UNIFIED ENERGY FOR INVARIANT AND INDEPENDENT DECODING IN DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion Language Models (DLMs) enable parallel text generation by iteratively denoising a full sequence, offering attractive flexibility compared to auto-regressive (AR) decoding. However, existing methods fail to fully capture token relationships, leading to a performance gap relative to AR baselines, especially as the degree of parallelism increases. In this paper, we give a systematic analysis of the gap, identifying three key factors: (i) model capacity, (ii) dependency, and (iii) invariance. To address these issues, we first propose an invariant energy (Inv-E) together with an effective sampling-based estimator to handle the invariance issue. By further combining with the independent energy (Ind-E), we obtain a unified energy (Uni-E), that accounts for all these factors. Uni-E enjoys a unique advantage: it can be computed exactly without sampling-based partition estimation. Besides, Uni-E is model agnostic and can therefore be scaled to models of arbitrary size. We further prove that Uni-E can correct the distribution shift caused by dependency and invariance. Extensive experiments across Diffusion Language Models (DLMs) and Diffusion Large Language Models (DLLMs) demonstrate the effectiveness of the proposed Uni-E.

1 INTRODUCTION

Diffusion Language Models (DLMs) Li et al. (2025); Yi et al. (2024) have emerged as a compelling alternative to the dominant auto-regressive (AR) paradigm in text generation, where tokens are generated in a fixed left-to-right order Zhao et al. (2023); Chang et al. (2024); Zou et al. (2023). Unlike AR-based methods, DLMs iteratively decode tokens across the entire sequence in parallel, leading to higher flexibility, diversity, and efficiency. However, traditional DLMs Sahoo et al. (2024); Austin et al. (2021a); Lou et al. (2023); Arriola et al. (2025); Sahoo et al. (2025) still underperform AR-based generation when decoding in parallel, as they assume tokens can be decoded independently and invariantly, failing to model their relationships. To handle this limitation, one line of research uses score guidance Ye et al. (2025); Wu et al. (2025); Li et al. (2026); Luo et al. (2026) or proxy models Liu et al. (2024); Xu et al. (2024) to capture token relationships. Another line of research Wang et al. (2025a); Yao (2026); Zhai et al. (2026) proposes correcting decoding errors in later steps through remasking strategies. Nevertheless, these approaches either lack a systematic understanding of DLM limitations or primarily focus on modeling dependencies among decoded tokens, providing only partial improvements.

To fully address these limitations, we first conduct a theoretical analysis of DLMs. We model the expressive limitation of DLMs as a distribution shift between the diffusion process and the ground-truth. We find that this shift is governed by three factors: (i) the *capacity factor*: the model's ability to fit the target distribution; (ii) the *independency factor*: the model's ability to select independent tokens at each decoding step; and (iii) the *invariance factor*: the model's ability to select tokens that can be totally determined by the current context and remain invariant to masked positions. Limitations in any of these factors can lead to distribution deviation and result in unreasonable outputs. We provide two simple examples in the top left panel of Figure 1: (i) when the model lacks dependency awareness, it fails to capture relationships between tokens (e.g., "cat and mat" or "dog and bone"), leading to an implausible output such as "cat sits on the bone"; and (ii) when the model lacks invariance, it may decode "cat" without sufficient context, which should instead be "car". The first factor can be alleviated by increasing model scale, but a unified solution for all three factors is still lacking.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Based on this insight, we design an unified solution via energy function for all three factors. **First**, we focus on the invariance factor and generalize the re-weighting principle Zhou et al. (2022) from causal inference to text generation. This principle encourages the model to learn invariant mappings by down-weighting non-invariant samples and emphasizing invariant ones. Specifically, we propose an invariant energy (Inv-E) function to implement this idea. Inv-E uses a proxy model to estimate possible future decoding over the entire sequence to capture the token invariance, where it assigns low energy to tokens that remain invariant under the current context and high energy otherwise. We further prove that this estimation can be achieved with a bounded error. **Second**, based on the principle of energy additivity LeCun et al. (2006); Haarnoja et al. (2017), we integrate an existing energy function for the independency factor (Ind-E) Xu et al. (2024) with our Inv-E, resulting in an unified energy function (Uni-E). Uni-E is model-agnostic and can be parameterized at different scales using pretrained parameters (Uni-EDLM-AR) or NCE fine-tuning (Uni-EDLM-NCE). We also show that Uni-E has a unique advantage: its partition function can be computed analytically, enabling accurate and efficient energy computation without Markov Chain Monte Carlo (MCMC) sampling. **Third**, we theoretically analyze the gap to the ground-truth distribution and show that Uni-EDLM achieves a strictly smaller gap than existing DLMs in non-independent or non-invariant scenarios. To validate our method, we conduct experiments on multiple benchmark datasets for both DLMs and Diffusion Large Language Models (DLLMs). For DLMs, Uni-E consistently outperforms baselines and achieves around a 22% speedup while maintaining generation quality comparable to AR models. For DLLMs, Uni-E improves performance across several benchmarks and significantly mitigates degradation as parallelism increases, consistently enhancing performance at different decoding speeds. The code is provided via anonymous link: `https://anonymous.4open.science/r/Uni-E-1CC2`.

2 Related Work

Diffusion Language Models. Diffusion Language Models (DLMs) are proposed as an alternative to auto-regressive (AR) text generation, allowing multiple tokens to be generated in parallel Li et al. (2025). Early work mainly focused on extending diffusion process from continuous data to discrete text Austin et al. (2021a); Hoogeboom et al. (2021); Campbell et al. (2022), which led to the development of DLMs Lou et al. (2023); Austin et al. (2021a); Sahoo et al. (2024); Arriola et al. (2025). Building on this, later studies scaled these models to larger sizes, developing Diffusion Large Language Models (DLLMs) that achieve performance close to Large Language Models (LLMs) Nie et al. (2025); Ye et al. (2025); Bie et al. (2025). However, their performance still lags behind AR-based methods, especially when reducing the decoding steps. This is because they predict multiple tokens independently, making it hard to capture the relationships between them Li et al. (2026).

Dependency-aware Parallel Decoding. To address this, one line of research uses score guidance Ye et al. (2025); Li et al. (2026); Luo et al. (2026) or leverages a proxy model Liu et al. (2024); Xu et al. (2024); Israel et al. (2025) to estimate token relationships. For example, EDLM Xu et al. (2024) introduces energy-based guidance from an AR model to compensate for dependency effects, while speculative decoding Israel et al. (2025) combines AR logits to improve generation. However, these methods mainly focus on dependency while overlooking another crucial factor: invariance. Another line of work designs remask strategy Wang et al. (2025a); Yao (2026); Zhai et al. (2026) to fix the decoding "error" in future steps, but these these methods are heuristic and unstable without theoretical guarantees, and generally require large decoding steps to perform well. We provide a detailed comparision with remask methods in Appendix A.8. In this work, we provide a systematic analysis of the performance gap and propose a unified solution to capture all these factors.

3 Preliminaries

Diffusion Language Model (DLM): We denote the token sequence as $\mathbf{x}$ and we use subscript to denote the diffision time step and superscript to denote the position. For example, $\mathbf{x}_t^i$ represents the i-th token in the sequence at time step t. As for the diffusion process, we consider the time step range from 0 to 1 where $\mathbf{x}_0$ denotes the clean data and $\mathbf{x}_1$ denotes the fully noised data. Given discrete data $\mathbf{x}_{t-1}$ at time $t-1$, one diffusion step is defined as $q(\mathbf{x}_t|\mathbf{x}_{t-1}) \coloneqq Cat(\mathbf{x}_t; p = \mathbf{x}_{t-1}\mathbf{Q}_t)$ where $Cat(.; p)$ is a categorical distribution parameterized by p, and $\mathbf{Q}_t$ is the transition matrix. The forward process at time t is defined as $q(\mathbf{x}_t|\mathbf{x}_0) \coloneqq Cat(\mathbf{x}_t; p = \mathbf{x}_0\bar{\mathbf{Q}}_t)$, where $\bar{\mathbf{Q}}_t \coloneqq \mathbf{Q}_1\mathbf{Q}_2...\mathbf{Q}_t$.

The reverse process at time t is defined as:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \coloneqq \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} = Cat(\mathbf{x}_{t-1}; p = \frac{\mathbf{x}_t\mathbf{Q}_t^T \odot \mathbf{x}_0\bar{\mathbf{Q}}_{t-1}}{\mathbf{x}_0\bar{\mathbf{Q}}_t\mathbf{x}_t^T}). \tag{1}$$

In this paper, we focus on masked diffusion language models (MDLMs) where the forward process replaces each unmasked token with a mask token $\mathbf{m}$ with probability $(1-\alpha_t)$ at time t, thus the process between any two time steps $0 < s < t < 1$ is simplified as $q(\mathbf{x}_t|\mathbf{x}_s) = Cat(\mathbf{x}_t; \alpha_{t/s}\mathbf{x}_s + (1-\alpha_{t/s})\mathbf{m})$ where $\alpha_{t/s} = \alpha_t/\alpha_s$. The backward process is computed from $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ as:

$$q(\mathbf{x}_s|\mathbf{x}_t, \mathbf{x}_0) = \begin{cases} Cat(\mathbf{x}_s; \mathbf{x}_t), & \text{if } \mathbf{x}_t \neq \mathbf{m} \\ Cat(\mathbf{x}_s; \dfrac{(1-\alpha_s)\mathbf{m} + (\alpha_s - \alpha_t)\mathbf{x}_0}{1-\alpha_t}). & \text{if } \mathbf{x}_t = \mathbf{m} \end{cases} \tag{2}$$

MDLMs parameterize the backward process using a denoising model $\mathbf{f}_\theta(\mathbf{x}_t, t)$ to approximate $\mathbf{x}_0$ in Eq. 2. In practice, they adopt an independence assumption: $\mathbf{f}_\theta(\mathbf{x}_t, t) = \prod_{l=1}^{L} \mathbf{f}_\theta(\mathbf{x}_t^l, t)$, which leads to $p_\theta(\mathbf{x}_s|\mathbf{x}_t) = \prod_{l=1}^{L} p_\theta(\mathbf{x}_s^l|\mathbf{x}_t)$ where L is the sequence length.

Distribution of DLM: Given a sequence $\mathbf{x}_0$, we consider K intermediate time steps ($t = 1, 2, \ldots, K$), and the deriving decoding order is denoted as $\{Z_1, Z_2, ..., Z_K\}$. Each $Z_t = \{i | i \in \{1, 2, ..., L\}\}$ is an index set representing the decoding positions at step t. Based on the framework proposed in Lavenant & Zanella (2025), we model the distribution of DLM by two parts: a decoding planner μ_ϕ and a distribution learner p_θ. The decoding planner μ_ϕ determines the decoding order, while the distribution learner p_θ predicts the token distribution. The overall DLM distribution across all possible decoding orders is defined as $P_{dlm}(\mathbf{x}_0) = \prod_{t=0}^{K} \mu_\phi(Z_t|Z_{>t}, \mathbf{x}_{t+1}) p_\theta(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1})$, where $Z_{>t}$ is the union of $Z_{t+1}, Z_{t+2}, ..., Z_K$ and $\mathbf{x}_t^{Z_t}$ denotes the tokens of $\mathbf{x}_t$ positioned by Z_t. For a specific decoding order $\Psi = \{Z_1, Z_2, ..., Z_K\}$ the distribution learner is written as $p_\theta(\mathbf{x}_0, \Psi) = \prod_{t=0}^{K} p_\theta(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1})$.

4 METHOD

In this section, we first conduct a theoretical analysis about the limitation of Diffusion Language Models (DLMs). We then propose the invariant energy (Inv-E) as the re-weighting function for invariant decoding. Next, we combine Inv-E with the existing independent energy (Ind-E) to obtain a unified energy (Uni-E), which can be easily parameterized and eliminates the need for partition estimation. Finally, we theoretically analyze the effectiveness of our method. The overall framework of our Uni-E is shown in Figure 1.

