Review, Remask, Refine (R3): Process-Guided Block Diffusion for Text Generation

Nikita Mounier [*1] **Parsa Idehpour** [*2]

Abstract

A key challenge for iterative text generation is enabling models to efficiently identify and correct their own errors. We propose Review, Remask, Refine (R3), a relatively simple yet elegant framework that requires no additional model training and can be applied to any pre-trained masked text diffusion model (e.g., LLaDA or BD3-LM). In R3, a Process Reward Model (PRM) is utilized for the **Review** of intermediate generated blocks. The framework then translates these PRM scores into a **Remask** strategy: the lower a block's PRM score, indicating potential mistakes, the greater the proportion of tokens within that block are remasked. Finally, the model is compelled to **Refine** these targeted segments, focusing its efforts more intensively on specific sub-optimal parts of past generations, leading to improved final output.

1. Introduction

Recent advancements in generative modeling have seen the rise of masked diffusion models (MDMs) for text generation, offering a compelling alternative to traditional autoregressive approaches (Nie et al., 2025). These models operate by iteratively refining a sequence, often starting from a masked state and progressively unmasking or de-noising tokens to produce coherent text. This iterative, non-autoregressive (or partially autoregressive when using block diffusion techniques (Arriola et al., 2025)) nature presents unique opportunities for controlled and high-quality text synthesis, allowing for the potential to revisit and improve discrete parts of the sequence.

However, a significant challenge lies in effectively guiding this iterative refinement process. Specifically, enabling these models to efficiently identify and correct their own earlier, less optimal generations is essential for maximizing output quality. While general remasking capabilities allow models to revisit previously generated tokens (Wang et al., 2025), the strategy determining *what* to remask is paramount for efficiency and impact.

To address this, we leverage the capabilities of Process Reward Models (PRMs). Unlike outcome-based reward models that only evaluate the final generated output, PRMs provide fine-grained feedback on the intermediate steps or components of a generation process ((Zheng et al., 2024), (Zhang et al., 2025)). This makes PRMs exceptionally well-suited for guiding iterative models, as they can offer nuanced assessments of partially generated sequences or specific text blocks.

In this paper, we introduce **Review, Remask, Refine (R3)**, a novel, relatively simple yet elegant framework that combines the strengths of masked text diffusion models with the targeted guidance of PRMs. R3 facilitates a cycle of self-correction within the generation process:

1. **Review:** A Process Reward Model (PRM) is utilized for the **Review** of the quality of currently generated text blocks.

2. **Remask:** Based on the PRM scores obtained during the review, R3 implements a remasking strategy. Critically, the lower a block's PRM score – indicating potential issues or "mistakes" – the greater the proportion of tokens *within* that block are remasked.

[*]Equal contribution [1]Jerome Fisher M&T Program, University of Pennsylvania, Philadelphia, USA [2]Computer Information Sciences, University of Pennsylvania, Philadelphia, USA. Correspondence to: Nikita Mounier <nikita.mounier@seas.upenn.edu>, Parsa Ideahpour <parsa.ideahpour@seas.upenn.edu>.

Accepted at Methods and Opportunities at Small Scale (MOSS), ICML 2025, Vancouver, Canada.

3. **Refine:** The masked diffusion model is then prompted to refine these remasked segments, thereby focusing its generative capacity on the areas identified as weakest by the PRM.

A key advantage of the R3 framework is that it requires **no additional training or fine-tuning of the underlying masked diffusion model** and can be readily integrated with various existing pre-trained models (e.g., LLaDA or BD3-LM). By strategically directing the model to revisit and improve specific sub-optimal segments of its past generations, R3 offers an efficient pathway to higher-quality text. Our contributions are: (1) the R3 framework for PRM-guided iterative refinement in masked text diffusion; (2) the specific mechanism of PRM-score-dependent proportional remasking for targeted error correction; and (3) a demonstration of how this approach can enhance text generation without requiring model retraining.

2. Related Work

Our Review, Remask, Refine (R3) framework integrates concepts from masked diffusion models for text, process-guided supervision, and iterative refinement strategies.

Masked and Block Diffusion Models Recent text generation approaches include masked diffusion models like LLaDA (Nie et al., 2025), which iteratively denoise sequences, and Block Diffusion (BD3-LMs) (Arriola et al., 2025), which combine autoregressive block-wise generation with intra-block discrete diffusion. These models offer iterative refinement capabilities. R3 is designed to leverage these models by intelligently guiding their block-wise refinement process.

