

OPTIMIZING DECODING PATHS IN MASKED DIFFUSION MODELS BY QUANTIFYING UNCERTAINTY

Anonymous authors

Paper under double-blind review

ABSTRACT

Masked Diffusion Models (MDMs) offer flexible, non-autoregressive generation, but this freedom introduces a challenge: final output quality is highly sensitive to the decoding order. We are the first to formalize this issue, attributing the variability in output quality to the cumulative predictive uncertainty along a generative path. To quantify this uncertainty, we introduce Denoising Entropy, a computable metric that serves as an internal signal for evaluating generative process. Leveraging this metric, we propose two algorithms designed to optimize the decoding path: a post-hoc selection method and a real-time guidance strategy. Experiments demonstrate that our entropy-guided methods significantly improve generation quality, substantially boosting accuracy on challenging reasoning, planning, and code benchmarks. Our work establishes Denoising Entropy as a principled tool for understanding and controlling generation, effectively turning the uncertainty in MDMs from a liability into a key advantage for discovering high-quality solutions.

1 INTRODUCTION

In language modeling, Masked Diffusion Models (MDMs) (Austin et al., 2021a; Lou et al., 2023; Sahoo et al., 2024) are rapidly emerging as powerful counterparts to the dominant Auto-Regressive Models (ARMs). While ARMs rely on the next-token prediction paradigm and are constrained to a fixed left-to-right generation path (Bengio et al., 2003; Sutskever et al., 2014), MDMs learn to denoise sequences via random-order masking, enabling them to construct sequences in any order in principle. Each unique sequence construction process can be viewed as a *decoding path*, and different paths typically entail varying generation difficulty that shapes the final output. The generation flexibility of MDMs thus has a theoretical potential: within the vast space of possible decoding paths, there must exist an optimal path that yields a results no worse than the single, rigid sequential path of ARMs.

However, this theoretical potential often remains untapped in practice, as MDMs always underperform compared to ARMs (Feng et al., 2025). Beyond the model capability, decoding strategy acts as a critical bottleneck determining the path taken (Kim et al., 2025). The default random-order strategy, for instance, treats all paths as equally probable (Austin et al., 2021a), reducing the search for an optimal path to mere chance. More sophisticated approaches employ greedy token-level heuristics, such as unmasking the token with the highest confidence (Chang et al., 2022), but are still myopic: a sequence of locally optimal steps provides no guarantee of a globally optimal path. The fundamental limitation underlying these strategies is their lack of a global perspective, as they operate without a mechanism to assess the quality of the overall generative path.

We draw inspiration from uncertainty quantification in ARMs, where metrics like entropy reliably link predictive uncertainty to generative quality (Xu et al., 2020; Kuhn et al., 2023), and uncertainty-aware decoding has shown effect for enhancing global coherence (Arora et al., 2023; Zhu et al., 2024). Adapting this insight to MDMs, we formalize **Path Uncertainty**: MDM’s cumulative predictive uncertainty along a complete decoding path, capturing the global quality of the entire generative process. We attribute MDM quality variability to path uncertainty. High cumulative uncertainty harms output consistency, while low uncertainty indicates reliable paths (Press et al., 2024).

We introduce **Denoising Entropy**, a novel and internally computable metric designed to quantify the inherent uncertainty of generative paths in MDMs. As illustrated in Figure 1, Denoising Entropy comprises two complementary variants: State Entropy (h_{DE}), which captures uncertainty at a single

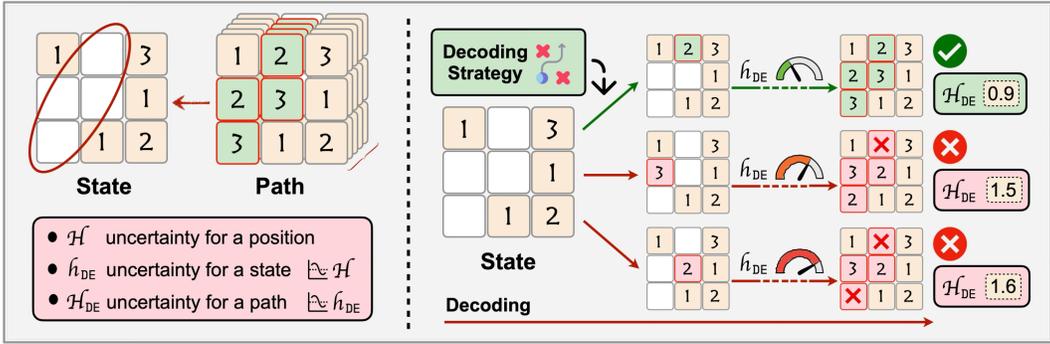


Figure 1: **Quantifying Path Uncertainty in MDMs with Denoising Entropy.** **Left:** State Entropy (h_{DE}) measures per-state uncertainty, computed as the mean Shannon Entropy over the predictive distributions for all masked positions. h_{DE} is then aggregated over the entire path to form the Path Entropy (H_{DE}). **Right:** Decoding process shows how different paths yield outputs of varying quality. Our key insight is that the lower H_{DE} indicates path yielding better output, providing a potent internal signal for generation quality.

decoding state, and Path Entropy (H_{DE}), which integrates this uncertainty across an entire trajectory to measure cumulative generative uncertainty. This formulation provides a principled framework for evaluating the reliability of different generative paths.

Minimizing Denoising Entropy offers a powerful new objective for steering MDM decoding towards more reliable outcomes. We operationalize this principle through two search algorithms: a post-hoc selection method that reranks sampled paths to find the one with the minimum Path Entropy (H_{DE}), and a real-time guidance method that uses State Entropy (h_{DE}) to actively steer generation towards lower-uncertainty regions of the state space. By shifting the focus from myopic, token-level decisions to holistic, path-level optimization, our approach significantly boosts the performance of MDMs on complex open-ended generation and reasoning tasks. **We summarize our key findings below:**

- **Formalizing Path Uncertainty** (Section 3.1): We identify and formalize Path Uncertainty, the cumulative predictive uncertainty of a MDM along a single decoding path, which drives the variability in output quality.
- **Denoising Entropy as Uncertainty Proxy** (Section 3.2): We introduce two novel and computable metrics, $h_{DE}(t)$ and H_{DE} , as a theoretically-grounded toolkit to evaluate entire generative states and paths, providing signals for path-aware guidance.
- **Path Search Algorithms** (Section 3.3): We propose two path search algorithms, Entropy-based Best-of-N and Entropy-guided Sequential Monte Carlo, that leverage Denoising Entropy as an internal signal to optimize the decoding process and improve the quality of generated sequences.

2 PRELIMINARIES

Let \mathcal{V} be a vocabulary of size K , and consider a sequence of L tokens $\mathbf{x}_0 = (x_0^1, \dots, x_0^L)$. For any position $\ell \in \{1, \dots, L\}$, the token x_0^ℓ is represented by a one-hot vector $\mathbf{x}_0^\ell \in \{0, 1\}^K$, whose component corresponding to the index of x_0^ℓ is 1.

2.1 FORWARD PROCESS AND TRAINING OBJECTIVE

The training of MDMs relies on a continuous-time forward corruption process. For any time $t \in [0, 1]$, a latent variable \mathbf{z}_t is generated from \mathbf{x}_0 . This is achieved probabilistically: each token x_0^ℓ is independently replaced by a special [MASK] token with probability $1 - \alpha_t$, and remains unchanged with probability α_t . There $\alpha_t \in [0, 1]$ is a strictly decreasing noise schedule with $\alpha_0 \approx 1$ and $\alpha_1 \approx 0$ (Sahoo et al., 2024). Let \mathcal{M}_t be the set of indices that are masked at time t .

An MDM with denoising network p_θ , parameterized by θ , is trained to reverse this process. Given the corrupted variable \mathbf{z}_t and the time t , the MDM predicts the probability distribution $\hat{\mathbf{p}}_0$ for each token in the original sequence as an estimate of \mathbf{x}_0 :

$$\hat{\mathbf{p}}_t^\ell = p_\theta(X_t^\ell | \mathbf{z}_t, t) \in \Delta^K, \quad \forall \ell \in \{1, \dots, L\}, \quad (1)$$

where X_t^ℓ is the random variable for the token at position ℓ and time t , Δ^K is the K -simplex.

Parameters θ are optimized by minimizing the Negative Evidence Lower Bound (NELBO)(Lou et al., 2023; Shi et al., 2024; Sahoo et al., 2024). This objective simplifies to an integral of the weighted negative log-likelihood over the masked tokens:

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0)} \left[\int_0^1 w(t) \mathbb{E}_{\mathbf{z}_t | \mathbf{x}_0} \left[\sum_{\ell \in \mathcal{M}_t} -\log p_\theta(x_0^\ell | \mathbf{z}_t, t) \right] dt \right], \quad (2)$$

where $q(\mathbf{x}_0)$ is the posterior data distribution, $1 - \alpha_t$ is the unmasking probability at time t and $w(t) = \left| \frac{d\alpha_t}{dt} \right| \frac{1}{1-\alpha_t}$ is the weighting function.

2.2 REVERSE PROCESS FOR GENERATION

The inference process generates a sequence \mathbf{x}_0 by iteratively denoising from a fully masked sequence. The procedure starts with \mathbf{z}_1 , where all tokens are [MASK]. It then proceeds backward in time along a discrete schedule $1 = t_N > t_{N-1} > \dots > t_0 = 0$.

At each step i (from N down to 1), the MDM p_θ takes the current state \mathbf{z}_{t_i} and time t_i as input to predict the original token probability distribution $\hat{\mathbf{p}}_0^\ell$ for each position ℓ . Then, the next state $\mathbf{z}_{t_{i-1}}$ is sampled from the reverse transition kernel $p(\mathbf{z}_{t_{i-1}} | \mathbf{z}_{t_i})$. This distribution allows a masked token to either be unmasked to a content token (sampled according to $\hat{\mathbf{p}}_0^\ell$) or to remain masked, governed by the noise levels α_{t_i} and $\alpha_{t_{i-1}}$. This iterative refinement continues until $t_0 = 0$, yielding the final generated sequence.

3 MODELING PATH UNCERTAINTY WITH DENOISING ENTROPY

3.1 PATH UNCERTAINTY

A decoding path $\tau = (\mathbf{z}_{t_N}, \dots, \mathbf{z}_{t_0})$ is a sequence of latent states traversed during the generative process, guided by a decoding strategy π . The strategy π determines the transition from \mathbf{z}_{t_i} to $\mathbf{z}_{t_{i-1}}$ at each step, based on the model transition kernel $p(\cdot | \cdot)$.

We define **Path Uncertainty** as the model cumulative predictive uncertainty along a *single* decoding path τ , which reflects how *confident* the model is throughout the entire generation of a sequence.

3.2 DENOISING ENTROPY

To quantify path uncertainty, we introduce **Denoising Entropy**. We first define *Oracle State Uncertainty* as the output uncertainty of the joint prediction distribution at any given state.

Definition 1 (Oracle State Uncertainty). Let $\mathbf{X}_{\mathcal{M}_t} = \{X_0^\ell\}_{\ell \in \mathcal{M}_t}$ be the random vector of the true tokens at all [MASK] positions. Oracle State Uncertainty is the Shannon entropy of the model’s joint predictive distribution over this vector, conditioned on the latent state \mathbf{z}_t :

$$H_{\text{oracle}}(\mathbf{z}_t) \triangleq H(\mathbf{X}_{\mathcal{M}_t} | \mathbf{z}_t). \quad (3)$$

Then, the *State Entropy* is defined to measure the average instantaneous prediction uncertainty.

Definition 2 (State Entropy). For a given latent sequence \mathbf{z}_t at time $t \in [0, 1]$, State Entropy (h_{DE}) is defined as the average Shannon entropy of the model predictive distributions over the set of masked positions \mathcal{M}_t :

$$h_{\text{DE}}(\mathbf{z}_t) \triangleq \frac{1}{|\mathcal{M}_t|} \sum_{\ell \in \mathcal{M}_t} H(p_\theta(X_0^\ell | \mathbf{z}_t, t)), \quad (4)$$

where $H(\cdot)$ denotes the Shannon entropy and the definition is valid for $|\mathcal{M}_t| > 0$.

The integral of *State Entropy* across the entire path then yields *Path Entropy*, which quantifies the cumulative uncertainty of the generative process.

