

On the Role of Discreteness in Diffusion LLMs

Ziqi Jin^{1,2}, Bin Wang^{*1}, Xiang Lin¹, Lidong Bing¹, Aixin Sun^{*2}

¹Miromind AI, ²Nanyang Technological University, Singapore

Abstract

Diffusion models offer appealing properties for language generation, such as parallel decoding and iterative refinement, but the discrete and highly structured nature of text challenges the direct application of diffusion principles. In this paper, we revisit diffusion language modeling from the view of diffusion process and language modeling, and outline five properties that separate diffusion mechanics from language-specific requirements. We first categorize existing approaches into *continuous diffusion in embedding space* and *discrete diffusion over tokens*. We then show that each satisfies only part of the five essential properties and therefore reflects a structural trade-off. Through analyses of recent large diffusion language models, we identify two central issues: (i) uniform corruption does not respect how information is distributed across positions, and (ii) token-wise marginal training cannot capture multi-token dependencies during parallel decoding. These observations motivate diffusion processes that align more closely with the structure of text, and encourage future work toward more coherent diffusion language models. ¹

¹*Correspond: bin.wang@miromind.ai, axsun@ntu.edu.sg

1 Introduction

Language modeling is to learn a probability distribution over word sequences, $P(x_{1:n})$. Autoregressive (AR) models, which predict tokens from left to right, is currently trending. Recent work explores diffusion language models (DLMs) as an alternative modeling method. With diffusion process, DLMs generate text by reversing a noise-adding process. This raises a natural question: *can diffusion be beneficial in language modeling?*

Diffusion offers potential advantages that AR models do not naturally support, illustrated in Figure 1. AR models generate strictly from left to right and add one token per forward pass. In AR decoding, the model naturally extends a prefix instead of inserting, deleting, or rewriting earlier sequences. Diffusion, by updating many positions at once, can support *flexible editing* such as insertion, deletion, and span rewriting more directly (Kim et al., 2025). AR decoding also ties computation directly to output length, since each new token needs one forward pass, whereas diffusion can revise multiple tokens together and adjust the number of refinement steps so that harder cases use more steps and easier ones use fewer. Finally in the training stage, AR sees

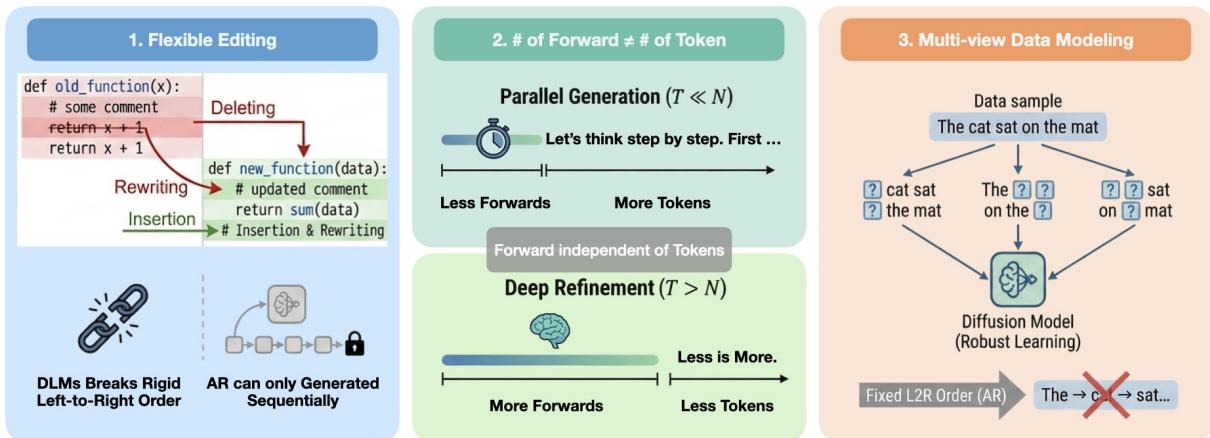


Figure 1: Core Advantages of Diffusion Language Models (DLMs) over Autoregressive (AR) Models.

each token once in a fixed left-to-right order, which makes the task easy to fit but leads to faster overfitting in low-data settings; diffusion training instead shows each example at different noise levels and patterns due to randomness, creating multiple augmented variations of the same sequence, which improves data efficiency in the long run where data is limited (Ni et al., 2025a,b). We discuss these advantages in more detail in Section 3.

