99.87	6.69
NLL (top-K)	1.0	2	1	-	99.75	9.14
NLL (dynamic)	0.1	-	1	0.9	99.74	13.30
NLL (top-K)	0.1	1	8	-	99.02	16.97
NLL (top-K)	0.01	1	16	-	98.04	18.30
NLL (top-K)	0.1	1	16	-	98.03	21.57
NLL (top-K)	1.0	8	4	-	97.77	22.00
NLL (top-K)	0.1	4	8	-	97.61	22.71
NLL (dynamic)	0.01	-	8	0.9	96.74	24.10
NLL (top-K)	0.01	1	32	-	96.43	24.52
NLL (top-K)	0.1	1	32	-	96.42	26.82
NLL (top-K)	0.1	8	8	-	96.13	31.14
NLL (top-K)	1.0	2	64	-	93.41	33.31
NLL (top-K)	1.0	8	32	-	92.22	33.86
NLL (top-K)	0.01	2	64	-	91.75	41.61
NLL (top-K)	1.0	2	128	-	90.88	47.24
NLL (top-K)	1.0	4	128	-	88.64	50.02
NLL (top-K)	1.0	8	256	-	83.73	54.20

B.2 Pareto-optimal configurations

Table 5 reports the exact configurations corresponding to the three knee points highlighted in the Pareto plot (Fig. 3).

C Evaluation Metrics

To systematically assess the performance of the self-correction framework, a combination of metrics is employed, including intrinsic model confidence, fidelity, and LLM-Based evaluation. Each of the metrics will be introduced below. Let $\mathbf{x} = (x_1, \ldots, x_L)$ be a text sequence of length L, and $\mathcal{V}$ be the vocabulary size.

C.1 Quantitative Quality Metrics

ELBO-Based Self-Perplexity (Self-PPL) To evaluate the intrinsic quality of the generated and corrected sequences, we utilize the model's own probabilistic estimate. Since the exact marginal likelihood $p(\mathbf{x})$ is not tractable for diffusion models, we rely on the Evidence Lower Bound (ELBO) introduced in Section 2 as an alternative. We define **Self-Perplexity** as the exponential of the negative ELBO evaluated on the generated sequence itself.

Recall that the training objective maximizes the ELBO, which serves as a lower bound to the log-likelihood. For a fixed sequence $\mathbf{x}$, the negative log-likelihood (NLL) can be approximated by discretizing the continuous time integral over $t \in [\epsilon, 1-\epsilon]$. We employ a uniform discretization strategy with N time steps (consistent with the evaluation protocol in GIDD):

$$t_i = \epsilon + \frac{i-1}{N-1}(1-2\epsilon), \quad i = 1, \ldots, N \tag{12}$$

The per-sequence NLL is estimated by averaging the weighted reconstruction loss over these time points:

$$\text{NLL}(\mathbf{x}) \lesssim \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{z_{t_i}\sim q_{t_i}(\cdot|\mathbf{x})}\Big[w_{t_i}(z_{t_i}, \mathbf{x}) \cdot \mathcal{L}_{\text{KL+IS}}(\mathbf{x}_\theta(z_{t_i}, t_i), \mathbf{x})\Big] \tag{13}$$

where $w(t_i)$ is the importance weighting term derived from the noise schedule, and the expectation over z_{t_i} is approximated using single-sample estimation.

To obtain a token-level metric, we average this value over the valid sequence length. Let $m_j = \mathbb{I}(\mathbf{x}_j = \text{pad})$ be the indicator variable for padded tokens. The average per-token NLL is:

$$\overline{\text{NLL}} = \frac{\sum_{j=1}^{L} m_j \cdot \text{NLL}(\mathbf{x})_j}{\sum_{j=1}^{L} m_j} \tag{14}$$

Finally, the Self-Perplexity is defined as:

$$\text{Self-PPL} = \exp\left(\overline{\text{NLL}}\right) \tag{15}$$

Self-PPL is expected to decrease after the self-correction process, as the corrected sequence lies in a higher density region of the model's distribution.

