COEVOLUTIONARY CONTINUOUS DISCRETE DIFFUSION: MAKE YOUR DIFFUSION LANGUAGE MODEL A LATENT REASONER

Anonymous authors
Paper under double-blind review

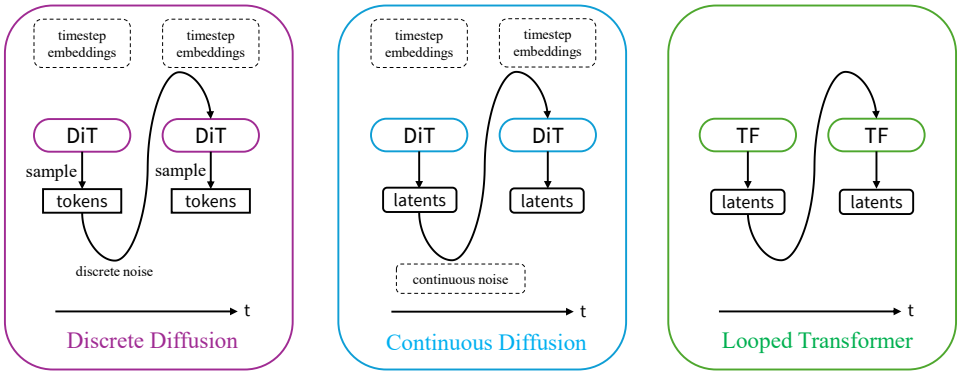
ABSTRACT

Diffusion language models, especially masked discrete diffusion models, have achieved great success recently. While there are some theoretical and primary empirical results showing the advantages of latent reasoning with looped transformers or continuous chain-of-thoughts, continuous diffusion models typically underperform their discrete counterparts. In this paper, we argue that diffusion language models do not necessarily need to be in the discrete space. In particular, we prove that continuous diffusion models have stronger expressivity than discrete diffusions and looped transformers. We attribute the contradiction between the theoretical expressiveness and empirical performance to their practical trainability: while continuous diffusion provides intermediate supervision that looped transformers lack, they introduce additional difficulty decoding tokens into the discrete token space from the continuous representation space. We therefore propose **C**oevolutionary **C**ontinuous **D**iscrete **D**iffusion (CCDD), which defines a joint multimodal diffusion process on the union of a continuous representation space and a discrete token space, leveraging a single model to simultaneously denoise in the joint space. By combining two modalities, CCDD is expressive with rich semantics in the latent space, as well as good trainability and sample quality with the help of explicit discrete tokens. We also propose effective architectures and advanced training/sampling techniques for CCDD, which reveals strong empirical performance in extensive language modeling experiments on real-world tasks.

1 INTRODUCTION

Recent years have seen great success of autoregressive (AR) large language models (LLMs) (Achiam et al., 2023; Yang et al., 2024b; Liu et al., 2024), especially their significant improvement in complex reasoning (Guo et al., 2025; Comanici et al., 2025). However, the **discrete** and **left-to-right** nature of these models still poses some fundamental difficulties. It is a known result in computation complexity theory that the expressivity of transformers – the architectural foundations of modern LLMs, are restricted in the class TC^0 even with logarithmic Chain-of-Thought (CoT) steps (Merrill & Sabharwal, 2025). This suggests that transformers cannot accurately address problems outside the TC^0 class such as *recognizing formal language* which measures state tracking capabilities, and *graph connectivity* that captures multistep reasoning ability, under standard complexity conjectures. Empirically, even state-of-the-art LLMs often struggle in a wide range of complex tasks requiring strong planning, parallel searching, and backtracking capabilities, such as *Sudoku*. To overcome these challenges, researchers have been working to develop new language modeling paradigms.

On the one hand, LLMs are shown to be benefit from latent reasoning through various ways, including looped transformers (LT) (Giannou et al., 2023) or continuous CoT (Hao et al., 2024). One line of research on *looped transformers* or *universal transformers* (Dehghani et al., 2018a) theoretically demonstrates that when a block of middle layers of a fixed transformer model is repeated for a variable number of times, its expressivity can be significantly improved (Saunshi et al., 2025; Merrill & Sabharwal, 2025; Fan et al., 2024). For example, Merrill & Sabharwal (2025) proves that a looped transformer with depth $\Theta(\log n)$ are in TC^1 and thus solve regular language recognition and graph connectivity problems with input context length n, which are intractable by LLMs with logarithmic CoT steps. Intuitively, LTs do not decode latents into discrete tokens until the last loop time, enabling