Process Reward Models (PRMs) for Guidance Unlike Outcome Reward Models (ORMs) that score only final outputs (Ouyang et al., 2022), PRMs (Zheng et al., 2024; Zhang et al., 2025) evaluate intermediate steps, providing granular feedback ideal for iterative generation. In R3, the "Review" stage employs a PRM, whose score S_b for a block b directly informs the subsequent "Remask" stage, focusing refinement on identified weaknesses.

Iterative Refinement and Guided Remasking The ability to correct prior generations is essential. **ReMDM** (Wang et al., 2025) allows already generated tokens in masked diffusion models to be remasked and updated at inference time. R3 advances this by introducing a PRM-guided remasking strategy. The proportion of tokens remasked within block b, $P_{\text{remask}}(b)$, is a decreasing function of its PRM score S_b (i.e., $P_{\text{remask}}(b) = f(S_b)$ with $f'(S_b) < 0$). This targeted approach, requiring no retraining of the base model, distinguishes R3 by focusing refinement efforts on PRM-identified suboptimal segments, offering a simple yet effective integration of process supervision. Other strategies to enhance reasoning in dLLMs, such as the d1 framework (Zhao et al., 2025), adapt pre-trained masked dLLMs using a combination of supervised finetuning (SFT) and reinforcement learning (RL). This contrasts with R3's approach, which aims to improve generation quality at inference time without requiring additional post-training of the base text diffusion model.

3. The R3 Framework

We introduce the Review, Remask, Refine (R3) framework, an iterative inference-time procedure designed to enhance the quality of text generated by masked diffusion models. R3 leverages Process Reward Models (PRMs) to provide targeted feedback for error correction and refinement, without requiring retraining of the base generative model or the PRM. The framework operates by cyclically generating text in discrete blocks, assessing these blocks with a PRM, selectively remasking segments of blocks identified as suboptimal, and then employing the base diffusion model to refine these remasked portions.

3.1. Core Components

The R3 framework is built upon two essential pre-trained models. The base Masked Diffusion Model (M_{diff}) is a generative model proficient at infilling masked regions within a text sequence. Given an input sequence x containing masked tokens x_{mask}, M_{diff} produces a prediction $\hat{x}_0 = M_{\text{diff}}(x; x_{\text{mask}})$ for the original content. The Process Reward Model (M_{PRM}) evaluates the quality of individual text segments. For a given text block x_b conditioned on prior context C_b, M_{PRM} outputs a scalar quality score $S_b = M_{\text{PRM}}(x_b|C_b)$, where $S_b \in [0, 1]$ and higher scores denote superior quality.

3.2. The R3 Iterative Process with Windowed Refinement

R3 constructs text block by block. A central feature of our approach is a windowed evaluation and batched refinement mechanism, allowing for efficient application of PRM guidance. The iterative process is as follows:

Let $X^{(j)}$ be the sequence generated up to block $j-1$. For each new block $j = 0, \ldots, N_{\text{total}} - 1$:

1. **Initial Block Generation (Extend):** Generate the content for the current block x_j using M_{diff}, conditioned on $X^{(j)}$. This step involves an iterative demasking procedure where initially masked tokens for block x_j are progressively filled. The sequence is updated: $X^{(j+1)} = X^{(j)} \oplus x_j$, where $\oplus$ denotes concatenation.

2. **Windowed Review (Score Blocks):** Periodically, typically every K blocks (the window size), the last K generated blocks $W_j = \{x_{\max(0,j-K+1)}, \ldots, x_j\}$ are evaluated. For each sequence in the batch, M_{PRM} assesses each block $x_b \in W_j$, yielding a set of scores $\mathcal{S}_{W_j} = \{S_{\max(0,j-K+1)}, \ldots, S_j\}$. These scores are stored.

3. **Refinement Trigger:** For each sequence in the batch, if its minimum score within the current window $\min(\mathcal{S}_{W_j})$ falls below a predefined threshold τ_{thresh}, a refinement cycle is initiated for its window W_j.