While the diffusion process can be applied naturally to visuals (Ho et al., 2020), applying diffusion to language introduces a core difficulty: text is discrete, but standard diffusion process assumes continuous data that can be corrupted with Gaussian noise smoothly. Because of this mismatch, existing DLMs only follow diffusion principles partially. Specifically, *continuous* DLMs add Gaussian noise in embedding space, but the final step must map continuous vectors back to discrete tokens. This mapping is discontinuous and breaks the diffusion interpretation (Li et al., 2022; Strudel et al., 2022; Dieleman et al., 2022). *Discrete* DLMs avoid this issue by operating directly on tokens, but they replace continuous noise with masking (Austin et al., 2021; He et al., 2023). Masking lacks the smooth, time-indexed corruption that diffusion relies on.

Most recent large-scale systems adopt masked discrete diffusion. While effective in practice, these models still leave open a basic question: *what properties should an ideal diffusion language model have?*

In this paper, we revisit diffusion and language separately. We identify five essential properties that a diffusion-based language model should satisfy in Section 4, separating general diffusion requirements from constraints that arise from discrete text. Through this lens, we show that both continuous and discrete DLMs satisfy only a subset of these requirements, and each involves trade-offs that lead to structural challenges for text diffusion. Our main contributions can be summarized as follows:

- We propose a framework that clarifies what diffusion requires and what language imposes, and use it to reorganize existing diffusion language models into continuous and discrete families with a unified analysis of their assumptions and limitations.
- We provide theoretical and empirical evidence of structural mismatches between diffusion and text, including position-dependent difficulty under uniform corruption and the gap between

marginal training and joint coherence.

- We outline several potential research directions related to these challenges, with the goal of informing future work on more complete and structurally aligned diffusion models for language.

2 Preliminary

Diffusion models (Ho et al., 2020) define a generative process through two coupled components: a *forward corruption* that gradually destroys information, and a *reverse model* that learns to recover data by iterative denoising. In the original continuous setting, the forward process injects small Gaussian noise so that a clean sample x_0 becomes increasingly noisy, and after T steps the state approaches a simple prior (typically a standard Gaussian).

A diffusion language model (DLM) applies the diffusion idea to text sequences $x_{1:n}$ over a vocabulary \mathcal{V} . The central design choice is the *state space* where corruption and denoising happen. This leads to two main families: **continuous DLMs**, which diffuse in real-valued spaces, and **discrete DLMs**, which diffuse directly over tokens. They differ primarily in (i) how noise is defined, and (ii) what the reverse model predicts at each step.

Continuous DLMs represent text as a real-valued sequence, such as token embeddings or other continuous vectors, and apply Gaussian diffusion to these representations (Li et al., 2022; Strudel et al., 2022; Dieleman et al., 2022). A typical workflow is:

- *State*: a continuous sequence z_0 derived from text (e.g., word embeddings, one-hot embeddings, or other continuous encodings).
- *Forward*: add Gaussian noise to obtain z_t at different noise levels.
- *Training*: learn a denoiser that predicts a clean target (often z_0 or an equivalent parameterization) from z_t .
- *Generation*: start from Gaussian noise z_T and iteratively denoise to z_0 , then convert the final continuous state into tokens.

The appeal is that continuous DLMs inherit the original diffusion structure: smooth corruption and joint refinement over all positions. At the same time, language information has a discrete component (token identity) that interacts with Gaussian noise differently from continuous geometry.

CANDI (Pynadath et al., 2025) highlights this tension: the signal that preserves token identity and the signal that supports smooth continuous denoising can become useful at different noise ranges, making it difficult for a single Gaussian diffusion trajectory to serve both roles well.

Discrete DLMs keep the state in the token domain and define corruption using masking or categorical transition kernels (Austin et al., 2021; He et al., 2023). A common instance is masked discrete diffusion: at higher noise levels, more positions are replaced by a special mask token. The workflow is:

- *State*: a token sequence $x_t \in \mathcal{V}^n$.
- *Forward*: increase uncertainty by replacing tokens with a mask (or by sampling from a categorical transition).
- *Training*: learn a denoiser that predicts token distributions for corrupted positions given the partially observed sequence.
- *Generation*: begin from a highly corrupted sequence (e.g., mostly masks) and iteratively fill in / refine tokens over multiple steps.

Discrete DLMs align naturally with the discreteness of language, but their corruption is inherently *stepwise*: a token is kept, replaced, or transitioned among categories, rather than being perturbed infinitesimally as in Gaussian diffusion. Recent work scales masked discrete diffusion to large models (Nie et al., 2025; Ye et al., 2025), and explores improved training objectives and discrete-time formulations that better fit categorical state spaces (Lou et al., 2024; Sahoo et al., 2024; Gat et al., 2024). Finally, hybrid systems inject diffusion-style updates into autoregressive decoding to combine parallel refinement with sequential structure (Han et al., 2023; Wu et al., 2023).