Theoretical Expressivity (Section 3.1): CDM $\succcurlyeq$ DDM CDM $\succcurlyeq$ LT

Empirical Trainability (Section 3.2): DDM $\succcurlyeq$ CDM $\succcurlyeq$ LT

Figure 1: Comparison of theoretical expressiveness and practical trainability of: discrete diffusion (left), continuous diffusion with optional continuous noise (middle), and looped transformer (right).

implicit conduction of complex reasoning such as planning and searching in the **continuous latent space** and efficient storage of these meaningful intermediate information efficiently for later loops. Taking advantage of a powerful latent space, scaling the looped depth of the model (even with uniform parameters) can be a more effective way for test-time scaling compared to increasing the length of reasoning steps with explicit CoT in the discrete, finite token space. Unfortunately, despite their strong theoretical expressive power, the LT scheme is not widely adopted in mainstream LLMs due to the limited practical performance. We attribute this to *the lack of supervision on intermediate states* – looping is an inference-time trick and the intermediate rollouts are not supervised at all in training. Therefore, when applied to practical LLMs at scale, looped transformers could encounter severe out-of-distribution (OOD) issues.

On the other hand, diffusion language models (DLMs) (Gong et al., 2024; Nie et al., 2025; Ye et al., 2025) have received considerable attention from researchers in recent years. The **non-autoregressive** nature of DLMs enables *any-order generation*, *self-correction*, and *parallel decoding* capabilities, leading to potentials in stronger expressiveness and superior generation efficiency. State-of-the-art DLMs outperform LLMs in complex or structured reasoning tasks such as Sudoku (Kim et al., 2025) and coding (Gong et al., 2025). There are two main families of DLMs: continuous diffusion models (CDMs) based on SDE or PF–ODE (Gong et al., 2022; Li et al., 2022; Sahoo et al., 2025), and discrete diffusion models (DDMs) based on the continuous-time Markov chain (CTMC) (Austin et al., 2021; Lou et al., 2023; Sahoo et al., 2024; Shi et al., 2024). Continuous DLMs, performing iterative denoising in either embedding space or probability space, emerged earlier but fell far behind AR LLMs in practical performance, until the recent success of discrete DLMs. Intriguingly, discrete diffusions with masked noises tend to outperform those with uniform noises (Amin et al., 2025), at the cost of losing self-correction capabilities. Analogously to LLMs, discrete DLMs also reason in the explicit token space and may partially lose information of previous decoding steps.

In this paper, we conceptually connect all the aforementioned models and algorithms, and propose a new language modeling paradigm that combines the advantages of previous methods. In Section 3, we systematically compare these models from the perspective of **theoretical expressivity** and **practical trainability** (Figure 1). We first show in Section 3.1 that: (i) continuous DLMs are more powerful than their discrete counterparts; (ii) continuous diffusion generalizes looped transformer, which is already partially more expressive than CoT. However, previous heuristics in performance seem to contradict the theoretical expressiveness, which we try to elucidate in Section 3.2 from the perspective of practical trainability. We argue that looped transformers face OOD issues in inference due to lack of intermediate supervision, while diffusion models supervise states in the whole probability path. We then attribute the insufficient trainability of continuous DLMs to the large decision space, the deficient and low-quality embeddings, and combinatorial complexity in decoding latents to tokens.

Based on these insights, in Section 4 we propose a new language modeling paradigm termed **Coevolutionary Continuous Discrete Diffusion** (CCDD), which combines the advantages of both continuous and discrete diffusion while eliminating the shortcomings via their complementarity. CCDD defines a joint diffusion process on both the discrete state space through CTMC and the

continuous probability or embedding space through SDE (Section 4.1). In the reverse process, one denoising model taking the partially noised tokens of both modalities as inputs learns to predict the data distribution in both spaces. Inspired by DiT (Peebles & Xie, 2023), MM-DiT (Esser et al., 2024), and MoE (Shazeer et al., 2017), we design several architectures for joint denoising with various parameter and computation efficiency (Section 4.2). In implementation, we adopt the contextualized embedding space from concurrent pretrained text embedding models such as Qwen3-Embedding (Zhang et al., 2025), which provides rich semnatics modeling joint distributions and injects knowledge from pretrained LMs via implicit representation guidance. CCDD further benefits from advanced training techniques such as asynchronous noises, and could adaptively balance between sampling quality (through inference-time scaling with SDE) and efficiency (through few-step sampling with ODE), thanks to improved sampling algorithms including representation classifier-free guidance (CFG) and consistency-based decoding. To summarize, CCDD features both the strong expressive power of continuous diffusion and the good trainability of discrete diffusion. We experimentally validate the effectiveness of CCDD through extensive text modeling tasks, showing that CCDD reduces over 25% compared with baselines of the same scale on LM1B dataset.

