RACC: Regret-Aware Confidence Calibration for Consistent Masked Discrete Diffusion Decoding

Qinglin Zeng, Jusheng Zhang, Jing Yang, Ningyuan Liu, Keze Wang*
Sun Yat-sen University
Guangzhou, China

Abstract

Masked Discrete Diffusion Models (MDMs) enable parallel generation via iterative refinement. However, we identify a critical decisional mismatch. The MDM architecture is inherently dynamic and capable of sensing context shifts. In contrast, prevailing decoding paradigms remain static and myopic. They treat each denoising step as an isolated snapshot, effectively discarding valuable temporal feedback that signals logical conflicts. To bridge this gap, we propose Regret-Aware Confidence Calibration (RACC). This training-free framework aligns decoding decisions with the model's latent self-correction capabilities. RACC introduces a *momentum anchor* to track confidence trajectories. When a token's probability drops abruptly below its historical trend, the system triggers a "regret" signal. Unlike expensive re-masking or lookahead search, RACC utilizes this signal to proactively demote unstable candidates. Extensive experiments on reasoning benchmarks, such as HumanEval and GSM8K, demonstrate that RACC significantly improves generation consistency. Crucially, RACC achieves these gains with zero additional inference overhead, effectively balancing decoding quality and efficiency.

1 Introduction

Diffusion Language Models (dLLMs) have emerged as formidable competitors to the dominant Autoregressive (AR) architectures, garnering significant attention for their superior bidirectional context awareness and potential for parallel generation. Theoretically, the bidirectional attention mechanism in dLLMs provides the model with a "global perspective," enabling it to dynamically evaluate the logical consistency of the entire sequence during each denoising step. This characteristic endows dLLMs with an inherent potential for **iterative self-correction**—a capability fundamentally absent in AR models. In each forward pass, the model re-examines the global context to reach an equilibrium of logical self-consistency (Nie et al., 2024).

However, we observe a critical mismatch between the underlying architecture of Masked Diffusion Models (MDMs) and their current decoding paradigms. Prevailing sampling methods remain entrenched in the linear decoding logic of AR models, treating each denoising step as an isolated, instantaneous inference. This paradigm assumes that the model can make optimal decisions solely based on the current probability distribution. Crucially, it overlooks a vital information source unique to dLLM decoding: while token confidence in AR models is fixed once generated, confidence in dLLMs is a **time-evolving variable** that fluctuates as new content is incorporated.

Existing paradigms, such as confidence-based sampling and Predictor-Corrector (PC) samplers, fail to escape this "linear" mindset, relying on local decisions from current-step distributions while ignoring temporal evolution features. Although re-masking methods (e.g., Re-MDM) attempt to alleviate contextual contradictions via partial reconstruction, they not only neglect key temporal dynamics but also introduce prohibitive computational overhead. Similarly, while Lookahead strategies mitigate logical conflicts through forward-path searching, they function as makeshift solutions that trade heavy computation for performance without addressing the fundamental misalignment between the decoding paradigm and the model architecture.

To activate the latent self-correction capabilities of dLLMs, we propose **RACC** (**R**egret-**A**ware **C**onfidence **C**alibration). RACC leverages a "free lunch" in the MDM inference process: the full-sequence logits that are automatically generated in each forward pass but typically discarded. RACC transforms these misaligned temporal evolution sig-

* Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 22656–22672

nals into negative feedback to guide generation, thereby reshaping the open-loop sampling process into a **Closed-loop Feedback System** with self-monitoring capabilities.

Specifically, RACC introduces **Momentum Anchors** to track the confidence trajectory of each position in real-time. When bidirectional attention detects logical inconsistencies, previously stable confidence scores drop below their historical averages. RACC identifies this drop as a **"Regret" signal**. This signal is then operationalized as an active conflict suppression mechanism: by applying a dynamic calibration penalty to candidates exhibiting high regret, they are naturally filtered out during the Top-K selection. Under this closed-loop architecture, the model's "reflection" on past decisions imposes real-time constraints on future sampling paths. Experimental results demonstrate that RACC significantly improves accuracy in logical reasoning tasks with **zero additional inference overhead**, achieving a Pareto optimal balance between decoding stability and execution efficiency.

2 Related Work

2.1 Masked Discrete Diffusion Models

Masked Discrete Diffusion Models (MDMs) generate sequences by simulating a reverse denoising process, transitioning from a fully masked state to the original text (Austin et al., 2021a; Lou et al., 2023). Compared to traditional Autoregressive (AR) models, the core advantage of MDMs lies in their bidirectional attention architecture, which enables the capture of global dependencies during each denoising iteration (Nie et al., 2025). Despite their excellence in parallel generation, leveraging this bidirectional capability to ensure logical consistency during decoding remains an open challenge.

2.2 Confidence-based Sampling and Calibration

The most fundamental decoding method for MDMs is **Confidence-based Sampling**. In this paradigm, the sampler evaluates the instantaneous predicted probabilities for all `[MASK]` positions at each denoising step t and selects the K positions with the highest confidence to be filled.

PC-Sampler and Calibration Techniques To rectify biases introduced by word frequency disparities and positional shifts in basic sampling, recent research proposed the **PC-Sampler** (Huang et al., 2025). This method calibrates raw confidence by introducing position-wise (P) and category-prior (C) weights. Such improvements enhance generation quality without increasing sampling complexity.

Limitations of Static Logic Despite these advancements, such methods essentially adhere to a **static sampling logic**: they make selections based solely on the instantaneous probability distribution of the current step. However, instantaneous confidence is often insufficient to fully characterize the robustness of a decision, as it ignores the **temporal evolution** of confidence across successive denoising trajectories which may signal potential logical conflicts.

2.3 Iterative Refinement and Inference-time Scaling

To compensate for the deficiencies of static sampling, recent studies have explored introducing additional computation during the inference stage to trade for generation quality.

Explicit Remasking Methods represented by **Re-MDM** (Wang et al., 2025a) allow the model to retract determined tokens and re-mask them during generation. However, the re-masking decisions in Re-MDM primarily rely on static confidence scores assigned at the token's "inception," rather than monitoring the continuous belief dynamics.

Lookahead Search Strategies such as **LookUM** (Lee et al., 2025) verify current decisions by simulating multiple future generation paths. While effective, the multi-path simulation incurs significant computational overhead. These methods typically function as post-hoc corrections or expensive lookaheads, leaving a gap for **zero-overhead preventive calibration** mechanisms.

2.4 Temporal Evolution in Diffusion Trajectories

Recent research has begun to exploit the temporal dimension of the generation process in MDMs, revealing that model predictions exhibit complex dynamical characteristics rather than simple linear convergence.

Temporal Oscillation and Post-hoc Aggregation. Wang et al. (2025b) uncovered the **temporal oscillation** phenomenon, wherein correct candidates frequently emerge during intermediate denoising stages but are subsequently overshadowed by erroneous late-stage predictions. To capture these

"intermediate truths," they proposed **Temporal Self-Consistency Voting**, which enhances accuracy through weighted voting across intermediate predictions within a single sampling trajectory.

Coherent Context and Trajectory Rectification. Addressing the unreliability of instantaneous metrics, **Coherent Contextual Decoding (CCD)** (Chen et al., 2025) introduces a **trajectory rectification** mechanism. By maintaining a **history buffer** (H_t) that stores recent prediction distributions, CCD identifies and rejects sampling paths that exhibit temporal instability.