Definition 3 (Path Entropy). Path Entropy (H_{DE}) provides measure for the total uncertainty of a denoising process with a decoding path τ . It is the integral of the h_{DE} over the generation time. In a discrete-time setting with N steps, it can be approximated by averaging the h_{DE} across all steps:

$$H_{DE}(\tau) \triangleq \int_0^1 h_{DE}(\mathbf{z}_t) dt \approx \frac{1}{N} \sum_{i=1}^N h_{DE}(\mathbf{z}_{t_i}). \quad (5)$$

3.3 ENTROPY-GUIDED DECODING

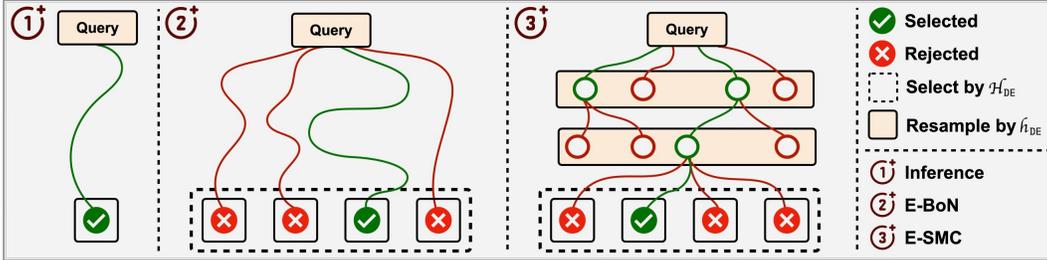


Figure 2: **An overview of our entropy-guided decoding algorithms.** While *standard inference* in subfigure (1) generates a single decoding path, our methods explore multiple candidate paths guided by Denoising Entropy. **E-BON** in subfigure (2) performs **post-hoc selection**, choosing the best path from multiple independent candidates based on path entropy H_{DE} . **E-SMC** in subfigure (3) provides **real-time guidance**, iteratively pruning high-entropy paths and replicating low-entropy ones based on state entropy h_{DE} of partial solutions.

A decoding strategy π induces a distribution $P(\tau|\pi)$ over paths and their corresponding path entropies $H_{DE}(\tau)$. This implies two properties: (i) a single stochastic strategy yields a variance in path uncertainty, $\mathbb{V}_{\tau \sim P(\cdot|\pi)}[H_{DE}(\tau)] > 0$; and (ii) distinct strategies, π_a and π_b , produce different expected uncertainties, $\mathbb{E}_{\pi_a}[H_{DE}(\tau)] \neq \mathbb{E}_{\pi_b}[H_{DE}(\tau)]$.

Hypothesizing that low path uncertainty lead to higher-quality outputs (Xu et al., 2020), generative decoding can be framed as an optimization problem. For a generative task, the objective is to find an optimal strategy π^* that minimizes the expected path entropy:

$$\pi^* = \arg \min_{\pi} \mathbb{E}_{\tau \sim P(\cdot|\pi)}[H_{DE}(\tau)]. \quad (6)$$

While finding the globally optimal strategy π^* is intractable due to the vast search space, we introduce two algorithms that leverage denoising entropy as an active guidance signal: (i) a *post-hoc selection method*, **Entropy-based Best-of-N (E-BON)**, and (ii) a *real-time path optimization strategy*, **Entropy-guided Sequential Monte Carlo (E-SMC)**, both designed to effectively approximate this objective. Figure 2 provides a conceptual overview E-BON, E-SMC, and standard single-path inference.

3.3.1 ENTROPY-BASED BEST-OF-N

H_{DE} can be used for post-hoc selection from a set of candidates. E-BON formalizes this principle. Given a population of M candidate decoding paths, $\{\tau^{(1)}, \dots, \tau^{(M)}\}$, E-BON identifies the single best path with minimum H_{DE} :

$$\tau^* = \arg \min_{m \in \{1, \dots, M\}} H_{DE}(\tau^{(m)}). \quad (7)$$

E-BON is straightforward, requiring the full generation of M samples before selection. However, it expends the full computational budget uniformly across all paths, without any mechanism to redirect resources during generation (Chatterjee & Diaconis, 2018).

3.3.2 ENTROPY-GUIDED SEQUENTIAL MONTE CARLO

To actively optimize the decoding path in real-time, we introduce **E-SMC**, a variant of Sequential Monte Carlo (Doucet et al., 2001) adapted for MDM reverse process (Singhal et al., 2025). E-SMC employs a guidance mechanism that operates on M parallel particles. By evaluating each partial

Algorithm 1 Entropy-guided Sequential Monte Carlo

```

216
217 1: Input: MDM  $\theta$ , number of particles  $M$ , temperature  $\lambda$ , resampling interval  $\Delta i_r$ .
218 2: Initialize: Sample initial particle set  $\mathcal{Z}_{t_N} = \{\mathbf{z}_{t_N}^{(m)}\}_{m=1}^M$  where  $\mathbf{z}_{t_N}^{(m)} \sim p(\mathbf{z}_{t_N})$ .
219 3: for  $i = N, N - 1, \dots, 1$  do
220 4:   Propagate: For each particle  $m \in \{1, \dots, M\}$ , sample  $\mathbf{z}_{t_{i-1}}^{(m)} \sim p_\theta(\cdot | \mathbf{z}_{t_i}^{(m)})$ .
221 5:   if  $(N - i + 1) \pmod{\Delta i_r} = 0$  and  $i > 1$  then
222 6:     Evaluate: For each particle  $m$ , compute potential  $G^{(m)} \leftarrow \Phi(\mathbf{z}_{t_{i-1}}^{(m)}; \lambda)$ .
223 7:     Normalize to get probabilities:  $w^{(m)} \leftarrow G^{(m)} / \sum_{j=1}^M G^{(j)}$  for all  $m$ .
224 8:     Resample: Draw ancestor indices  $\{a^{(m)}\}_{m=1}^M$  where  $a^{(m)} \sim \text{Categorical}(\{w^{(j)}\}_{j=1}^M)$ .
225 9:     Update particle set:  $\mathcal{Z}_{t_{i-1}} \leftarrow \{\mathbf{z}_{t_{i-1}}^{(a^{(m)})}\}_{m=1}^M$ .
226 10:   end if
227 11: end for
228 12: Return: Final population  $\mathcal{Z}_{t_0}$ .

```

path via State Entropy h_{DE} , E-SMC can dynamically reallocate computational budget and prune high-entropy paths while replicating low-entropy ones. This iterative search process is structured around three steps: *Propagation, Evaluation, and Resampling*:

- **Propagation.** At each reverse step $i \in \{N, \dots, 1\}$, we advance each particle $\mathbf{z}_{t_i}^{(m)}$ in the set \mathcal{Z}_{t_i} to a new state $\mathbf{z}_{t_{i-1}}^{(m)}$ by sampling from the proposal distribution given by MDM reverse kernel, $\mathbf{z}_{t_{i-1}}^{(m)} \sim p(\cdot | \mathbf{z}_{t_i}^{(m)})$. This process generates the subsequent particle population $\mathcal{Z}_{t_{i-1}}$.
- **Evaluation.** To actively guide the search, evaluation are performed periodically at fixed intervals of Δi_r steps. During evaluation, each particle $\mathbf{z}_{t_{i-1}}^{(m)}$ in current set $\mathcal{Z}_{t_{i-1}}$ is assessed using a potential function $\Phi(\mathbf{z}; \lambda)$. Φ is designed to assign a higher score to particles with lower $h_{\text{DE}}(\mathbf{z})$. Temperature λ modulates the sharpness of the resulting score distribution. Φ is detailed in Appendix A.
- **Resampling.** Potential scores computed during evaluation are normalized to form a categorical distribution over the current particle set $\mathcal{Z}_{t_{i-1}}$. This distribution assigns a selection probability $w^{(m)}$ to each particle. A new population is formed by drawing M particles with replacement from $\mathcal{Z}_{t_{i-1}}$ according to $w^{(m)}$. Resampling mechanism prunes high-entropy paths while replicating low-entropy ones, steering search towards more promising path space (Moral, 2004). Full process is detailed in Algorithm 1.

3.4 THEORETICAL JUSTIFICATION FOR DENOISING ENTROPY

The effectiveness of E-BON and E-SMC, is predicated on Denoising Entropy being a indicator for generation quality. This section validates the use of this metric as a decoding objective by establishing its theoretical basis through two key properties. First, we prove that State Entropy h_{DE} is a computable upper bound on the ideal, yet intractable, joint entropy of the masked tokens (Proposition 1). Second, we relate h_{DE} directly to the model’s training objective, showing it serves as a close proxy for the instantaneous loss (Proposition 2). Together, these results justify minimizing Denoising Entropy as a principled strategy for guiding the generative process toward more consistent and higher-quality outputs. We also provide a further analysis of Denoising Entropy properties in Appendix D.

h_{DE} as a bound on ideal uncertainty. h_{DE} is a well-posed measure of uncertainty by relating it to an ideal but intractable metric, *Oracle State Uncertainty* H_{oracle} , defined as the joint entropy over all [MASK] tokens. h_{DE} serves as a computable upper bound to this oracle.

Proposition 1 (h_{DE} as an upper bound). *Oracle State Uncertainty $H_{\text{oracle}}(\mathbf{z}_t)$ is upper-bounded by the sum of marginal entropies, which is directly proportional to State Entropy $h_{\text{DE}}(\mathbf{z}_t)$:*

$$H_{\text{oracle}}(\mathbf{z}_t) \leq \sum_{\ell \in \mathcal{M}_t} H(p_\theta(X_0^\ell | \mathbf{z}_t, t)) = |\mathcal{M}_t| \cdot h_{\text{DE}}(\mathbf{z}_t). \quad (8)$$

h_{DE} sums the uncertainty of each masked token independently, thereby ignoring contextual constraints that tokens impose on each other and forming a valid upper bound. Proof relying on the subadditivity of Shannon entropy (Shannon, 1948) is provided in Appendix B.1.

h_{DE} as a proxy for MDM loss. We define ϵ -accurate condition with the step-wise error.

Definition 4 (ϵ -Accurate Denoising Model). We say an MDM is ϵ -accurate if, for any time step t_i , the Kullback-Leibler(KL) divergence between the true posterior transition kernel q (conditioned on ground truth) and the model’s reverse kernel p_θ is bounded:

$$\mathbb{E}_{\mathbf{z}_{t_i} \sim q} \left[D_{\text{KL}} \left(q(\mathbf{z}_{t_{i-1}} | \mathbf{z}_{t_i}) \parallel p_\theta(\mathbf{z}_{t_{i-1}} | \mathbf{z}_{t_i}) \right) \right] \leq \epsilon, \quad (9)$$

where ϵ is a small constant reflecting the model’s training loss.

Under the assumption of ϵ -accuracy, h_{DE} serves as a direct proxy to approximate the instantaneous per-token training loss with error of $\mathcal{O}(\epsilon)$. We state the approximation using state entropy as follows:

Proposition 2 (Approximation of Normalized Loss by h_{DE}). Assume that MDM is ϵ -accurate. $h_{\text{DE}}(\mathbf{z}_t)$ provides an approximation to the instantaneous, per-token normalized training loss:

$$\frac{1}{|\mathcal{M}_t|} \sum_{\ell \in \mathcal{M}_t} (-\log p_\theta(x_0^\ell | \mathbf{z}_t, t)) - h_{\text{DE}}(\mathbf{z}_t) \leq \epsilon + \sqrt{\frac{\epsilon}{2}} \log K, \quad (10)$$

where K is the vocabulary set size.

Proof in Appendix B.2. While h_{DE} measures instantaneous difficulty, H_{DE} extends by integrating it throughout the path, measuring total generation challenge and reflecting distribution uncertainty.

Entropy gap as output quality bound. We denote the true and model’s path distribution as Pr and $\widehat{\text{Pr}}$ respectively for decoding path τ . Let $\mu_{\text{Pr}} = \mathbb{E}_{\tau \sim \text{Pr}}[H_{\text{DE}}(\tau)]$, $\mu_{\widehat{\text{Pr}}} = \mathbb{E}_{\tau \sim \widehat{\text{Pr}}}[H_{\text{DE}}(\tau)]$ be the expected Path Entropy evaluated on reference paths and model’s generations, respectively.

Proposition 3 (Path Entropy Gap as a Quality Bound). Assume that $|H_{\text{DE}}(\tau)| \leq B$ where B is a positive constant. The KL divergence between the reference path distribution Pr and the generated one $\widehat{\text{Pr}}$ is lower-bounded by the squared error in their expected Path Entropies:

$$D_{\text{KL}}(\text{Pr} \parallel \widehat{\text{Pr}}) \geq \frac{1}{2B^2} (\mu_{\widehat{\text{Pr}}} - \mu_{\text{Pr}})^2. \quad (11)$$

Making $\mu_{\widehat{\text{Pr}}}$ closer to μ_{Pr} is necessary for lower KL divergence, which represents higher output quality. Our algorithm E-BON and E-SMC both reduce the path entropy gap $|\mu_{\widehat{\text{Pr}}} - \mu_{\text{Pr}}|$, tightening the lower bound for the KL divergence. Details in Appendix C.2 and C.3.

4 EXPERIMENTS

This section empirically studies Path Uncertainty and Denoising Entropy in MDMs through three stages: (i) validating Denoising Entropy as a quality-aligned internal metric on MDMs; (ii) improving generation by using E-BON and E-SMC to optimize decoding paths; (iii) scaling our method to larger models and complex reasoning and planning tasks.