4. **Candidate Generation for Refinement (Remask & Propose):** For each sequence whose window W_j triggered refinement:

   - **Calculate Remasking Probabilities:** For each block $x_b \in W_j$ with score S_b, a remasking probability $P_R(S_b)$ is determined. $P_R(S_b)$ is a monotonically decreasing function of S_b; for instance, $P_R(S_b) \propto \exp(\alpha_B(1 - S_b))$, where α_B is a scaling parameter. This ensures that lower-quality blocks are more likely to be remasked more extensively.
   - **Token-level Remasking:** Within each block $x_b \in W_j$, a proportion of its tokens, $\rho_b = \beta_I \cdot \tilde{P}_R(S_b)$ (where $\tilde{P}_R(S_b)$ might be a normalized or clipped version of $P_R(S_b)$ and β_I is an intensity factor), are selected and replaced by '[MASK]' tokens. This results in a masked window $W_{j,\text{masked}}$.
   - **Propose Refinements:** N_S candidate refined versions of $W_{j,\text{masked}}$, denoted $\{\hat{W}_j^{(s)}\}_{s=1}^{N_S}$, are generated by applying M_{diff} to N_S copies of $X^{(j-K+1)} \oplus W_{j,\text{masked}}$. This step is typically batched for efficiency.

5. **Candidate Scoring (Review Refined Candidates):** Each of the N_S refined candidate windows $\hat{W}_j^{(s)}$ is scored using M_{PRM}, yielding sets of scores $\{\mathcal{S}_{W_j}^{(s)'}\}_{s=1}^{N_S}$.

6. **Selection and Update (Refine):** For each sequence that underwent refinement, the optimal candidate window $\hat{W}_j^{(s^*)}$ is selected from the N_S proposals. The selection is based on a metric $\mathcal{M}(\cdot)$ applied to the candidate scores, e.g., $s^* = \arg\max_s \mathcal{M}(\mathcal{S}_{W_j}^{(s)'})$, where $\mathcal{M}$ could be the product of scores or the minimum score within the window. The main sequence $X^{(j+1)}$ is updated by replacing its window W_j with the selected $\hat{W}_j^{(s^*)}$, and the stored scores for these blocks are updated accordingly. If refinement was not triggered or no candidate offers improvement, the original blocks W_j are retained.

This R3 cycle allows the model to progressively build text while continuously evaluating and correcting prior segments, thereby focusing refinement efforts on the passages most in need of improvement as identified by the PRM. The detailed pseudocode is presented in Algorithm 1 below.

3.3. Algorithm Pseudocode

The R3 framework with windowed refinement is formalized in Algorithm 1.

4. Experiments

We evaluate the efficacy of our Review, Remask, Refine (R3) framework, focusing on its ability to correct errors in mathematical reasoning tasks and comparing its performance and computational characteristics against relevant baselines.

4.1. Experimental Setup

We utilize LLaDA-8B-Instruct (Nie et al., 2025) as the base masked diffusion model (M_{diff}) for generation and refinement. For the "Review" stage (and for the BoN baseline), we employ Qwen2.5-Math-PRM-7B (Zhang et al., 2025) as our Process Reward Model (M_{PRM}).

Experiments were conducted on a subset of 127 questions from the MATH 500 dataset, focusing on problems requiring step-by-step derivations.

Algorithm 1 R3 Framework with Windowed Refinement

Input: Initial prompt $X^{(0)}$, Base Diffusion Model M_{diff}, PRM M_{PRM}
Parameters: Total blocks N_{total}, block length L_B, window size K, threshold τ_{thresh}, num samples N_S, intensity β_I, alpha α_B
Let $X \leftarrow X^{(0)}$; $S_{\text{all}} \leftarrow \emptyset$ (list of lists for scores per block per batch item)
for block index $j = 0$ **to** $N_{\text{total}} - 1$ **do**
  $x_j \leftarrow M_{\text{diff}}(X|\text{mask for block } j)$ {Generate current block}
  $X \leftarrow X \oplus x_j$
  Append placeholder for S_j to S_{all} for each batch item.
  if $(j+1) \bmod K == 0$ **or** $j == N_{\text{total}} - 1$ **then**
    Let window $W = \{x_{\max(0,j-K+1)}, \ldots, x_j\}$
    For each item b_{item} in batch: $S_W[b_{\text{item}}] \leftarrow \{M_{\text{PRM}}(x_b|X_{<b}) \text{ for } x_b \in W[b_{\text{item}}]\}$. Update S_{all}.
    for each item b_{item} in batch **do**
      if $\min(S_W[b_{\text{item}}]) < \tau_{\text{thresh}}$ **then**
        For each $x_b \in W[b_{\text{item}}]$, calculate remasking prob $P_R(S_b)$ using α_B.
        For $s = 1 \ldots N_S$:
        Create $X^{(s)}_{\text{masked}}$ by masking tokens in $W[b_{\text{item}}]$ of a copy of $X[b_{\text{item}}]$ based on $P_R(S_b)$ and β_I.
        $\hat{W}^{(s)} \leftarrow \{M_{\text{diff}}(X^{(s)}_{\text{masked}}|\text{mask for block } b) \text{ for each block } b \text{ in window}\}$ {Refine candidates}
        Score all $\{\hat{W}^{(s)}\}_{s=1}^{N_S}$ using M_{PRM} to get $\{\mathcal{S}_W^{(s)'}\}$.
        $s^* \leftarrow \arg\max_s \mathcal{M}(\mathcal{S}_W^{(s)'})$.
        Replace window in $X[b_{\text{item}}]$ with $\hat{W}^{(s^*)}$.
        Update corresponding scores in $S_{\text{all}}[b_{\text{item}}]$ with $\mathcal{S}_W^{(s*)'}$.
      end if
    end for
  end if
end for
Output: Final sequence X