While these works recognize the significance of the temporal dimension, they primarily rely on post-hoc processing (e.g., voting) or explicit, memory-intensive history buffers. **Crucially, the potential to operationalize these evolutionary signals into a proactive, real-time feedback loop remains largely unexplored. The field lacks a lightweight paradigm that can activate this intrinsic self-correction capability without the burden of additional inference overhead.**

3 Methodology

3.1 Probabilistic Modeling: AR vs. MDM

To clarify the unique advantages of MDMs, we first contrast the joint probability modeling of Autoregressive (AR) models and Masked Diffusion Models (MDMs).

AR models follow the chain rule, decomposing the probability of a sequence $\mathbf{x} = [x_1, \ldots, x_L]$ into a product of unidirectional conditional probabilities:

$$P_{AR}(\mathbf{x}) = \prod_{i=1}^{L} P(x_i|\mathbf{x}_{<i}) \tag{1}$$

This decoding process is monotonic and irreversible; the confidence of the i-th token is fixed at the moment of generation, precluding any updates in subsequent steps.

In contrast, MDMs are defined over a denoising trajectory $\mathbf{x}_T, \mathbf{x}_{T-1}, \ldots, \mathbf{x}_0$. The inference process is a gradual iteration over the variational distribution $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$:

$$P_{MDM}(\mathbf{x}_0) = \int_{\mathbf{x}_{1:T}} p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) d\mathbf{x}_{1:T} \tag{2}$$

At each sampling step $t \in \{T, \ldots, 1\}$, the model predicts the global conditional distribution $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$ via a bidirectional attention mechanism. Unlike AR models, the predicted probability $P(x_i|\mathbf{x}_t)$ for each token x_i evolves dynamically across different timesteps t. This property allows the model to utilize newly introduced context at step t to re-evaluate decisions made at step $t+1$.

3.2 Instantaneous Inference Bias in MDM Decoding

Despite the bidirectional potential of the architecture, current parallel sampling paradigms (e.g., PC-Sampler) suffer from **instantaneous inference bias**. Formally, let $\mathcal{C}_t \subset \{1, \ldots, L\}$ be the set of positions committed before step t. The decision function $\mathcal{D}$ at step t relies solely on the current probability snapshot P_t:

$$\mathbf{x}_{t-1} = \mathcal{D}(p_\theta(\mathbf{x}_0|\mathbf{x}_t)) \tag{3}$$

This paradigm neglects the temporal gradient of confidence. For a committed position $i \in \mathcal{C}_{t+1}$, if the newly filled content at step t conflicts with x_i, the bidirectional attention will trigger a significant collapse in confidence:

$$\Delta P_t^i = P(x_i|\mathbf{x}_t) - P(x_i|\mathbf{x}_{t+1}) \ll 0 \tag{4}$$

In traditional open-loop sampling, this signal reflecting logical dissonance is ignored because $\mathcal{D}$ lacks temporal memory. Consequently, the model is forced to continue generating within a spurious context where conditional probabilities have plummeted, leading to **error cascading**.

3.3 Regret-Aware Confidence Calibration (RACC)

RACC reshapes the decoding process from a sequence of instantaneous selections into a system of evolution monitoring through a temporal feedback loop.

3.3.1 Position-wise Momentum Tracking

To robustly quantify confidence fluctuations, RACC maintains a momentum anchor E_t^i for each position i in the sequence. Crucially, unlike tracking specific token identities, E_t^i monitors the **stability of the local belief state** at position i. This prevents the "moving target" problem where legitimate hypothesis switching might be conflated with regret. The anchor is updated via an Exponential Moving Average (EMA) of the probability of the locally optimal candidate $w_{t,i}^* = \text{argmax}\, p_t(\cdot|\mathbf{x}_t)$:

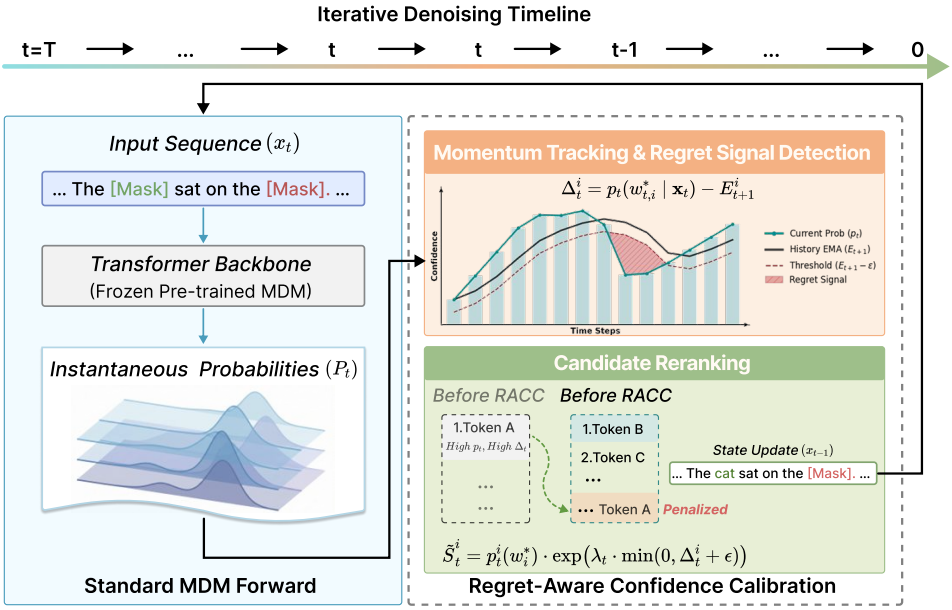


Figure 1: **Overview of the RACC framework.** RACC acts as a closed-loop system, monitoring confidence trajectories to detect Regret Signals (red shaded area). By dynamically penalizing unstable candidates (e.g., Token A), it actively steers the generation toward consistency without additional inference overhead.

$$E_t^i = \begin{cases} p_t(w_{t,i}^*|\mathbf{x}_t), & E_{t+1}^i = \emptyset \\ \beta E_{t+1}^i + (1-\beta)p_t(w_{t,i}^*|\mathbf{x}_t), & \text{otherwise} \end{cases} \tag{5}$$

where $\beta \in [0, 1]$ is the momentum coefficient. This anchor provides an adaptive reference frame: stable beliefs result in $E_t^i \approx p_t^i$, while sudden collapses in confidence (indicating logical conflict) manifest as $p_t^i \ll E_t^i$.

3.3.2 Dynamic Regret Signal and Penalty Scheduling

We define the **Regret Signal** Δ_t^i as the negative deviation of the current instantaneous probability from the historical momentum anchor:

$$\Delta_t^i = p_t(w_{t,i}^*|\mathbf{x}_t) - E_{t+1}^i \tag{6}$$

A significant negative Δ_t^i indicates that the bidirectional attention has detected a dissonance between the current context and the model's prior belief trajectory.

To balance exploration and consistency, we modulate the penalty intensity λ_t using a time-dependent strategy. Instead of a fixed penalty, we employ a **cosine schedule** where λ_t smoothly increases from $\lambda_{\min}$ to $\lambda_{\max}$ as the denoising process progresses ($t \to 0$). This design ensures that the "Temporal Gatekeeper" remains permissive during early-stage hypothesis formation—where uncertainty is naturally high—but becomes increasingly stringent in the final stages to enforce strict logical coherence.

3.4 The Temporal Gatekeeper: Regret-Driven Trajectory Reshaping

Unlike the fixed left-to-right decoding order in Autoregressive (AR) models, the decoding trajectory of MDMs is dynamically determined by the confidence ranking across all masked positions. We leverage this property to introduce the **Temporal Gatekeeper** mechanism, which transforms RACC from a local calibration tool into a global trajectory optimizer.