4.1 VALIDATING DENOISING ENTROPY AS A QUALITY METRIC

We study whether the internal metric H_{DE} , computed online during generation as a proxy for path uncertainty, aligns with external text quality evaluations.

Experimental setting. We use a MDLM trained on OpenWebText (Gokaslan et al., 2019) with approximately 130 million non-embedding parameters for evaluation (Sahoo et al., 2024). The MDLM generates sequences of length 1,024 under random-order unmasking. We vary the number of denoising steps $S = 2^i$ for $i \in [5, 10]$ and use GPT2-Large (Radford et al., 2019), a substantially

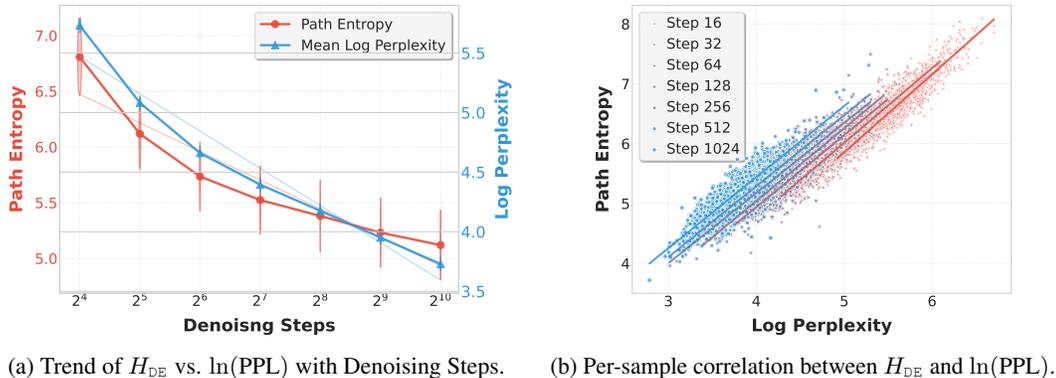


Figure 3: **Empirical validation of Path Entropy (H_{DE}) as an internal proxy for generation quality.** The aggregate trend in (a) shows that a more refined generation process reduces both the model’s internal uncertainty (H_{DE} , red) and the output perplexity ($\ln(PPL)$, blue). (b) provides a per-sample analysis, revealing a strong positive linear correlation between H_{DE} and $\ln(PPL)$. Details in Appendix E.3.

larger model than the MDLM we used, as ARM evaluator to compute Perplexity (PPL). Perplexity is a standard evaluation metric for generative uncertainty and text quality quantification. For each generated sample, we record H_{DE} and $\ln(PPL)$.

Results. *Across thousands of unconditional generations, we observe a clear, near-linear relationship between H_{DE} and $\ln(PPL)$ as shown in Figure 3. Lower H_{DE} aligns with lower perplexity, indicating outputs with higher quality. Both H_{DE} and $\ln(PPL)$ decrease as the number of denoising steps S increases, reflecting a reduction in path uncertainty and a improvement in sample quality. We use this case to show that H_{DE} is a reliable internal proxy for MDM-generated text quality: from its own denoising dynamics, MDM can anticipate whether it is on a high- or low-quality path.*

4.2 IMPROVING GENERATION BY USING DENOISING ENTROPY TO GUIDE PATH SEARCH

Building upon finding that H_{DE} serves as a internal proxy for text quality, we now investigate whether Denoising Entropy can be actively employed to mitigate Path Uncertainty and improve generation.

Experimental setting. We evaluate and compare three decoding types: vanilla single path uniform sampling, with E-BON and E-SMC. We utilize the same MDLM as in Section 4.1, while adopting GPT2-Large and Llama-3-8B as the ARM evaluators. We analyze performance measured with perplexity and diversity (Zheng et al., 2024) across different numbers of denoising steps (S), particles (K), and resampling intervals (Δi_r). In detail:

- Diversity is used to assess whether repetitive degeneration occurs in generated content. For a sequence of length L that contains K distinct tokens, with each token k occurring L_k times. Diversity D is computed as $D = -\sum_{k=1}^K \frac{L_k}{L} \log \frac{L_k}{L}$.
- For E-BON, we generate K independent particles and return the one with the minimum H_{DE} .
- For E-SMC, we steer K particles using weighted resampling with h_{DE} at intervals of Δi_r steps, and finally return the one with the minimum H_{DE} .
- We also implemented an entropy minimization method based on Greedy Search as a comparison to demonstrate the negative impact of over-optimization. Results are presented in Table 6.

Results. As shown in Table 1, both entropy-guided strategies substantially outperform the vanilla sampler, confirming the practical utility of Denoising Entropy as a control signal. E-SMC consistently achieves the best performance, underscoring the benefit of its active, online guidance mechanism. E-BON and E-SMC reduce perplexity while maintaining diversity comparable to the vanilla sampler. Under specific configurations, E-SMC can improve generation quality without compromising diversity. In contrast, other methods achieve significantly lower perplexity only by sacrificing diversity.

Table 1: **Comparison of decoding strategies via ablation studies on hyperparameters.** The table analyzes the impact of the number of denoising steps (S), number of particles (K), and resampling interval (Δi_r). Perplexity is evaluated on two models: **GPT2-Large** and **Llama-3-8B**. E-SMC consistently shows the best performance, which can be achieved without compromising diversity in certain configurations.

Configuration			Perplexity ↓			Diversity ↑		
S	K	Δi_r	Vanilla	E-BON	E-SMC	Vanilla	E-BON	E-SMC
128	4	32	85.3/84.5	66.4/66.1	63.4/61.9	5.53	5.47	5.48
256	4	32	68.5/66.9	52.3/55.2	47.7/51.5	5.45	5.39	5.40
128	4	32	85.3/84.5	66.4/66.1	63.4/61.9	5.53	5.47	5.48
128	8	32	85.3/84.5	60.3/59.8	56.0/59.0	5.53	5.42	5.43
256	2	32	68.5/66.9	60.0/59.1	55.5/57.0	5.45	5.43	5.43
256	4	32	68.5/66.9	52.3/55.2	47.7/51.5	5.45	5.39	5.40
256	8	32	68.5/66.9	47.5/51.1	40.4/45.1	5.45	5.37	5.31
256	4	8	68.5/66.9	52.3/55.2	43.9/44.4	5.45	5.39	5.31
256	4	16	68.5/66.9	52.3/55.2	45.1/48.1	5.45	5.39	5.35
256	4	32	68.5/66.9	52.3/55.2	47.7/51.5	5.45	5.39	5.40
256	4	64	68.5/66.9	52.3/55.2	50.2/51.8	5.45	5.39	5.42
256	4	128	68.5/66.9	52.3/55.2	53.4/54.1	5.45	5.39	5.46

The results affirm the scalability of our methods, as performance improves directly with the computational budget. Increasing the number of particles (K) significantly enhances outcomes, particularly for E-SMC, which better exploits the expanded search space (e.g., perplexity drops from 47.7 to 36.1 as K increases from 4 to 12). Furthermore, for E-SMC, more frequent resampling (a smaller Δi_r) provides a complementary and computationally efficient lever for improvement by actively pruning high-uncertainty paths.

4.3 SCALING ENTROPY GUIDANCE TO LARGE-SCALE REASONING TASKS

Experiments in Section 4.1 & 4.2 validated Denoising Entropy as a proxy for generation quality and demonstrated the efficacy of our proposed algorithms, E-BON and E-SMC, on a small-scale MDM using perplexity as the evaluation metric. To assess the generalizability and practical impact of our approach, we now scale to advanced MDMs and evaluate on a suite of challenging reasoning benchmarks. This shift enables evaluation beyond abstract quality to task-specific accuracy, testing whether entropy guidance improves problem-solving in large-scale MDMs.

Models and datasets. We conduct experiments on three large MDMs: LLaDA-8B-Instruct (Nie et al., 2025), LLaDA-1.5-8B (Zhu et al., 2025), and Open-dCoder-0.5B (Peng et al., 2025b). LLaDA models are evaluated across five diverse benchmarks spanning three reasoning domains: mathematical reasoning (GSM8K (Cobbe et al., 2021), MATH500 (Lightman et al., 2024)), scientific reasoning (GPQA (Rein et al., 2024)), and planning (Sudoku (Nolte et al., 2024) and Countdown (Ye et al., 2025a)). Open-dCoder-0.5B is an MDM for code generation, we evaluate its performance on HumanEval/HumanEval+ (Chen et al., 2021) and MBPP/MBPP+ (Austin et al., 2021b).

Decoding strategies. We evaluate our path search algorithms with a spectrum of established MDM decoding strategies. These include the standard uniform sampler (Austin et al., 2021a), uncertainty-based samplers (e.g., confidence (Chang et al., 2022), entropy (Ben-Hamu et al., 2025), and margin (Kim et al., 2025)), efficient samplers (Semi-AR (Nie et al., 2025), EB-Sampler (Ben-Hamu et al., 2025), and Fast-dLLM (Wu et al., 2025)), and two recently proposed advanced strategies: P2 (Peng et al., 2025a) and PC-Sampler (Huang et al., 2025a). For block-wise strategy including Semi-AR and Fast-dLLM, we set the number of decoding blocks to 8 for all datasets. A detailed description of each baseline strategy is provided in Appendix E.1.

Results: path search algorithms enhance decoding strategies. We demonstrate the efficacy of E-BON and E-SMC through two experiments. First, to establish their maximum potential, we apply them to PC-Sampler (Huang et al., 2025a) on LLaDA models. This combination sets a new

Table 2: Accuracy (%) of our path search algorithms on LLaDA models across five reasoning and planning benchmarks. When applied to the strong PC-Sampler baseline, both E-BoN and E-SMC consistently improve performance. Gains over the baseline are highlighted in red.

Strategies	GSM8K	MATH500	GPQA	Countdown	Sudoku	Avg.↑
LLaDA-Instruct-8B						
Uniform	48.8	15.0	29.0	14.4	2.2	21.9
Confidence	6.8	3.4	27.9	34.0	23.8	19.2
Entropy	2.2	3.8	28.4	33.8	1.6	14.0
Margin	11.1	1.8	28.4	33.9	26.6	20.4
P2	11.6	3.4	28.2	34.2	24.0	20.3
EB-Sampler	1.6	3.6	29.9	34.1	24.2	18.7
Semi-AR	77.9	27.6	27.7	32.6	0.0	33.2
Fast-dLLM	78.2	28.4	28.6	11.4	0.3	29.4
PC-Sampler	79.3	34.0	28.6	36.3	27.6	41.2
w/ E-BoN	0.5 ↑	1.0 ↑	0.2 ↑	5.9 ↑	0.6 ↑	1.6 ↑
w/ E-SMC	1.9 ↑	1.6 ↑	0.3 ↑	4.1 ↑	1.6 ↑	1.9 ↑
LLaDA-1.5-8B						
Uniform	52.7	20.0	28.1	15.8	3.4	24.0
Confidence	19.2	5.4	29.0	33.8	24.8	22.4
Entropy	12.1	5.0	28.8	34.7	0.2	16.2
Margin	27.9	6.4	28.6	31.8	33.6	25.7
P2	25.4	5.6	28.8	33.6	27.8	24.2
EB-Sampler	12.3	4.8	28.6	34.6	1.6	16.4
Semi-AR	80.7	34.2	26.1	32.4	0.0	34.7
Fast-dLLM	80.8	31.2	27.9	32.9	0.4	34.6
PC-Sampler	82.2	37.4	28.8	35.0	33.4	43.4
w/ E-BoN	0.7 ↑	0.8 ↑	0.1 ↑	5.2 ↑	0.8 ↑	1.5 ↑
w/ E-SMC	1.0 ↑	1.2 ↑	0.2 ↑	4.3 ↑	0.2 ↑	1.4 ↑

state-of-the-art, achieving substantial accuracy gains across five challenging benchmarks as shown in Table 2. Second, to validate their broad applicability, we integrate our methods with five different baseline samplers on Open-dCoder model. As shown by the average accuracy improvements in Figure 4, both algorithms consistently boost every tested sampler, confirming they act as versatile enhancers for a wide array of decoding strategies.

Results: path-level optimization is effective for complex reasoning and planning. The gains are most pronounced on benchmarks requiring multi-step reasoning and planning. For example, on LLaDA-Instruct-8B, E-SMC improves GSM8K accuracy from 79.3% to 81.2% (+1.9%) and the Countdown planning task from 36.3% to 40.4% (+4.1%). These results demonstrate that success on such tasks hinges on a global perspective of the entire solution. Our path search algorithms provide this view, leveraging Denoising Entropy to assess the coherence of the entire generation path, not just a single step.