2 Preliminary

Notation. Let $\Omega = \{1, \ldots, V\}$ be a vocabulary ($|\Omega| = V$) and L the sequence length. Suppose we have the discrete sequence data $x_0 \in \Omega^L$. A fixed encoder $\mathcal{E}$ maps tokens to continuous embeddings $z_0 = \mathcal{E}(x_0) \in \mathbb{R}^{L\times d}$, which can be either one-hot on the simplex $\Delta^{V-1} := \{p \in \mathbb{R}^V_{\geq 0} : \mathbf{1}^\top p = 1\}$ (namely $d = V$) or representations of any pretrained model / LLM – hence, we may use the terms "logits" and "representations" interchangeably. We write $t \in [0, 1]$ for continuous time or $t \in \{1, \ldots, T\}$ for discrete steps, where the latter can also be converted to discretized $\tilde{t} \in [0, 1]$. Denote corrupted variables as z_t and x_t, and in Gaussian settings $\epsilon_t \sim \mathcal{N}(0, I)$ denotes the standard noise used to synthesize z_t. A *single* time–conditioned network $f_\theta(\cdot, t, \mathrm{cond})$ (typically a transformer-based model) is called at every step/instant, where cond is the optional condition (often omitted).

Continuous diffusion. The forward/noising dynamics for continuous data $z_t \in \mathbb{R}^{L\times d}$ are

$$dz_t \;=\; a_t(z_t)\,dt \;+\; g_t\,dW_t, \tag{1}$$

with drift $a_t(\cdot)$, scalar (or matrix) diffusion $g_t \geq 0$, and Wiener process W_t. The marginals $q_t(z_t)$ satisfy the Fokker–Planck PDE $\partial_t q_t = -\nabla\cdot(a_t q_t) + \frac{1}{2} g_t^2 \Delta q_t$. A standard instance is the variance preserving (VP) schedule: $dz_t = -\frac{1}{2}\beta_t z_t\,dt + \sqrt{\beta_t}\,dW_t$, yielding a closed-form forward marginals:

$$z_t \;=\; \alpha_t\, z_0 + \sigma_t\, \epsilon_t,\;\; \epsilon_t \sim \mathcal{N}(0, I),\;\; \alpha_t = \exp\big(-\tfrac{1}{2}\int_0^t \beta_\tau\, d\tau\big),\;\; \sigma_t = \sqrt{1-\alpha_t^2}. \tag{2}$$

The reverse process is based on the reverse SDE in equation 3 or the PF–ODE in equation 4:

$$dz_t \;=\; \big(a_t(z_t) - g_t^2\, s_\theta(z_t, t)\big)\,dt \;+\; g_t\,d\bar{W}_t, \tag{3}$$

$$\dot{z}_t \;=\; a_t(z_t) - \tfrac{1}{2} g_t^2\, s_\theta(z_t, t). \tag{4}$$

where $s_\theta(\cdot, t) \approx \nabla_z \log q_t(\cdot)$ is produced by f_θ (up to a known scaling). Under equation 2, we have closed form sampling update rules as in DDPM (Ho et al., 2020) and DDIM (Song et al., 2020). In practice, common equivalent heads include ϵ-pred: $\epsilon_\theta(z_t, t)$, x_0-pred: $\hat{z}_{0,\theta}(z_t, t)$, v-pred: $v_\theta(z_t, t)$, which are equivalent up to linear transformations. A typical VP training loss with ϵ-prediction and time-dependent weight $\lambda_{\mathrm{cont}}(t)$ is

$$\mathcal{L}_{\text{cont}} = \mathbb{E}_{t, z_0, \epsilon}\big[\lambda_{\mathrm{cont}}(t)\, \|\epsilon - \epsilon_\theta(\alpha_t z_0 + \sigma_t \epsilon,\, t)\|^2\big]. \tag{5}$$

Discrete diffusion. For a single token ($L = 1$ for notation), let $q_t \in \Delta^{V-1}$ be the column vector of token marginals. In the forward process, a time-inhomogeneous continuous-time Markov chain (CTMC) with generator $G_t \in \mathbb{R}^{V\times V}$ evolves as follows ($\mathcal{T}$ the normalizing constant):

$$\dot{q}_t = G_t\, q_t, \quad P_{s\to t} = \mathcal{T}\exp\big(\int_s^t G_\tau\, d\tau\big), \quad q_t = P_{0\to t} q_0. \tag{6}$$