3 Advantages of DLMs

Diffusion language models offer several advantages that are difficult to realize with autoregressive (AR) models. These benefits stem from the fact that diffusion updates all positions jointly rather than committing to tokens one at a time. We highlight three central advantages: flexible editing, decoupled compute-length scaling, and improved data efficiency.

Flexible Editing. AR models generate text strictly left-to-right; once a token is produced, it cannot be revised without regenerating the entire sequence. This constraint prevents global adjustments and limits interactive editing. Diffusion models, by contrast, treat the entire sequence as mutable throughout the denoising trajectory. Each step jointly refines all positions, enabling natural support for infilling, rewriting, and any-order generation.

This property is particularly valuable for more structured text domains such as programming codes. A modification to a function signature or variable declaration may require coordinated changes across multiple distant locations. Diffusion’s joint refinement allows such global consistency updates without restarting generation, offering a practical advantage over causal models that can only append tokens.

Token generation is not one-to-one with forward operations. AR inference requires one forward pass per token, coupling computation to sequence length and preventing dynamic allocation of compute based on task difficulty. Diffusion breaks this constraint by introducing a refinement process with T steps independent of the output length N , which includes two aspects:

- *Parallel generation ($T \ll N$)*: A long sequence can be generated simultaneously, providing substantial speedups for long-context synthesis.
- *Deep refinement with more generation steps than the number of tokens ($T > N$)*: For tasks with high information density such as multi-step reasoning, long-form planning, or drafting structured reports, the model can spend more refinement steps per token. This matches the intuition that harder problems demand more “thinking time,” aligning with test-time scaling practices without excessively increasing the number of tokens.

Multi-View Data Modeling. Recent work shows that DLMs can use data more efficiently than autoregressive models, especially when the training set is small (Ni et al., 2025a,b). This arises not only because noise acts as data augmentation, but also because diffusion does not constrain the model to a single left-to-right factorization of the sequence.

If we train causal transformers with different token orders, left-to-right (L2R) and right-to-left

Criterion	AR	Continuous DLMs	Discrete DLMs
Diffusion Properties (D)			
(D1) Smooth Corruption	✗ (Not diffusion)	✓ (Infinitesimal)	✗ (Stepwise/Coarse)
(D2) Tractable Intermediate States	✓ (Prefix state)	✓ (Gaussian)	✓ (Categorical)
(D3) Iterative Refinement	✓ (Token-by-token)	✓ (Latent trajectory)	✓ (Token trajectory)
Language Properties (L)			
(L1) Discreteness	✓ (Tokens)	✗ (Continuous states)	✓ (Tokens)
(L2) Structural Dependency	✗ (Casual)	✗ (Implicit)	✗ (Implicit)

Table 1: AR and diffusion language models (continuous and discrete) viewed through **Diffusion Properties** and **Language Properties**. AR satisfies language properties but does not define a diffusion-style smooth corruption path. Continuous DLMs preserve smooth diffusion but operate in continuous states; discrete DLMs remain token-based but use stepwise corruption and marginal denoising.

(R2L) models learn at almost the same speed, while random orders learn much more slowly (Cunxiao et al., 2025). This suggests that L2R is not a fundamental property of language, but a modeling choice that happens to work well because it matches local word-to-word dependencies. Using a fixed order (L2R or R2L) makes the prediction task very regular: every example is always seen as a prefix followed by a target, which makes it easy for the model to fit the training data, but also makes it easier to memorize.

Diffusion breaks this fixed ordering with random masking. Each sequence is seen many times with different masked positions, so the model must recover tokens from many kinds of partial context instead of a single, fixed prefix. This acts as a strong form of data augmentation. In practice, autoregressive models usually reach good performance faster but start to overfit earlier, while diffusion models improve more slowly and keep benefiting from additional epochs, which matches the idea that their supervision signal is broader and more challenging.

4 Properties for Diffusion LLMs

Diffusion models are defined by a tight link between how data are *corrupted* and how they are *recovered*. When we move from images to text, the same idea remains attractive, but it must be reconciled with what diffusion assumes and what language is. To keep the discussion concrete, we summarize five properties that are used throughout the paper: three describe the diffusion mechanism itself, and two come from the nature of text.