Mechanism. For each position $i \in \{1, \ldots, L\}$ in the [MASK] state, RACC calibrates the sampling score of its most probable candidate w_i^*. The calibrated score $\tilde{S}_t^i$ is formally defined as:

$$\tilde{S}_t^i = p_t^i(w_i^*) \cdot \exp\left(\lambda_t \cdot \min(0, \Delta_t^i + \epsilon)\right) \tag{7}$$

where Δ_t^i is the regret signal captured in Section 3.2, ϵ represents the tolerance threshold, and λ_t is the dynamic penalty intensity derived from the schedule. This operator acts as a **soft admission filter**: it preserves the scores of tokens that maintain temporal stability while exponentially suppressing those exhibiting logical inconsistency.

Inter-position Re-ranking. While the calibration is computed locally, its primary impact manifests in the **global competition for commitment**.

In parallel sampling paradigms (e.g., LLaDA), the set of positions $\mathcal{P}_t$ to be de-masked at step t is selected based on their relative confidence scores. Formally:

$$\mathcal{P}_t = \text{ArgTop-}K\left(\{\tilde{S}_t^i \mid i \in \mathcal{M}_t\}\right) \quad (8)$$

where $\mathcal{M}_t$ is the set of indices currently in the `[MASK]` state. By penalizing tokens with high temporal regret, RACC effectively **demotes the global priority** of logically unstable positions. This ensures that the limited budget of K tokens per step is allocated only to positions where the model's bidirectional context has reached a consensus.

Regret-Driven Trajectory Reshaping. This mechanism achieves what we term **Regret-Driven Trajectory Reshaping**. Unlike explicit re-masking methods that retract committed tokens, RACC implements a preventive **priority deferral** strategy. Traditional samplers are often "greedy," committing to high-confidence but logically premature tokens that cause error cascading. In contrast, RACC recognizes these "regretful" decisions before they are finalized.

By penalizing tokens that exhibit a high regret signal ($\Delta_t^i \ll 0$), the calibration term $\exp(\cdot)$ acts as a **ranking suppressor**. It drastically reduces $\tilde{S}_t^i$, causing logically unstable positions to plummet in the global priority list. Consequently, the sampler is steered to bypass these high-risk regions in the current step, allowing other positions with stable, coherent confidence trajectories to be prioritized for filling. This strategic postponement empowers the model to "pause" on contentious positions and wait for more definitive global context to emerge in subsequent iterations, transforming dLLM decoding from an instantaneous probabilistic selection into a globally coordinated, time-aware reasoning process.

3.5 Complexity and Efficiency

RACC achieves superior logical robustness with **negligible computational cost** by adopting an **information reuse** strategy.

Unlike inference-time scaling methods (e.g., Re-MDM) which require ΔT additional forward passes, or Lookahead Search (e.g., LookUM) which multiplies FLOPs by a branching factor B, RACC utilizes the full-sequence logits inherently produced during the standard forward pass. The extra computation involves only element-wise operations with $O(L)$ complexity for momentum tracking.

Compared to the $O(L^2)$ complexity of self-attention and the massive parameter multiplications in Transformers, these scalar operations are computationally invisible in practice. As verified in Section 4.3, RACC maintains the same wall-clock throughput as the baseline sampler, effectively offering a "free lunch" for decoding consistency.

4 Experiments

4.1 Pilot Study: Unveiling the Decisional Mismatch

Before evaluating the performance of RACC, we first conduct a comprehensive pilot study to empirically verify the existence of the "decisional mismatch" in MDM decoding. We analyze the evolution of confidence from both qualitative and quantitative perspectives.

Qualitative Observation: Temporal Confidence Collapse We visualize the confidence trajectories across denoising steps for a representative failed reasoning case using a spatiotemporal heatmap, as shown in Figure 2. Vertical slices of the heatmap reveal a critical phenomenon: certain tokens are initially committed with high confidence; however, as the bidirectional context becomes more complete in subsequent steps, their predictive probabilities undergo a significant **"post-commitment collapse."** This observation confirms that while the MDM's internal architecture is capable of perceiving logical conflicts via bidirectional attention, existing static samplers lack the feedback mechanism to react to these emergent signals.

Quantitative: Regret as a Failure Predictor. We perform a statistical analysis on HumanEval to quantify this phenomenon. Our results reveal two insights: (1) **High Correlation with Failure.** Regret signals are triggered in 57.6% of failed samples, compared to only 40.6% in correct ones (30% lower), indicating a strong link between internal regret and logical inconsistency. (2) **Signal Magnitude Disparity.** The average intensity of regret in erroneous samples is $1.1\times$ **higher** than in correct ones, suggesting that logical contradictions induce a more profound internal "dissonance" than typical generative variance.

Key Insight The gathered evidence highlights a fundamental bottleneck: ***the model knows it has***

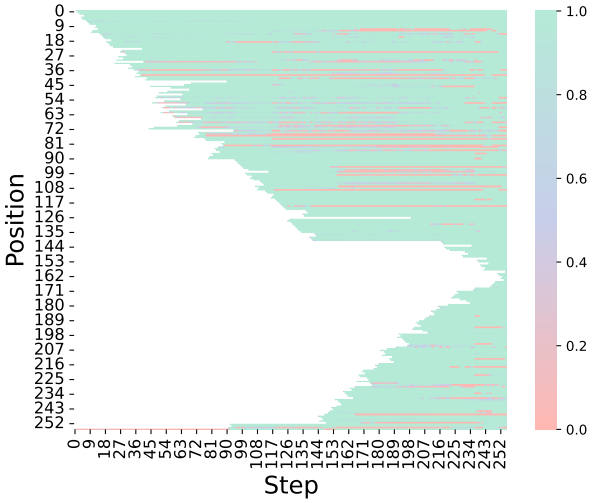


Figure 2: **Confidence Evolution Heatmap.** Green and red denote high and low confidence, respectively. The appearance of red streaks within initially stable (green) regions highlights **post-commitment probability collapses**, visually confirming the emergent *logical regret* as global context evolves.

erred, yet the sampler remains blind. This disconnect between internal feedback (regret signals) and external decision-making (static Top-K selection) serves as the primary driver of error cascading. This observation provides the direct physical motivation for our proposed RACC closed-loop feedback mechanism.

4.2 Experimental Setup

Datasets and Metrics. We evaluate RACC on a diverse suite of reasoning-intensive benchmarks: **HumanEval** (Chen, 2021) and **MBPP** (Austin et al., 2021b) for code generation; **GSM8K** (Cobbe et al., 2021) and **MATH** (Hendrycks et al., 2021) for mathematical reasoning; and **Sudoku** and **Countdown** for constrained symbolic reasoning. We report Pass@1 accuracy for code and math tasks, and standard accuracy for symbolic tasks.

Baselines. We benchmark RACC against six representative decoding paradigms for MDMs: (1) **Confidence**: The vanilla confidence-based sampling; (2) **Margin**: A variation selecting tokens based on the gap between the top two probabilities; (3) **Entropy**: A strategy utilizing predictive entropy to identify low-confidence positions; (4) **PC-Sampler** (Huang et al., 2025): A method using position and category priors for calibration; (5) **Re-MDM** (Wang et al., 2025a): An iterative refinement approach with 5 explicit re-masking steps; (6) **LookUM** (Lee et al., 2025): A heavy-computation strategy using forward-path lookahead search.