4.1 EXPRESSIVE LIMITATION OF DLMS

To better understand the expressive limitations of DLMs, we analyze the distribution shift from the ground-truth distribution, defined as the Kullback–Leibler (KL) divergence between the model distribution and the ground-truth distribution: $KL(\pi(\mathbf{x}_0)||P_{dlm}(\mathbf{x}_0)) = \mathbb{E}_{\pi(\mathbf{x}_0)}[log\frac{\pi(\mathbf{x}_0)}{P_{dlm}(\mathbf{x}_0)}] = \mathbb{E}_{\pi(\mathbf{x}_0),\mu_\phi(\mathbf{x}_0,\Psi)}[log\frac{\pi(\mathbf{x}_0)}{p_\theta(\mathbf{x}_0,\Psi)}]$, where $\pi(.)$ denotes the ground-truth distribution. Given a specific order $\Psi = \{Z_1, Z_2, ..., Z_K\}$, the KL divergence can be expanded as:

$$\begin{aligned} log\frac{\pi(\mathbf{x}_0)}{p_\theta(\mathbf{x}_0, \Psi)} &= \sum_t log\frac{\pi(\mathbf{x}_t^{Z_t}|\mathbf{x}_0 - \mathbf{x}_t^{Z_t})}{p_\theta(\mathbf{x}_t^{Z_t}|\mathbf{x}_0 - \mathbf{x}_t^{Z_t})} = \sum_t \left[log\frac{\pi(\mathbf{x}_t^{Z_t}|\mathbf{x}_0 - \mathbf{x}_t^{Z_t})}{\pi(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1})} + log\frac{\pi(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1})}{\prod_{i\in Z_t}\pi(\mathbf{x}_t^i|\mathbf{x}_{t+1})} \right. \\ &\left. + log\frac{\prod_{i\in Z_t}\pi(\mathbf{x}_t^i|\mathbf{x}_{t+1})}{\prod_{i\in Z_t}p_\theta(\mathbf{x}_t^i|\mathbf{x}_{t+1})} + log\frac{\prod_{i\in Z_t}p_\theta(\mathbf{x}_t^i|\mathbf{x}_{t+1})}{p_\theta(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1})} + log\frac{p_\theta(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1})}{p_\theta(\mathbf{x}_t^{Z_t}|\mathbf{x}_0 - \mathbf{x}_t^{Z_t})} \right], \end{aligned} \tag{3}$$

where $\mathbf{x}_0 - \mathbf{x}_t^{Z_t}$ denotes the relative complement of $\mathbf{x}_t^{Z_t}$ to $\mathbf{x}_0$. We find that the KL divergence is governed by the following three factors:

(i) **Capacity Factor**: This factor corresponds to the third term of the expanded KL divergence, which describes the model's capacity to fit the target distribution:

$$L_{lr} \coloneqq log\frac{\prod_{i\in Z_t}\pi(\mathbf{x}_t^i|\mathbf{x}_{t+1})}{\prod_{i\in Z_t}p_\theta(\mathbf{x}_t^i|\mathbf{x}_{t+1})} = \sum_{i\in Z_t} log\frac{\pi(\mathbf{x}_t^i|\mathbf{x}_{t+1})}{p_\theta(\mathbf{x}_t^i|\mathbf{x}_{t+1})}. \tag{4}$$

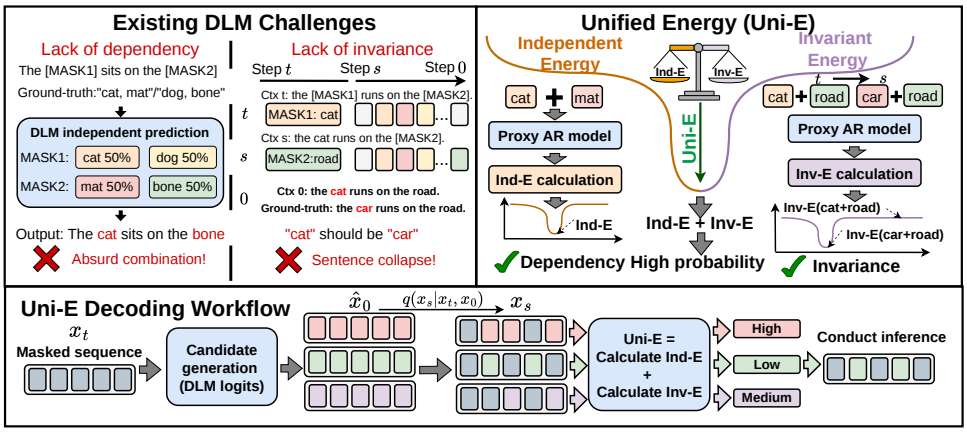

Figure 1: Challenges of Diffusion Language Models and the framework of our unified energy (Uni-E). Uni-E uses a proxy model to capture both dependency and invariance to guide the decoding process.

(ii) **Dependency Factor**: This factor corresponds to the second and fourth terms:

$$L_{dp} := log\frac{\pi(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1})}{p_\theta(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1})} + log\frac{\prod_{i\in Z_t} p_\theta(\mathbf{x}_t^i|\mathbf{x}_{t+1})}{\prod_{i\in Z_t}\pi(\mathbf{x}_t^i|\mathbf{x}_{t+1})} = log\frac{\pi(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1})}{\prod_{i\in Z_t}\pi(\mathbf{x}_t^i|\mathbf{x}_{t+1})} + log\frac{\prod_{i\in Z_t} p_\theta(\mathbf{x}_t^i|\mathbf{x}_{t+1})}{p_\theta(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1})}. \tag{5}$$

One way to reduce this factor is to simultaneously reduce: i) the divergence between $p_\theta(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1})$ and $\pi(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1})$; and ii) the divergence between $p_\theta(\mathbf{x}_t^i|\mathbf{x}_{t+1})$ and $\pi(\mathbf{x}_t^i|\mathbf{x}_{t+1})$ for all positions. However, this is almost infeasible because modeling the joint distribution $p_\theta(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1})$ is difficult, and sampling from $\pi(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1})$ is intractable. An alternative way is to identify independent tokens that can be decoded in one denoising step, such that $\pi(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1}) = \prod_{i\in Z_t}\pi(\mathbf{x}_t^i|\mathbf{x}_{t+1})$, and a well-optimized p_θ should also satisfy $p_\theta(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1}) = \prod_{i\in Z_t} p_\theta(\mathbf{x}_t^i|\mathbf{x}_{t+1})$. Therefore, this factor is totally bounded by the independency of Z_t.

(iii) **Invariance Factor**: This factor corresponds to the first and last terms:

$$L_{iv} := log\frac{\pi(\mathbf{x}_t^{Z_t}|\mathbf{x}_0 - \mathbf{x}_t^{Z_t})}{p_\theta(\mathbf{x}_t^{Z_t}|\mathbf{x}_0 - \mathbf{x}_t^{Z_t})} + log\frac{p_\theta(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1})}{\pi(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1})} = log\frac{\pi(\mathbf{x}_t^{Z_t}|\mathbf{x}_0 - \mathbf{x}_t^{Z_t})}{\pi(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1})} + log\frac{p_\theta(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1})}{p_\theta(\mathbf{x}_t^{Z_t}|\mathbf{x}_0 - \mathbf{x}_t^{Z_t})}. \tag{6}$$

Due to the same reasons of sampling and joint distribution modeling, we require that the decoded tokens at each step can be fully determined. This leads to $\pi(\mathbf{x}_t^{Z_t}|\mathbf{x}_0 - \mathbf{x}_t^{Z_t}) = \pi(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1})$. And a well-optimized model should satisfy $p_\theta(\mathbf{x}_t^{Z_t}|\mathbf{x}_0 - \mathbf{x}_t^{Z_t}) = p_\theta(\mathbf{x}_t^{Z_t}|\mathbf{x}_{t+1})$. This factor describes invariance in decoding: the distribution of decoded tokens (e.g. $\mathbf{x}_t^{Z_t}$) should be fully determined by the current context (e.g. $\mathbf{x}_{t+1}$) and remain unchanged in subsequent steps.

4.2 Energy Function for Invariant Decoding

In order to address the expressive limitations of DLMs, we first propose an effective way to guide invariant decoding for the invariance factor. Our solution is inspired by the re-weighting principle from causal inference Zhou et al. (2022), whose idea is that models can learn invariant representations across different feature parts (e.g., objects and background) by reweighting samples using clustering, graph partitioning, or neural network guidance. We generalize this principle to text generation in two ways: (i) we treat tokens in the sequence as basic units and consider their invariance, and (ii) we adjust the token distribution using a re-weighting function to ensure that the posterior distribution of the current decoding is close to the posterior distribution of its invariant part. Specifically, for the i-th token at step t, we assume its invariant part is $\mathcal{V}_i$, satisfying $\pi(\mathbf{x}_t^i|\mathbf{x}_0 - \mathbf{x}_0^i) = \pi(\mathbf{x}_t^i|\mathbf{x}_0^{\mathcal{V}_i})$. This re-weighting function enforces $\mathcal{W}(\mathbf{x}_t^i, \mathbf{x}_{t+1})p_\theta(\mathbf{x}_t^i|\mathbf{x}_{t+1}) = \pi(\mathbf{x}_t^i|\mathbf{x}_0^{\mathcal{V}_i})$. We define the current context intersection as $\mathcal{V}_i^+ := \mathcal{V}_i \cap Z_{>t}$ and the current context difference as $\mathcal{V}_i^- := \mathcal{V}_i - Z_{>t}$. The re-weighting function can then be derived as:

$$\mathcal{W}(\mathbf{x}_t^i, \mathbf{x}_{t+1}) = \frac{\pi(\mathbf{x}_t^i|\mathbf{x}_0^{\mathcal{V}_i})}{p_\theta(\mathbf{x}_t^i|\mathbf{x}_{t+1})} = \frac{\pi(\mathbf{x}_t^i|\mathbf{x}_0^{\mathcal{V}_i^+}, \mathbf{x}_0^{\mathcal{V}_i^-})}{p_\theta(\mathbf{x}_t^i|\mathbf{x}_{t+1})} = \frac{\pi(\mathbf{x}_0^i, \mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_0^{\mathcal{V}_i^+})}{\pi(\mathbf{x}_0^i|\mathbf{x}_0^{\mathcal{V}_i^+})\pi(\mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_0^{\mathcal{V}_i^+})}\frac{\pi(\mathbf{x}_0^i|\mathbf{x}_0^{\mathcal{V}_i^+})}{p_\theta(\mathbf{x}_t^i|\mathbf{x}_{t+1})}. \tag{7}$$

Here, $\mathbf{x}^{\mathcal{V}_i^+}$ is difficult to obtain directly, so we use a proxy AR-based model $p_{AR}(.|\mathbf{x}_{t+1})$ to estimate $\pi(.|\mathbf{x}^{\mathcal{V}_i^+})$. The reasons behind are: (i) $\mathbf{x}_t^i$ remains invariant to the unmasked tokens in $\mathbf{x}_{t+1}$ except those in $\mathcal{V}_i^+$, so $\mathbf{x}_{t+1}$ can serve as a valid conditioning context; (ii) an AR model pretrained on clean data $\mathbf{x}_0$ is well-suited to model this distribution, and its posterior can be computed efficiently as $p_{AR}(\mathbf{x}_0|\mathbf{x}_t) = p_{AR}(\mathbf{x}_t, \mathbf{x}_0 - \mathbf{x}_t)/p_{AR}(\mathbf{x}_t) = p_{AR}(\mathbf{x}_0)/p_{AR}(\mathbf{x}_t)$ where $p_{AR}(\mathbf{x}_t) = \sum_{Z_t} p_{AR}(\mathbf{x}_0^{Z_t}, \mathbf{x}_0 - \mathbf{x}_t)$ since the unmasked tokens are fixed and unchanged. We represent the re-weighting function as an energy function, called invariant energy (Inv-E), which is defined as: $\mathcal{W}(\mathbf{x}_t^i, \mathbf{x}_{t+1}) = \frac{exp(-E_{inv}(\mathbf{x}_t^i, \mathbf{x}_{t+1}))}{F(\mathbf{x}_t)}$. Based on Eq. 7 we obtain the following energy formulation:

$$E_{inv}(\mathbf{x}_t^i, \mathbf{x}_{t+1}) = log\frac{p_\theta(\mathbf{x}_t^i|\mathbf{x}_{t+1})}{p_{AR}(\mathbf{x}_0^i|\mathbf{x}_{t+1})} + log\frac{p_{AR}(\mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_{t+1})}{p_{AR}(\mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_0^i, \mathbf{x}_{t+1})} - logF(\mathbf{x}_t). \tag{8}$$

Another question is that the term $logp_{AR}(\mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_{t+1})/p_{AR}(\mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_0^i, \mathbf{x}_{t+1})$ cannot be calculated directly, since the difference part $\mathcal{V}_i^-$ is unknown. Fortunately, it can be effectively estimated through Lemma 1. The detailed proof is provided in Appendix A.4.