In continuous domains, diffusion uses a forward noising process and a learned reverse denoising pro-

cess. The forward process adds small noise to gradually increase uncertainty, and intermediate noisy states can be sampled at arbitrary noise levels without simulating the full chain. The reverse process starts from a simple noise prior and iteratively denoises to recover a clean sample. We refer to these diffusion-side properties as **D1: smooth corruption**, meaning the time index corresponds to gradual, continuous noise changes rather than abrupt jumps; **D2: tractable intermediate states**, meaning the marginal corruption distribution $q(x_t | x_0)$ is available in closed form or through an analytic procedure so training can sample x_t directly; and **D3: iterative reverse generation**, meaning generation starts from a simple noise prior and repeatedly applies learned reverse updates to refine the same state over multiple steps. Text, in contrast, introduces two properties that are independent of any modeling choice: **L1: discreteness**, since text is composed of discrete symbols and changing a token is a jump rather than an infinitesimal perturbation, and **L2: structural dependency**, since syntax and semantics impose long-range constraints that couple positions.

4.1 How Current DLMs Fit the Properties

Table 1 summarizes how autoregressive (AR) models and diffusion language models (DLMs) align with the diffusion-side properties (D1–D3) and the language-side properties (L1–L2).

AR models generate sequences by a sequential factorization, predicting each token conditioned on its prefix. They therefore satisfy discreteness (L1) by operating directly on tokens. For structural dependency (L2), AR imposes an explicit *causal* dependency assumption: it hard-codes one spe-

cific factorization of dependencies (left-to-right), which captures an important class of linguistic constraints, but does not represent the full space of global dependencies in a symmetric way. On the diffusion side, AR is not diffusion in the sense of smooth corruption (D1), but it does have a simple and tractable notion of intermediate state: the current prefix together with the un-generated suffix (D2). Generation is also iterative by construction (D3), refining the sequence token by token.

Continuous DLMs apply diffusion to continuous representations derived from text, such as embeddings, latent vectors, or token-distribution states (Ho et al., 2020; Li et al., 2022; Strudel et al., 2022; Dieleman et al., 2022; Gong et al., 2022). They closely follow the original diffusion mechanism: Gaussian-style perturbations yield smooth corruption (D1); intermediate marginals are tractable (D2) in the continuous state space; and sampling proceeds through iterative refinement (D3) along a latent denoising trajectory. On the language side, however, the diffusion state is continuous rather than symbolic, so discreteness (L1) is not satisfied. For structural dependency (L2), these models typically do not impose an explicit dependency factorization; instead, any multi-token constraints must be learned implicitly through the denoising network.

Discrete DLMs define diffusion directly over tokens using masking or categorical transition kernels (Austin et al., 2021; He et al., 2023). They satisfy discreteness (L1) by construction. On the diffusion side, because the corruption process is specified by explicit transition matrices, the intermediate corruption distribution is tractable (D2), and generation follows iterative refinement (D3) with token-level intermediate states. However, smooth corruption (D1) is approximated: even with continuous-time parameterizations (Austin et al., 2021; Lou et al., 2024), the underlying state changes remain step-wise due to the discrete space, and recent analysis suggests that t often behaves like a proxy for the number of masked tokens rather than a smooth signal-to-noise ratio (Zheng et al., 2024). For structural dependency (L2), discrete DLMs also rely mainly on implicit learning rather than an explicit dependency assumption: the corruption is usually position-symmetric, and dependency structure is expected to emerge from the denoiser rather than from the generative factorization itself.

Trade-offs. Viewed through these properties, continuous and discrete DLMs reflect a recurring trade-off. Continuous methods stay close to classical diffusion and preserve (D1 – D3), but they move away from symbolic text (L1) and usually leave (L2) to be learned implicitly. Discrete methods stay faithful to symbols (L1) and keep tractable corruption (D2), but they sacrifice smoothness (D1) and do not build in a dependency structure (L2). Soft-Masked Diffusion (Hersche et al., 2025) illustrates how improving one property can weaken another: by replacing binary masks with soft mixtures over candidate tokens, it makes the corruption trajectory more gradual and improves (D1), but the intermediate corruption distribution becomes model-dependent and is no longer available in closed form, weakening (D2) and requiring a two-pass training scheme. Overall, we conclude that property-level Trade-offs should be explicitly considered when designing methods that incorporate both language and diffusion.

5 Core Challenges for DLMs

The previous sections motivate diffusion for language and summarize the key properties in Table 1. We now focus on the two properties that are most problematic in practice and largely determine the current gap between diffusion and text: **smooth corruption** (D1) and **structural dependency** (L2). We first discuss why designing a “smooth” corruption process for discrete sequences is non-trivial (Section 5.1), and then analyze why token-wise, parallel denoising struggles to capture multi-token constraints (Section 5.2).