4.4 OTHER RESULTS AND ABLATION STUDIES

Path, H_{DE} , and accuracy. We validate Denoising Entropy as a proxy for task performance on Sudoku, a benchmark sensitive to decoding order. We use the PC-Sampler hyperparameter λ to control decoding sequentiality, thus generating a spectrum of distinct paths. Smaller λ encourage more random decoding orders, which benefit Sudoku solving, while larger λ values lead to more sequential decoding, harming performance. Figure 5 reveals a negative correlation between H_{DE} and final accuracy, confirming that higher entropy signals a lower-quality generation path.

Budget-Efficiency. Under the same computational budget (using LLaDA-Instruct-8B with 5 particles), our entropy-guided methods outperform Majority Vote. Specifically, on the Sudoku and Countdown task, Majority Vote improves accuracy by (-0.8%, +1.0%), while E-BoN and E-SMC achieve gains of (+0.6%, +5.9%) and (+1.6%, +4.1%), respectively. These results indicate that the

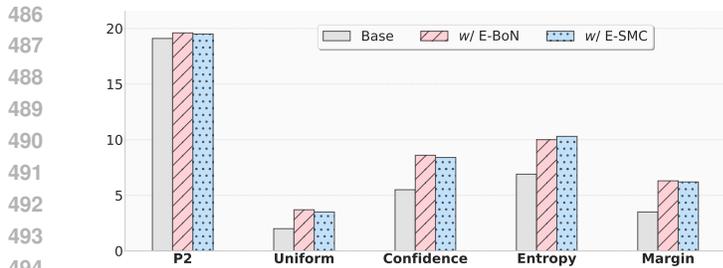


Figure 4: Average accuracy on code benchmarks. E-BON and E-SMC consistently enhance performance of various baseline decoding strategies, demonstrating their broad applicability.

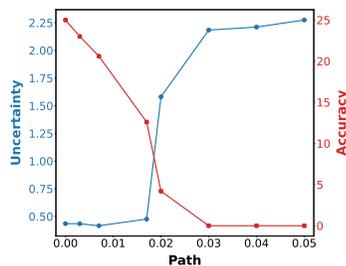


Figure 5: Relationship between decoding path, path uncertainty, and accuracy on Sudoku benchmark.

entropy-guided approach utilizes the sampling budget more effectively than standard ensembling, suggesting greater potential for scaling.

5 RELATED WORK

Masked Diffusion Models. Recent years have seen the extension of diffusion models, originally successful in continuous domains (Ho et al., 2020; Song et al., 2020; Dhariwal & Nichol, 2021; Rombach et al., 2022), to discrete generation (Austin et al., 2021a). A key branch is Masked Diffusion Models, which was formalized in D3PM (Austin et al., 2021a) and explored in text and sequence generation in DiffusionBERT (He et al., 2023). The field matured with substantial theoretical progress simplifying the training objective, making the paradigm more stable and efficient (Zheng et al., 2023; Sahoo et al., 2024; Shi et al., 2024; Ou et al., 2024; Gat et al., 2024). This progress has culminated in the development of large-scale MDMs (Nie et al., 2025; Zhu et al., 2025; Ye et al., 2025b; Gong et al., 2025a;b) that achieve performance competitive with autoregressive counterparts, establishing MDMs as a viable and powerful paradigm for language generation.

Decoding Strategies for Language Models. Decoding strategy is critical for language models. Unlike ARMs, which only select the next token (Graves, 2012), MDMs face a more complex two-dimensional problem: choosing both which position to unmask and what token to generate. Uncertainty-based greedy approaches are most common, where the next position is selected based on local confidence signals like maximum probability (Chang et al., 2022), minimum entropy (Koh et al., 2024; Ben-Hamu et al., 2025), or largest margin (Kim et al., 2025), often building upon a simple random-selection baseline (Austin et al., 2021a). Other approaches impose structure through a semi-autoregressive block-wise order (Han et al., 2023; Nie et al., 2025; Arriola et al., 2025), combining positional bias with content-aware confidence scores (Huang et al., 2025a), or explicitly training a planner (Huang et al., 2025b) to achieve more global control. Some methods improve flexibility by allowing already-decoded tokens to be remasked (Wang et al., 2025; Peng et al., 2025a).

6 CONCLUSION AND DISCUSSION

Generation uncertainty has remained a critical but unquantified aspect for MDMs. Our work is the first to address this gap, introducing Denoising Entropy as a metric designed to explicitly quantify the path-level uncertainty of MDM outputs. We also proposed E-BON and E-SMC, two algorithms that leverage Denoising Entropy to actively guide the generation process toward paths with lower uncertainty. Our experiments demonstrate that these methods significantly improve both the general quality of MDM outputs and performance on challenging benchmarks. Beyond the specific algorithms proposed, we believe Denoising Entropy serves as a foundational tool for uncertainty quantification in MDMs, opening directions such as developing more calibrated decoding strategies and providing internal reward signals for reinforcement learning.

540 **Ethics statement** We confirm that this research does not raise any ethical concerns. Our work
 541 presents a methodological contribution that does not involve human subjects, sensitive data, or
 542 potential societal harms.

543
 544 **Reproducibility Statement** To facilitate the reproducibility of our results, we have included
 545 detailed experimental configurations and hyperparameters in Appendix E.2. We also provide source
 546 code and scripts in the supplementary materials to replicate the empirical results. Our implementation
 547 is designed to be compatible with existing open-source codebases, which we believe will ease the
 548 process of replication and future research.

549

550 REFERENCES

- 551 Kushal Arora, Timothy J O’Donnell, Doina Precup, Jason Weston, and Jackie CK Cheung. The stable
 552 entropy hypothesis and entropy-aware decoding: An analysis and algorithm for robust natural
 553 language generation. *arXiv preprint arXiv:2302.06784*, 2023.
- 554
 555 Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Sub-
 556 ham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between au-
 557 toregressive and diffusion language models. In *The Thirteenth International Conference on*
 558 *Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025. URL <https://openreview.net/forum?id=tyEyYT267x>.
- 559
 560 Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured
 561 denoising diffusion models in discrete state-spaces. *Advances in neural information processing*
 562 *systems*, 34:17981–17993, 2021a.
- 563
 564 Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan,
 565 Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language
 566 models. *arXiv preprint arXiv:2108.07732*, 2021b.
- 567
 568 Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas Nolte, and Brian Karrer. Accelerated sampling from
 569 masked diffusion models via entropy bounded unmasking. *ArXiv preprint*, 2025.
- 570
 571 Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic
 572 language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.
- 573
 574 Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative
 575 image transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 11305–11315, 2022. URL <https://doi.org/10.1109/CVPR52688.2022.01103>.
- 576
 577 Sourav Chatterjee and Persi Diaconis. The sample size required in importance sampling. *The Annals*
 578 *of Applied Probability*, 28(2):1099–1135, 2018.
- 579
 580 Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared
 581 Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large
 582 language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.
- 583
 584 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,
 585 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve
 586 math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- 587
 588 Imre Csiszár and János Körner. *Information theory: coding theorems for discrete memoryless systems*.
 589 Cambridge University Press, 2011.
- 590
 591 Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances*
 592 *in neural information processing systems*, 34:8780–8794, 2021.
- 593
 594 Arnaud Doucet, Nando De Freitas, and Neil Gordon. An introduction to sequential monte carlo
 595 methods. In *Sequential Monte Carlo methods in practice*, pp. 3–14. Springer, 2001.
- 596
 597 Guhao Feng, Yihan Geng, Jian Guan, Wei Wu, Liwei Wang, and Di He. Theoretical benefit and
 598 limitation of diffusion language model. *arXiv preprint arXiv:2502.09622*, 2025.

- 594 Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and
595 Yaron Lipman. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37:
596 133345–133385, 2024.
- 597
- 598 Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. Openwebtext corpus. [http://](http://SkyLion007.github.io/OpenWebTextCorpus)
599 SkyLion007.github.io/OpenWebTextCorpus, 2019.
- 600
- 601 Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An,
602 Peilin Zhao, Wei Bi, Jiawei Han, et al. Scaling diffusion language models via adaptation from
603 autoregressive models. In *The Thirteenth International Conference on Learning Representations*,
604 2025a.
- 605
- 606 Shansan Gong, Ruixiang Zhang, Huangjie Zheng, Jiatao Gu, Navdeep Jaitly, Lingpeng Kong, and
607 Yizhe Zhang. Diffucoder: Understanding and improving masked diffusion models for code
608 generation. *arXiv preprint arXiv:2506.20639*, 2025b.
- 609
- 610 Alex Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*,
611 2012.
- 612
- 613 Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. SSD-LM: semi-autoregressive simplex-
614 based diffusion language model for text generation and modular control. In *Proceedings of*
615 *the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*
616 *Papers)*, *ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 11575–11596, 2023. URL <https://doi.org/10.18653/v1/2023.acl-long.647>.
- 617
- 618 Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu.
619 Diffusionbert: Improving generative masked language models with diffusion models. In *Pro-*
620 *ceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume*
621 *1: Long Papers)*, *ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 4521–4534, 2023. URL
622 <https://doi.org/10.18653/v1/2023.acl-long.248>.
- 623
- 624 Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint*
625 *arxiv:2006.11239*, 2020.
- 626
- 627 Pengcheng Huang, Shuhao Liu, Zhenghao Liu, Yukun Yan, Shuo Wang, Zulong Chen, and Tong
628 Xiao. Pc-sampler: Position-aware calibration of decoding bias in masked diffusion models. *arXiv*
629 *preprint arXiv:2508.13021*, 2025a.
- 630
- 631 Zemin Huang, Zhiyang Chen, Zijun Wang, Tiancheng Li, and Guo-Jun Qi. Reinforcing the diffusion
632 chain of lateral thought with diffusion language models. *arXiv preprint arXiv:2505.10446*, 2025b.
- 633
- 634 Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham Kakade, and Sitan Chen. Train for the worst, plan
635 for the best: Understanding token ordering in masked diffusions. *arXiv preprint arXiv:2502.06768*,
636 2025.
- 637
- 638 Hyukhun Koh, Minha Jhang, Dohyung Kim, Sangmook Lee, and Kyomin Jung. Plm-based discrete
639 diffusion language models with entropy-adaptive gibbs sampling. *arXiv preprint arXiv:2411.06438*,
640 2024.
- 641
- 642 Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for
643 uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*, 2023.
- 644
- 645 Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan
646 Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. In *The Twelfth*
647 *International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11,*
2024, 2024. URL <https://openreview.net/forum?id=v8L0pN6EOi>.
- Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios
of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.
- Pierre Moral. *Feynman-Kac formulae: genealogical and interacting particle systems with applica-*
tions. Springer, 2004.

- 648 Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-
649 Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*,
650 2025.
- 651 Niklas Nolte, Ouail Kitouni, Adina Williams, Mike Rabbat, and Mark Ibrahim. Transformers can
652 navigate mazes with multi-step prediction. *arXiv preprint arXiv:2412.05117*, 2024.
- 653
654 Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li.
655 Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In
656 *The Thirteenth International Conference on Learning Representations*, 2024.
- 657
658 Fred Zhangzhi Peng, Zachary Bezemek, Sawan Patel, Jarrid Rector-Brooks, Sherwood Yao,
659 Avishek Joey Bose, Alexander Tong, and Pranam Chatterjee. Path planning for masked diffusion
660 model sampling. *arXiv preprint arXiv:2502.03540*, 2025a.
- 661
662 Fred Zhangzhi Peng, Shuibai Zhang, Alex Tong, and contributors. Open-dllm: Open
663 diffusion large language models. [https://github.com/pengzhangzhi/
664 Open-dLLM](https://github.com/pengzhangzhi/Open-dLLM), 2025b. Blog: [https://oval-shell-31c.notion.site/
665 Open-Diffusion-Large-Language-Model-25e03bf6136480b7a4ebe3d53be9f68a?
666 pvs=74](https://oval-shell-31c.notion.site/Open-Diffusion-Large-Language-Model-25e03bf6136480b7a4ebe3d53be9f68a?pvs=74), Model: <https://huggingface.co/fredzpz/open-dcoder-0.5B>.
- 667
668 Ori Press, Ravid Shwartz-Ziv, Yann LeCun, and Matthias Bethge. The entropy enigma: Success and
669 failure of entropy minimization. *arXiv preprint arXiv:2405.05012*, 2024.
- 670
671 Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language
672 models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- 673
674 David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani,
675 Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In
676 *First Conference on Language Modeling*, 2024.
- 677
678 Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-
679 resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF confer-
680 ence on computer vision and pattern recognition*, pp. 10684–10695, 2022.
- 681
682 Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu,
683 Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language
684 models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.
- 685
686 Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27
687 (3):379–423, 1948.
- 688
689 Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized
690 masked diffusion for discrete data. *Advances in neural information processing systems*, 37:
691 103131–103167, 2024.
- 692
693 Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and
694 Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion
695 models. *arXiv preprint arXiv:2501.06848*, 2025.
- 696
697 Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv
698 preprint arXiv:2010.02502*, 2020.
- 699
700 Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks.
701 *Advances in neural information processing systems*, 27, 2014.
- 702
703 Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete
704 diffusion models with inference-time scaling. *arXiv preprint arXiv:2503.00307*, 2025.
- 705
706 Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song
707 Han, and Enze Xie. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache
708 and parallel decoding. *ArXiv preprint*, 2025.