Key hyperparameters for R3: sampling temperature 0.8, PRM threshold $\tau_{\text{thresh}} = 0.8$, backmasking intensity $\beta_I = 0.8$, candidate samples $N_S = 5$, score mapping factor $\alpha_B = 10.0$. Sequences used 16 blocks of 32 tokens (512 total) with 128 demasking steps. Window size K was varied across experiments.

We compare R3 against two main baselines. First, pass@1 represents standard sequential block-by-block generation from LLaDA-8B-Instruct with a sampling temperature of 0.8. No PRM guidance or re-evaluation is used. Second, Block-wise Best of N (B-BoN) generates $N_S = 5$ candidate versions for each block in the sequence using M_{diff}. Each candidate block is then scored by M_{PRM}, and the highest-scoring candidate block is selected to extend the sequence before proceeding to the next block. This represents a strong baseline that heavily utilizes the PRM at each step.

4.2. Qualitative Example: Error Correction Cycle

The R3 framework identifies and corrects errors by iteratively reviewing, remasking, and refining generated blocks. An illustrative example of this error correction cycle on an intermediate step of a trigonometric problem is provided in Appendix A, specifically in Figure 1.

4.3. Quantitative Results

We evaluated the accuracy (number of correctly solved problems) on the 127-question MATH subset. The results are summarized in Table 1.

4.4. Discussion

R3 with $K = 8$ substantially improves over the simple diffusion baseline, increasing accuracy from 29.13% to 42.52%, demonstrating the benefit of PRM-guided iterative refinement.

Table 1. Accuracy on MATH 500 subset (127 questions). K is the R3 window size; $N_S = 5$ for R3 and BoN.

METHOD	CORRECT / TOTAL	ACCURACY (%)
SIMPLE DIFFUSION (PASS@1)	37 / 127	29.13%
R3 ($K = 4$)	42 / 127	33.07%
R3 ($K = 5$)	44 / 127	34.65%
R3 ($K = 8$)	**54 / 127**	**42.52%**
BLOCK-WISE BEST OF N (BON)	61 / 127	48.03%

While block-wise BoN achieves the highest accuracy (48.03%), it requires $N_{\text{total}} \times N_S$ PRM calls (e.g., $16 \times 5 = 80$ calls for a 16-block sequence). R3 is significantly more efficient: in the best case, it makes only $\lceil N_{\text{total}}/K \rceil$ calls (e.g., 2 for $K = 8$); in the worst case, $2 \times \lceil N_{\text{total}}/K \rceil$ calls (4 for $K = 8$). Thus, R3 achieves strong accuracy with far fewer PRM calls (2-4 vs. 80) by strategically targeting only problematic windows.

Larger window sizes improve performance: increasing K from 4 to 8 enhanced accuracy, suggesting that evaluating larger segments allows better contextual assessment and more impactful refinements. R3 offers a compelling balance between performance gains and computational efficiency.

5. Conclusion

We introduced Review, Remask, Refine (R3), an inference-time framework that enhances masked diffusion models using Process Reward Models (PRMs). R3 iteratively generates text blocks, reviews them with a PRM, strategically remasks low-quality tokens, and refines these segments. Its windowed evaluation strategy enables targeted error correction without retraining.

Our experiments on MATH 500 showed R3 significantly improves accuracy over simple diffusion baselines, increasing from 29.13% to 42.52% with $K = 8$. R3 is also computationally efficient, requiring far fewer PRM calls than block-wise Best of N while achieving strong performance. Larger review windows proved beneficial, suggesting that more context enables better quality improvements.