5.1 Smooth Corruption does not mean Even Information Loss

In Section 4, we defined (D1) *Smooth Corruption* from a modeling perspective. Here we view it from an information perspective: a good noise schedule should make the information about the target decay gradually as the noise level t increases. In other words, when t changes a little, the amount of recoverable information should also change a little, both at the sequence level and at the token level.

For language, this is challenging because information is not evenly distributed across tokens. Some tokens carry most of the meaning and strongly constrain the rest of the sentence, while others are easier to infer. This is also why attention matters: it lets the model weigh context by use-

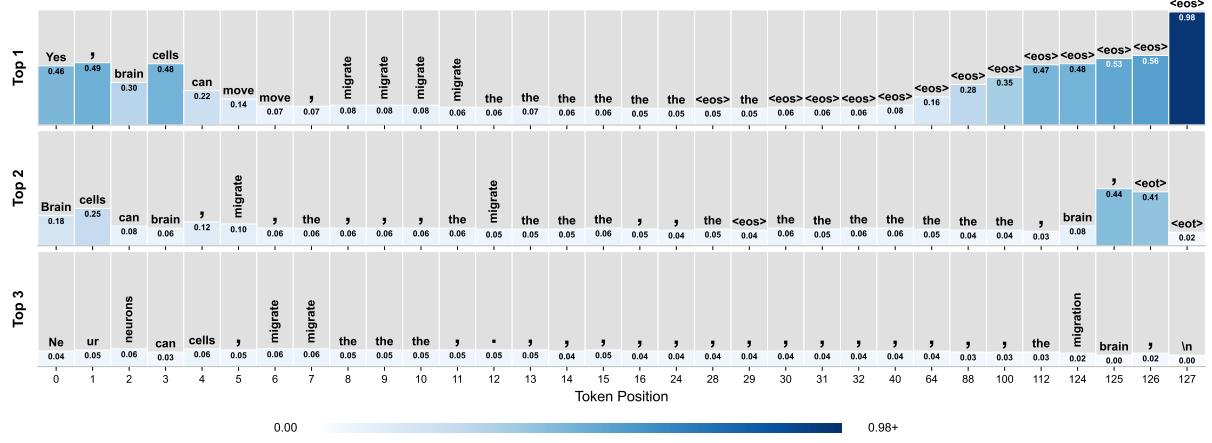


Figure 2: An example from LIMA dataset (Zhou et al., 2023) illustrating the probability distribution analysis of top-3 tokens by LLaDA-Instruct with 128 masked positions when prompt with "***Can brain cells move? By movement I mean long distance migration (preferably within the brain only).***". We found early [MASK] are more deterministic, while distant ones collapse to frequency-dominant. The Experiment Setting is detailed in Appendix A.

fulness instead of treating all positions equally. If corruption applies the same rule everywhere, then “the same noise level” does not mean “the same information loss”. Corrupting a few high-impact tokens can destroy the core meaning early, while corrupting low-impact tokens may change little. We examine this mismatch at both the sequence level and the token level.

At the sequence level, discrete and continuous diffusion behave quite differently. In masked discrete diffusion, the forward process gradually increases the number of masked positions. Even at high mask ratios, a subset of tokens may remain visible, so some coarse information about the sequence is preserved. However, this preservation is blind to where information actually lies: all positions are equally likely to be masked, regardless of whether they carry crucial content or redundant tokens. The same noise level t can therefore correspond to very different amounts of remaining information, depending on which positions were masked. Continuous diffusion approaches apply Gaussian noise to every coordinate in an embedding or one-hot space. From a mathematical standpoint this is perfectly smooth, but CANDI (Pynadath et al., 2025) shows that, for large vocabularies, the discrete identity of a token becomes unrecoverable at relatively low noise levels. The model’s ability to distinguish the correct token from tens of thousands of alternative tokens collapses quickly, even though the continuous signal still changes slowly. Thus, if we measure smoothness in terms of *recoverable information* rather than variance of

the noise, the corruption is not gradual: token identity is lost in a few steps.

At the token level, the situation is more uneven. In discrete diffusion, corruption is often described as “binary” (keep or mask). However, from an information viewpoint, a masked token is not always equally unknown. If its neighbors are visible, local context strongly constrains what the token can be; if its neighbors are also masked, the same token effectively experiences a much higher noise level. Uniform masking therefore produces a wide spread of effective noise across positions: some tokens remain easy to recover, while others quickly become almost impossible.