Some common choices of G_t include: (i) *Uniform noise (USDM)*, where state j jumps to any $i \neq j$ uniformly at rate u_t, resulting in $(G_t)_{ij} = \frac{u_t}{V-1}\ (i \neq j),\ (G_t)_{jj} = -u_t$; (ii) *Masked (absorbing)*

noise, where we Augment Ω with a mask state [MASK]. For any $j \neq$ [MASK], it jumps to the mask state with rate u_t, leading to $(G_t)_{[\texttt{MASK}],j} = u_t,\ (G_t)_{jj} = -u_t,\ (G_t)_{\cdot,[\texttt{MASK}]} = 0$. The marginal of both processes can be expressed by an interpolation between the clean data and a noise distribution π_t,

$$q_t(x_t|x_0) = \text{Cat}(\eta_t x_0 + (1-\eta_t)\pi_t) \tag{7}$$

where $\pi_t = \boldsymbol{m}$ the one-hot vector for [MASK] for the absorbing noise, and $\pi_t = \frac{1}{V}\mathbf{1}$ for the uniform noise. For sequences of length L, corruptions are exerted independently per-position. In the reverse process, the denoising network predicts the clean data distribution $\mathbf{x}_\theta := \hat{\pi}_\theta(x_0|x_t,t) = \mathrm{softmax}(\ell_\theta(x_t,t))$ where ℓ_θ the output logits. A Bayesian form of posterior is

$$p_\theta(x_s|x_t) = q_{t|s}(x_t|x_s)\frac{q_s(x_s|\hat{\pi}_\theta)}{q_t(x_t|\hat{\pi}_\theta)} \tag{8}$$

The training loss is usually calculated as with weights $\lambda_{\text{disc}}(t,x_t,x_0)$ derived from Rao-Blackwellized likelihood bounds and would be zero for unmasked token in masked diffusion:

$$\mathcal{L}_{\text{disc}} = \mathbb{E}_{t,x_0}\left[\lambda_{\text{disc}}(t,x_t,x_0)\log\langle\hat{\pi}_\theta(x_t,t),x_0\rangle\right] \tag{9}$$

Looped transformer. In the standard setting of looped transformer, a *single* n-layer transformer (Φ_θ) block with shared parameters θ is rolled out adaptive T steps. Let $h_k \in \mathbb{R}^{L\times d}$ be the hidden state after k steps.

$$h_{k+1} \;=\; \Phi_\theta\big(h_k\big), \qquad k = 0,\dots,T-1. \tag{10}$$

The transformer layers can be either encoder-based (bidirectional attention) or decoder-based (causal attention) with residual connections. A readout $R(h_T) : \mathbb{R}^{L\times d} \to \mathbb{R}^{L\times V}$ yields logits $\ell_{\theta,T} = R(h_T)$ and token samples $x_T \sim \mathrm{Cat}(\mathrm{softmax}(\ell_{\theta,T}))$. Traditionally, looped transformers receive supervision on the final outputs using standard cross-entropy loss.

3 Theoretical Expressivity and Practical Trainability Analysis

3.1 Theoretical Expressivity Analysis

In this subsection, we analyze the theoretical expressivity of CDM, DDM and LT. Throughout the paper, for expressiveness comparison we assume the *same* architectures and parameter counts in networks with *finite capacity* as in common practice. By default, we consider Transformers (TF) (Vaswani et al., 2017) and Diffusion Transformers (DiT) (Peebles & Xie, 2023), up to slight differences in the first encoding layer and the last decoding layer. Proofs are available in Appendix B.

Continuous diffusion dominates discrete diffusion. We first compare the families of trajectory laws and terminal distributions induced by CDM on $\mathcal{Z} := \mathbb{R}^{L\times d}$ and DDM on $\mathcal{X} := \Omega^L$ embedded into $\mathcal{Z}$ via a bijective encoder $\mathcal{E}$. Denote the distribution induced from the reverse SDE (equation 3) as $p_t(z_t) \in \mathcal{P}(\mathcal{Z})$, and $p_t(x_t) \in \mathcal{P}(\mathcal{X})$ produced by the posterior of CTMC (equation 8).