Generative PPL To objectively evaluate the quality of generated text, we use an external language model p_{LM} (GPT-2 Large) to compute the generative perplexity. Let $M = \sum_{j=1}^{L} m_j$ denote the count of tokens excluding padding. The metric is computed as:

$$\text{Gen-PPL}(\mathbf{x}) = \exp\left(-\frac{1}{M} \sum_{i=1}^{L} m_i \cdot \log p_{\text{LM}}(x_i | x_{<i}) \right) \tag{16}$$

Sequence Entropy Sequence Entropy is computed to measure the diversity of a given sequence by proxy of empirical entropy of the token distribution for each sequence. We denote c_v the count of token v in $\mathbf{x}$, and $\hat{p}(v) = c_v / (\sum c_v)$ its observed frequency. It is important to note that the summation is taken exclusively over the unique tokens appearing in $\mathbf{x}$, not the entire vocabulary. The entropy is given by the following:

$$H(\mathbf{x}) = -\sum_{v \in \mathcal{V}_{\mathbf{x}}} \hat{p}(v) \ln \hat{p}(v) \tag{17}$$

We monitor the average entropy per sequence to keep track whether the model falls into repetitive loops (low entropy) or maintains diversity.

C.2 Fidelity Metrics

BLEU Score To quantify the lexical fidelity, we compute the BLEU score (Papineni et al., 2002), which measures the n-gram overlap between the original generation $\mathbf{x}_{\text{orig}}$ and the corrected sequence $\mathbf{x}_{\text{corr}}$. The metric is defined as the geometric mean of modified n-gram precisions, adjusted by a brevity penalty BP to penalize overly short generations.

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{4} w_n \cdot \log p_n \right), \tag{18}$$

where $w_n = 1/4$ are uniform weights.

Token Change Count We calculate the Hamming distance between the original and corrected sequences to quantify the magnitude of change, with m_i again denoting the padding indicator variable:

$$\Delta(\mathbf{x}_{\text{orig}}, \mathbf{x}_{\text{corr}}) = \sum_{i=1}^{L} m_i \cdot \mathbb{I}(x_{\text{orig},i} \neq x_{\text{corr},i}) \tag{19}$$

Semantic Cosine Similarity Beyond BLEU score which measures surface-level wording, the preservation of semantic meaning also needs to be quantified. We uses **Jina Embeddings v3** (see Appendix E; Sturua et al., 2024). Let $E(\cdot)$ be the embedding function. The similarity is defined as the cosine of the angle between the sentence embeddings:

$$\text{Sim}(\mathbf{x}_{\text{orig}}, \mathbf{x}_{\text{corr}}) = \frac{E(\mathbf{x}_{\text{orig}}) \cdot E(\mathbf{x}_{\text{corr}})}{\|E(\mathbf{x}_{\text{orig}})\| \|E(\mathbf{x}_{\text{corr}})\|} \tag{20}$$

D GENERATOR MODEL CHECKPOINT DETAILS

The samples generated in this thesis utilize the **GIDD-Base** model trained with a uniform noise probability of $p_{\text{unif}} = 0.2$. The model weights, configuration files, and usage instructions are publicly hosted on the Hugging Face Hub.

- **Model ID:** `dvruette/gidd-base-p_unif-0.2`
- **URL:** `https://huggingface.co/dvruette/gidd-base-p_unif-0.2`

To reproduce the generation results, the model can be loaded directly using the Transformers library or the GIDD codebase provided in our supplementary material.

E SEMANTIC EMBEDDING MODEL DETAILS

For the semantic similarity evaluation, we employ **jina-embeddings-v3**, a high-performance multilingual text embedding model developed by Jina AI.

- **Model ID:** `jinaai/jina-embeddings-v3`
- **URL:** `https://huggingface.co/jinaai/jina-embeddings-v3`

F QUALITATIVE JUDGE MODEL DETAILS

For the qualitative evaluation, we utilize the 12-billion parameter variant of the Gemma 3 family.

- **Model ID:** `gemma3:12b`
- **Inference engine:** Ollama, 4-bit quantized (GGUF Q4_K_M)
- **URL:** `https://ollama.com/library/gemma3:12b`

G COMPREHENSIVE BREAKDOWN OF QUALITATIVE METRICS

Table 6 provides a comprehensive evaluation of all experimental configurations at 128 inference steps. We report the LLM-based quality scores across five dimensions (clarity, grammaticality, factuality, style, creativity) for various combinations of sampling temperatures (τ) and tokens per step (K).