This explains the “frequency collapse” observed in practice. Let x_i be a token and $x_{\mathcal{O}}$ the set of positions that survive corruption. A diffusion LM estimates $p(x_i \mid x_{\mathcal{O}})$. As more local context is removed around x_i , the Mutual Information $I(x_i; x_{\mathcal{O}})$ rapidly decreases. When the remaining context carries almost no information about x_i , we have

$$\lim_{I(x_i; x_{\mathcal{O}}) \rightarrow 0} p(x_i \mid x_{\mathcal{O}}) = p(x_i), \quad (1)$$

so the optimal prediction becomes the marginal distribution. As a result, the model prefers very common tokens (such as “the”, punctuation, or `<eos>`) because, given the corrupted input, these are simply the statistically safest guesses.

Empirical Study. To illustrate such token-level locality, our analysis of how current DLMs treats masked token makes this effect visible (Figure 2).

We prompt the model with “*Can brain cells move? By movement I mean long distance migration (preferably within the brain only).*” and then append 128 masked positions. We inspect the top-3 predictions at each position.

At positions closest to the prompt (0–2), the predictions are confident and semantically appropriate, such as “Yes” and “brain”. Here, nearby context provides strong constraints, so the model can effectively recover the token. As we move further away (positions 12–29), the influence of the prompt fades. The predictions become uncertain and collapse toward high-frequency tokens such as “the” and punctuation, even though the semantic question has not changed. At even more distant positions, the model assigns high probability to `<eos>`, reflecting only the dataset’s length statistics. A similar pattern appears in other DLMs such as Dream-7B: early positions are tightly constrained, while distant ones drift toward the unigram prior.

This experiment highlights that, under a uniform corruption schedule, local information disappears much faster than the nominal noise level would suggest. Positions that are equally “noised” according to t can have very different amounts of usable information, depending on how much local context remains.

We found that some current methods have tried to mitigate such effect. For example, Dream-7B explicitly addresses this mismatch (Ye et al., 2025). Its Context-Adaptive noise Rescheduling at Token-level (CART) scales the training loss by a geometric function of the distance to the nearest unmasked token. Positions with nearby anchors are upweighted, encouraging the model to learn from contexts where recovery is feasible, while positions surrounded by masks are downweighted, preventing the model from overfitting to signals that are effectively just word frequency. In our words, CART modifies the training objective to not over-punish the mask with little context.

Future Direction. These observations suggest that satisfying (D1) for language requires more than a continuous noise schedule; the corruption must be smooth in terms of information, not only in terms of variance.

For discrete DLMs, one direction is to move beyond binary masking and define transition kernels that change tokens in smaller, structured steps, for example along semantic or categorical axes (specific → general → [MASK]). Such processes could

keep token identity recoverable for longer while still increasing noise.

For continuous DLMs, hybrids such as CANDI (Pynadath et al., 2025) decouple discrete identity from continuous refinement. They maintain a corruption process that preserves which token is present, while a separate continuous channel learns smooth gradients. Both lines of work share the same goal: to align the corruption process with how information is distributed across positions, so that information loss becomes more gradual and even, at both the sequence and token level.

5.2 Absence of Structural Dependency

Masked discrete diffusion models typically learn token-wise conditionals given the visible context. At a denoising step, the model outputs a separate distribution for each masked position, $p(x_i | x_{\mathcal{O}})$, and the training loss is a sum of per-token cross-entropies. Under this objective, the model is not directly trained to represent how multiple unknown tokens should constrain one another. As a result, the model can match the *marginal* distribution at each position while still missing *joint* constraints required by language (L2), such as agreement and phrase-level compatibility.

Conditions. Importantly, this limitation is most visible under two practical choices that are common in current masked DLMs.

(1) *Committed intermediate states.* Many implementations represent intermediate states as *partially filled token sequences*: once a token is sampled (or greedily chosen) at some step, it becomes part of the visible context for later steps. This commitment makes the model sensitive to early inconsistent choices, because later updates must condition on them rather than revising them jointly. By contrast, an idealized refinement process that keeps states *uncommitted* until convergence (e.g., operating on soft token distributions instead of hard tokens) does not force such early irreversible decisions, and therefore reduces the chance of forming incompatible multi-token combinations.