702 Jiacheng Xu, Shrey Desai, and Greg Durrett. Understanding neural abstractive summarization models
703 via uncertainty. *arXiv preprint arXiv:2010.07882*, 2020.
704

705 Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong.
706 Beyond autoregression: Discrete diffusion for complex reasoning and planning. In *The Thirteenth
707 International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*,
708 2025a. URL <https://openreview.net/forum?id=NRygUzSPZz>.

709 Jiacheng Ye, Zihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng
710 Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025b.
711

712 Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked
713 diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical
714 sampling. *arXiv preprint arXiv:2409.02908*, 2024.

715 Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. A reparameterized discrete diffusion model for
716 text generation. In *First Conference on Language Modeling*, 2023.
717

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

721 Wenhong Zhu, Hongkun Hao, Zhiwei He, Yiming Ai, and Rui Wang. Improving open-ended text
722 generation via adaptive decoding. *arXiv preprint arXiv:2402.18223*, 2024.
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755

756	CONTENTS	
757		
758	1 Introduction	1
759		
760	2 Preliminaries	2
761	2.1 Forward Process and Training Objective	2
762	2.2 Reverse Process for Generation	3
763		
764	3 Modeling Path Uncertainty with Denoising Entropy	3
765	3.1 Path Uncertainty	3
766	3.2 Denoising Entropy	3
767	3.3 Entropy-Guided Decoding	4
768	3.3.1 Entropy-based Best-of-N	4
769	3.3.2 Entropy-guided Sequential Monte Carlo	4
770	3.4 Theoretical Justification for Denoising Entropy	5
771		
772	4 Experiments	6
773	4.1 Validating Denoising Entropy as a Quality Metric	6
774	4.2 Improving Generation by Using Denoising Entropy to Guide Path Search	7
775	4.3 Scaling Entropy Guidance to Large-Scale Reasoning Tasks	8
776	4.4 Other Results and Ablation studies	9
777		
778	5 Related Work	10
779		
780	6 Conclusion and Discussion	10
781		
782	A E-SMC Implementation Details	16
783		
784	B Theoretical Analysis	16
785	B.1 Proof of Proposition 1	16
786	B.2 Proof and Corollary of Proposition 2	17
787		
788	C Theoretical Analysis of Decoding Algorithms	18
789	C.1 Distributions over Paths	18
790	C.2 Entropy Drift and The Divergence Bounds	19
791	C.2.1 Proof of Proposition 3	19
792	C.2.2 Entropy Gap Lower Bounds Divergence	20
793	C.3 Entropy Drift in Decoding Algorithms	20
794		
795	D Theoretical Properties of State Entropy	21
796		
797	E Experiments	23
798	E.1 Decoding Strategies	23
799	E.2 Experimental Configurations	24
800	E.3 Results	25
801		
802	F Use of Large Language Models	28
803		
804		
805		
806		
807		
808		
809		

810 A E-SMC IMPLEMENTATION DETAILS

811 This appendix provides specific formulas for the **Evaluation** step of the E-SMC algorithm, which
812 were abstracted in the main text for clarity.

813 **Reward Definition.** The raw State Entropy, $h_{\text{DE}}(\mathbf{z}) \in [0, \log_2 V]$ where V is the vocabulary size,
814 is first converted into a normalized reward signal $r(\mathbf{z}) \in [0, 1]$. This is done by inverting and scaling
815 the entropy:

$$816 r(\mathbf{z}) \triangleq \frac{\log_2 V - h_{\text{DE}}(\mathbf{z})}{\log_2 V} = 1 - \frac{h_{\text{DE}}(\mathbf{z})}{\log_2 V}. \quad (12)$$

817 This formulation ensures that a state with minimum entropy (0) receives the maximum reward (1),
818 and a state with maximum entropy ($\log_2 V$) receives the minimum reward (0).

819 **Potential Function.** The reward $r(\mathbf{z})$ is then used to compute the potential score $G(\mathbf{z})$ for each
820 particle, which determines its fitness for resampling. We employ a Gibbs potential function, controlled
821 by a temperature parameter $\lambda > 0$:

$$822 G(\mathbf{z}) \triangleq \exp(\lambda \cdot r(\mathbf{z})). \quad (13)$$

823 The parameter λ controls the selection pressure. A higher λ value more strongly favors particles with
824 high rewards (low entropy), leading to a more aggressive pruning of undesirable trajectories.

825 **Resampling Probabilities.** The probability $\pi^{(k)}$ of selecting particle $\mathbf{z}_{t_{i-1}}^{(k)}$ during the resampling
826 step is given by its normalized potential score:

$$827 \pi^{(k)} = \frac{G(\mathbf{z}_{t_{i-1}}^{(k)})}{\sum_{j=1}^K G(\mathbf{z}_{t_{i-1}}^{(j)})} = \frac{\exp(\lambda \cdot r(\mathbf{z}_{t_{i-1}}^{(k)}))}{\sum_{j=1}^K \exp(\lambda \cdot r(\mathbf{z}_{t_{i-1}}^{(j)}))}. \quad (14)$$

828 This is equivalent to applying a softmax function to the scaled rewards of all particles in the population.
829 The new population is then formed by drawing K samples with replacement from the current
830 population according to these probabilities.

831 B THEORETICAL ANALYSIS

832 B.1 PROOF OF PROPOSITION 1

833 As the sample mean of marginal entropy over the masked set \mathcal{M}_t , $h_{\text{DE}}(\mathbf{z}_t)$ is the expected entropy
834 for the single position ℓ chosen uniformly from \mathcal{M}_t :

$$835 h_{\text{DE}}(\mathbf{z}_t) \triangleq \frac{1}{|\mathcal{M}_t|} \sum_{\ell \in \mathcal{M}_t} H(p_{\theta}(X_0^{\ell} | \mathbf{z}_t, t)) = \mathbb{E}_{\ell \sim \text{Unif}(\mathcal{M}_t)} [H(p_{\theta}(X_0^{\ell} | \mathbf{z}_t, t))].$$

836 This establishes h_{DE} as a measure of average uncertainty over masked tokens.

837 Our proof relies on the subadditivity property of Shannon entropy. We state it here as a lemma.

838 **Lemma 1 (Subadditivity of Conditional Entropy)** . For any set of random variables
839 $\{Y_1, \dots, Y_n\}$ and a conditioning variable Z , the joint conditional entropy is bounded by the
840 sum of the individual conditional entropy:

$$841 H(Y_1, \dots, Y_n | Z) \leq \sum_{i=1}^n H(Y_i | Z).$$

842 Equality holds if and only if the variables $\{Y_i\}$ are conditionally independent given Z .

843 We now restate and prove Proposition 1.

844 **Proposition 1.** Oracle State Uncertainty $H_{\text{oracle}}(\mathbf{z}_t)$ is upper-bounded by the sum of the marginal
845 entropies, which is directly proportional to State Entropy $h_{\text{DE}}(\mathbf{z}_t)$:

$$846 H_{\text{oracle}}(\mathbf{z}_t) \leq |\mathcal{M}_t| \cdot h_{\text{DE}}(\mathbf{z}_t).$$

864 *Proof.* By definition, $H_{\text{oracle}}(\mathbf{z}_t)$ is the conditional joint entropy $H(\mathbf{X}_{\mathcal{M}_t}|\mathbf{z}_t)$. We can derive the
865 bound as follows:
866

$$\begin{aligned}
 867 \quad H_{\text{oracle}}(\mathbf{z}_t) &= H(\mathbf{X}_{\mathcal{M}_t}|\mathbf{z}_t) && \text{by Definition 1} \\
 868 \quad &\leq \sum_{\ell \in \mathcal{M}_t} H(X_0^\ell|\mathbf{z}_t) && \text{by Lemma 1} \\
 869 \quad &= \sum_{\ell \in \mathcal{M}_t} H(p_\theta(X_0^\ell|\mathbf{z}_t, t)) && \text{by definition of MDM output} \\
 870 \quad &= |\mathcal{M}_t| \cdot \left(\frac{1}{|\mathcal{M}_t|} \sum_{\ell \in \mathcal{M}_t} H(p_\theta(X_0^\ell|\mathbf{z}_t, t)) \right) \\
 871 \quad &= |\mathcal{M}_t| \cdot h_{\text{DE}}(\mathbf{z}_t). && \text{by Definition 2}
 \end{aligned}$$

872
873 The inequality in the second step is strict if the random variables $\{X_0^\ell\}_{\ell \in \mathcal{M}_t}$ are not conditionally
874 independent given the latent state \mathbf{z}_t . In the context of natural language, where token occurrences are
875 highly correlated, this condition for strict inequality is almost always met. \square
876

877 B.2 PROOF AND COROLLARY OF PROPOSITION 2

878 Here, we provide the detailed proof for Proposition 2 and discuss the correlation between negative
879 evidence lower bound \mathcal{L} and H_{DE} .
880

881 The proof relies on the decomposition of cross-entropy loss:
882

883 **Lemma 2 (Decomposition of Cross-Entropy Loss)**. *Let $q(X)$ be the true data distribution and*
884 *$p_\theta(X)$ be the model's estimation distribution. For any random variable X , the cross-entropy loss*
885 *can be decomposed as:*

$$886 \quad \mathbb{E}_{X_0^\ell \sim q(\cdot|\mathbf{z}_t)}[-\log p_\theta(X_0^\ell|\mathbf{z}_t, t)] = H(q) + D_{\text{KL}}(q \| p_\theta(X_0^\ell|\mathbf{z}_t, t)),$$

887 where $q = q(X_0^\ell|\mathbf{z}_t)$ denotes the true posterior distribution, $H(\cdot)$ is the Shannon entropy and
888 $D_{\text{KL}}(\cdot \| \cdot)$ is the Kullback-Leibler divergence.
889

890 The proposition states that with the assumption of ϵ -accurate model, the gap between h_{DE} and training
891 loss can be upper-bounded:
892

$$893 \quad \frac{1}{|\mathcal{M}_t|} \sum_{\ell \in \mathcal{M}_t} (-\log p_\theta(x_0^\ell|\mathbf{z}_t, t)) - h_{\text{DE}}(\mathbf{z}_t) \leq \epsilon + \sqrt{\frac{\epsilon}{2}} \log K. \quad (15)$$

894
895 *Proof.* For an ϵ -accurate model, the KL divergence between the posterior distribution $q(x_0^\ell|\mathbf{z}_t)$ and
896 the estimation distribution $p_\theta(x_0^\ell|\mathbf{z}_t, t)$, which is less than $D_{\text{KL}}(q(\mathbf{z}_{t_{i-1}}|\mathbf{z}_{t_i}) \| p_\theta(\mathbf{z}_{t_{i-1}}|\mathbf{z}_{t_i}))$, is
897 upper-bounded by ϵ :
898

$$899 \quad D_{\text{KL}}(q(x_0^\ell|\mathbf{z}_t) \| p_\theta(x_0^\ell|\mathbf{z}_t, t)) \leq \epsilon. \quad (16)$$

900 With the upper bound as above, we turn to bound the gap between entropy $H(p_\theta(x_0^\ell|\mathbf{z}_t, t))$ and
901 $H(q(x_0^\ell|\mathbf{z}_t))$. We use q, p_θ for simplification.
902

903 Let $\delta(q, p_\theta) \triangleq \frac{1}{2} \sum_{x \in \mathcal{V}} |q(x) - p_\theta(x)|$ be the total variation distance on the size- K vocabulary set \mathcal{V} .
904 Since $D_{\text{KL}}(q \| p_\theta) \leq \epsilon$, we obtain $\delta(q, p_\theta) \leq \sqrt{\frac{\epsilon}{2}}$ based on Pinsker's Inequality (Csiszár & Körner,
905 2011).
906

907 Using Audenaert-Fannes Inequality, we obtain the bound of shannon entropy gap:
908

$$909 \quad |H(q) - H(p_\theta)| \leq \delta \log(K - 1) + h(\delta), \quad (17)$$

910 where $h(\delta)$ is binary entropy function $h(\delta) = -\delta \log \delta - (1 - \delta) \log(1 - \delta)$. Since $h(\delta)$ (bounded
911 with $\delta \leq \sqrt{\frac{\epsilon}{2}}$) is small term that approaches 0 when ϵ approaches 0, we omit this term in the following
912 inequality. Thus, we have $|H(q) - H(p_\theta)| \leq \delta \log(K)$.
913

Putting the above inequalities together, we obtain the final result:

$$\mathbb{E}_{X_0^\ell \sim q(\cdot|\mathbf{z}_t)}[-\log p_\theta(X_0^\ell|\mathbf{z}_t, t)] \leq h_{\text{DE}}(\mathbf{z}_t) + \epsilon + \sqrt{\frac{\epsilon}{2}} \log K. \quad (18)$$

By the law of large numbers, the average loss over masked positions approximates the expectation:

$$\begin{aligned} \frac{1}{|\mathcal{M}_t|} \sum_{\ell \in \mathcal{M}_t} (-\log p_\theta(x_0^\ell|\mathbf{z}_t, t)) &\approx \mathbb{E}_{X_0^\ell \sim q(\cdot|\mathbf{z}_t)}[-\log p_\theta(X_0^\ell|\mathbf{z}_t, t)], \\ \frac{1}{|\mathcal{M}_t|} \sum_{\ell \in \mathcal{M}_t} (-\log p_\theta(x_0^\ell|\mathbf{z}_t, t)) - h_{\text{DE}}(\mathbf{z}_t) &\leq \epsilon + \sqrt{\frac{\epsilon}{2}} \log K. \end{aligned}$$

□

Justification for the Correlation between \mathcal{L} and H_{DE} . The full objective (negative evidence lower bound, NELBO) \mathcal{L} and our metric $H_{\text{DE}}(\tau)$ are strongly correlated, which is direct corollary of Proposition 2.