Lemma 1 (Invariance Sampling Estimation)**.** *The invariance divergence term can be estimated by sampling $Z_{<t}$. Given the sampled index set is Z_s, the average estimation over all possible Z_s satisfies the following property:*

$$\mathbb{E}_{Z_s}\left\{log\frac{p_{AR}(\mathbf{x}_0^{Z_s}|\mathbf{x}_{t+1})}{p_{AR}(\mathbf{x}_0^{Z_s}|\mathbf{x}_0^i, \mathbf{x}_{t+1})}\right\} = \beta log\frac{\pi(\mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_{t+1})}{\pi(\mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_0^i, \mathbf{x}_{t+1})} + \gamma, \tag{9}$$

where β is the approximation degree falling into $(0, 1]$ and γ is the approximation error between $p_{AR}(.)$ and $\pi(.)$. If the sampling number is r and the index length of $\mathcal{V}_i^-$ and $Z_{<t} - \mathcal{V}_i^-$ is x and y respectively, then the average β is $\frac{r}{x+y}$.

For each time step t, we randomly select another time step s satisfying $0 < s < t < 1$. The forward diffusion produces two noised data $\mathbf{x}_s$ and $\mathbf{x}_t$, where $\mathbf{x}_t$ carries all masked tokens from $\mathbf{x}_s$. (i.e., $\mathbf{x}_0 \xrightarrow{mask} \mathbf{x}_s \xrightarrow{mask} \mathbf{x}_t$). We treat $\mathbf{x}_s$ as the "next step" of $\mathbf{x}_t$ and the decoding of the masked tokens in $\mathbf{x}_s$ is regarded as "future decoding"(i.e., invariance sampling Z_s). Based on the energy formulation in Eq. 8 and Lemma 1, Inv-E can be estimated as:

$$E_{inv}(\mathbf{x}_0, \mathbf{x}_s, \mathbf{x}_t) = log\frac{p_\theta(\mathbf{x}_s|\mathbf{x}_t)}{p_{AR}(\mathbf{x}_s|\mathbf{x}_t)} + log\frac{p_{AR}(\mathbf{x}_0|\mathbf{x}_t)}{p_{AR}(\mathbf{x}_0|\mathbf{x}_s)} - logF_{inv}(\mathbf{x}_t), \tag{10}$$

where $F_{inv}(\mathbf{x}_t)$ is the partition function. Intuitively, the first term $log\frac{p_\theta(\mathbf{x}_s|\mathbf{x}_t)}{p_{AR}(\mathbf{x}_s|\mathbf{x}_t)}$ corresponding to term $log\frac{p(\mathbf{x}_t^i|\mathbf{x}_{t+1})}{p_{AR}(\mathbf{x}_0^i|\mathbf{x}_{t+1})}$ in Eq. 8 describes the model ability to predict the next step distribution. The second term $log\frac{p_{AR}(\mathbf{x}_0|\mathbf{x}_t)}{p_{AR}(\mathbf{x}_0|\mathbf{x}_s)}$ acts as the *Invariance Sampling Estimation* describing the invariance, since if the decoding is invariant it will have no impact on the rest tokens (i.e., $p_{AR}(\mathbf{x}_0|\mathbf{x}_t) \approx p_{AR}(\mathbf{x}_0|\mathbf{x}_s)$).

4.3 Unified Energy for Independent and Invariant Decoding

Based on our Inv-E, we further propose an unified energy (Uni-E) for all the factors. For the dependency factor, we adopt the independent energy (Ind-E) proposed in Xu et al. (2024), which is defined as:

$$E_{ind}(\mathbf{x}_0, \mathbf{x}_t) = log\frac{p_\theta(\mathbf{x}_0|\mathbf{x}_t)}{p_{AR}(\mathbf{x}_0|\mathbf{x}_t)} - logF_{ind}(\mathbf{x}_t), \tag{11}$$

where $F_{ind}(\mathbf{x}_t)$ is the partition function. The idea behind Ind-E is that the independence of parallel decoded tokens can be captured by an AR model, as it operates on the entire sequence and decodes tokens sequentially. More details can be found in Xu et al. (2024). We argue that Ind-E can be seamlessly integrated with Inv-E for two reasons: (i) Ind-E acts as the residual term for the DLM distribution, which has a similar form to the re-weighting function used in Inv-E; and (ii) energy functions can be naturally combined due to the principle of additivity LeCun et al. (2006). This leads to the unified energy (Uni-E):

$$E_{uni}(\mathbf{x}_0, \mathbf{x}_s, \mathbf{x}_t) = E_{inv} + E_{ind} = log\frac{p_\theta(\mathbf{x}_s|\mathbf{x}_t)}{p_{AR}(\mathbf{x}_s|\mathbf{x}_t)} + log\frac{p_\theta(\mathbf{x}_0|\mathbf{x}_t)}{p_{AR}(\mathbf{x}_0|\mathbf{x}_s)} - logF(\mathbf{x}_t), \tag{12}$$

where $F(\mathbf{x}_t)$ is the new partition function. This unified formulation simultaneously captures both dependency and invariance through the term $log\frac{p_\theta(\mathbf{x}_0|\mathbf{x}_t)}{p_{AR}(\mathbf{x}_0|\mathbf{x}_s)}$: the AR guidance models the dependency relationships (i.e., $p_\theta(\mathbf{x}_0|\mathbf{x}_t) \approx p_{AR}(\mathbf{x}_0|\mathbf{x}_t)$), while the decoding from $\mathbf{x}_s$/$\mathbf{x}_t$ reflects the invariance (i.e., $p_{AR}(\mathbf{x}_0|\mathbf{x}_t) \approx p_{AR}(\mathbf{x}_0|\mathbf{x}_s)$). We parameterize the unified energy-based model (Uni-EBM) as:

$$E_\phi(\mathbf{x}_0, \mathbf{x}_s, \mathbf{x}_t) = log\frac{p_\theta(\mathbf{x}_s|\mathbf{x}_t)}{p_{AR}(\mathbf{x}_s|\mathbf{x}_t)} + log\frac{p_\theta(\mathbf{x}_0|\mathbf{x}_t)}{p_{AR}(\mathbf{x}_0|\mathbf{x}_s)}, \tag{13}$$

Partition Function Estimation: Estimating the partition function is notoriously intractable. Traditional methods like Markov Chain Monte Carlo (MCMC) sampling or importance sampling require a large number of samples, which is not only inefficient but also inaccurate. Here, we show that *the partition function of E_{uni} can be calculated analytically, avoiding intractable sampling and estimation*. Based on the definition $logF(\mathbf{x}_t) = log\sum_{\mathbf{x}_0} p_\theta(\mathbf{x}_0|\mathbf{x}_t)exp(-E_\phi(\mathbf{x}_0, \mathbf{x}_s, \mathbf{x}_t))$, we derive it as:

$$logF(\mathbf{x}_t) = log\sum_{\mathbf{x}_0} p_\theta(\mathbf{x}_0|\mathbf{x}_t)\frac{p_{AR}(\mathbf{x}_s|\mathbf{x}_t)}{p_\theta(\mathbf{x}_s|\mathbf{x}_t)}\frac{p_{AR}(\mathbf{x}_0|\mathbf{x}_s)}{p_\theta(\mathbf{x}_0|\mathbf{x}_t)} = log\left\{\sum_{\mathbf{x}_0} p_{AR}(\mathbf{x}_0|\mathbf{x}_s)\right\} + log\frac{p_{AR}(\mathbf{x}_s|\mathbf{x}_t)}{p_\theta(\mathbf{x}_s|\mathbf{x}_t)}, \tag{14}$$

where $\sum_{\mathbf{x}_0} p_{AR}(\mathbf{x}_0|\mathbf{x}_s) = 1$, and the partition function is $log\frac{p_{AR}(\mathbf{x}_s|\mathbf{x}_t)}{p_\theta(\mathbf{x}_s|\mathbf{x}_t)}$. This closed-form partition provides a unique advantage: Uni-EBM avoids the need for intractable estimation and enables accurate computation of the evidence lower bound (ELBO). After applying the Uni-EBM $E_\phi(\mathbf{x}_0, \mathbf{x}_s, \mathbf{x}_t)$ to the DLM, we obtain the unified energy-based DLM (Uni-EDLM) as:

$$p_{\phi,\theta}(\mathbf{x}_0|\mathbf{x}_t) = \frac{exp(-E_\phi(\mathbf{x}_0, \mathbf{x}_s, \mathbf{x}_t))}{F(\mathbf{x}_t)} p_\theta(\mathbf{x}_0|\mathbf{x}_t). \tag{15}$$

Uni-EDLM Parameters: Similar to the independent energy-based diffusion language model Xu et al. (2024), there are two ways to obtain the model parameters of Uni-EDLM. The first is to utilize a pretrained AR model and a DLM to directly calculate $E_\phi(\mathbf{x}_0, \mathbf{x}_s, \mathbf{x}_t)$, and we refer to the resulting model as Uni-EDLM-AR. The second one is to fine-tune the pretrained parameters using the Noise Contrastive Estimation (NCE) loss Gutmann & Hyvärinen (2010). In this setting, we freeze the pretrained AR model and fine-tune the parameters of the DLM. We use the clean data $\mathbf{x}_0$ together with $\mathbf{x}_s$, $\mathbf{x}_t$ as positive samples, and we sample $\hat{\mathbf{x}}_0$ from $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$ and derive $\hat{\mathbf{x}}_s$ as negative samples. We refer to the fine-tuned model as Uni-EDLM-NCE. The loss is described as follows, and the training algorithm is in Appendix A.1.

$$\begin{aligned}\mathcal{L}_{NCE}(\phi, \theta) = \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t \sim \pi, q}\{&\mathbb{E}_{\mathbf{x}_s \sim q(.|\mathbf{x}_t, \mathbf{x}_0)} log\frac{1}{1 + exp(E_\phi(\mathbf{x}_0, \mathbf{x}_s, \mathbf{x}_t))} \\ &+ \mathbb{E}_{\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_s \sim p_\theta(.|\mathbf{x}_t), q(.|\mathbf{x}_t, \hat{\mathbf{x}}_0)} log\frac{1}{1 + exp(-E_\phi(\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_s, \mathbf{x}_t))}\}.\end{aligned} \tag{16}$$

Inference of Uni-EDLM: We adapt the self-normalized importance sampling strategy (Hammersley, 2013; James, 1980; Xu et al., 2024) to Uni-EDLM, whose framework is shown at the bottom of Figure 1. At each time step t, we randomly select time step $s < t$. The importance sampling involves: (i) sampling multiple $\mathbf{x}_0$ candidates $\{\mathbf{x}_0^j\}_{j=1}^k$ from the DLM in parallel; (ii) sampling $\mathbf{x}_s^j$ from $\mathbf{x}_0^j$ and $\mathbf{x}_t$ in parallel via the backward process in Eq. 2; (iii) feeding these tuples $\{(\mathbf{x}_0^j, \mathbf{x}_s^j, \mathbf{x}_t)\}_{j=1}^k$ into the energy function in batch to calculate their energies; and (iv) selecting the $\mathbf{x}_0^j$ with the lowest energy. The selected $\mathbf{x}_0^j$ is then fed into $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ to conduct one-step inference. In practice, we also define a sampling window with size w ($w = 0.2$ by default), and apply importance sampling only within the interval $(1 - w, 1]$. The full inference algorithm is provided in Appendix A.2.