(2) *Parallel updates with fewer steps than tokens.* If the model updates many masked positions in parallel and uses a small number of denoising steps ($T \ll N$), then multiple dependent tokens must be decided at the same time, without an external factorization that enforces their compatibility. In contrast, if decoding updates only *one token at a time* (or uses $T \geq N$ with a sequential schedule),

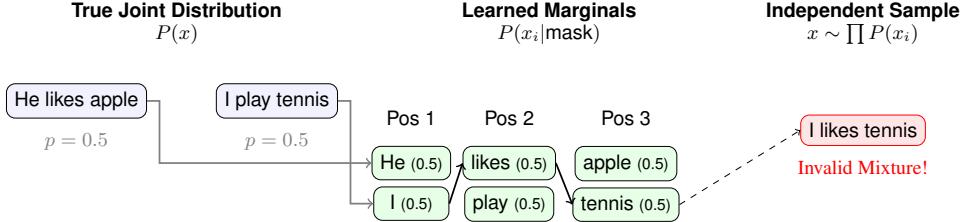


Figure 3: The “Marginal Trap”: A toy example shows that the model learns from “He likes apple” and “I play tennis” (50% each). However, parallel decoding samples them independently resulting each position to be 50% for both samples at each token position. Directly sampling from these distribution may create a path (I → likes → tennis) that never existed in the training data.

then joint constraints can be enforced implicitly through conditioning: later choices depend on earlier chosen tokens, reducing inconsistent mixtures. This resembles the role played by sequential factorization in AR decoding.

Example of the “Marginal Trap.” Figure 3 shows a toy dataset containing only two sentences: “*He likes apple*” and “*I play tennis*”. The optimal token-wise model learns the correct marginals. Each prediction is correct on its own, but sampling these marginals independently can produce invalid mixtures like “*I likes tennis*”. This illustrates the core gap: marginals are correct, but the joint distribution is not modeled.

Empirical Evidence. The same pattern appears in large masked DLMs. In LLaDA-8B-Instruct (Figure 2), two nearby masked positions may both assign non-trivial probability to the same token (e.g., “*brain*”), so parallel sampling can yield local duplications such as “*brain brain*”. This is not because each position is unreasonable in isolation; it is because the model has no explicit mechanism to couple the decisions across positions during a parallel update.

In practice, large masked DLMs often rely on generating one token per forward or extremely low temperatures (Nie et al., 2025; Ye et al., 2025) to reduce these inconsistent combinations. Alternatively, one can reduce the issue by decoding with more sequential schedules (e.g., updating one token per step), but this partially sacrifices the parallelism that motivates diffusion decoding.

Future Direction. Addressing (L2) requires mechanisms that couple multiple positions beyond token-wise losses. One direction is to move from per-token cross-entropy toward sequence-level or structured objectives that score joint configurations, so inconsistent multi-token outcomes are directly penalized (e.g., energy-based or contrastive for-

mulations (Yang et al., 2021)). A complementary direction is to use state representations that delay commitment, such as keeping intermediate states as soft token distributions, so later steps can revise earlier choices and correct inconsistencies jointly instead of being locked into early decisions. Finally, parallel decoding can be made more reliable by explicitly pushing the model toward convergence of single path. For example, dParallel trains a DLLM with a certainty-forcing distillation objective, encouraging many masked positions to reach high confidence earlier (Chen et al., 2025).

6 Conclusion

In this paper, we revisited diffusion language models through a unified lens based on **Diffusion Properties (D1 to D3)** and **Language Properties (L1 and L2)**. This framework shows that current diffusion LLMs occupy only parts of the ideal design landscape. Continuous approaches satisfy the mathematical form of diffusion but lose contact with the discrete and dependency-rich structure of text. Discrete approaches preserve the state space of language but must approximate diffusion through coarse masking and independent token predictions.

Our empirical analysis demonstrates that these structural gaps have direct impact during inference. In Section 5.1, we documented the phenomenon of frequency collapse: uniform corruption ignores how information is distributed across positions, causing recoverability to drop abruptly and pushing predictions toward the unigram prior. In Section 5.2, we highlighted the marginal trap: training objectives that operate on individual tokens cannot enforce multi-token dependencies, which leads parallel decoding to produce sequences that are locally plausible but globally inconsistent. Together, these findings show that the typical diffusion assumptions of uniform corruption and marginal denoising are not naturally aligned with the structure of text.

Limitations

Our analysis is primarily conceptual and aims to clarify definitions, assumptions, and trade-offs in diffusion language modeling. As a result, several limitations should be noted. First, the “five properties” is an abstraction that compresses a wide range of diffusion formulations into a small set of criteria; different choices of definitions may yield alternative interpretation. Second, our discussion reflects common design patterns in existing continuous and discrete DLMs, but does not exhaust all variants (e.g., alternative state spaces, transition kernels, or decoding schedules). Third, we did not take all variants of DLMs into consideration and discuss in detail due to the number of variation. Finally, we do not quantify end-to-end trade-offs among speed, quality, and controllability under unified implementations, which would be required for strong engineering recommendations.