R3 offers a promising approach for improving masked diffusion models by integrating process supervision at inference time. Future work could explore adaptive window sizes and sophisticated remasking strategies. A key research direction involves developing self-correction capabilities within the base model during post-training, potentially reducing reliance on external PRMs. Applying R3 to broader text generation tasks and diffusion architectures warrants further exploration.

References

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

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

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Wang, G., Schiff, Y., Sahoo, S., and Kuleshov, V. Remasking discrete diffusion models with inference-time scaling. *arXiv preprint arXiv:2503.00307*, 2025.

Zhang, Z., Zheng, C., Wu, Y., Zhang, B., Lin, R., Yu, B., Liu, D., Zhou, J., and Lin, J. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*, 2025.

Zhao, S., Gupta, D., Zheng, Q., and Grover, A. d1: Scaling reasoning in diffusion large language models via reinforcement learning, 2025. URL https://arxiv.org/abs/2504.12216.

Zheng, C., Zhang, Z., Zhang, B., Lin, R., Lu, K., Yu, B., Liu, D., Zhou, J., and Lin, J. ProcessBench: identifying process errors in mathematical reasoning. *arXiv preprint arXiv:2412.06559*, 2024.

A. Qualitative Example of R3 Error Correction Cycle

Figure 1 illustrates the R3 error correction cycle. The framework identifies an initial error in a generated block, uses the PRM score to guide targeted remasking, and then refines the remasked segment to produce a corrected output.

1. LLM Output – Before R3 Intervention (Initial Error)	**2. Conceptual State – After R3 Remasking (Based on Low PRM Score)**	**3. LLM Output – After R3 Refinement**
Step 4: Determine the value of b The period T of the sine function $y = a\sin(bx + c) + d$ is given by: $T = \frac{2\pi}{b}$ From Step 3, we know $T = \pi$. Therefore, we can set up the equation: $\pi = \frac{2\pi}{b}$ Solving for b: $b = \frac{2\pi}{\pi} = \mathbf{3}$	[MASK] Step 4: Determine the value of (b) The period (T) of the sine function ($y = a\sin(bx + c) + d$) is given by: $T = \frac{2\pi}{b}$ [MASK][MASK][MASK] From [MASK][MASK] 3, [MASK][MASK] ([MASK] = [MASK]). Therefore [MASK][MASK][MASK] set [MASK] the equation: $\pi = \frac{2\pi}{b}$ [MASK][MASK][MASK] [MASK] Solving for [MASK] b [MASK][MASK][MASK]	Step 4: Determine the value of b The period T of the sine function $y = a\sin(bx + c) + d$ is given by: $T = \frac{2\pi}{b}$ From Step 3, we know $T = \pi$. Therefore, we can set up the equation: $\pi = \frac{2\pi}{b}$ Solving for b: $b = \frac{2\pi}{\pi} = \mathbf{2}$

Figure 1. Qualitative example of the R3 error correction cycle. (1) An initial error in calculation. (2) Conceptual representation of the block after the PRM "Review" leads to targeted "Remasking". (3) The "Refine" stage produces the corrected calculation.

B. PRM Score to Remasking Probability Calculation

We detail the conversion of a PRM score $S_b \in [0, 1]$ for a block x_b into a remasking probability $P_R(S_b)$. This probability determines the likelihood and extent of remasking for that block if its window triggers refinement.

1. An intermediate quality value q_b is first computed from the PRM score S_b using an exponential decay function: $q_b = \exp(-\alpha_B \cdot S_b)$. The hyperparameter α_B (backmasking alpha, e.g., 10.0 in our experiments) controls the steepness of this transformation; a higher α_B makes the q_b value more sensitive to small differences in S_b, especially penalizing higher scores more rapidly.

2. These q_b values for all blocks within the current scoring window W are then normalized to a probability range $[p_{\text{min}}, 1]$. Let $\{q_{b'}\}_{b' \in W}$ be the set of these intermediate values for the window. The final remasking probability for block x_b is:

$$P_R(S_b) = p_{\text{min}} + (1 - p_{\text{min}}) \cdot \frac{q_b - \min_{b' \in W} q_{b'}}{\max_{b' \in W} q_{b'} - \min_{b' \in W} q_{b'} + \epsilon}$$

   where p_{min} is a minimum remasking probability (e.g., 0.01) to ensure even high-scoring blocks have a small chance of being re-evaluated, and ϵ is a small constant for numerical stability. This formula effectively maps lower PRM scores (which result in higher q_b) to higher remasking probabilities.