Effectiveness Discussion: We analyze the effectiveness of Uni-EDLM with respect to the three limitation factors. First we ask: *Can Uni-EDLM reduce the model distribution shift in non-independent or non-invariant decoding scenarios?* Our answer is **yes**. The effectiveness of Uni-EDLM on dependency and invariance factors is formalized in Lemma 2, with the proof provided in Appendix A.5. Besides, Uni-EBM can be integrated with any scale models to obtain Uni-EDLMs with various size. This addresses the limitation related to the capacity factor.

Lemma 2 (Effectiveness of Uni-EDLM)**.** *Across all the possible decodings, the Kullback-Leibler divergence between Uni-EDLM and the ground-truth distribution is strictly lower than that between DLM and the ground-truth distribution in non-independent or non-invariant scenarios.*

5 EXPERIMENTS

5.1 EXPERIMENTS ON DIFFUSION LANGUAGE MODELS (DLMS)

Settings: Following prior works Xu et al. (2024); Sahoo et al. (2025; 2024); Arriola et al. (2025), we use OpenWebText (OWT) Gokaslan & Cohen (2019) for fine-tuning and evaluation, along with seven zero-shot datasets: PTB Marcus et al. (1993), Wikitext Merity et al. (2016), LM1B Chelba et al. (2013), Lambada Paperno et al. (2016), AG News Zhang et al. (2015), Pubmed and Arxiv Cohan et al. (2018). The baselines include: (i) Transformer (AR) Vaswani et al. (2017); (ii) traditional DLMs: MDLM Sahoo et al. (2024), SEDD Lou et al. (2023), Duo Sahoo et al. (2025) and Block Diffusion Arriola et al. (2025); (iii) dependency-aware DLMs: EDLM-(AR and NCE) Xu et al. (2024) and DCD Liu et al. (2024); and (iv) remask: ReMDM Wang et al. (2025a). Perplexity (PPL) and Generative Perplexity (Gen PPL) are used to measure effectiveness and we follow Xu et al. (2024) to use average wall-clock time to evaluate efficiency. More details are provided in Appendix A.6.

Implementation Details: We follow common practice and use GPT-2-small Radford et al. (2019); Sahoo et al. (2024); Xu et al. (2024) as proxy AR for our method and baselines. We use the pretrained MDLM Sahoo et al. (2024) as the diffusion backbone. Uni-EDLM-AR directly uses these pretrained models, while Uni-EDLM-NCE freezes the AR model and fine-tunes MDLM on 4xH200 GPUs for up to 1,000,000 steps, with a learning rate of 3e-4 and a per-GPU batch size of 16. We save checkpoint every 10,000 steps and select the best model based on the dev set (randomly select 0.1% data). By default, we set the importance sampling candidate size $k = 2$ and window size $w = 0.2$.

Table 1: Perplexity results. The Lower is better. * denotes reproduced results, † taken from Xu et al. (2024), ‡ taken from Sahoo et al. (2025). The best/second-best is in **bold**/underlined (except AR).

Methods	OWT	PTB	WikiText	LM1B	Lambada	AG News	Pubmed	Arxiv
AR†	17.56	82.05	25.75	51.25	51.28	52.09	49.01	41.73
SEDD†	24.56	100.09	34.28	68.20	49.86	62.09	44.53	38.48
MDLM†	23.83	95.26	32.83	67.01	47.52	61.15	41.89	37.37
Duo‡	23.25	89.35	33.57	73.86	49.78	67.81	44.48	40.39
Block Diffusion*	21.27	97.23	31.82	61.39	49.50	62.16	42.18	38.74
DCD*	23.33	95.69	35.20	66.51	<u>46.71</u>	61.35	47.65	40.09
EDLM-NCE†	21.52	93.21	30.77	63.19	46.92	60.02	<u>41.80</u>	<u>36.63</u>
EDLM-CoAR†	17.58	89.73	28.31	<u>60.23</u>	50.04	**57.94**	46.31	39.02
Uni-EDLM-AR	<u>17.41</u>	<u>87.36</u>	<u>26.62</u>	60.51	**46.62**	<u>58.40</u>	**41.35**	37.37
Uni-EDLM-NCE	**16.54**	**86.18**	**24.01**	**58.14**	48.26	58.66	42.74	**35.53**

Table 2: Generative Perplexity under Llama2, Llama3, and GPT2. ↓ means the lower is better. The best results are highlighted with **bold**. * denotes reproduced results, † taken from Xu et al. (2024).

Method	Timesteps	Llama2↓	Llama3↓	GPT2↓	Entropy
Data	-	7.0	9.4	14.7	7.7
AR†	1024	22.9	40.3	35.7	8.1
SEDD†	128 / 256 / 512 / 1024	46.3 / 36.1 / 32.5 / 27.3	86.6 / 65.0 / 54.3 / 43.7	78.5 / 64.8 / 52.2 / 41.5	8.1 / 7.8 / 7.7 / 7.6
MDLM†	128 / 256 / 512 / 1024	48.5 / 37.2 / 30.6 / 27.6	86.7 / 66.9 / 52.6 / 44.6	78.8 / 66.8 / 52.4 / 42.6	8.1 / 7.9 / 7.8 / 7.6
Duo*	128 / 256 / 512 / 1024	45.8 / 35.6 / 29.5 / 25.4	85.3 / 65.4 / 51.9 / 43.7	74.1 / 63.2 / 48.6 / 36.6	7.9 / 7.7 / 7.6 / 7.6
Block Diffusion*	128 / 256 / 512 / 1024	47.1 / 37.9 / 31.1 / 26.7	89.8 / 69.6 / 54.1 / 45.0	83.9 / 66.7 / 51.8 / 40.8	7.7 / 7.5 / 7.3 / 7.1
DCD*	128 / 256 / 512 / 1024	43.8 / 36.6 / 28.5 / 22.2	82.4 / 64.1 / 48.4 / 40.9	74.3 / 63.5 / 47.3 / 38.7	7.9 / 7.6 / 7.6 / 7.5
ReMDM*	128 / 256 / 512 / 1024	47.2 / 35.4 / 29.6 / 21.8	86.4 / 64.0 / 42.8 / 33.0	76.4 / 64.1 / 44.6 / 26.9	8.1 / 7.7 / 7.7 / 7.5
EDLM-AR†	128 / 256 / 512 / 1024	43.2 / 34.7 / 26.8 / 19.0	83.2 / 62.2 / 44.4 / 28.8	71.3 / 62.1 / 42.0 / 25.5	8.0 / 7.9 / 7.6 / 7.2
EDLM-NCE†	128 / 256 / 512 / 1024	43.2 / 35.7 / 26.3 / 19.6	83.5 / 62.9 / 44.1 / 28.8	71.7 / 61.7 / 42.5 / 25.5	8.0 / 7.9 / 7.6 / 7.3
Uni-EDLM-AR	128 / 256 / 512 / 1024	**40.4** / 32.4 / 23.2 / 13.4	**79.9** / **56.5** / 37.6 / 19.2	**68.2** / 55.5 / 37.6 / 19.2	8.1 / 7.9 / 7.5 / 7.2
Uni-EDLM-NCE	128 / 256 / 512 / 1024	40.6 / **32.2** / **22.8** / **13.1**	80.0 / 57.3 / **37.5** / **18.9**	69.4 / **54.3** / **37.5** / **18.9**	8.1 / 7.9 / 7.5 / 7.2

Perplexity (PPL): The PPL results are shown in Table 1. Uni-EDLM significantly outperforms traditional DLMs. In particular, compared to the backbone MDLM, it achieves a 26.9% improvement on OWT without fine-tuning. After fine-tuning, Uni-EDLM even achieves lower perplexity than the AR model. In addition, Uni-EDLM performs better than other energy baselines under both pre-training and fine-tuning settings. These results demonstrate the effectiveness of Uni-EDLM.

Generative Perplexity (Gen PPL): The Gen PPL under different oracle Large Language Models (LLMs), along with entropy, is reported in Table 2. In this evaluation we set energy-based sampling window size w as 1 to show the full power of Uni-E. Uni-EDLM consistently outperforms the baselines across all timesteps. Moreover, Uni-EDLM achieves comparable (22.8 vs 22.9 on Llama2) or better (37.5 vs 40.3 on Llama3) performance than the AR model while using only half of the timesteps (512). This indicates that Uni-E improves generation quality and reduces the number of required timesteps. Furthermore, the improvement over EDLM increases as the number of timesteps grows. A possible reason is that Inv-E reduces non-invariant error accumulation, which benefits more for longer timestep. The remask strategy can only perform well in large timesteps as it needs more steps to fix "error". As for entropy, Uni-EDLM maintains a similar level of diversity as others.

Ablation Studies: We use Uni-EDLM-AR as an example to conduct an ablation study to examine the influence of different energy terms. To show the full effect, we also set the window size w as 1. The Gen PPL via Llama2 is shown in Figure 2. No energy has the same performance as the diffusion backbone MDLM. Using only the invariant energy (Inv-EDLM) provides limited benefits at small timesteps but yields significant improvements at larger timesteps. This observation suggests that invariant decoding is more beneficial at larger timesteps. In contrast, at smaller timesteps, independence becomes more important, as more tokens are decoded at each step. Using only the independent energy (Ind-EDLM) shows a larger gap compared to Uni-EDLM at larger timesteps, due to the lack of invariance modeling.

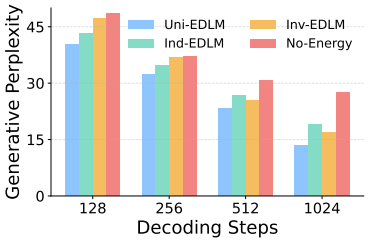

Figure 2: Ablation studies. Inv-EDLM only uses invariant energy. Ind-EDLM only uses independent energy.

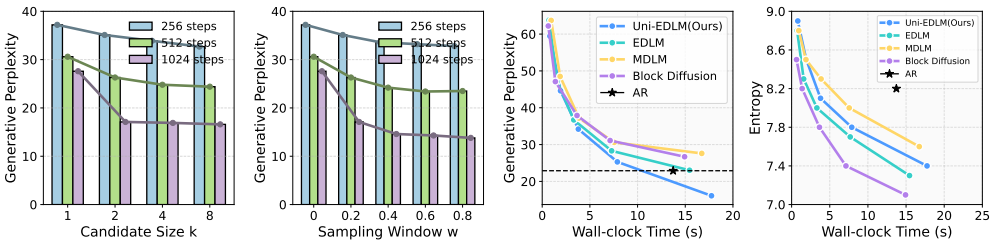

Figure 3: Hyper-parameter and efficiency analysis. From left to right, the first two figures show the influence of candidate size k and sampling window w respectively; the last two figures show the trade-off between Gen PPL/entropy and wall-clock time. Llama2 is used as the oracle model.