References

- Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. In *Proceedings of NeurIPS*.
- Zigeng Chen, Gongfan Fang, Xinyin Ma, Ruonan Yu, and Xinchao Wang. 2025. dparallel: Learnable parallel decoding for dllms. *arXiv:2509.26488*.
- Du Cunxiao, Yang Xinyu, Zhang Fengzhuo, Hou Yunlong, Yu Sicheng, Lin Min, and Du Chao. 2025. Understanding the limitations of diffusion llms through a probabilistic perspective.
- Sander Dieleman, Laurent Sartran, Arman Roshan-nai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, et al. 2022. Continuous diffusion for categorical data. *arXiv:2211.15089*.
- Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. 2024. Discrete flow matching. *Proceedings of NeurIPS*, 37:133345–133385.
- Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2022. Diffuseq: Sequence to sequence text generation with diffusion models. In *Proceedings of ICLR*.
- Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. 2023. Ssd-lm: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. In *Proceedings of ACL*, pages 11575–11596.
- Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. 2023. Diffusionbert: Improving generative masked language models with diffusion models. In *Proceedings of ACL*.
- Michael Hersche, Samuel Moor-Smith, Thomas Hofmann, and Abbas Rahimi. 2025. Soft-masked diffusion language models.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Proceedings of NeurIPS*, 33:6840–6851.
- Jaeyeon Kim, Lee Cheuk-Kit, Carles Domingo-Enrich, Yilun Du, Sham Kakade, Timothy Ngotiaoco, Sitan Chen, and Michael Albergo. 2025. Any-order flexible length masked diffusion.
- Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022. Diffusion-lm improves controllable text generation. *Proceedings of NeurIPS*, 35:4328–4343.
- Aaron Lou, Chenlin Meng, and Stefano Ermon. 2024. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of ICML*.
- Jinjie Ni, Qian Liu, Longxu Dou, Chao Du, Zili Wang, Hang Yan, Tianyu Pang, and Michael Qizhe Shieh. 2025a. Diffusion language models are super data learners. *arXiv:2511.03276*.
- Jinjie Ni, Qian Liu, Chao Du, Longxu Dou, Hang Yan, Zili Wang, Tianyu Pang, and Michael Qizhe Shieh. 2025b. Training optimal large diffusion language models.
- Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Large language diffusion models. In *Proceedings of NeurIPS*.
- Patrick Pynadath, Jiaxin Shi, and Ruqi Zhang. 2025. Candi: Hybrid discrete-continuous diffusion models. *arXiv:2510.22510*.
- Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. 2024. Simple and effective masked diffusion language models. *Proceedings of NeurIPS*.
- Robin Strudel, Corentin Tallec, Florent Altché, Yilun Du, Yaroslav Ganin, Arthur Mensch, Will Grathwohl, Nikolay Savinov, Sander Dieleman, Laurent Sifre, et al. 2022. Self-conditioned embedding diffusion for text generation. *arXiv:2211.04236*.
- Tong Wu, Zhihao Fan, Xiao Liu, Hai-Tao Zheng, Yeyun Gong, Jian Jiao, Juntao Li, Jian Guo, Nan Duan, Weizhu Chen, et al. 2023. Ar-diffusion: Auto-regressive diffusion model for text generation. *Proceedings of NeurIPS*, 36:39957–39974.

Ziyi Yang, Yinfei Yang, Daniel Cer, Jax Law, and Eric Darve. 2021. Universal sentence representation learning with conditional masked language model. In *Proceedings of EMNLP*.

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

Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. 2024. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. In *Proceedings of ICLR*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. In *Proceedings of NeurIPS*.

A Experiment Details

To produce Figure 2, we probe a masked DLM with a single forward pass under a fully-masked answer span. Concretely, we first format the user question using the model’s chat template, and then append $N = 128$ mask tokens to represent an unknown response. The resulting input has the form

```
input = chat_template(user) || [MASK]128.
```

We run the model once and extract the output distribution at each masked position. For every position $i \in \{0, \dots, 127\}$, we apply softmax to the logits and record the top-3 tokens and their probabilities. Figure 2 visualizes these top-3 predictions across positions for one representative prompt.

Beyond the single example shown, we repeat the same procedure on 100 different prompts from LIMA’s training dataset (Zhou et al., 2023), and observe the same qualitative pattern: early masked positions tend to have sharper, more content-specific distributions, while distant positions become increasingly dominated by high-frequency tokens and special symbols such as punctuation or <eos>.