Definition 1 (Trajectory families and embedded discrete family)**.** *Define the trajectory family of continuous diffusion* $\mathsf{F}_{\text{cont}}(\theta)$*, discrete diffusion* $F_{\text{disc}}(\theta)$*, and the* embedded discrete family $\widetilde{\mathsf{F}}_{\text{disc}}(\theta) \subset \mathcal{P}(\mathcal{Z})$ *(the pushforward by the fixed encoder* $\mathcal{E}$*) as follows:*

$$\mathsf{F}_{\text{cont}}(\theta) := \Big\{\{p_t(z_t)\}_{t\in[0,1]}\Big\},\ \mathsf{F}_{\text{disc}}(\theta) := \Big\{\{p_t(x_t)\}_{t\in[0,1]}\Big\},\ \widetilde{\mathsf{F}}_{\text{disc}}(\theta) := \mathcal{E}_\sharp\mathsf{F}_{\text{disc}}(\theta) \tag{11}$$

Theorem 1 (Strict trajectory-level gap)**.** *At any fixed* $t \in [0,1]$*, we have the following strict inclusion*

$$\widetilde{\mathsf{F}}_{\text{disc}}(\theta) \;\subsetneq\; \mathsf{F}_{\text{cont}}(\theta) \;\subseteq\; \mathcal{P}(\mathcal{Z}) \tag{12}$$

The key insight here is the fact that the input of denoising network in DDM is always discrete and *supported on a finite set* (Lemma 3), while the Fokker-Planck equation in CDM would yield *absolutely continuous marginals* (Lemma 4). The inclusion holds for the entire sampling trajectories so as the terminal distributions. Intuitively, considering both two models operating on the probability simplex – analogously for other embedding spaces given the encoder bijective. The "logits→sample→embed" operation in discrete diffusion sampling loop *quantizes* the cross-step memory into a single token per step, losing access to the full logits. However, the "logits→logits" procedure in continuous diffusion propagates a continuous state, retaining fine-grained uncertainty and historical memory. The discrete scheme imposes a hard finite-support bottleneck with information loss at every step (Lemma 6), making it strictly dominated by the continuous counterpart producing non-atomic outputs.

Remark 1 ("Finite-combination" viewpoint)**.** *Discrete diffusion operates over convex combinations of finitely many basis states in Δ^{V-1}. This is a strict subset of the continuous family, which admits general smooth/multimodal densities via equation 1. Lemma 6 further shows that per-step token sampling discards the full logit geometry, whereas continuous samplers propagate full z_t (ODE deterministically or SDE stochastically) without compulsory quantization at intermediate times.*

Continuous diffusion generalizes looped transformer. It is known that looped transformers are already partially more expressive than CoT (Saunshi et al., 2025; Merrill & Sabharwal, 2025; Fan et al., 2024). We now further show that a continuous diffusion, in principle, can *simulate* any looped transformer with the same architecture and parameter count, hence is at least as expressive as the powerful looped transformer (and potentially even more expressive).

Proposition 2 (Continuous diffusion sampler can simulate looped rollouts)**.** *Fix any looped transformer Φ_θ and any roll out times $T \in \mathbb{N}$, there exists a continuous diffusion sampler for the reverse PF–ODE by the explicit Euler method with step size $1/T$ that exactly reproduces the looped rollout.*

The crucial intuition is one can always construct a reverse PF–ODE with grid outpoints matching the looped transformer roll outs, and is realizable with a denoising network with the same architecture and parameter budget as the looped transformer. The contrary does not hold: a deterministic looped rollout cannot simulate any non-degenerate stochastic path produced by a continuous diffusion with $g_t > 0$ in the reverse SDE, more details refer to Theorem 9. In addition, Xu & Sato (2024) proves that timestep embeddings improve expressiveness, while standard looped transformers tend to discard.

3.2 Empirical Performance and Practical Trainability

Intriguingly, despite their strong theoretical expressiveness, previous looped transformers tend to exhibit limited empirical performance compared with SOTA LLMs. Meanwhile, continuous DLMs typically underperform their discrete counterparts, contradicting to the expressivity inclusion. In this subsection, we analyze these empirical observations through the lens of practical trainability.

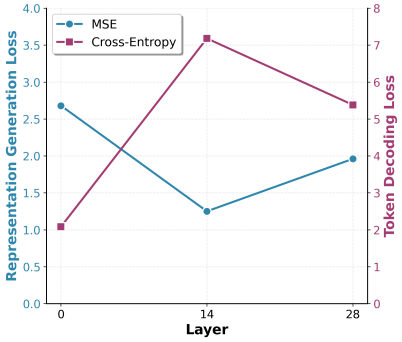


Figure 2: Comparison of validation losses when using representations from different layers of Qwen3-Embedding-0.6B as the latent spaces for CDMs.

Advantages of intermediate supervision. We point out a drawback in classical looped transformer training: they are typically trained as standard transformers (i.e., depth $T = 1$), or trained with a fixed depth and only supervised on the last roll out. Consequently, LT would encounter out-of-distribution (OOD) problems when rolled out with a different time from training, since supervisions on the intermediate states of these depths are never received.