We begin with the definition of the NELBO objective:

$$\mathcal{L}(\mathbf{x}_0) = \int_0^1 w(t) \left[\sum_{\ell \in \mathcal{M}_t} -\log p_\theta(x_0^\ell|\mathbf{z}_t, t) \right] dt, \quad (19)$$

where the weighting function is $w(t) = \left| \frac{d\alpha_t}{dt} \right| \frac{1}{1-\alpha_t}$. Assume that $\epsilon \rightarrow 0$, we take the approximation by h_{DE} from Proposition 2:

$$\sum_{\ell \in \mathcal{M}_t} -\log p_\theta(x_0^\ell|\mathbf{z}_t, t) \approx |\mathcal{M}_t| \cdot h_{\text{DE}}(\mathbf{z}_t). \quad (20)$$

Substituting this back into the $\mathcal{L}(\mathbf{x}_0)$:

$$\mathcal{L} \approx \int_0^1 w(t) \cdot |\mathcal{M}_t| \cdot h_{\text{DE}}(\mathbf{z}_t) dt. \quad (21)$$

In expectation, $|\mathcal{M}_t| \approx L(1 - \alpha_t)$, where L is the sequence length. Thus, the term $w(t) \cdot |\mathcal{M}_t| \approx L \cdot \left| \frac{d\alpha_t}{dt} \right|$ is a strictly positive weighting function of time t , which we denote as $w'(t)$. Then the objective is approximated as follows:

$$\mathcal{L} \approx \int_0^1 w'(t) \cdot h_{\text{DE}}(\mathbf{z}_t) dt. \quad (22)$$

In parallel, our metric is the unweighted integral:

$$H_{\text{DE}}(\tau) = \int_0^1 h_{\text{DE}}(\mathbf{z}_t) dt. \quad (23)$$

Since both \mathcal{L} and $H_{\text{DE}}(\tau)$ are integrals of the same underlying function, $h_{\text{DE}}(\mathbf{z}_t)$, with one being uniformly weighted and the other weighted by a positive function $w'(t)$, a strong positive correlation between their values is mathematically expected. This provides rigorous justification for the remark.

C THEORETICAL ANALYSIS OF DECODING ALGORITHMS

In this section, we provide a rigorous theoretical justification for minimizing Path Uncertainty as a proxy for improving generation quality. We establish a direct link between our proposed metric and the divergence from the true data distribution.

C.1 DISTRIBUTIONS OVER PATHS

To rigorously compare the uncertainty of generated samples against real data, we first align their metrics. While $H_{\text{DE}}(\tau)$ is defined over a decoding path $\tau = (\mathbf{z}_{t_N}, \dots, \mathbf{z}_{t_0})$, real data only consists of static sample sequence \mathbf{x}_0 . We bridge this gap by defining the **True Path Distribution** via the forward diffusion process.

Definition 5 (Path Distributions) . We consider two distributions over the space of paths:

- **Model Distribution** ($\widehat{\mathcal{T}}$): The distribution of paths generated by the model’s reverse process, starting from Gaussian noise:

$$\widehat{\text{Pr}}(\tau) = p(\mathbf{z}_{t_N}) \prod_{i=N}^1 p_{\theta}(\mathbf{z}_{t_{i-1}} | \mathbf{z}_{t_i}), \quad (24)$$

where the starting state is sampled from the prior $\mathbf{z}_{t_N} \sim p(\cdot)$.

- **Reference (True) Distribution** (\mathcal{T}^*): The distribution of paths induced by applying the forward corruption process $q(\mathbf{z}_t | \mathbf{x}_0)$ to ground-truth data $\mathbf{x}_0 \sim \mathcal{D}_{real}$:

$$\text{Pr}(\tau) = q(\mathbf{x}_0) \prod_{i=1}^N q(\mathbf{z}_{t_i} | \mathbf{z}_{t_{i-1}}, \mathbf{x}_0), \quad (25)$$

where $\mathbf{x}_0 \sim q(\cdot)$ is sampled from the real dataset \mathcal{D}_{real} .

Here, $\text{Pr}(\tau)$ represents the distributions of the "ideal" paths that stay perfectly on the manifold of the forward process. For a well-trained model, evaluating H_{DE} on the reversed reference paths should yield low uncertainty, as the states are consistent with the training objective. Conversely, generated paths $\tau \sim \widehat{\text{Pr}}$ that deviate from this manifold (e.g., via hallucinations) are expected to induce higher model uncertainty.

The discrepancy between the generated paths and the true data manifold originates from imperfect approximation at each denoising step. Assume that MEM is ϵ -accurate. Although ϵ is small locally, the generation process involves a sequence of N transitions. For the full path distributions defined as $\text{Pr}(\tau)$ (Reference) and $\widehat{\text{Pr}}(\tau)$ (Model), the chain rule of KL divergence implies that errors accumulate additively:

$$D_{\text{KL}}(\text{Pr}(\tau) \parallel \widehat{\text{Pr}}(\tau)) = \sum_{i=N}^1 \mathbb{E}_{\mathbf{z}_{t_i} \sim q} [D_{\text{KL}}(q(\cdot | \mathbf{z}_{t_i}) \parallel p_{\theta}(\cdot | \mathbf{z}_{t_i}))] \leq N \cdot \epsilon. \quad (26)$$

Equation 26 establishes that even a well-trained model ($\epsilon \rightarrow 0$) can exhibit a non-negligible path divergence $N\epsilon$ due to the cumulative nature of the generative process. This macroscopic divergence is the root cause of the "drift" phenomenon.

C.2 ENTROPY DRIFT AND THE DIVERGENCE BOUNDS

C.2.1 PROOF OF PROPOSITION 3

We formally quantify the entropy discrepancy as **Entropy Drift**. Let $\mu_{\text{Pr}} = \mathbb{E}_{\tau \sim \text{Pr}}[H_{\text{DE}}(\tau)]$ be the expected Path Entropy evaluated on reference paths (representing the model’s uncertainty when guided by ground truth), and let $\mu_{\widehat{\text{Pr}}} = \mathbb{E}_{\tau \sim \widehat{\text{Pr}}}[H_{\text{DE}}(\tau)]$ be the expected Path Entropy of the model’s autonomous generations.

Proof. We now show that the gap between the two Path Entropy quantities lower-bounds the divergence between the generated distribution and the true distribution.

Lemma 3 (Pinsker’s Inequality) . Let P and Q be two distributions over the path space. For any bounded measurable functional $f(\tau)$ such that $|f(\tau)| \leq B$, the difference in expectations is bounded by the KL divergence:

$$|\mathbb{E}_{\tau \sim P}[f(\tau)] - \mathbb{E}_{\tau \sim Q}[f(\tau)]| \leq B \sqrt{2D_{\text{KL}}(P \parallel Q)}. \quad (27)$$

Applying Lemma 3 with $f(\tau) := H_{\text{DE}}(\tau)$, we obtain the following bound:

$$D_{\text{KL}}(\text{Pr} \parallel \widehat{\text{Pr}}) \geq \frac{1}{2B^2} (\mu_{\widehat{\text{Pr}}} - \mu_{\text{Pr}})^2.$$

□

Remark. Equation above establishes a necessary condition for high-fidelity generation. Ideally, we expect the generated distribution to match the reference distribution ($D_{\text{KL}} \rightarrow 0$), but Theorem 3 implies that it is impossible unless the entropy of the generated paths matches the entropy of the reference paths ($\mu_{\widehat{\text{Pr}}} \rightarrow \mu_{\text{Pr}}$). Since reference paths Pr correspond to ground-truth data evolution where the model is typically confident (low μ_{Pr}), and autonomous generations often drift into high-uncertainty regions (high $\mu_{\widehat{\text{Pr}}}$), minimizing the expected Path Entropy $\mu_{\widehat{\text{Pr}}}$ effectively reduces the gap, implicitly constraining the generative process closer to the true data manifold.

C.2.2 ENTROPY GAP LOWER BOUNDS DIVERGENCE

Since computing the high-dimensional divergence $D_{\text{KL}}(\text{Pr} \parallel \widehat{\text{Pr}})$ directly is intractable, we propose using Path Entropy as a measurable proxy. We formally show that the KL divergence is upper-bounded with ϵ -accurate condition and lower-bounded with the measurable gap in expected path entropy.

Theorem 1 (Bounds of KL Divergence) . *Let Pr and $\widehat{\text{Pr}}$ be the reference and generated distributions over path space; $\mu_{\text{Pr}} = \mathbb{E}_{\tau \sim \text{Pr}}[H_{\text{DE}}(\tau)]$ and $\mu_{\widehat{\text{Pr}}} = \mathbb{E}_{\tau \sim \widehat{\text{Pr}}}[H_{\text{DE}}(\tau)]$ be the Path Entropy expectation on the reference and model distributions, respectively.*

Assume that MDM is ϵ -accurate for the model’s reverse kernel p_θ and true transition kernel q , the KL-divergence of distributions Pr and $\widehat{\text{Pr}}$ is upper-bounded as in Eq. 26. By applying Pinsker’s Inequality to path functionals, the divergence is lower-bounded by the squared entropy gap:

$$N \cdot \epsilon \geq D_{\text{KL}}(\text{Pr} \parallel \widehat{\text{Pr}}) \geq \frac{1}{2B^2} (\mu_{\widehat{\text{Pr}}} - \mu_{\text{Pr}})^2 . \quad (28)$$

Implication: This inequality chain reveals the full mechanism:

- **Source:** Local approximation errors (ϵ) accumulate over N steps.
- **State:** This accumulation creates a distributional shift $D_{\text{KL}}(\text{Pr} \parallel \widehat{\text{Pr}})$.
- **Observation:** This shift forces drift in observable statistics, specifically creating a gap $|\mu_{\widehat{\text{Pr}}} - \mu_{\text{Pr}}|$.

Therefore, if we observe a large entropy gap (Entropy Drift), it implies that the model has deviated significantly from the reference path. Conversely, our algorithms aim to reduce this entropy gap $|\mu_{\widehat{\text{Pr}}} - \mu_{\text{Pr}}| \rightarrow 0$, which is a necessary condition for tightening the divergence bound and mitigating the accumulated error.

C.3 ENTROPY DRIFT IN DECODING ALGORITHMS

Motivated by Theorem 3, our algorithms, E-BON and E-SMC, aim to bridge the divergence gap by aligning the expected path entropy of the generated distribution $\mu_{\widehat{\text{Pr}}}$ with that of the reference μ_{Pr} .

E-BON. E-BON approximates a truncated distribution $\widehat{\text{Pr}}_{\text{E-BON}}$ by rejecting paths with high H_{DE} . A potential concern is that the algorithm might retain path samples with extremely low uncertainty (e.g., repetition). For a well-designed model $\widehat{\text{Pr}}$, the sampling probability of such paths is low. The model tends to sample problematic sequences with high entropy or sequences with normal entropy; then the finite candidate path set $S = \{\tau_1, \dots, \tau_N\}$ is unlikely to contain path samples with extremely low entropy. E-BON reduces the expectation $\mathbb{E}_{\tau \sim \widehat{\text{Pr}}_{\text{E-BON}}}[H_{\text{DE}}(\tau)]$. Considering that S consists solely of samples from the region where $\mu_{\widehat{\text{Pr}}} \leq \mu_{\text{Pr}}$, the selection process brings it closer to the reference baseline μ_{Pr} . By Theorem 3, bridging this path entropy gap is a prerequisite for reducing the distributional divergence.