Implementation Details. We conduct experiments using the **LLaDA-8B-Instruct** and **LLaDA-1.5** backbones. All evaluations are performed on a server with 8 $\times$ NVIDIA A100 (64GB) GPUs using CUDA 12.7. To ensure a rigorous comparison of inference efficiency, each task is executed on a single GPU in an isolated environment. For RACC, we set $\beta = 0.8$, $[\lambda_{\min}, \lambda_{\max}] = [1.0, 5.0]$, and $\epsilon = 0.08$ as the default configuration.

4.3 Main Results and Discussion

As shown in Table 1, RACC establishes a new state-of-the-art for MDM decoding across all tested backbones and benchmarks.

Superiority in Logical and Code Reasoning. The most striking performance leaps occur in code generation tasks (HumanEval and MBPP). For instance, on LLaDA-1.5, RACC achieves a Pass@1 of **45.73%** on HumanEval, nearly doubling the performance of the vanilla Confidence sampler (**23.70%**) and significantly outperforming the next best method, LookUM (**33.50%**). Similarly, in MBPP, RACC leads the second-best baseline by a margin of over **5%**. These results suggest that RACC's closed-loop calibration is uniquely effective at maintaining the long-range logical coherence required for algorithmic tasks.

Efficiency and Accuracy Pareto Frontier. A key takeaway is RACC's ability to outperform "search-heavy" methods like LookUM without any additional computational cost. While LookUM provides competitive results by simulating multiple future paths, RACC consistently surpasses it in symbolic tasks like Countdown (e.g., **35.70%** vs **25.40%** on 8B) and Sudoku. This proves that the internal bidirectional attention of dLLMs already encodes sufficient signals for error detection; RACC successfully activates this "latent' self-correction without the need for redundant forward passes.

Comparison with Static Calibration. Baseline methods such as Margin, Entropy, and PC-Sampler, which rely on instantaneous probability distributions, show limited gains or even performance degradation in complex tasks like HumanEval. This confirms our hypothesis in Section 4.1: instantaneous snapshots of confidence are insufficient to capture logical conflicts. By incorporating the *temporal evolution* of confidence via momentum

Table 1: **Main Results.** Comparison of RACC against state-of-the-art decoding methods. Results are reported in Pass@1 (%) for code/math tasks and Accuracy (%) for others. **Bold** indicates the best performance among all methods.

Model	Method	HumanEval	GSM8K	MBPP	MATH	Sudoku	Countdown
LLaDA-8B	Confidence	32.00	76.70	28.40	32.40	23.80	20.30
	Margin	31.00	76.10	28.40	34.40	21.80	19.10
	Entropy	19.50	75.40	24.40	33.00	12.00	21.90
	PC-Sampler	30.50	73.70	25.20	32.40	23.20	26.50
	Re-MDM	29.30	77.90	28.40	33.00	15.40	25.30
	LookUM	35.90	79.30	36.20	34.60	25.00	25.40
	Ours (RACC)	**44.51**	**79.76**	**47.31**	**34.80**	**25.80**	**35.70**
LLaDA-1.5	Confidence	23.70	79.40	38.60	32.60	27.40	23.40
	Margin	27.40	78.30	38.10	35.00	27.80	14.00
	Entropy	23.20	77.00	36.50	32.20	23.00	12.00
	PC-Sampler	23.80	77.30	39.60	32.20	27.20	19.10
	Re-MDM	29.80	80.10	39.30	34.00	22.80	19.90
	LookUM	33.50	82.30	43.60	35.80	28.00	17.90
	Ours (RACC)	**45.73**	**82.60**	**48.95**	**36.20**	**29.40**	**34.40**

Table 2: **Efficiency Profile on HumanEval.** Measured on a single A100 ($T = 256$). We report Throughput and Relative Latency in a unified column. RACC matches the baseline speed (17.76 vs 17.92 tok/s), while search-based methods exhibit significant real-world slowdowns.

Method	NFE / Sample (Ratio)	Throughput (Latency) tok/s (Relative)
Confidence	256 (1.0×)	17.92 (1.00×)
PC-Sampler	256 (1.0×)	17.85 (1.00×)
RACC (Ours)	**256 (1.0×)**	**17.76** (1.01×)
Re-MDM (5-iter)	384 (1.5×)	12.97 (1.38×)
LookUM (k=2)	512 (2.0×)	11.43 (1.57×)

anchors, RACC provides a much more robust discriminative signal for high-quality generation.

4.3.1 Synergy with Global Temporal Aggregation

Recent research has identified the **"temporal oscillation"** phenomenon in MDMs, where correct answers frequently emerge during intermediate denoising steps but are subsequently overwritten by erroneous late-stage predictions. To address this, Wang et al. (2025b) proposed **Temporal Self-Consistency Voting (TSCV)**, a post-hoc strategy that selects the final output by aggregating predictions $\{x_0^t\}_{t=1}^T$ across the sampling trajectory. In this section, we investigate whether RACC, as a real-time interceptor, provides complementary benefits to such global voting mechanisms.

Experimental Setup. We evaluate the synergy between RACC and TSCV on three reasoning benchmarks: **GSM8K**, **SVAMP**, and **MATH500**. We utilize LLaDA-8B-Instruct as the backbone with 64 total denoising steps. We treat the original TSCV implementation as the *Baseline* and compare it against the RACC-enhanced version. We report both the *Final Accuracy* (at the last step) and the *Voting Accuracy* (aggregated via TSCV) to demonstrate how RACC improves the quality of the entire trajectory.

Table 3: **Synergy between RACC and Temporal Voting.** We compare the performance of standard TSCV when applied to vanilla trajectories (*Baseline*) vs. trajectories calibrated by RACC. **Dark green** values denote absolute improvements over the respective baseline. RACC consistently elevates the voting performance upper bound.

Dataset	Method	Final Acc (%)	Vote Acc (%)
GSM8K	Baseline (TSCV)	67.70	69.60
	RACC + TSCV	**68.23** +0.53	**70.20** +0.60
SVAMP	Baseline (TSCV)	82.00	83.33
	RACC + TSCV	**82.67** +0.67	**84.67** +1.34
MATH	Baseline (TSCV)	27.80	28.60
	RACC + TSCV	**28.80** +1.00	**29.40** +0.80

Analysis: RACC as a Trajectory Optimizer. As shown in Table 3, the integration of RACC consistently yields performance gains across all datasets. This synergy stems from two key factors. **First, optimizing the signal source.** The effectiveness of TSCV relies on correct intermediate predictions. By suppressing logically inconsistent tokens in real-time, RACC ensures that more intermediate steps $\{x_0^t\}$ stay within the correct reasoning manifold, providing a higher-quality "source" for voting. **Second, robustness against oscilla-**

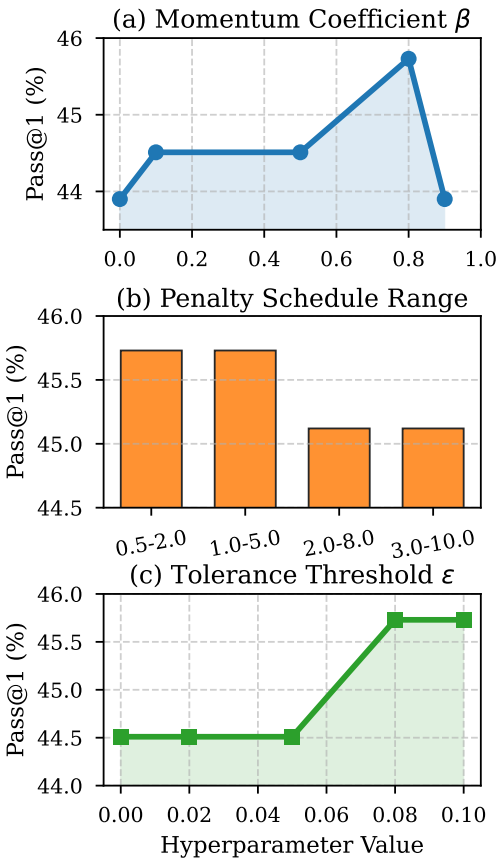


Figure 3: **Parameter Sensitivity Analysis.** We investigate the impact of (a) momentum β, (b) penalty schedule ranges, and (c) regret threshold ϵ on HumanEval performance. The results demonstrate a robust operating window for RACC.

tion. While TSCV helps recover correct answers lost to oscillation, RACC proactively reduces the occurrence of such oscillations by penalizing "regretful" paths. The fact that the voting accuracy increases—reaching **84.67%** on SVAMP—confirms that RACC and TSCV are orthogonal and complementary.