Fortunately, continuous diffusion models naturally address this problem. During training, all continuous time instances (or sufficiently dense discrete timesteps) would be sampled and supervised by denoising loss, so the model is able to model all intermediate timesteps along the probability path. The progressively denoising parameterization also enables flexible number of function evaluations (NFEs) and the sampling timesteps in inference, which is hard for looped transformers without sophisticated special design for this purpose. Combining the advantage brought by **intermediate supervision** and the theoretical expressivity inclusion in Proposition 2, we conclude that instead of LTs, one can actually train CDMs which are expressive and easier to optimize.

Limitations in trainability of continuous diffusion. While theoretically expressive, previous CDMs typically underperform their discrete counterparts and AR LLMs in practice, necessitating us to rethink the reasons behind. In addition to the gap in engineering efforts, we argue that there are some fundamental challenges of existing continuous diffusion, including the larger decision space, the ambiguity and combinatorial complexity in decoding, and the deficient generation space.

We summarize three generation spaces for CDMs in Table 1 and leave more rigorous definitions and detailed discussions to Appendix E. In fact, all of them adhere **larger decision spaces** compared with DDMs, which brings (perhaps in fact marginal) expressiveness gain yet incur harder optimization.

Table 1: Comparison between generation space of continuous diffusion.

	Simplex Δ^{V-1}	**Token-wise** $\mathbb{R}^d$	**Contextualized** $\mathbb{R}^d$
Dimensionality	$V-1$ (high)	$d \leq V$ (often $\ll V$)	$d \leq V$ (often $\ll V$)
Geometry	Constrained manifold	Euclidean; codebook cells	Euclidean; contextual manifold
Target smoothness	Low (near vertices)	Atomic, non-smooth	Higher (good embedding models)
Calibration	Natural	Requires decoder	Requires decoder (context)
Expressivity (terminal)	Baseline	$\not\geq$simplex (Prop. 17)	$\geq$ simplex if decoder strong
Decoding ambiguity	Low	Medium (NN/energy)	High if not sufficient
Optimization	Hard (constraints)	Boundary brittle	Complex but smoother targets

In particular, the probability simplex adopted by Han et al. (2022); Sahoo et al. (2025) is usually **high-dimensional** with hard constraints for valid distributions. Token-wise embedding space is the most common choice of early CDMs (Gong et al., 2022; Li et al., 2022), which we argue is a **deficient representation space** as it is not more expressive than the simplex with dimension $d \leq V$ (Proposition 17). Moreover, the generation target is the atomic codebook representations (essentially a set in $\mathbb{R}^d$, such as the first embedding layer of a LM), posing difficulty for a continuous diffusion to generate. By contrast, contextualized embeddings (where the token embeddings depends on the contexts in the sequence, such as hidden features in LLMs) provide more semantic information of the contexts and serve as a smoother generation target – especially for those high-quality representations from well-pretrained LLMs. However, the complicated and ambiguous contextualized embeddings in turns present more difficulties in **decoding featuring combinatorial complexity**. The analysis above is supported by quantitative experimental results demonstrated in Figure 2: utilizing the 0-th layer of Qwen3-Embedding as the generation space (which produces essentially token-wise embeddings) results in the smallest cross-entropy but the largest generation MSE, while the 28-th layer (which gives contextualized embeddings) lead to moderately small MSE and larger classification loss.

4 Coevolutionary Continuous Discrete Diffusion

Based on the insights in Section 3, we aim to **bridge the gap between expressivity and learnability**. It is natural to ask: can we overcome the dilemma so that one could leverage the powerful embedding space while keep it decodable? In this section, we give a positive answer by combining the continuous representation space with discrete diffusion. Remarkably, the discrete state reduces the uncertainty and complexity of input space, making continuous space easier to optimize and decode. The continuous space enlarges expressivity upper bound, and those representations from well-pretrained LLMs significantly improve the generation quality.

4.1 Joint Continuous-Discrete Diffusion

Now we introduce Coevolutionary Continuous Discrete Diffusion (CCDD), a diffusion model on the joint of discrete and continuous space $\mathcal{X} \times \mathcal{Z}$ (Figure 3). In particular, we consider a *joint* corruption process $(x_t, z_t) \in \mathcal{X} \times \mathcal{Z}$ that applies noise individually to each component, and a denoising process that *conditions* on both (x_t, z_t) but updates each component with its own modality-specific rule, i.e., both forward and backward process are parametrically factored.