Analysis: We conduct a hyperparameter and efficiency analysis using Uni-EDLM-AR, with default settings of $k = 2$ and $w = 0.2$. The results are shown in Figure 3. Increasing the candidate size k mainly improves generation quality at smaller timesteps as additional sampling improves decoding independency, but provides limited gains at larger timesteps. For the sampling window w, applying the energy function in the first 40% decoding steps significantly improves generation quality, while the remaining 60% yields marginal gains (less than 3 points). This aligns with the observation in Xu et al. (2024) that most errors occur early in decoding steps. Based on this, we argue that applying Uni-E only in the first 20% of steps with a candidate size of 2 is sufficient. In terms of efficiency trade-off, Uni-EDLM achieves comparable Gen PPL to the AR model with approximately 22% speedup in wall-clock time (10.5s vs. 13.5s). Moreover, all DLMs exhibit greater generative diversity than the AR model, as measured by entropy.

Case Study: Case study is given in Appendix A.12 to show the effects of invariance and independency.

5.2 Experiments on Diffusion Large Language Models (DLLMs)

Uni-E is model agnostic and can be easily integrated into DLLMs: for $\mathbf{x}_t$ we generates k clean candidates $\{\mathbf{x}_0^j\}_{j=1}^k$ from the DLLM logits. To better align with the decoding trajectory of DLLMs, we use the highest logit of tokens as the weight to select m unmasked positions for each $\mathbf{x}_0^j$, masking the remaining positions in $\mathbf{x}_0^j$ to generate $\mathbf{x}_s^j$. We calculate energy via Eq. 13 and select the $\mathbf{x}_s^j$ with lowest energy as the decoding result. The algorithm of this process are provided in Appendix A.3.

This integration is similar to speculative decoding Israel et al. (2025); Gao et al. (2025), but the key difference is that *existing speculative decodings are heuristic and ignore the decoding invariance.* We provide a detailed discussion in Appendix A.7.

Settings: Following Israel et al. (2025), we use Qwen2.5-0.5B Yang et al. (2025) as the proxy AR model, and LLaDa-8B-Instruct Nie et al. (2025) and Dream-7B-Instruct Ye et al. (2025) as the DLLM backbones. We use these models to compute Uni-E directly. We set $k = 3$ and $m = 4$ for energy-based decoding, and apply energy decoding only in the first 20% of decoding steps. We evaluate on the following benchmark datasets: MATH500 Hendrycks et al. (2021), GSM8K Cobbe et al. (2021), MBPP Austin et al. (2021b), LiveCodeBench V2(LCB V2) Jain et al. (2024), LiveBench White et al. (2024), MMLU Hendrycks et al. (2020) and HellaSwag Zellers et al. (2019). We compare against (i) Qwen2.5-0.5B, Qwen2.5-7B Yang et al. (2025), and Llama3.1-8B Grattafiori et al. (2024); (ii) Speculative decoding APD Israel et al. (2025); (iii) Dependency decoding DAWN Luo et al. (2026); and (iv) Remask CORE Zhai et al. (2026). The detailed settings are provided in Appendix A.9.

Table 3: Benchmark results. † taken from Wang et al. (2025b). ‡ taken from Yang et al. (2025). * means reproduced results.

Model	MATH500	GSM8K	MBPP	LCB V2	LiveBench	MMLU	HellaSwag
Qwen2.5-7B†	74.0	89.9	74.9	26.9	31.1	74.2	80.2
Qwen2.5-0.5B‡	33.8	41.6	39.3	10.4	12.1	47.5	52.1
LLaDa-8B*	35.2	78.1	34.4	5.8	4.1	57.3	68.2
+DAWN*	38.3 (+3.1)	79.0 (+0.9)	33.6 (-0.8)	5.5 (-0.3)	4.8 (+0.7)	58.6 (+1.3)	68.8 (+0.6)
+APD*	36.8 (+1.6)	77.5 (-0.6)	32.6 (-1.8)	7.1 (+1.3)	6.3 (+2.2)	60.5 (+3.2)	69.6 (+1.4)
+CORE*	35.6 (+0.4)	78.7 (+0.6)	33.9 (-0.5)	4.9 (-0.9)	3.0 (-1.1)	56.5 (-0.8)	68.0 (-0.2)
+Ind-Energy	37.6 (+2.4)	79.5 (+1.4)	34.9 (+0.5)	9.2 (+3.4)	8.6 (+4.5)	62.7 (+5.4)	69.9 (+1.7)
+Inv-Energy	36.5 (+1.3)	78.7 (+0.6)	35.0 (+0.6)	7.8 (+2.0)	5.7 (+1.6)	59.4 (+2.1)	69.0 (+0.8)
+Uni-Energy	39.2 (+4.0)	80.1 (+2.0)	35.2 (+0.8)	10.5 (+4.7)	10.9 (+6.8)	66.5 (+9.2)	71.2 (+3.0)
Dream-7B*	34.4	69.7	55.4	7.5	7.1	61.8	70.6
+APD*	35.9 (+1.5)	71.4 (+1.7)	55.8 (+0.4)	9.0 (+1.5)	8.2 (+1.1)	61.3 (-0.5)	71.3 (+0.7)
+CORE*	34.6 (+0.2)	70.1 (+0.4)	56.0 (+0.6)	6.3 (-1.2)	6.5 (-0.6)	62.2 (+0.4)	69.8 (-0.8)
+Ind-Energy	35.7 (+1.3)	71.3 (+1.6)	56.7 (+1.3)	9.8 (+2.3)	9.3 (+2.2)	62.4 (+0.6)	72.6 (+2.0)
+Inv-Energy	35.2 (+0.8)	70.2 (+0.5)	56.3 (+0.9)	8.1 (+0.6)	8.8 (+1.7)	62.7 (+0.9)	71.4 (+0.8)
+Uni-Energy	36.4 (+2.0)	71.9 (+2.2)	57.2 (+1.8)	11.1 (+3.6)	9.9 (+2.8)	63.2 (+1.4)	73.7 (+3.1)

Results: Table 3 summarizes the benchmark results, with full results in Appendix A.11. Uni-E consistently improves DLLM performance, outperforming APD and DAWN. This advantage is likely from jointly modeling independence and invariance. CORE shows limited gains since the remask strategy is unstable. Across MATH500, GSM8K, LCB V2, MMLU, and HellaSwag, Uni-E outperforms both the AR model and the corresponding DLLM. In contrast, using only independent energy (Ind-E) or invariant energy (Inv-E) yields limited improvements, which is consistent with the observations in Figure 2, suggesting that modeling only dependency or invariance alone is insufficient.

Analysis: Following Israel et al. (2025), we evaluate efficiency using throughput, measured as the average number of tokens generated per second. The experiment is counducted on MATH500. As shown in Figure 4, increasing decoding parallelism substantially degrades DLLM performance. Incorporating APD or Uni-E mitigates this degradation, as they partially capture token dependencies. Notably, Uni-E consistently achieves better performance at the same throughput. We provide the hyperparameter analysis in Appendix A.10.

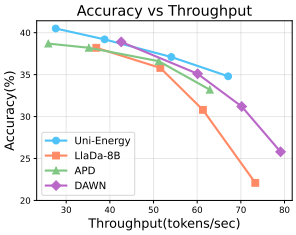

Figure 4: Efficiency Analysis

6 Conclusion

In this paper, we first analyze the limitation factor of Diffusion Language Models (DLMs): (i) model capacity, (ii) dependency, and (iii) invariance. To address these factors, we propose a unified energy (Uni-E) that combines invariant energy (Inv-E) and independent energy (Ind-E). We theoretically demonstrate that Uni-E can be computed efficiently and accurately while mitigating distribution shifts caused by dependent or non-invariant decoding. Extensive experiments on both DLMs and Diffusion Large Language Models (DLLMs) validate its effectiveness.

7 AI USE STATEMENT

In this work, we used generative AI tools only for polishing the paper writing, including fixing grammar mistakes, making expressions more easy to understand. We have not used generative AI tools for developing our research idea and experiments. The applicable and not applicable details are as follows:

Applicable AI usage for this work: (i) Grammar, spelling mistake finding and fixing; (ii) Expression and writing polishing, AI is used to paraphrase some of my sentences to make it clear and easy to understand, while exactly keeping the main content of the original sentences. (iii) Give suggestions on the paper formatting, including the position of our experimental Tables, Figures.

Not Applicable AI usage for this work: We do not use AI for our research idea developing, including: (i) AI is not used for reviewing related works; (ii) AI is not used for developing methodology and theory of this work; (iii) AI is not used for collecting experimental data; (iv) AI is not used for implementing our method codes;

8 ETHICS STATEMENT

This work does not involve ethical issues. The main focus of this work is on the research of natural language frontier models. The content of the work does not involve the release of harmful data, harmful content, or issues related to conflicts of interest.

9 REPRODUCIBILITY STATEMENT

We declare that all the work presented in this paper is reproducible. Firstly, in terms of methodology, we have provided rigorous theoretical proofs for each theorem/lemma. Secondly, in terms of the software code, we have made the relevant code of our work public via anonymous link. Furthermore, we adopt the publicly available datasets, data processing methods, and evaluation metrics. This means that the experimental and theoretical results of this work can all be reproduced.

REFERENCES

Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*, 2025.

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

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

Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, et al. Llada2.0: Scaling up diffusion language models to 100b. *arXiv preprint arXiv:2512.15745*, 2025.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45, 2024.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*, 2013.

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

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 615–621, 2018.

Yifeng Gao, Ziang Ji, Yuxuan Wang, Biqing Qi, Hanlin Xu, and Linfeng Zhang. Self speculative decoding for diffusion large language models. *arXiv preprint arXiv:2510.04147*, 2025.

Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`, 2019.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pp. 1352–1361. PMLR, 2017.

John Hammersley. *Monte carlo methods*. Springer Science & Business Media, 2013.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

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

Emiel Hoogeboom, Alexey A Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. *arXiv preprint arXiv:2110.02037*, 2021.

Daniel Israel, Guy Van den Broeck, and Aditya Grover. Accelerating diffusion llms via adaptive parallel decoding. *arXiv preprint arXiv:2506.00413*, 2025.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

Frederick James. Monte carlo theory and practice. *Reports on progress in Physics*, 43(9):1145–1189, 1980.

Hugo Lavenant and Giacomo Zanella. Error bounds and optimal schedules for masked diffusions with factorized approximations. *arXiv preprint arXiv:2510.25544*, 2025.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Ian Li, Zilei Shao, Benjie Wang, Rose Yu, Guy Van den Broeck, and Anji Liu. Breaking the factorization barrier in diffusion language models. *arXiv preprint arXiv:2603.00045*, 2026.

Tianyi Li, Mingda Chen, Bowei Guo, and Zhiqiang Shen. A survey on diffusion language models. *arXiv preprint arXiv:2508.10875*, 2025.

Anji Liu, Oliver Broadrick, Mathias Niepert, and Guy Van den Broeck. Discrete copula diffusion. *arXiv preprint arXiv:2410.01949*, 2024.

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

Lizhuo Luo, Zhuoran Shi, Jiajun Luo, Zhi Wang, Shen Ren, Wenya Wang, and Tianwei Zhang. Dawn: Dependency-aware fast inference for diffusion llms. *arXiv preprint arXiv:2602.06953*, 2026.

Mitch Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

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

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 1525–1534, 2016.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin Chiu, and Volodymyr Kuleshov. The diffusion duality. *arXiv preprint arXiv:2506.10892*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

Yinjie Wang, Ling Yang, Bowen Li, Ye Tian, Ke Shen, and Mengdi Wang. Revolutionizing reinforcement learning framework for diffusion large language models. *arXiv preprint arXiv:2509.06949*, 2025b.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Siddartha Naidu, et al. Livebench: A challenging, contamination-free llm benchmark. *arXiv preprint arXiv:2406.19314*, 4:2, 2024.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *arXiv preprint arXiv:2505.22618*, 2025.

Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon, and Arash Vahdat. Energy-based diffusion language models for text generation. *arXiv preprint arXiv:2410.21357*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Lin Yao. Remask, don't replace: Token-to-mask refinement in masked diffusion language models. *arXiv preprint arXiv:2604.18738*, 2026.

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

Qiuhua Yi, Xiangfan Chen, Chenwei Zhang, Zehai Zhou, Linan Zhu, and Xiangjie Kong. Diffusion models in text generation: a survey. *PeerJ Computer Science*, 10:e1905, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

Kevin Zhai, Sabbir Mollah, Zhenyi Wang, and Mubarak Shah. Core: Context-robust remasking for diffusion language models. *arXiv preprint arXiv:2602.04096*, 2026.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2):1–124, 2023.

Xiao Zhou, Yong Lin, Renjie Pi, Weizhong Zhang, Renzhe Xu, Peng Cui, and Tong Zhang. Model agnostic sample reweighting for out-of-distribution learning. In *International conference on machine learning*, pp. 27203–27221. PMLR, 2022.

Hao Zou, Zae Myung Kim, and Dongyeop Kang. A survey of diffusion models in natural language processing. *arXiv preprint arXiv:2305.14671*, 2023.

A APPENDIX

APPENDIX CONTENTS

A.1 Training Algorithm 15
A.2 Inference Algorithm 15
A.3 Inference Algorithm for Diffusion Large Language Models (DLLMs) 15
A.4 Proof of Invariance Divergence Estimation 16
A.5 Proof of Uni-EDLM's Effectiveness 17
A.6 Settings on Diffusion Language Model Experiments 18
A.6.1 Metrics Details 18
A.6.2 Baseline Settings 19
A.7 Comparison with Speculative Decodings 19
A.8 Comparison with Remask Strategies 19
A.9 Settings of Baselines on Diffusion Large Language Models (DLLMs) Experiments 20
A.10 Hyper-parameter Analysis on Diffusion Large Language Model 20
A.10.1 Candidate Size Study 20
A.10.2 Energy-based Decoding Ratio Study 20
A.11 Benchmark Performance of Diffusion Large Language Models (DLLMs) 21
A.12 Case Study 21

A.1 Training Algorithm

We provide the NCE training algorithm for our Uni-EDLM in Algorithm 1.

Algorithm 1 Training Uni-EDLM with Noise Contrastive Estimation (NCE)

Require: Training dataset $\mathcal{D}$, AR model p_{AR}, diffusion model p_θ, learning rate η

Freeze parameters of p_{AR}
while not converged **do**
  Sample clean data $\mathbf{x}_0 \sim \mathcal{D}$ and diffusion timestep $t \sim \mathcal{U}(0, 1)$
  Sample $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_0)$
  Sample s with $s < t$ and $\mathbf{x}_s \sim q(\mathbf{x}_s|\mathbf{x}_t, \mathbf{x}_0)$
  Compute energy:

$$E_\phi(\mathbf{x}_0, \mathbf{x}_s, \mathbf{x}_t) = \log \frac{p_\theta(\mathbf{x}_s|\mathbf{x}_t)}{p_{\text{AR}}(\mathbf{x}_s|\mathbf{x}_t)} + \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_t)}{p_{\text{AR}}(\mathbf{x}_0|\mathbf{x}_s)}$$

  Sample $\hat{\mathbf{x}}_0 \sim p_\theta(\mathbf{x}_0|\mathbf{x}_t)$
  Sample $\hat{\mathbf{x}}_s \sim q(\mathbf{x}_s|\mathbf{x}_t, \hat{\mathbf{x}}_0)$
  Compute energy:

$$E_\phi(\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_s, \mathbf{x}_t) = \log \frac{p_\theta(\hat{\mathbf{x}}_s|\mathbf{x}_t)}{p_{\text{AR}}(\hat{\mathbf{x}}_s|\mathbf{x}_t)} + \log \frac{p_\theta(\hat{\mathbf{x}}_0|\mathbf{x}_t)}{p_{\text{AR}}(\hat{\mathbf{x}}_0|\mathbf{x}_s)}$$

  Compute loss:

$$\mathcal{L}_{\text{NCE}} = \log\left(1 + \exp(E_\phi(\mathbf{x}_0, \mathbf{x}_s, \mathbf{x}_t))\right) + \log\left(1 + \exp(-E_\phi(\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_s, \mathbf{x}_t))\right)$$

  Update $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{NCE}}$
end while
return trained parameters θ

A.2 Inference Algorithm

We provide the inference of Uni-EDLM in Algorithm 2.

Algorithm 2 Uni-EDLM Inference with Importance Sampling

Require: Full masked sequence $\mathbf{x}_{\mathcal{T}_1}$, diffusion timesteps $\mathcal{T}_1 > \mathcal{T}_2 > ... > \mathcal{T}_T$, candidate size k, window size w, proxy AR model p_{AR}, diffusion model p_θ

for $t = \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_T$ **do**
  if $t \in (1 - w, 1]$ **then**
    Sample $\{\mathbf{x}_0^1, .., \mathbf{x}_0^k\} \sim p_\theta(\mathbf{x}_0|\mathbf{x}_t)$ and sample timestep $s < t$
    Sample $\{\mathbf{x}_s^1, .., \mathbf{x}_s^k\} \sim q(\mathbf{x}_s|\mathbf{x}_t, \mathbf{x}_0^{(j)})$ with $j \in \{1, 2, .., k\}$
    Compute energy $E^{(j)} = E_\phi(\mathbf{x}_0^{(j)}, \mathbf{x}_s^{(j)}, \mathbf{x}_t)$ for each $(\mathbf{x}_0^{(j)}, \mathbf{x}_s^{(j)})$ pair
    $j^* = \arg\min_j E^{(j)}$
    $\tilde{\mathbf{x}}_0 = \mathbf{x}_0^{(j^*)}$
  else
    Sample $\tilde{\mathbf{x}}_0 \sim p_\theta(\mathbf{x}_0|\mathbf{x}_t)$
  end if
  Sample $\mathbf{x}_{t-1} \sim q(\mathbf{x}_{t-1}|\mathbf{x}_t, \tilde{\mathbf{x}}_0)$
end for
return $\mathbf{x}_0$

A.3 Inference Algorithm for Diffusion Large Language Models (DLLMs)

We provide the inference algorithm when integrating with Diffusion Large Language Models (DLLMs) in Algorithm 3.

Algorithm 3 Energy-Guided Inference for Diffusion Large Language Models (DLLMs)

Require: Full masked sequence $\mathbf{x}_1$ with max length L, candidate size k, number of unmask positions m for each step, window size w, DLLM model p_θ, AR model p_{AR}

for $t = 1, 1 - \frac{m}{L}, 1 - \frac{2m}{L}, ..., 0$ **do**
  Obtain token logits from DLLM: $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$
  if $t \in (1 - w, 1]$ **then**
    for $j = 1$ to k **do**
      Sample clean candidate: $\mathbf{x}_0^{(j)} \sim p_\theta(\mathbf{x}_0|\mathbf{x}_t)$
      Compute token-level logits $\{\ell_i^{(j)}\}_{i=1}^L$ for $\mathbf{x}_0^{(j)}$
      Select index set $\mathcal{I}^{(j)}$ corresponding to top-m logits
      Generate $\mathbf{x}_s^{(j)}$:

$$\mathbf{x}_{s,i}^{(j)} = \begin{cases} \mathbf{x}_{0,i}^{(j)}, & i \in \mathcal{I}^{(j)} \\ \text{[MASK]}, & \text{otherwise} \end{cases}$$

      Compute energy for each $(\mathbf{x}_0^{(j)}, \mathbf{x}_s^{(j)})$ pair

$$E^{(j)} = \log \frac{p_\theta(\mathbf{x}_s^{(j)}|\mathbf{x}_t)}{p_{\text{AR}}(\mathbf{x}_s^{(j)}|\mathbf{x}_t)} + \log \frac{p_\theta(\mathbf{x}_0^{(j)}|\mathbf{x}_t)}{p_{\text{AR}}(\mathbf{x}_0^{(j)}|\mathbf{x}_s^{(j)})}$$

    end for
    $j^* = \arg\min_j E^{(j)}$
    $\mathbf{x}_t = \mathbf{x}_s^{(j^*)}$
  else
    Unmask m tokens with highest logits within $\mathbf{x}_t$ from $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$
  end if
end for
return $\mathbf{x}_0$

A.4 Proof of Invariance Divergence Estimation

Here we give a formal proof to the Lemma 1 as follows:

Proof. According to the invariance defination, we have:

$$\pi(\mathbf{x}_0^i|\mathbf{x}_{t+1}, \mathbf{x}_0^{\mathcal{V}_i^-}, \mathbf{x}_0^{Z<t} - \mathbf{x}_0^{\mathcal{V}_i^-}) = \pi(\mathbf{x}_0^i|\mathbf{x}_{t+1}, \mathbf{x}_0^{\mathcal{V}_i^-}) \tag{17}$$

Then we have:

$$\frac{\pi(\mathbf{x}_0^i, \mathbf{x}_0^{Z<t} - \mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_{t+1}, \mathbf{x}_0^{\mathcal{V}_i^-})}{\pi(\mathbf{x}_0^{Z<t} - \mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_{t+1}, \mathbf{x}_0^{\mathcal{V}_i^-})} = \pi(\mathbf{x}_0^i|\mathbf{x}_{t+1}, \mathbf{x}_0^{\mathcal{V}_i^-}) \tag{18}$$

Therefore we have:

$$\pi(\mathbf{x}_0^{Z<t} - \mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_0^i, \mathbf{x}_{t+1}, \mathbf{x}_0^{\mathcal{V}_i^-}) = \pi(\mathbf{x}_0^{Z<t} - \mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_{t+1}, \mathbf{x}_0^{\mathcal{V}_i^-}) \tag{19}$$

The intuitive understanding is that the i-th token has no influence on the rest tokens beyond invariant ones. Given the invriance part defination: $\pi(\mathbf{x}_0^i|\mathbf{x}_0 - \mathbf{x}_0^i) = \pi(\mathbf{x}_0^i|\mathbf{x}_0^{\mathcal{V}_i}) = \pi(\mathbf{x}_0^i|\mathbf{x}_{t+1}, \mathbf{x}_0^{\mathcal{V}_i^-})$, we know that $\pi(\mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_0^i, \mathbf{x}_{t+1}, \mathbf{x}_0^{Z<t} - \mathbf{x}_0^{\mathcal{V}_i^-}) = \pi(\mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_0^i, \mathbf{x}_{t+1})$, this means the invariant tokens $\mathbf{x}_0^{\mathcal{V}_i^-}$ of $\mathbf{x}_0^i$ is also invariant to $\mathbf{x}_0^{Z<t} - \mathbf{x}_0^{\mathcal{V}_i^-}$. This indicate a key insight that **the decoding of $\mathbf{x}_0^i$ will only influence the invariant tokens and this influence will not passed to the non-invariant parts**.