4.4 Ablation Study and Analysis

To verify the contribution of RACC's core components and assess its robustness, we analyze the performance impact of varying three key hyperparameters on LLaDA-1.5 (HumanEval). As visualized in Figure 3:

(a) Momentum Coefficient β (Impact of EMA): The performance curve follows an inverted U-shape, confirming the necessity of long-term history tracking. Specifically, setting $\beta = 0$ reduces RACC to a local-only calibration (referencing only the previous step). While this still outperforms the standard baseline (43.9% vs 23.7%), it lags behind the EMA-enhanced version ($\beta = 0.8$, **45.73%**). This gap empirically validates that tracking the *historical trajectory* via momentum allows for more accurate regret detection than immediate snapshots.

(b) Penalty Schedule λ (Impact of Dynamic Gating): We compare different intensity ranges $[\lambda_{\min}, \lambda_{\max}]$ for the cosine schedule. The results show that moderate penalty ranges (e.g., $[1.0, 5.0]$) yield optimal performance. Ranges that are too aggressive (e.g., $[3.0, 10.0]$) cause a performance drop ($45.73\% \rightarrow 45.12\%$), indicating that over-penalizing high-entropy positions hinders valid exploration. This supports our design of using a dynamic schedule to balance early-stage diversity with late-stage consistency.

(c) Tolerance Threshold ϵ (Robustness to Noise): Higher thresholds ($\epsilon \in [0.08, 0.10]$) outperform lower ones ($\epsilon \approx 0.0$). This indicates that the "Gatekeeper" mechanism benefits from being lenient towards minor probability fluctuations (noise), intervening only when significant confidence collapses signal a true logical conflict.

5 Conclusion

In this work, we revisited the decoding paradigm of Masked Discrete Diffusion Models, identifying a critical **decisional mismatch**: while the model architecture is bidirectional and dynamic, the prevailing sampling strategies remain static and myopic. To bridge this gap, we introduced **Regret-Aware Confidence Calibration (RACC)**, a training-free framework that operationalizes the model's internal "regret" signals into a real-time feedback mechanism.

Unlike prior approaches that rely on computationally expensive lookahead search or explicit remasking, RACC achieves **implicit trajectory replanning** with **zero inference overhead**. Our extensive experiments demonstrate that RACC not only establishes new state-of-the-art performance across logic-intensive benchmarks but also acts as a robust "source optimizer" that synergizes effectively with post-hoc temporal aggregation strategies.

Ultimately, our findings suggest a paradigm shift in diffusion decoding: moving from treating generation as a sequence of isolated, irreversible steps to viewing it as a **continuous, self-rectifying temporal trajectory**. We hope RACC inspires future research to further exploit the rich, latent temporal dynamics inherent in bidirectional attention mechanisms.

Limitations

While RACC demonstrates significant improvements in decoding consistency without additional overhead, several limitations warrant further investigation.

First, RACC functions as a **decisional filter** rather than a creative generator; its efficacy is fundamentally bounded by the backbone model's internal awareness. If a model fails to exhibit any confidence fluctuations (i.e., no regret signal) during logical collapses due to undertraining, RACC's calibration mechanism remains inactive (Wang et al., 2025b).

Second, the optimal configuration of hyperparameters, such as the regret threshold ϵ and the penalty schedule λ_t, was empirically determined. While we observed a robust operating window across several benchmarks, these heuristics may require recalibration for radically different architectures or non-linguistic modalities.

Lastly, the effectiveness of RACC's **priority deferral** is constrained by the remaining sampling budget. Should a regret signal emerge in the final steps of the denoising process, the limited number of unfilled positions may restrict the sampler's ability to re-plan the reasoning trajectory effectively. Future work could explore integrating RACC with explicit re-masking strategies to handle such late-stage logical conflicts.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. 2021a. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021b. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Kecheng Chen, Ziru Liu, Xijia Tao, Hui Liu, Xinyu Fu, Suiyun Zhang, Dandan Tu, Lingpeng Kong, Rui Liu, and Haoliang Li. 2025. Beyond confidence: Adaptive and coherent decoding for diffusion language models. *arXiv preprint arXiv:2512.02044*.

Mark Chen. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Zhengcong Fei, Mingyuan Fan, Changqian Yu, Debang Li, Youqiang Zhang, and Junshi Huang. 2024. Dimba: Transformer-mamba diffusion models. *arXiv preprint arXiv:2406.01159*.

Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. 2025. Deep think with confidence. *arXiv preprint arXiv:2508.15260*.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. *arXiv preprint arXiv:1904.09324*.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2023. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*.

Pengcheng Huang, Shuhao Liu, Zhenghao Liu, Yukun Yan, Shuo Wang, Zulong Chen, and Tong Xiao. 2025. Pc-sampler: Position-aware calibration of decoding bias in masked diffusion models. *arXiv preprint arXiv:2508.13021*.

Adam Tauman Kalai and Santosh S Vempala. 2024. Calibrated language models must hallucinate. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 160–171.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Sanghyun Lee, Seungryong Kim, Jongho Park, and Dongmin Park. 2025. Lookahead unmasking elicits accurate decoding in diffusion language models. *arXiv preprint arXiv:2511.05563*.

Aaron Lou, Chenlin Meng, and Stefano Ermon. 2023. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. 2024. Scaling up masked diffusion models on text. *arXiv preprint arXiv:2410.18514*.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Large language diffusion models. *arXiv preprint arXiv:2502.09992*.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. 2024. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. 2025a. Remasking discrete diffusion models with inference-time scaling. *arXiv preprint arXiv:2503.00307*.

Wen Wang, Bozhen Fang, Chenchen Jing, Yongliang Shen, Yangyi Shen, Qiuyu Wang, Hao Ouyang, Hao Chen, and Chunhua Shen. 2025b. Time is a feature: Exploiting temporal dynamics in diffusion language models. *arXiv preprint arXiv:2508.09138*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Qinglin Zeng, Jing Yang, and Keze Wang. 2025. Reflective confidence: Correcting reasoning flaws via online self-correction. *arXiv preprint arXiv:2512.18605*.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, and 1 others. 2025. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. *arXiv preprint arXiv:2505.19223*.

Supplementary Material for RACC: Regret-Aware Confidence Calibration for Consistent Masked Discrete Diffusion Decoding

A Detailed Experimental Settings

A.1 Hyperparameter Configuration

We maintain a consistent experimental setup across two backbones: `LLaDA-8B-Instruct` and `LLaDA-1.5`. For all experiments, the momentum coefficient is set to $\beta = 0.8$. We adapt the penalty schedule range $[\lambda_{\min}, \lambda_{\max}]$ and tolerance threshold ϵ based on the task characteristics (e.g., symbolic vs. natural language). Table 4 details the exact generation configurations.