Forward process. Let the forward law be the *product* of a CTMC on $\mathcal{X}$ and an SDE on $\mathcal{Z}$, both possibly time-inhomogeneous, and independent *conditional on* (x_0, z_0), which gives the factored conditional forward kernels:

$$q_t(x_t, z_t \mid x_0, z_0) = q_t^{\text{disc}}(x_t \mid x_0)\, q_t^{\text{cont}}(z_t \mid z_0). \tag{13}$$

For instance, considering the representative forward process for continuous diffusion in equation 2 and discrete diffusion in equation 7, the corresponding (x_t, z_t) follows:

$$z_t \sim \mathcal{N}(\alpha_t z_0, \sigma_t^2 I), \quad x_t \sim \text{Cat}(\eta_t x_0 + (1-\eta_t)\pi_t). \tag{14}$$

Reverse process. A single time–conditioned network $f_\theta(\cdot, t)$ receives inputs (x_t, z_t) and outputs modality-specific heads. We adopt the following *factored* reverse update at a time step $t \to s$ $(s < t)$, conditioned on the *multimodal pair* (x_t, z_t):

$$p_\theta(x_s, z_s \mid x_t, z_t) \;=\; p_\theta^{\text{disc}}(x_s \mid x_t, z_t)\; p_\theta^{\text{cont}}(z_s \mid x_t, z_t). \tag{15}$$

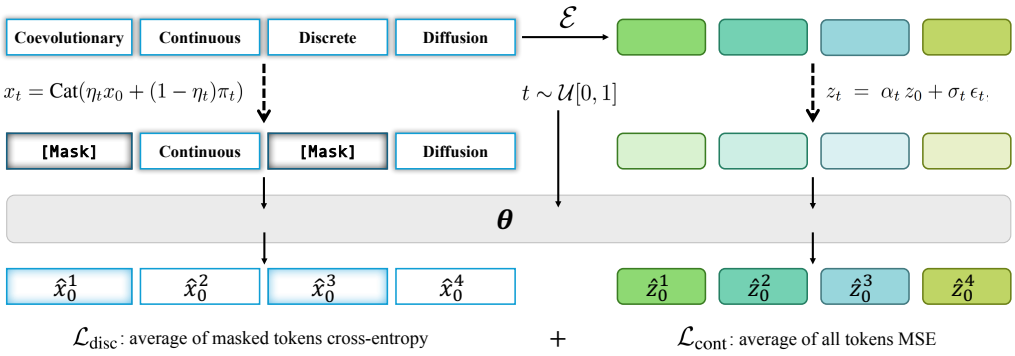


Figure 3: Framework of Coevolutionary Continuous Discrete Diffusion.

For simplicity, we illustrate with x_0-prediction, while other standard parameterizations such as ϵ-prediction and v-prediction are equivalent. While the estimation for each modality depends on both states, i.e., $\hat{x}_{0,\theta} = \hat{x}_{0,\theta}(x_t, z_t, t)$, $\hat{z}_{0,\theta} = \hat{z}_{0,\theta}(x_t, z_t, t)$, the following updates are carried on separately based on the original rules such as DDPM (equation 3) or DDIM (equation 4) for z_t and the Bayes posterior (equation 8) for x_t; see Algorithm 2 for algorithmic description.

Intuitively, this reverse process combines the *radical* discrete decoder with high confidence, and the *conservative* continuous decoder with full uncertainty information. Thanks to the continuous component, CCDD is able to preserve full semantics in previous denoising steps and later leverage these historical information, which would mostly be discarded by masked DDM. CCDD is also capable of striking the balance between inference efficiency through few-step ODE sampling, and generation quality through test-time scaling with SDE.

Based on the established ELBOs for continuous and discrete diffusion, we calculate the loss for two modalities according to equation 5 and equation 9, respectively. As illustrated in Algorithm 1, the CCDD training loss is a (weighted) sum of two losses:

$$\mathcal{L}_{\text{CCDD}} = \gamma_{\text{cont}} \cdot \mathcal{L}_{\text{cont}} + \gamma_{\text{disc}} \cdot \mathcal{L}_{\text{disc}} \tag{16}$$

Remark 2 (Conditioning vs. factorization)**.** *Although equation 15 factorizes the* kernel *at each step, each factor is allowed to depend on* both *inputs* (x_t, z_t)*. Thus cross-modal coupling is injected via conditioning:* x*-updates see* z_t *and vice versa. In other words, the factorization provides an efficient parameterization without making* x_t *and* z_t *to be independent in the reverse process. In fact, the factored forward processes admit semigroups (Theorem 11), and the factored reverse kernels with sufficiently small steps yield aymptotically the same expressivity as fully coupled kernels (Theorem 12).*

4.2 Implementation Techniques

Architecture design. Based on DiT (Peebles & Xie, 2023), MM-DiT (Esser et al., 2024), and MoE (Shazeer et al., 2017), we design several effective architectures (termed MDiT, MMDiT, and MoEDiT) for joint denoising with various parameter and computation efficiency (Figure 4). More details of parameters and computation complexity are available in Appendix C.1.