E-SMC. E-SMC modifies the transition kernel by re-weighting particles with $w \propto \exp(-\lambda h_{\text{DE}})$ at fixed intervals of Δ_{i_r} steps, which can be viewed as constructing a calibrated distribution $\widehat{\text{Pr}}_{\text{E-SMC}}$ that penalizes transitions into high-entropy states. By actively avoiding regions where the model is uncertain (which are characteristic of $\widehat{\text{Pr}}$ but rare in Pr), E-SMC promotes that the accumulated Path Entropy remains low. A similar potential concern is that when the entropy of $\mu_{\widehat{\text{Pr}}}$ is lower than the reference μ_{Pr} , reducing the entropy $\mu_{\widehat{\text{Pr}}}$ would increase the gap. There we can set the temperature λ near the optimal one λ^* , which enables $\mu_{\widehat{\text{Pr}}} = \mu_{\text{Pr}}$ and the entropy gap is zero. The existence of λ^* can be verified as follows:

Theorem 2 (Existence and Uniqueness of Optimal Temperature). Let $\mu_{\widehat{\mathcal{P}}_r}(\lambda)$ denote the expected path entropy of the generative process under the E-SMC algorithm with temperature parameter $\lambda \in [0, \infty)$. There exists a unique optimal temperature $\lambda^* \in (0, \infty)$ such that the expected generated entropy matches the reference entropy:

$$\mu_{\widehat{\mathcal{P}}_r}(\lambda^*) = \mu_{\mathcal{P}_r}. \quad (29)$$

Proof. Consider a set of M particles with state entropies $\{H_1, \dots, H_M\}$ before evaluation, the m -th particle is retained with probability $\pi_m(\lambda) = \frac{\exp(-\lambda H_m)}{\sum_{j=1}^M \exp(-\lambda H_j)}$ in the E-SMC resampling step. The expected entropy after resampling is given by the function $\tilde{H} : [0, \infty) \rightarrow \mathbb{R}$:

$$\tilde{H}(\lambda) = \sum_{m=1}^M \pi_m(\lambda) H_m = \frac{\sum_{m=1}^M H_m \exp(-\lambda H_m)}{\sum_{j=1}^M \exp(-\lambda H_j)}. \quad (30)$$

Continuity. Since the denominator $\sum_{j=1}^M \exp(-\lambda H_j)$ is a sum of strictly positive terms for all $\lambda \in \mathbb{R}$, $\tilde{H}(\lambda)$ is continuous and differentiable on $[0, \infty)$.

Monotonicity. Since $\frac{d}{d\lambda} e^{-\lambda H} = -H e^{-\lambda H}$, we obtain:

$$\frac{d}{d\lambda} \tilde{H}(\lambda) = - \left(\sum_{m=1}^M \pi_m(\lambda) H_m^2 - \left(\sum_{m=1}^M \pi_m(\lambda) H_m \right)^2 \right) = -\text{Var}\pi(\lambda)(H). \quad (31)$$

Since the variance $\text{Var}\pi(\lambda)(H)$ is non-negative, the derivative $\frac{d}{d\lambda}$ is non-positive. Assuming the particle entropies $\{H_m\}$ are not all identical, the variance is strictly positive, implying $\tilde{H}(\lambda)$ is strictly decreasing with respect to λ .

- **Case $\lambda = 0$:** The weights become uniform with $\pi_m(0) = 1/M$. The output distribution is consistent with the original model. Given the Entropy Drift phenomenon, we have $\mu_{\widehat{\mathcal{P}}_r} > \mu_{\mathcal{P}_r}$ when $\lambda = 0$.
- **Case $\lambda \rightarrow \infty$:** The retaining probability of resampling steps concentrates on the particle with the minimum entropy, that is,

$$\lim_{\lambda \rightarrow \infty} \pi_m(\lambda) = \begin{cases} 1, & \text{if } H_m = \min_i H_i, \\ 0, & \text{otherwise.} \end{cases}$$

This corresponds to Model Collapse, where $\lim_{\lambda \rightarrow \infty} \mu_{\widehat{\mathcal{P}}_r}(\lambda) < \mu_{\mathcal{P}_r}$.

The function $\mu_{\widehat{\mathcal{P}}_r}(\lambda)$ is continuous and strictly monotonically decreasing on the domain $[0, \infty)$. The reference entropy $\mu_{\mathcal{P}_r}$ lies strictly between the boundary values $\lim_{\lambda \rightarrow \infty} \mu_{\widehat{\mathcal{P}}_r}(\lambda)$ and $\mu_{\widehat{\mathcal{P}}_r}(0)$. By the **Intermediate Value Theorem**, there exists a λ^* such that $\mu_{\widehat{\mathcal{P}}_r}(\lambda^*) = \mu_{\mathcal{P}_r}$. Furthermore, due to the strict monotonicity of the function, this λ^* is unique. \square

Remark. While Theorem 2 establishes λ as a primary control for entropy, the resampling interval Δi_r also governs the final uncertainty. Since the autonomous generative process inherently exhibits *Entropy Drift*, uncertainty accumulates between resampling steps. Consequently, a larger Δi_r allows greater accumulation of drift before resampling occurs, leading to a higher expectation of entropy. Thus, λ (resampling strength) and Δi_r (resampling frequency) jointly determine the accumulated entropy of the generation.

D THEORETICAL PROPERTIES OF STATE ENTROPY

We present theoretical analysis to verify the validity of the state entropy $h_{\text{DE}}(\mathbf{z}_t)$. We begin by verifying its asymptotic behavior at the diffusion process boundaries. Then, we establish its monotonicity with respect to context information.

Theorem 3 (Asymptotic Behavior of state Entropy) . Let $\hat{\mathbf{p}}_\theta(\cdot|\mathbf{z}_t, t)$ be a denoising model that approximates the true posterior distribution $q(\mathbf{z}_t|x_0)$. Assume the following limits exist, then the expected state entropy $\mathbb{E}[h_{\text{DE}}(\mathbf{z}_t)]$ satisfies:

- **Fully-Noised State** ($t \rightarrow 1$): As $t \rightarrow 1$, $\alpha_t \rightarrow 0$ and \mathbf{z}_1 becomes independent of the original sequence \mathbf{x}_0 . The entropy converges to the marginal distribution of tokens in the training data, p_{data}

$$\lim_{t \rightarrow 1} \mathbb{E}_{\mathbf{z}_t \sim q(\mathbf{z}_t|x_0)}[h_{\text{DE}}(\mathbf{z}_t)] = H(p_{\text{data}}). \quad (32)$$

- **Noise-Free State** ($t \rightarrow 0$): As $t \rightarrow 0$, $\alpha_t \rightarrow 1$ and the input z_0 is the clean data x_0 . The model performs reconstruction, and the entropy converges to 0:

$$\lim_{t \rightarrow 0} \mathbb{E}_{\mathbf{z}_t \sim q(\mathbf{z}_t|x_0)}[h_{\text{DE}}(\mathbf{z}_t)] = 0. \quad (33)$$

Proof. The proof examines the two temporal boundaries of the diffusion process.

Asymptotic behavior as $t \rightarrow 1$. As $t \rightarrow 1$, the noise schedule satisfies $\lim_{t \rightarrow 1} \alpha_t = 0$. The forward process marginal, $q(z_t|x_0) = \text{Cat}(z_t; \alpha_t x_0 + (1 - \alpha_t)\mathbf{m})$, converges to a categorical distribution concentrated entirely on the mask token:

$$\lim_{t \rightarrow 1} q(\mathbf{z}_t|x_0) = \text{Cat}(\mathbf{z}_t; \mathbf{m}), \quad (34)$$

where $\text{Cat}(\cdot; \boldsymbol{\pi})$ denotes the categorical distribution with probability vector $\boldsymbol{\pi}$, and \mathbf{m} is the one-hot vector representing the [MASK] token. Consequently, $\mathbf{z}_1 = [\mathbf{m}, \dots, \mathbf{m}]$ is deterministic and independent of \mathbf{x}_0 , with all positions masked ($\mathcal{M}_1 = \{1, \dots, L\}$). For an optimally trained denoising model, the prediction at every masked position converges to the marginal data distribution:

$$\forall \ell \in \mathcal{M}_1, \quad \lim_{t \rightarrow 1} \mathbf{p}_\theta^\ell(\mathbf{z}_t, t) = p_{\text{data}}. \quad (35)$$

The state entropy therefore satisfies:

$$h_{\text{DE}}(\mathbf{z}_1) = \frac{1}{L} \sum_{\ell=1}^L H(p_{\text{data}}) = H(p_{\text{data}}). \quad (36)$$

As \mathbf{z}_1 is a deterministic state, the expectation is equal to the value itself.

Asymptotic behavior as $t \rightarrow 0$. In the opposite limit $t \rightarrow 1$, $\alpha_t \rightarrow 1$. The forward process marginal converges to the clean data:

$$\lim_{t \rightarrow 0} q(\mathbf{z}_t|x_0) = \text{Cat}(\mathbf{z}_t, \mathbf{x}_0), \implies z_0 = x. \quad (37)$$

In this regime, the model has access to nearly the entire ground-truth context. For any masked position $\ell \in \mathcal{M}_t$ where t is small, an optimal model's prediction $p_\theta^\ell(x_0|z_t, t)$ is conditioned on a context that is almost entirely the ground truth, $\{x_0^j \mid j \notin \mathcal{M}_t\}$. Formally, the predictive distribution converges to a point mass on the true token:

$$\lim_{t \rightarrow 0} \hat{\mathbf{p}}_\theta^\ell(\mathbf{z}_t, t) = \delta_{k, x_0^\ell}. \quad (38)$$

The Shannon entropy of a Dirac delta distribution is zero:

$$\lim_{t \rightarrow 0} H(\hat{x}_\theta^\ell(z_t, t)) = H(\delta_{k, x_0^\ell}) = 0. \quad (39)$$

As this holds for every masked position, the average entropy $h_{\text{DE}}(\mathbf{z}_t)$ converges to zero. The convergence of the expectation follows from the deterministic nature of the limit and the boundedness of the entropy function.

This completes the verification of both asymptotic behaviors. \square

The asymptotic analysis confirms that h_{DE} behaves intuitively at process boundaries. We now establish its monotonicity with respect to context information.

Lemma 4 (Concavity of Entropy). *The Shannon entropy $H(p)$ is a concave function on the probability simplex Δ^{K-1} .*

Proof. The entropy function $H(p) = -\sum_{i=1}^K p_i \log p_i$ is a sum of terms $f(p_i) = -p_i \log p_i$. Since $f''(p_i) = -1/p_i < 0$ for $p_i > 0$, each $f(p_i)$ is strictly concave. The sum of concave functions remains concave. \square

We now turn to investigating the context-sensitivity property of the state entropy. Let $\mathcal{O}_t \subseteq \{1, \dots, L\}$ denote the set of indices corresponding to unmasked tokens a time t .

Theorem 4 (Context-Sensitivity of State Entropy). *Let \mathbf{z}_t and \mathbf{z}'_t be latent variables with observed(unmasked) position sets \mathcal{O}_t and \mathcal{O}'_t respectively, where $\mathcal{O}_t \subset \mathcal{O}'_t$. For any masked position $\ell \notin \mathcal{O}'_t$, revealing additional context does not increase the prediction entropy:*

$$\mathbb{E}[H(\hat{\mathbf{p}}_\theta^\ell(\mathbf{z}'_t, t))] \leq \mathbb{E}[H(\hat{\mathbf{p}}_\theta^\ell(\mathbf{z}_t, t))]. \quad (40)$$

Proof. For an optimal denoising model, the prediction $\hat{p}_\theta^\ell(z_t, t)$ approximates the true posterior distribution $p(x_0^\ell | \{x_0^j\}_{j \in \mathcal{O}_t})$, where \mathcal{O}_t is the set of observed indices in z_t . Let $X = x_0^\ell$ be the target token at position ℓ , $Y = \{x_0^j\}_{j \in \mathcal{O}_t}$ be the initial context, and $Z = \{x_0^j\}_{j \in \mathcal{O}'_t \setminus \mathcal{O}_t}$ be the additional context revealed in z'_t .

By the information-theoretic property that conditioning reduces entropy:

$$H(X|Y, Z) \leq H(X|Y), \quad (41)$$

the entropy of an optimal model’s prediction should converge to the true conditional entropy of the data:

$$\mathbb{E}[H(\hat{\mathbf{p}}_\theta^\ell(\mathbf{z}_t, t))] \rightarrow H(X|Y), \quad (42)$$

$$\mathbb{E}[H(\hat{\mathbf{p}}_\theta^\ell(\mathbf{z}'_t, t))] \rightarrow H(X|Y, Z). \quad (43)$$

Combining these inequalities yields the desired result Equation 40. \square

Theorem 4 demonstrates that h_{DE} properly decreases with increasing context information, validating its use as a measure of uncertainty in masked diffusion models.