Dataset	Steps (T)	Gen Len	Block Len	Penalty (λ)	Threshold (ϵ)	Momentum (β)
HumanEval	256	256	256	[1.0, 5.0]	0.08	0.8
MBPP	256	256	256	[1.0, 5.0]	0.08	0.8
GSM8K	256	256	256	[1.0, 5.0]	0.08	0.8
MATH	1024	1024	1024	[1.0, 5.0]	0.08	0.8
Sudoku	128	128	128	[1.0, 5.0]	0.03	0.8
Countdown	128	128	128	[1.0, 5.0]	0.08	0.8

Table 4: **Hyperparameter configurations** for different benchmarks. The penalty schedule follows a cosine curve within the specified range.

A.2 Prompt Templates

To ensure reproducibility, we provide the exact prompt templates used for each benchmark. All prompts are formatted to be compatible with the LLaDA instruction format.

HumanEval (Code Generation)

```
Role: You are a professional Python coding assistant
Task: Complete the follow function implementation strictly and clearly without any additional comments or explanations.
{func}
```

MBPP (Code Generation)

```
Role: You are a professional Python coding assistant
Task: Complete the follow function implementation strictly and clearly without any additional comments or explanations.
{func}
the function name and the parameters is {function_signature}
```

GSM8K (Math Reasoning)

Please solve the new question step by step just like the following examples. For this question:
1. Break down the problem into logical steps
2. Show all intermediate calculations
3. Conclude with "So the answer is..." format

Examples:
question: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?
target: There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been $21 - 15 = 6$. The answer is 6.
question: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?
target: There are originally 3 cars. 2 more cars arrive. $3 + 2 = 5$. The answer is 5.

New Question: {question}
Solution:

MATH (Math-500)

Q: Let

$$f(x) = \begin{cases} ax + 3, & \text{if } x > 2, \\ x - 5, & \text{if } -2 \le x \le 2, \\ 2x - b, & \text{if } x < -2. \end{cases}$$

Find $a + b$ if the piecewise function is continuous (which means that its graph can be drawn without lifting your pencil from the paper).
A: <think>
For the piecewise function to be continuous, the cases must "meet" at 2 and -2. For example, $ax + 3$ and $x - 5$ must be equal when $x = 2$. This implies $a(2) + 3 = 2 - 5$, which we solve to get $2a = -6 \Rightarrow a = -3$. Similarly, $x - 5$ and $2x - b$ must be equal when $x = -2$. Substituting, we get $-2 - 5 = 2(-2) - b$, which implies $b = 3$. So $a + b = -3 + 3 = \boxed{0}$.
</think>
<answer>
0
</answer>

Q: A rectangular band formation is a formation with m band members in each of r rows, where m and r are integers. A particular band has less than 100 band members. The director arranges them in a rectangular formation and finds that he has two members left over. If he increases the number of members in each row by 1 and reduces the number of rows by 2, there are exactly enough places in the new formation for each band member. What is the largest number of members the band could have?
A: <think>
Let x be the number of band members in each row for the original formation, when two are left over. Then we can write two equations from the given information:

$$rx + 2 = m$$

$$(r - 2)(x + 1) = m$$

Setting these equal, we find:

$$rx + 2 = (r - 2)(x + 1) = rx - 2x + r - 2$$

$$2 = -2x + r - 2 \implies 4 = r - 2x$$

We know that the band has less than 100 members. Based on the first equation, we must have rx less than 98. We can guess and check some values of r and x in the last equation. If $r = 18$, then $x = 7$, and $rx = 126$ which is too big. If $r = 16$, then $x = 6$, and $rx = 96$, which is less than 98. Checking back in the second formation, we see that $(16 - 2)(6 + 1) = 14 \cdot 7 = 98$ as it should. This is the best we can do, so the largest number of members the band could have is $\boxed{98}$.
</think>
<answer>
98
</answer>

Q: What is the degree of the polynomial $(4 + 5x^3 + 100 + 2\pi x^4 + \sqrt{10}x^4 + 9)$?
A: <think>
This polynomial is not written in standard form. However, we don't need to write it in standard form, nor do we need to pay attention to the coefficients. We just look for the exponents on x. We have an x^4 term and no other term of higher degree, so $\boxed{4}$ is the degree of the polynomial.
</think>
<answer>
4
</answer>

Q: Evaluate $\left\lceil 3\left(6 - \frac{1}{2}\right)\right\rceil$.
A: <think>
Firstly, $3\left(6 - \frac{1}{2}\right) = 18 - 1 - \frac{1}{2} = 17 - \frac{1}{2}$. Because $0 \leq \frac{1}{2} < 1$, we have $\left\lceil 17 - \frac{1}{2}\right\rceil = \boxed{17}$.
</think>
<answer>
17
</answer>

Q: {question}
A:

Sudoku

```
Solve this 4x4 Sudoku puzzle represented as a 16-digit string (read left-to-right, top-to-bottom) where '0'=empty cell.

Requirements:
1. Replace ALL '0's with digits 1-4
2. Follow STRICT Sudoku rules:
   - Rows: Each must contain 1-4 exactly once
  - Columns: Each must contain 1-4 exactly once
  - 2x2 Boxes: Each must contain 1-4 exactly once
3. Format answer as:
<answer>
-digit solution



</answer>

Please solve the following 4x4 Sudoku puzzle...

Puzzle: 4100
0001
1300
2000
<answer>
4132
3241
1324
2413
</answer>

Puzzle: 0004
0321
0203
3002
<answer>
2134
4321
1243
3412
</answer>

Puzzle: 4123
0000
0402
2300
<answer>
4123
3214
1432
2341
</answer>

Puzzle: 1432
0041
3000
4000
<answer>
1432
2341
3214
4123
</answer>

Puzzle: 0020
0341
0210
1002
<answer>
4123
2341
3214
1432
</answer>

Puzzle: {puzzle_str}
<answer>
```

Countdown

For the given numbers, find a sequence of arithmetic operations that results in the target number. Show your reasoning and conclude with "The answer is: [formula]".

Examples:
question: 15,44,79,50

Solution: Let's try to combine 15 and 44. $44 - 15 = 29$. Now we have 29 and the remaining number 79. We need to reach the target 50. Let's try $79 - 29 = 50$. This works. The answer is: 44-15=29,79-29=50

question: 1,2,12,25
Solution: We have 1, 2, 12 and the target is 25. Let's try multiplying 2 and 12. $2 * 12 = 24$. Now we have 24 and the remaining number 1. We need to reach 25. $24 + 1 = 25$. This is correct. The answer is: 2*12=24,1+24=25

question: 3,85,5,30
Solution: The numbers are 3, 85, 5 and the target is 30. Let's try adding 85 and 5. $85 + 5 = 90$. Now we have 90 and the remaining number 3. We need to reach 30. $90/3 = 30$. That's the target. The answer is: 85+5=90,90/3=30

New Question: {question}
Solution:

B Baseline Implementation Details

We compare RACC against the following baselines, adopting their official or recommended configurations to ensure a fair evaluation:

- **PC-Sampler** (Huang et al., 2025): We utilize the official implementation with the Gaussian corrector hyperparameter set to $w = 1.2$. This method calibrates the logits using position-based and category-based priors to mitigate selection bias.

- **Re-MDM** (Wang et al., 2025a): We adopt the inference-time scaling setting with $R = 5$ refinement iterations. In each iteration, the tokens with the lowest confidence (bottom 10%) are re-masked and regenerated, allowing the model to iteratively self-correct committed errors.