H QUALITATIVE EVALUATION PROMPTS

To ensure reproducibility, we provide the exact prompts used to query the Gemma 3 12B judge model via the Ollama API.

H.1 SYSTEM INSTRUCTION

The system prompt enforces strict output formatting to facilitate automated parsing:

```
You are a strict grader. Return ONLY valid JSON.
Do not include markdown fences or extra text.
```

H.2 USER PROMPT TEMPLATE

The user prompt template is used for each evaluation is given in Figure 5. The placeholder `{text}` is replaced by the actual text sample generated by the GIDD model.

Table 6: Full breakdown of qualitative metrics for all configurations at 128 inference steps.

Configuration	**Clar.**	**Gram.**	**Fact.**	**Style**	**Creat.**	**Avg.**	**Sim. (%)**
Generated (no correction)	1.59	1.66	1.71	1.50	2.30	1.75	100
Confidence-based							
$K = 1, \tau = 0.1$ **(baseline)**	1.72	1.89	1.90	1.66	2.41	1.92	96.0
$K = 8, \tau = 0.1$	2.09	2.24	2.39	1.89	2.35	2.19	84.3
NLL (top-K; $\tau = 0.01$)							
$K = 1$	2.14	2.58	2.42	2.11	2.57	2.37	90.6
$K = 2$	2.12	2.61	2.42	2.12	2.57	2.37	90.0
$K = 4$	2.21	2.61	2.49	2.17	2.57	2.41	88.8
$K = 8$	2.20	2.58	2.48	2.14	2.48	2.38	87.4
NLL (top-K; $\tau = 0.1$)							
$K = 1$	2.13	2.64	2.41	2.14	2.54	2.37	90.5
$K = 2$	2.24	2.68	2.46	2.21	2.59	2.44	89.7
$K = 4$	2.27	2.71	2.56	2.23	2.54	2.46	88.5
$K = 8$	2.34	2.73	2.65	2.22	2.52	2.49	86.9
NLL (top-K; $\tau = 1.0$)							
$K = 1$	2.05	2.51	2.34	2.04	2.63	2.31	92.6
$K = 2$ **(preferred)**	2.34	2.83	2.63	2.32	**2.77**	2.58	90.9
$K = 4$	2.42	2.91	2.69	2.38	2.73	**2.63**	88.6
$K = 8$	**2.42**	**2.92**	**2.71**	**2.36**	2.62	2.61	86.2
NLL (dynamic; $\alpha = 0.9$)							
$\tau = 0.01$	2.20	2.70	2.56	2.21	2.55	2.44	88.3
$\tau = 0.1$	2.26	2.74	2.58	2.23	2.56	2.48	87.8
$\tau = 1.0$	2.32	2.75	2.58	2.28	2.69	2.53	88.2

User Prompt Template

```
1. Clarity and coherence: Keeping in mind that the text may be cut
off in the beginning and at the end due to it being an excerpt, how
 clear and understandable is the text?
2. Grammaticality: Are there any grammatical errors in the text?
3. Factuality: If applicable, is the factually verifiable
information stated in the text (e.g. facts about geography, history
, etc.) accurate and reliable?
4. Writing style: How well is the text written in terms of style
and fluency? Do the sentences flow well, is the vocabulary
appropriate?
5. Creativity: How original and creative is the text?
For each category, give a short justification before providing the
final score. Your answer should be following the JSON format, with
one top-level key for each aspect ('clarity', 'grammaticality', '
factuality', 'style', and 'creativity').
Each aspect, in turn, should be a JSON object consisting of a '
reasoning' and 'score' key in that order. The 'reasoning' key
should contain a short justification for the score, and the 'score'
 key should contain the score itself.
Please keep the following in mind:
- Give your justification first before deciding on a final score.
- Only output the JSON containing the justifications and scores and
 nothing else.
- Keep in mind that the presented paragraph may be an excerpt from
a longer document, so it may not be fully self-contained. Do not
deduct points for issues arising from this.
The text to be graded is as follows:
'''
{text}
'''
```

Figure 5: User prompt used for the LLM-based evaluation of sample quality.