Assuming in time step t, we decode a token $\mathbf{x}_0^i$, the rest part of $\mathbf{x}_0$ distribution before and after decoding is shown as the first two row in Table 4, where T_i^j denotes the j-th invariant token and T_n^k denote the k-th non-invariant token. The decoding has no influence on the non-invariant tokens while the invariant one will have distribution shift from p_i to q_i. We can easily conclude that:

	T_i^1	T_i^2	...	T_i^x	T_n^1	T_n^2	...	T_n^y
Before Decoding	p_1	p_2	...	p_x	g_1	g_2	...	g_y
After Decoding	q_1	q_2	...	q_x	g_1	g_2	...	g_y
After Sampling	q_1	q_2	...	p_x	g_1	g_2	...	g_y

Table 4: The distribution shift before and after decoding token $\mathbf{x}_0^i$

$$log\frac{\pi(\mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_{t+1})}{\pi(\mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_i,\mathbf{x}_{t+1})} = log\frac{\pi(\mathbf{x}_0^{Z_{<t}}|\mathbf{x}_{t+1})}{\pi(\mathbf{x}_0^{Z_{<t}}|\mathbf{x}_i,\mathbf{x}_{t+1})} = \sum_i^x KL(p_i||q_i) + \sum_i^y KL(g_i||g_i) = \sum_i^x KL(p_i||q_i) \tag{20}$$

Therefore if we sample all the $Z_{<t}$ we will have the full esitmation of invariant part distribution. Next we will prove the estimation when sampling. Here we sample r tokens from $Z_{<t}$ and get index Z_s, and after sampling the distribution is shown as the third line in Table 4. Suppose we sample the first m invariant and the rest n tokens are non-invariant($m+n=r$), the first m distribution will be influenced by $\mathbf{x}_0^i$ when calculating estimation, which is $q_i(i \le m)$ and the remains are the same as the non-decoding $\mathbf{x}_0^i$(The "Before Decoding" line in Table 4). The sampling estimation is:

$$log\frac{\pi(\mathbf{x}_0^{Z_s}|\mathbf{x}_{t+1})}{\pi(\mathbf{x}_0^{Z_s}|\mathbf{x}_i,\mathbf{x}_{t+1})} = \sum_{i=1}^m KL(p_i||q_i) + \sum_{i=m+1}^x KL(p_i||p_i) + \sum_{i=1}^y KL(g_i||g_i) = \sum_i^m KL(p_i||q_i) \tag{21}$$

Therefore, we have:

$$log\frac{\pi(\mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_{t+1})}{\pi(\mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_i,\mathbf{x}_{t+1})} - log\frac{\pi(\mathbf{x}_0^{Z_s}|\mathbf{x}_{t+1})}{\pi(\mathbf{x}_0^{Z_s}|\mathbf{x}_i,\mathbf{x}_{t+1})} = \sum_{i=m+1}^x KL(p_i||q_i) \tag{22}$$

Considering the Equation 22 just describe one specific situation that we select the first m-th invariant tokens, when we traverse all the possible sampling of Z_s and calculate the average, it can be deduced easily that the possibility of picking each token is $\frac{r}{x+y}$, leading to the average residual term as:

$$\mathbb{E}_{Z_s}\left\{log\frac{\pi(\mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_{t+1})}{\pi(\mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_i,\mathbf{x}_{t+1})} - log\frac{\pi(\mathbf{x}_0^{Z_s}|\mathbf{x}_{t+1})}{\pi(\mathbf{x}_0^{Z_s}|\mathbf{x}_i,\mathbf{x}_{t+1})}\right\} = (1-\frac{r}{x+y})\sum_{i=1}^x KL(p_i||q_i) \tag{23}$$

Therefore we have:

$$\mathbb{E}_{Z_s}\left\{log\frac{\pi(\mathbf{x}_0^{Z_s}|\mathbf{x}_{t+1})}{\pi(\mathbf{x}_0^{Z_s}|\mathbf{x}_i,\mathbf{x}_{t+1})}\right\} = (\frac{r}{x+y})\sum_{i=1}^x KL(p_i||q_i) = (\frac{r}{x+y})log\frac{\pi(\mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_{t+1})}{\pi(\mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_i,\mathbf{x}_{t+1})} \tag{24}$$

So if we use $p_{AR}(.)$ as the proxy, we finally get:

$$\mathbb{E}_{Z_s}\left\{log\frac{p_{AR}(\mathbf{x}_0^{Z_s}|\mathbf{x}_{t+1})}{p_{AR}(\mathbf{x}_0^{Z_s}|\mathbf{x}_i,\mathbf{x}_{t+1})}\right\} = (\frac{r}{x+y})log\frac{\pi(\mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_{t+1})}{\pi(\mathbf{x}_0^{\mathcal{V}_i^-}|\mathbf{x}_i,\mathbf{x}_{t+1})} + \gamma \tag{25}$$

Therefore, we prove Lemma 1.

□

A.5 Proof of Uni-EDLM's Effectiveness

Here we give a formal proof to the Lemma 2 as follows:

Proof. Considering all the possible decoding s and t pairs on clean data $\mathbf{x}_0$, the average KL Divergence of diffusion language model is defined as:

$$KL(\pi||p_\theta) = \mathbb{E}_{\mathbf{x}_t}\left\{\sum_{s(s<t)}\Omega(\mathbf{x}_s|\mathbf{x}_t)log\frac{\pi(\mathbf{x}_s|\mathbf{x}_t)}{p_\theta(\mathbf{x}_s|\mathbf{x}_t)}\right\}, \tag{26}$$

where the $\Omega(\mathbf{x}_s|\mathbf{x}_t)$ is the ground truth decoding planner distribution describing the true possibility of decoding position in $\mathbf{x}_s$ based on $\mathbf{x}_t$, the average KL Divergence for Uni-EDLM is:

$$KL(\pi||p_{\phi,\theta}) = \mathbb{E}_{\mathbf{x}_t}\left\{\sum_{s(s<t)} \Omega(\mathbf{x}_s|\mathbf{x}_t) log \frac{\pi(\mathbf{x}_s|\mathbf{x}_t)}{p_{\phi,\theta}(\mathbf{x}_s|\mathbf{x}_t)}\right\}, \tag{27}$$

Now giving specific t, we consider the situation that the decoding positions do not fulfill the independence or invariance. Supposing such decodings in descendent violation degree are Z_1, Z_2, ..., Z_n(Z_n is the most "independent & invariant" one and Z_1 is the most "non-independent&non-invariant" one), we have:

$$\Omega(\mathbf{x}_s^{Z_1}|\mathbf{x}_t) < \Omega(\mathbf{x}_s^{Z_2}|\mathbf{x}_t) < ... < \Omega(\mathbf{x}_s^{Z_n}|\mathbf{x}_t), \tag{28}$$

since $\Omega(.)$ represents the real possiblity of decoding order. We also assume that the diffusion language model unfortunately can not distinguish them, where:

$$p_\theta(\mathbf{x}_s^{Z_1}|\mathbf{x}_t) \approx p_\theta(\mathbf{x}_s^{Z_2}|\mathbf{x}_t) \approx ... \approx p_\theta(\mathbf{x}_s^{Z_n}|\mathbf{x}_t), \tag{29}$$

According to the Uni-Energy, it will allocate lower energy to the independent and invariant decodings while allocate higher energy to the non-independent or non-invariant decodings, thus:

$$E_{uni}(\mathbf{x}_0, \mathbf{x}_s^{Z_1}, \mathbf{x}_t) > E_{uni}(\mathbf{x}_0, \mathbf{x}_s^{Z_2}, \mathbf{x}_t) > ... > E_{uni}(\mathbf{x}_0, \mathbf{x}_s^{Z_n}, \mathbf{x}_t), \tag{30}$$

Thus we have:

$$\begin{aligned} &\frac{exp(-E_{uni}(\mathbf{x}_0, \mathbf{x}_s^{Z_1}, \mathbf{x}_t))}{F(\mathbf{x}_t)} p_\theta(\mathbf{x}_s^{Z_1}|\mathbf{x}_t) < \frac{exp(-E_{uni}(\mathbf{x}_0, \mathbf{x}_s^{Z_2}, \mathbf{x}_t))}{F(\mathbf{x}_t)} p_\theta(\mathbf{x}_s^{Z_2}|\mathbf{x}_t) \\ &< ... < \frac{exp(-E_{uni}(\mathbf{x}_0, \mathbf{x}_s^{Z_n}, \mathbf{x}_t))}{F(\mathbf{x}_t)} p_\theta(\mathbf{x}_s^{Z_n}|\mathbf{x}_t), \end{aligned} \tag{31}$$

We also know that all the possibility are normalized, which means:

$$\sum_{s,i} \frac{exp(-E_{uni}(\mathbf{x}_0, \mathbf{x}_s^{Z_i}, \mathbf{x}_t))}{F(\mathbf{x}_t)} p_\theta(\mathbf{x}_s^{Z_i}|\mathbf{x}_t) = 1, \tag{32}$$

and

$$\sum_{s,i} p_\theta(\mathbf{x}_s^{Z_i}|\mathbf{x}_t) = 1, \tag{33}$$

Therefore we have:

$$\sum_{s,i} \Omega(\mathbf{x}_s^{Z_i}|\mathbf{x}_t) log \frac{exp(-E_{uni}(\mathbf{x}_0, \mathbf{x}_s^{Z_i}, \mathbf{x}_t))}{F(\mathbf{x}_t)} p_\theta(\mathbf{x}_s^{Z_i}|\mathbf{x}_t) > \sum_{s,i} \Omega(\mathbf{x}_s^{Z_i}|\mathbf{x}_t) log p_\theta(\mathbf{x}_s^{Z_i}|\mathbf{x}_t) \tag{34}$$

Therefore, we have:

$$KL(\pi||p_{\phi,\theta}) < KL(\pi||p_\theta). \tag{35}$$

□

A.6 Settings on Diffusion Language Model Experiments

A.6.1 Metrics Details

Perplexity (PPL) is used to evaluate the quality of a language model by measuring how well it predicts a sequence of tokens. Given a sequence $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_L)$ of length L, PPL is calculated as: $PPL(\mathbf{x}) = exp(-\frac{1}{L}\sum_{i=1}^{L} log p_\theta(\mathbf{x}_i))$.

Generative Perplexity (Gen PPL) evaluates the quality of text generated by a model using a reference language model. Instead of conditioning on ground-truth prefixes, it measures how likely a generated sequence is. Given a generated sequence $\hat{\mathbf{x}} = (\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, ..., \hat{\mathbf{x}}_L)$, Gen PPL is calculated as: $GenPPL(\hat{\mathbf{x}}) = exp(-\frac{1}{L}\sum_{i=1}^{L} log p_{eval}(\hat{\mathbf{x}}_i|\hat{\mathbf{x}}_{<i}))$, where p_{eval} is the reference model and usually it's a Large Language Model.

A.6.2 BASELINE SETTINGS

As for auto-regressive Transformer (AR) Vaswani et al. (2017), SEDD Lou et al. (2023), MDLM Sahoo et al. (2024) and EDLM Xu et al. (2024), we use the official model and dataset settings. Therefore, we directly report Perplexity (PPL) from Xu et al. (2024). For Generative Perplexity (Gen PPL), we reproduce results with 128 timesteps, while the remaining results are taken from Xu et al. (2024).

Duo Sahoo et al. (2025) uses the same dataset settings as ours, and we report its official PPL results. As for Generative Perplexity (Gen PPL), we vary the decoding steps (128, 256, 512, and 1024) and use the official checkpoint, which is trained on the same dataset (OpenWebText).

For Block Diffusion Arriola et al. (2025), we set the block size as 4, following the default setting in the original paper, which generally achieves better performance than larger block sizes (e.g., 8 or 16). For Gen PPL, we use the published checkpoint trained on OpenWebText, and evaluate with decoding timesteps of 128, 256, 512, and 1024 while keeping the block size fixed at 4.

For DCD Liu et al. (2024), we use GPT-2-small Radford et al. (2019) as the AR proxy (copula) model and MDLM Sahoo et al. (2024) as the diffusion backbone.

For ReMDM Wang et al. (2025a), we use MDLM as backbone and set the ReMDM-cap parameter as 0.05 and ReMDM-rescale as 0.5 because it generally leads to better performance in its official report.

A.7 COMPARISON WITH SPECULATIVE DECODINGS

When integrating with Diffusion Large Language Models (DLLMs), our Uni-E is similar to speculative decoding methods Israel et al. (2025); Gao et al. (2025), which also use a proxy model to calculate scores or logits to accelerate decoding or improve generation quality. However, the key difference is that *existing speculative decoding methods for DLLMs are generally heuristic and integrate the proxy model without guarantees for invariant decoding*.