- **LookUM** (Lee et al., 2025): Following the main experimental setup in the original paper, we set the lookahead depth to $d = 1$ with a branching factor of $k = 2$ (i.e., 2 selected paths). We utilize a candidate pool size of $|\mathcal{P}_t| = 5$ and employ *Average Negative Entropy* as the verifier metric to score and select the optimal unmasking trajectory.

- **CCD** (Chen et al., 2025): We implement Coherent Contextual Decoding (CCD) following the official configuration for the LLaDA backbone. Specifically, we set the sliding-window history length to $d = 2$ and the candidate set size to $V = 4$. The method constructs a historical buffer to track the top-V most confident tokens across the last d iterations. Tokens that consistently appear in this buffer are prioritized based on their marginalized probability over the trajectory, serving as a temporal consistency filter.

C Comparison with Trajectory-Aware Methods

We provide a direct comparison with Coherent Contextual Decoding (CCD) (Chen et al., 2025), a recent state-of-the-art method that also leverages historical trajectories. Unlike CCD, which requires maintaining a memory-intensive buffer of probability distributions ($O(d \cdot V)$), RACC employs a lightweight scalar momentum anchor ($O(1)$) to achieve superior performance.

Table 5: **Comparison with CCD on LLaDA-8B.** RACC outperforms CCD across key reasoning benchmarks while being more memory-efficient. (* indicates results reproduced using official settings: $d = 2, V = 4$).

Method	HumanEval	GSM8K	MBPP
Baseline (Confidence)	32.00	76.70	28.40
CCD (Chen et al., 2025)*	38.41	75.30	41.20
RACC (Ours)	**44.51**	**79.76**	**47.31**

D Impact of Penalty Schedule Strategies

The dynamic penalty schedule λ_t is a critical component governing the trade-off between exploration (early-stage) and consistency (late-stage). In this section, we investigate how different functional forms of the schedule impact the model's ability to navigate the denoising trajectory, focusing on shape characteristics and intensity robustness.

D.1 Influence of Functional Forms (Shape)

To identify the optimal modulation strategy, we compare five representative functional forms: **Constant** ($\lambda_t = \lambda_{\max}$), **Step Function** (discrete jump at 50%), **Concave** (Square-root growth), **Linear** (Uniform increase), and **Cosine**. All schedules operate within the same intensity range $[\lambda_{\min}, \lambda_{\max}] = [1.0, 5.0]$.

Analysis. Table 6 indicates that the functional shape plays a pivotal role in decoding stability.

Table 6: **Performance by Schedule Shape.** Comparison of different functional forms on HumanEval and GSM8K. The Cosine schedule yields superior performance by closely mirroring the signal-to-noise ratio (SNR) evolution inherent in diffusion models.

Schedule Shape	HumanEval (Pass@1)	GSM8K (Acc)
Constant ($\lambda = 5.0$)	40.85	79.90
Concave (Sqrt Growth)	42.07	80.10
Step Function (50%/50%)	43.29	81.20
Linear Growth	44.51	81.95
Cosine	**45.73**	**82.60**

Early-Stage Tolerance. The **Concave** schedule underperforms (42.07%) because it increases the penalty too aggressively in the early noisy stages, potentially suppressing valid diverse hypotheses before they can mature.

Smooth Transition. The **Step Function** lags behind continuous schedules (Linear/Cosine), suggesting that the transition from exploration to consistency should be gradual rather than abrupt to align with the continuous nature of the denoising process.

Optimality of Cosine. The **Cosine** schedule achieves the best results (45.73%). Its derivative is near-zero at $t \approx T$, effectively creating a "safety buffer" that permits sufficient exploration in the initial phase before strictly enforcing consistency constraints as the sequence structure solidifies.

D.2 Robustness to Intensity Variations

We further assess whether the superiority of the Cosine shape holds under aggressive parameter settings. We increase the maximum penalty intensity $\lambda_{\max}$ from 5.0 to 10.0 to simulate extreme operating conditions.

Table 7: **Intensity Robustness Analysis (HumanEval).** Performance degradation when increasing the maximum penalty $\lambda_{\max}$ from 5.0 to 10.0. The Cosine shape exhibits higher stability compared to Linear and Constant strategies.

$\lambda_{\max}$ Value	Constant	Linear	Cosine
$\lambda_{\max} = 5.0$	40.85	44.51	**45.73**
$\lambda_{\max} = 7.0$	37.80	43.29	**45.12**
$\lambda_{\max} = 10.0$	35.37	41.46	**44.51**
Drop	-5.48	-3.05	**-1.22**

Robustness Findings. As shown in Table 7, the **Constant** strategy is highly sensitive to intensity, collapsing to 35.37% under high penalty. In contrast, the **Cosine** schedule demonstrates remarkable resilience, maintaining a high performance (44.51%) even at $\lambda_{\max} = 10.0$. This stability suggests that the gradual onset of the Cosine function effectively mitigates the risk of over-penalization, making it a theoretically robust choice for diffusion decoding.

E Qualitative Analysis and Case Studies

To better understand how RACC improves code generation reliability, we analyze specific failure cases from the Baseline (Standard PC-Sampler) that were successfully corrected by RACC. We also visualize the confidence evolution heatmaps to demonstrate the stability improvements (Figure 4).

E.1 Case Study 1: Logical Ordering (HumanEval/44)

Task: `pairs_sum_to_zero` requires finding if two *distinct* elements sum to zero.

Analysis. The Baseline fails on inputs like `[0]` because it adds the number to the set *before* checking for its complement. This causes `0` to match with itself. RACC corrects this by enforcing the logical check *before* the state update.

Baseline (Fail). Incorrect order leads to false positives.

Baseline Code

```
seen = set()
for num in l:
    seen.add(num)       # <--- Wrong Order:
    Adds before check
    if -num in seen:
        return True
```

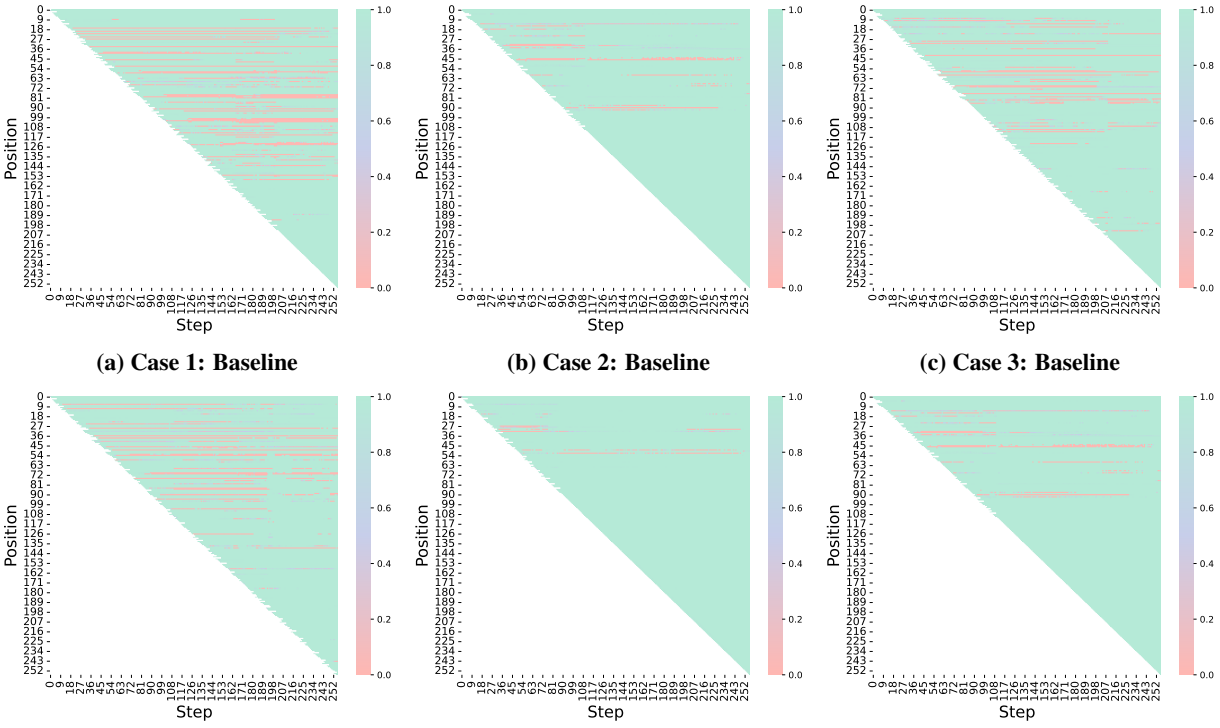
RACC (Success). Corrects causality.