E EXPERIMENTS

E.1 DECODING STRATEGIES

Here we provide a detailed description of the decoding strategies we evaluate.

Uniform (Austin et al., 2021a) At each step, this sampler unmask a fixed number of tokens at positions chosen uniformly at random from the set of currently masked tokens. This is the vanilla sampler in MDMs.

Confidence (Chang et al., 2022) Sampler with confidence score selects tokens for positions where MDM prediction is most confident, defined as the position ℓ that maximizes the probability of the most likely token, i.e., $\arg \max_{\ell \in \mathcal{M}_t} \max(\hat{\mathbf{p}}_0^\ell)$.

Entropy (Ben-Hamu et al., 2025) Sampler with entropy selects tokens for positions where the MDM prediction is least ambiguous, defined as the position ℓ that minimizes the Shannon entropy of the predicted probability distribution, i.e., $\arg \min_{\ell \in \mathcal{M}_t} H(\hat{\mathbf{p}}_0^\ell)$.

Margin (Kim et al., 2025) Sampler with margin selects tokens for positions with the clearest distinction between the top two candidates, defined as the position ℓ that maximizes the margin $p_{(1)}^\ell - p_{(2)}^\ell$, where $p_{(1)}^\ell$ and $p_{(2)}^\ell$ are the highest and second-highest probabilities in $\hat{\mathbf{p}}_0^\ell$.

EB-Sampler (Ben-Hamu et al., 2025) Entropy-Bounded sampler is an adaptive method that unmask a variable number of tokens, k , at each step. It selects the k tokens with the lowest entropy, where k is dynamically determined by ensuring the cumulative entropy of the selected tokens remains bounded by a predefined threshold γ .

Semi-AR (Nie et al., 2025) Semi-AutoRegressive sampler partitions the sequence into contiguous blocks. It generates tokens within each block in a parallel, non-autoregressive manner, while the blocks themselves are generated sequentially from left to right.

Fast-dLLM (Wu et al., 2025) This sampler accelerates the generation of each block by adaptively unmasking a variable number of tokens at each step. The number of tokens is determined based on whether their prediction confidence exceeds a given threshold, allowing a block to be completed in fewer steps than prescheduled.

PC-Sampler (Huang et al., 2025a) Positional-Confidence sampler selects tokens based on a score that combines a confidence measure (derived from the cross-entropy against a background token distribution) with an exponentially decaying positional bias, thus prioritizing tokens that are confidently predicted and appear earlier in the sequence.

P2 (Peng et al., 2025a) This is a multi-stage sampler that employs a self-correction mechanism. It first generates a high-proportion draft of the sequence by filling the most confident positions. It then enters an iterative refinement phase, where in each step it identifies the *least* confident generated tokens, re-masks them, and immediately re-predicts their content based on the updated context. This process allows MDM to revise and improve its initial predictions.

E.2 EXPERIMENTAL CONFIGURATIONS

Here we provide a detailed description of the configurations for each experiment. Table 3 summarizes the hyperparameters used for experiments on LLaDA model. Table 4 summarizes the hyperparameters used for experiments on Open-dCoder model.

Note on Selection Temperature Parameter. LLaDA experiments utilize a selection temperature parameter ($T_{\text{sel}} = 0.1$) while Open-dCoder experiments do not require this parameter. This difference is because of the generation temperature settings and their impact on path diversity.

In LLaDA experiments, generation temperature is set to 0, making the token prediction process deterministic. Under such deterministic conditions, multiple runs would yield identical paths, preventing path exploration for E-BoN and E-SMC algorithms. The selection temperature introduces controlled randomness during the position selection phase of decoding strategies.

Specifically, when selecting the next k positions to unmask based on confidence scores $\mathbf{s} = [s_1, s_2, \dots, s_n]$, the selection temperature modifies the standard top- k selection through stochastic sampling:

$$\text{Top-}2k \text{ candidates: } \mathbf{s}_{\text{top}} = \text{topk}(\mathbf{s}, 2k) \quad (44)$$

$$\text{Temperature-scaled probabilities: } p_i = \frac{\exp(s_i/T_{\text{sel}})}{\sum_{j \in \text{top-}2k} \exp(s_j/T_{\text{sel}})} \quad (45)$$

$$\text{Stochastic selection: } \text{positions} \sim \text{Multinomial}(\mathbf{p}, k) \quad (46)$$

where T_{sel} controls the randomness level: lower values favor high-confidence positions (approaching deterministic top- k as $T_{\text{sel}} \rightarrow 0$), while higher values increase selection diversity.

Open-dCoder experiments use a generation temperature of 0.8, which introduces stochasticity in the token prediction process and the generation process already satisfies the diversity requirements for path search algorithms.

Table 3: Experimental Configurations for Different Tasks of LLaDA models.

Parameter	GSM8K	MATH500	GPQA	Countdown	Sudoku
Task					
Few-shot examples	4	4	5	3	5
Generation					
Generation length	256	1024	256	128	128
Steps	256	1024	256	128	128
Temperature	0	0	0	0	0
CFG scale	0	0	0	0	0
PC-Sampler					
λ	0.25	0.25	0.25	0.5	0.0
α	10	10	10	10	10
E-BoN & E-SMC					
Number of particles	5	5	5	5	5
λ_{weight}	5.0	5.0	5.0	10.0	5.0
Selection temperature	0.1	0.1	0.1	0.1	0.1
E-SMC					
Resample interval	64	256	64	32	32

Table 4: Experimental Configurations for Different Tasks of Open-dCoder model.

Parameter	HumanEval	HumanEval+	MBPP	MBPP+
Generation				
Generation length	128	128	128	128
Steps	128	128	128	128
Temperature	0.8	0.8	0.8	0.8
E-BoN & E-SMC				
Number of particles	5	5	5	5
λ_{weight}	5.0	5.0	5.0	5.0
E-SMC				
Resample interval	32	32	32	32

E.3 RESULTS

In this section, we present supplementary experimental results that provide further details supporting the findings discussed in the main text.

Table 5 reports the complete raw data from Figure 3.

Table 6 compares decoding algorithms with Greedy Search, where c represents the number of candidates and s represents the beam size.

Table 7 presents the runtime analysis of different decoding algorithms.

Table 8 provides the detailed numerical values underlying Figure 4.

Figure 6 visually illustrates the evolution of state entropy across different particles throughout the decoding process.

Table 5: Mean H_{DE} , Mean Log PPL, and their Correlation at different steps.

Step	Mean H_{DE}	Std H_{DE}	Mean Log PPL	Correlation
16	6.8080	0.4320	5.731	0.891
32	6.1183	0.3927	5.083	0.863
64	5.7352	0.3887	4.661	0.852
128	5.5242	0.3823	4.396	0.853
256	5.3815	0.4018	4.176	0.851
512	5.2318	0.3891	3.951	0.847
1024	5.1197	0.3924	3.729	0.854

Table 6: **Comparison with Greedy Search Baselines.** Perplexity is reported for **GPT2-Large** and **Llama-3-8B**. While Greedy Search achieves lower perplexity by restricting the search space, it significantly degrades diversity compared to E-SMC.

Experiment	Perplexity ↓	Diversity ↑
Vanilla	68.5/66.9	5.45
E-BON ($K = 4$)	52.3/55.2	5.39
E-SMC ($K = 4, \Delta i_r = 128$)	53.4/54.8	5.46
Greedy Search ($c = 2, s = 1$)	32.7/35.9	5.17
Greedy Search ($c = 4, s = 1$)	19.9/24.2	4.83
Greedy Search ($c = 8, s = 1$)	14.0/18.7	4.59
Greedy Search ($c = 16, s = 1$)	10.6/15.7	4.40
Greedy Search ($c = 8, s = 2$)	11.0/15.7	4.42
Greedy Search ($c = 8, s = 4$)	8.8/13.7	4.09
Greedy Search ($c = 8, s = 8$)	7.9/12.1	3.84

Table 7: **Runtime analysis comparison.** We report the computation time (in seconds) across different hyperparameter configurations ($S, K, \Delta i_r$) on a single NVIDIA L40 GPU with MDLM. Parallel implementations (**Par.**) reduce latency compared to sequential baselines (**Seq.**) as the number of particles K increases.

Configuration			Baseline	E-BON		E-SMC	
S	K	Δi_r	Vanilla	Seq.	Par.	Seq.	Par.
256	2	64	7.995	13.273	8.091	13.635	8.650
256	4	64	7.995	26.981	11.236	27.330	12.066
256	8	64	7.995	53.718	17.943	54.617	19.329
1024	2	256	20.256	53.848	32.790	54.652	34.644
1024	4	256	20.256	106.605	46.928	108.457	47.771
1024	8	256	20.256	213.582	75.134	215.317	76.474

Table 8: Comparison of Sampling Strategies with E-BoN and E-SMC on Code Generation Benchmarks with Open-dCoder.

Strategies	HumanEval	HumanEval+	MBPP	MBPP+	Avg.↑
P2	19.3	17.0	16.3	23.7	19.1
w/ E-BoN	0.4 ↑	0.9 ↑	0.5 ↑	0.2 ↑	0.5 ↑
w/ E-SMC	0.2 ↑	0.7 ↑	0.4 ↑	0.3 ↑	0.4 ↑
Uniform	2.7	2.6	1.1	1.6	2.0
w/ E-BoN	2.2 ↑	1.6 ↑	1.3 ↑	1.5 ↑	1.7 ↑
w/ E-SMC	1.8 ↑	1.3 ↑	1.5 ↑	1.3 ↑	1.5 ↑
Confidence	7.0	6.3	2.9	5.9	5.5
w/ E-BoN	3.4 ↑	2.7 ↑	2.2 ↑	3.8 ↑	3.1 ↑
w/ E-SMC	3.1 ↑	2.4 ↑	2.1 ↑	4.0 ↑	2.9 ↑
Entropy	7.6	6.4	6.1	7.5	6.9
w/ E-BoN	3.4 ↑	3.8 ↑	2.7 ↑	2.4 ↑	3.1 ↑
w/ E-SMC	3.7 ↑	4.2 ↑	3.0 ↑	2.6 ↑	3.4 ↑
Margin	4.3	3.5	2.2	3.9	3.5
w/ E-BoN	3.7 ↑	3.3 ↑	2.1 ↑	2.3 ↑	2.8 ↑
w/ E-SMC	3.3 ↑	3.2 ↑	2.2 ↑	2.0 ↑	2.7 ↑

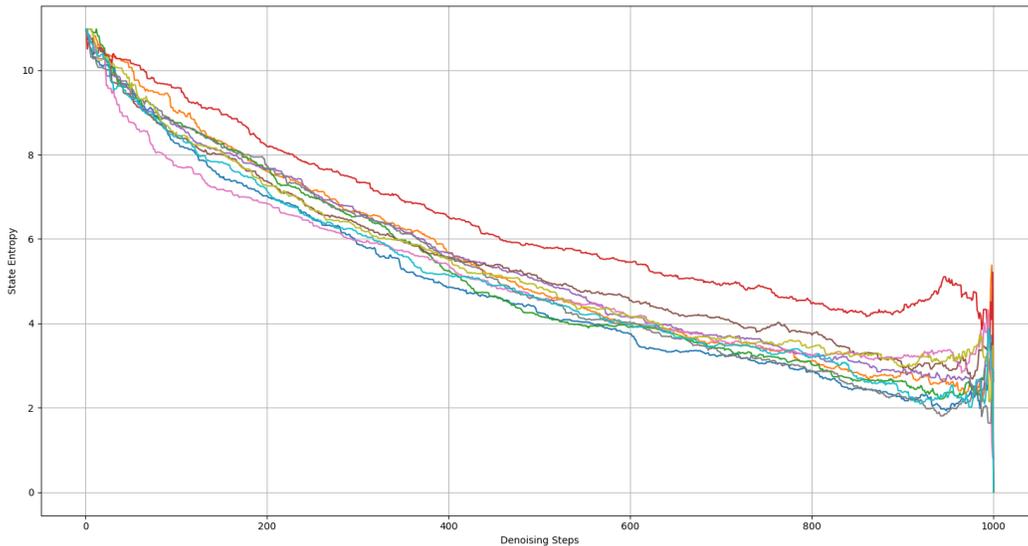


Figure 6: Paths of state entropy over denoising steps for 8 particles, where different colors represent distinct particles. It visually illustrates how the state entropy evolves across each particle throughout the decoding process.

1458 F USE OF LARGE LANGUAGE MODELS
1459

1460 Large language models were used solely as a tool to assist in the writing and polishing of this
1461 manuscript. They were occasionally employed for tasks such as: i) *refining sentence structure for*
1462 *better readability*; ii) *correcting grammatical errors and typos*; and iii) *polishing the phrasing of*
1463 *certain paragraphs*. The core intellectual content, including research ideas, analyses, experimental
1464 designs, and results, did not involve the use of LLMs.

1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511