Taking APD Israel et al. (2025) as an example, it combines the AR proxy model with the DLLM in a linear manner, where the model produces logits as: $model_logits = Softmax(R * p_\theta(\mathbf{x}_t|\mathbf{x}_{<t}) + (1-R) * p_{AR}(\mathbf{x}_t|\mathbf{x}_{<t}))$, where $p_\theta(.)$ and $p_{AR}(.)$ are the DLLM and AR proxy model, respectively. This design can model dependencies to some extend, but cannot capture invariance (i.e., future decoding effects).

In contrast, is explicitly designed to model both dependency and invariance. Moreover, the formulation of our energy function has a clear theoretical foundation, showing that it can bridge the gap between the ground-truth distribution and the diffusion model distribution (see Eq. 7 and Eq. 11). A comparison between Uni-E and speculative decoding is summarized in Table 5.

Table 5: Comparison between Speculative Decoding and Uni-E.

Method	AR Proxy	Dependency	Invariance	Theoretical Guarantee
Speculative Decoding	Yes	Yes	No	No
Uni-E	Yes	Yes	Yes	Yes

A.8 COMPARISON WITH REMASK STRATEGIES

To address token relationship issues, several methods Wang et al. (2025a); Yao (2026); Wang et al. (2025a) have proposed using a remask strategy to correct decoding errors. The basic idea behind these methods is to remask previously unmasked tokens during future decoding steps, allowing errors caused by dependency or invariance issues to be corrected. However, these approaches are generally heuristic and lack a theoretical foundation to guarantee dependency or invariance modeling. The selection of tokens to be corrected is also highly dependent on heuristic strategies or manually designed hyperparameters, without theoretical guarantee. For example, ReMDM Wang et al. (2025a) uses a hyper-parameter and a scale-score to remask tokens at each step, which view the tokens equally important. Core Wang et al. (2025a) remask based on neighbor stability test, which only consider the nearby tokens. None of these methods have theoretical guarantee that the remasking can truly fix the dependency and invariance error. In addition, these methods usually

require more decoding timesteps to achieve good performance, since remasking and re-predicting tokens introduces additional decoding steps. A detailed comparison between these methods and our Uni-E is summarized in Table 6.

Table 6: Comparison between Remask Strategy and Uni-E.

Method	AR Proxy	Dependency	Invariance	Theoretical Guarantee(Stable)	Efficiency
Remask	No	Yes	Yes	No	No
Uni-E	Yes	Yes	Yes	Yes	Yes

A.9 Settings of Baselines on Diffusion Large Language Models (DLLMs) Experiments

For all the AR models (Qwen2.5-0.5B, Qwen2.5-7B Yang et al. (2025), Llama3.1-8B Grattafiori et al. (2024)) we use the official model parameters and evaluate them within the TraceRL framework Wang et al. (2025b). We adopt a static decoding strategy and, similar to Uni-E, decode 4 tokens at each step.

For APD Israel et al. (2025), we follow the original paper and use Qwen2.5-0.5B as the proxy model, decoding 4 tokens per step.

For DAWN Luo et al. (2026), we set the confidence thresholds for graph construction as follows: the high-confidence threshold is 0.9, the low-confidence threshold is 0.7, and the anchor-based threshold is 0.7.

For CORE Zhai et al. (2026), we use their official implementation and adapt the decoding number as 4 at each step (including the decoding for the remasked tokens).

A.10 Hyper-parameter Analysis on Diffusion Large Language Model

A.10.1 Candidate Size Study

We study the impact of the candidate size k on model performance. In this experiment, we use MATH500 as the evaluation benchmark and adopt LLaDa-8B as the base diffusion large language model (DLLM). The results are summarized in Table 7. We observe that varying the candidate size has only a marginal effect on performance. However, increasing k leads to a noticeable reduction in decoding speed.

Table 7: Influence of candidate size k.

	Accuracy	Throughoutput
k=2	38.6	40.1
k=3	39.2	38.8
k=4	40.1	35.5
k=5	39.8	31.7

A.10.2 Energy-based Decoding Ratio Study

By default, we apply energy-based sampling to the first 20% of the inference steps. To study the effect of this ratio, we vary the energy sampling ratio w and evaluate performance when applying energy sampling to the first 10%, 20%, 40%, and 80% of the steps. We conduct the experiments on MATH500 using LLaDa-8B as the base model. The results are presented in Table 8. We find that increasing the sampling ratio generally improves performance. However, this comes at the cost of significantly reduced decoding efficiency. Therefore, we select $w = 0.2$ as a balanced setting between performance and speed.

Table 8: Influence of energy sampling ratio w.

	Accuracy	Throughoutput
w=0.1	37.0	43.9
w=0.2	39.2	38.8
w=0.4	41.5	27.4
w=0.8	42.3	11.8

A.11 Benchmark Performance of Diffusion Large Language Models (DLLMs)

We provide the full results for DLLMs in Table 9.

Table 9: Several benchmark results. Improvements of applying Uni-Energy to their backbone models are shown in green. † taken from Wang et al. (2025b). ‡ taken from Yang et al. (2025). * denotes reproduced results.

Model	MATH500	GSM8K	MBPP	LCB V2	LiveBench	MMLU	HellaSwag
Llama3.1-8B†	51.9	84.5	42.4	20.0	19.7	66.6	76.7
Qwen2.5-7B†	74.0	89.9	74.9	26.9	31.1	74.2	80.2
Qwen2.5-0.5B‡	33.8	41.6	39.3	10.4	12.1	47.5	52.1
LLaDa-8B*	35.2	78.1	34.4	5.8	4.1	57.3	68.2
+DAWN*	38.3 (+3.1)	79.0 (+0.9)	33.6 (-0.8)	5.5 (-0.3)	4.8 (+0.7)	58.6 (+1.3)	68.8 (+0.6)
+APD*	36.8 (+1.6)	77.5 (-0.6)	32.6 (-1.8)	7.1 (+1.3)	6.3 (+2.2)	60.5 (+3.2)	69.6 (+1.4)
+CORE*	35.6 (+0.4)	78.7 (+0.6)	33.9 (-0.5)	4.9 (-0.9)	3.0 (-1.1)	56.5 (-0.8)	68.0 (-0.2)
+Ind-Energy	37.6 (+2.4)	79.5 (+1.4)	34.9 (+0.5)	9.2 (+3.4)	8.6 (+4.5)	62.7 (+5.4)	69.9 (+1.7)
+Inv-Energy	36.5 (+1.3)	78.7 (+0.6)	35.0 (+0.6)	7.8 (+2.0)	5.7 (+1.6)	59.4 (+2.1)	69.0 (+0.8)
+Uni-Energy	39.2 (+4.0)	80.1 (+2.0)	35.2 (+0.8)	10.5 (+4.7)	10.9 (+6.8)	66.5 (+9.2)	71.2 (+3.0)
Dream-7B*	34.4	69.7	55.4	7.5	7.1	61.8	70.6
+DAWN*	35.8 (+1.4)	70.5 (+0.8)	55.6 (+0.2)	7.1 (-0.4)	7.3 (+0.2)	61.1 (-0.7)	71.9 (+1.3)
+APD*	35.9 (+1.5)	71.4 (+1.7)	55.8 (+0.4)	9.0 (+1.5)	8.2 (+1.1)	61.3 (-0.5)	71.3 (+0.7)
+CORE*	34.6 (+0.2)	70.1 (+0.4)	56.0 (+0.6)	6.3 (-1.2)	6.5 (-0.6)	62.2 (+0.4)	69.8 (-0.8)
+Ind-Energy	35.7 (+1.3)	71.3 (+1.6)	56.7 (+1.3)	9.8 (+2.3)	9.3 (+2.2)	62.4 (+0.6)	72.6 (+2.0)
+Inv-Energy	35.2 (+0.8)	70.2 (+0.5)	56.3 (+0.9)	8.1 (+0.6)	8.8 (+1.7)	62.7 (+0.9)	71.4 (+0.8)
+Uni-Energy	36.4 (+2.0)	71.9 (+2.2)	57.2 (+1.8)	11.1 (+3.6)	9.9 (+2.8)	63.2 (+1.4)	73.7 (+3.1)

A.12 Case Study

To visualize the influence of invariance and independency to text generation, we conduct a case study. We use Uni-EDLM-AR and MDLM for this study. We find that Uni-E generally capture the invariance and the independency correctly and we provide two cases to demonstrate. Besides, we also provides two cases where DLM fails to model the invariance and the independency thus leads to strange generation.

Invariance Case: The invariant decoding case of our Uni-E is shown in Figure 5. In this case, the "sad" can not be fully decided without considering the later context: People "don't have opinions" and city is "not going to pay", thus the decision can not be carried out, then we can deduce the "sad". Uni-E can totally capture the invariance of these tokens and generate the reasonable text.

Independency Case: The independent decoding case of our Uni-E is shown in Figure 6. In this case, although the "not allow" and the "comply with" are adjacent, these tokens are non-independent (If "allow it", then it will "break against" the rule; If "not allow it", then it will "comply with the rule"). Uni-E can capture the dependency between these tokens and correctly model the relationship that "not allow it" leads to "comply with".

Non-invariance Case: We give one case where MDLM fails to capture the invariance. The case is shown in Figure 7. We observe that the text generated by MDLM is weird since there is conflict between "delayed the meeting" and "had a conversation", and we check the decoding order and find that the "delayed the meeting" are decoded before the "had a conversation when ... met with ..." are decoded, which breaks the invariance that the "delayed the meeting" can not be decided before considering the later context. This case shows the limitation of DLM in capturing the invariance.

Non-independent Case: We give one case where MDLM fails to capture the dependency. The case is shown in Figure 8. The generated text is unreasonable because "keep the wealth" conflicts with "increase the money lose" and also conflicts with the word "fails". We check the decoding order which shows that the "keep", the "increase" and the "lose" are decoded simultaneously but they are non-independent with each other. Although the model succeeds modeling the invariance between the "fails" and the "increase the money lose", the text is still weird since the "fail" conflicts with "keep the wealth". This case shows the limitation of DLM in capturing the dependency.

Invariance Case for Uni-E Decoding

Case snippet:

“I am [4] sad about it. The city is just [2] not going to pay for it,” she said. “I don’t think all the time, the time is up’’ and, “when you can’t make the decision, you do it. I wish I could make the decision that I made, that I’m very proud of. It’s a decision that’s my decision [3] and not my budget.” And yet, it’s not happening. “I know we can’t carry that out,” Bouton said. “It’s not happening. A lot of the council members, people don’t [1] have an opinion…”

Decoding order:

"dont have an opinion" -> "not going to pay" -> "cant carry out" -> "sad"

Figure 5: The invariant decoding case of Uni-E.

Dependency Case for Uni-E Decoding

Case snippet:

“... However, Justice Kirby said, the Government will not allow it, and will comply with the ruling that Government’s request. If that is agreed with and the Government would support it, we will not proceed and will go forward with that application and without regard to any costs or difficulties," said Kirby. ...”

Decoding order:

"not allow" -> "will comply with"

Figure 6: The independent decoding case of Uni-E.

Non-invariance Case for DLM Decoding

Case snippet:

“The Prime Minister understands that a meeting with Mr Oakes had taken place between the president, the Secretary, and the Minister in the Oval Office. The President and Mr Oakes delayed the meeting on Monday, and had a conversation when Mr Oakes met with the Prime Minister at the Oval Office…”

Non-invariant order:

"delayed the meeting" -> "had a conversation"

Figure 7: The non-invariant decoding case of MDLM.

Non-independency Case for DLM Decoding

Case snippet:

“If the person fails on one point, they will keep the wealth but also increase the money lose at the same time, and this is why the situation becomes confusing for everyone involved.”

Non-independent order:

"keep, increase, lose" -> "fails"

Figure 8: The non-independent decoding case of MDLM.