RACC Code

```
seen = set()
for num in l:
    if -num in seen:    # <--- Correct
    Order: Check first
        return True
    seen.add(num)
```

E.2 Case Study 2: Edge Case Coverage (HumanEval/28)

Task: `flip_case` requires flipping case while preserving non-alphabetic characters.



(a) Case 1: Baseline (b) Case 2: Baseline (c) Case 3: Baseline

(d) Case 1: RACC (e) Case 2: RACC (f) Case 3: RACC

Figure 4: **Visualization of Confidence Dynamics across Case Studies.** Top Row: Baseline (PC-Sampler) exhibits chaotic fluctuations and low confidence (red regions). Bottom Row: RACC stabilizes the generation process, maintaining higher confidence (green regions). Columns correspond to Case Studies 1, 2, and 3 respectively.

Analysis. The Baseline employs an `if-elif` structure covering only alphabets, inadvertently dropping symbols. RACC leverages the robust library function.

Baseline (Fail). Explicit branching misses non-alpha characters.

Baseline Code

```
flipped_string = ''
for char in string:
    if char.islower():
        flipped_string += char.upper()
    elif char.isupper():
        # <--- Missing 'else' branch for
    symbols/spaces
        flipped_string += char.lower()
return flipped_string
```

RACC (Success). Uses robust library function.

RACC Code

```
# swapcase() automatically handles
    non-alphabetic chars
return string.swapcase()
```

E.3 Case Study 3: State Maintenance (HumanEval/7)

Task: `parse_nested_parens` requires tracking maximum nesting depth.

Analysis. The Baseline updates the max depth using the value *popped* from the stack, which represents the previous level, not the current peak. RACC uses `len(stack)` before popping to capture the true depth.

Baseline (Fail). Updates `max_level` with stale state.

Baseline Code

```
if char == ')':
    current_level -= 1
    if stack:
        # Incorrect: popping returns the
    OLD level
        max_level = max(max_level,
    stack.pop())
```

RACC (Success). Uses immediate state.

RACC Code

```
elif char == ')':
    # Correct: capture depth before popping
```

```
    max_level = max(max_level, len(stack))
    if stack:
        stack.pop()
```

F Extended Discussion: The Evolution of Confidence Utilization

In this section, we analyze the divergence in uncertainty utilization between Autoregressive (AR) models and Masked Discrete Diffusion Models (MDMs). We argue that while AR methods are constrained by their monotonic nature to rely on *external* interventions (filtering or prompting), MDMs offer a unique opportunity for *internal* dynamic calibration, which RACC exploits.

F.1 Confidence in AR: Static Snapshots and External Interventions

In AR models (Achiam et al., 2023; Dubey et al., 2024), decoding is monotonic. The confidence of a token x_i is derived instantly from $P(x_i|x_{<i})$. Since the model cannot intrinsically revisit past decisions without external mechanisms, improvements largely rely on "patching" the output via two paradigms:

Paradigm 1: Post-Hoc Filtering (Rejection). Strategies like **Self-Consistency** (Wang et al., 2022) attempt to bypass uncertainty by aggregating multiple independent samples. This is theoretically grounded in the finding that calibrated models must hallucinate when the data distribution is complex (Kalai and Vempala, 2024), necessitating external verification. Furthermore, Min et al. (2022) highlight that in-context learning is often driven by format rather than ground truth, reinforcing the need for output-side validation. To address linguistic variation, **Semantic Uncertainty** (Kuhn et al., 2023) refines aggregation by clustering meanings. Building on this, **DeepConf** (Fu et al., 2025) introduces a confidence-aware pruning mechanism. By monitoring the joint probability of reasoning traces, it proactively discards low-confidence paths. However, these methods are **subtractive**: they treat low confidence as a "stop signal," discarding potentially recoverable reasoning chains.

Paradigm 2: Verbal Reflection (Prompting). To enable correction, methods like **Reflexion** (Shinn et al., 2023) and **Self-Refine** (Madaan et al., 2023) rely on iterative prompting, building on the foundational **Chain-of-Thought** (Wei et al., 2022) paradigm. More advanced frameworks like **Tree of Thoughts** (Yao et al., 2023) explicitly search the reasoning space to identify optimal paths. Recently, **Reflective Confidence** (Zeng et al., 2025) proposed automating this by using low confidence as a trigger for self-inquiry. Despite these advances, Huang et al. (2023) argue that LLMs struggle to intrinsically self-correct without external feedback. Consequently, frameworks like **ReAct** (Yao et al., 2022) and **CRITIC** (Gou et al., 2023) integrate external tools to verify outputs. Nevertheless, these approaches are **additive**: they improve accuracy but incur significant overhead by expanding the context window, slowing down inference.

F.2 Confidence in MDMs: Temporal Dynamics as a Feature

MDMs (Sahoo et al., 2024; Fei et al., 2024; Lou et al., 2023; Zhu et al., 2025) generate sequences via iterative denoising ($x_T \rightarrow \cdots \rightarrow x_0$), transforming confidence from a scalar into a **temporal trajectory**. Recent works have begun to exploit this non-monotonic property.

Temporal Aggregation. "Time Is a Feature" (Wang et al., 2025b) observes that correct tokens often oscillate during generation. They propose a voting mechanism across timesteps to retrieve lost answers. Similarly, **CCD** (Chen et al., 2025) enforces consistency by aggregating predictions from a history buffer. While these methods utilize history for *aggregation*, RACC utilizes the *derivative* (Regret) for *control*. We do not just average the past; we actively penalize deviations from the expected momentum.

F.3 Future Horizons: Regret-Driven Control

The "Regret Signal" identified in this work opens avenues for more granular control in non-autoregressive generation, potentially linking back to early iterative refinement ideas like **Mask-Predict** (Ghazvininejad et al., 2019).

Dynamic Re-masking. Currently, RACC suppresses the probability of unstable tokens. Future work could leverage high-regret signals to explicitly **re-mask** spatially adjacent tokens. For instance, if x_i exhibits high regret, the model could reset x_{i-1} and x_{i+1} to `[MASK]`, triggering a localized regeneration to resolve semantic conflicts contextually.

Adaptive Step Scheduling. A sudden system-wide spike in regret could serve as an indicator

that the current diffusion step size is too large. An adaptive scheduler could respond by slowing down the denoising process (adding intermediate steps) specifically when high regret is detected, allocating more compute dynamically to difficult reasoning transitions.