Selection of continuous space: representation learning perspective. Based on the above analysis, we select contextualized embedding space obtained from well-pretrained LLM-based text encoders, such as Qwen3-Embedding (Zhang et al., 2025). The *contextualized* embeddings provides rich sequence-level semantics and is easier to generate, while the discrete diffusion helps in decoding. Moreover, from a representation learning perspective, the high-quality latents serve as representation regularization that accelerates the convergence of training (Yu et al., 2024; Wang et al., 2025), and also the high-level conditioning or guidance in inference (Li et al., 2024a; Kouzelis et al., 2025). The procedure of continuous diffusion learning to reconstruct the high-quality representations in pretrained LLMs can also be viewed as a sort of knowledge distillation.

Classifier-free guidance. The representation-enhanced diffusion training perspective views the continuous representations as the self-generated guidance for the discrete token generation. Then

Table 2: Validation perplexity on LM1B. We use Qwen3-Embedding-0.6B as the continuous generation space for CCDD, and reimplement the baselines with the same Qwen-2 tokenizer. The numbers of trained tokens and non-embedding parameter counts are also reported for fair comparison. CCDD with the same number of parameters significantly outperforms the discrete-only MDLM baseline.

Model	Train. toks.	# params.	Validation PPL (↓)
MDLM (Sahoo et al., 2024) (reimpl.)	33B	92.1M	≤ 39.17
CCDD-MDiT w/ Qwen3 (ours)	33B	92.1M	≤ 29.22
CCDD-MMDiT w/ Qwen3 (ours)	33B	216.2M	≤ 25.76

anologously to classifier-free guidance (CFG) (Ho & Salimans, 2022), we treat the dual-modality forward as the conditional model (whose output is written as logits_c), and the discrete-only forward as the unconditional model (whose output is denoted as logits_ϕ). In training, we randomly drop continuous embeddings with probability p_{drop} by zero-in and zero-out all continuous tokens within the sample, so that the model forward with only discrete states. At inference time, the logits per-step with CFG are computed as $\text{logits} = w \cdot \text{logits}_c + (1 - w) \cdot \text{logits}_\phi$ with the guidance scale w.

Asynchronous noise schedule. In addition to approximately synchronous signal-noise ratios in two modalities by adjusting the coefficients α_t and η_t, we propose asynchronous noise schedules of the forward process in light of the "representation guidance" spirit. We set the information decay rate in continuous space slower than the discrete modality, so that in the reverse process the model would generate the latent representations faster (playing the role of implicit planning and reasoning), which serve as the high-level guidance for token decoding. More details are available in Appendix B.2.2.

To summarize, CCDD is a novel language modeling regime that combines multimodal spaces to generate unimodal texts. CCDD generalizes CDM, DDM and LT, featuring high expressivity. The learnability is also improved through synergy: the discrete component actually provides auxiliary capabilities for decoding, and the continuous components also have practical engineering benefits from the representation learning and knowledge distillation perspective.

5 Experiments

Experimental setup. We pretrain our models on the widely used One Billion Words Dataset (LM1B) (Chelba et al., 2013) and OpenWebText (OWT) (Gokaslan & Cohen, 2019) dataset, following most settings in prior work (von Rütte et al., 2025; Shi et al., 2024; Lou et al., 2023; Sahoo et al., 2024). For LM1B, we use the standard split, and train models using sequence length $L = 128$ with sentence packing. For OWT, following Sahoo et al. (2024); von Rütte et al. (2025), we reserve the last 100K documents as the validation set, and adopt sequence length $L = 512$ with sentence packing. Instead of bert-base-uncased tokenizer on LM1B and GPT-2 (Radford et al., 2019) tokenizer on OWT, for both datasets we use GPT-2 tokenizer when train CCDD with RoBERTa, and use Qwen-2 (Yang et al., 2024a) tokenizer (also adopted by Qwen-3 series of models) when leverage Qwen3-Embedding (Zhang et al., 2025) representations. Notably, perplexity calculated with different vocabulary sizes are not comparable: Qwen-2 tokenizer has approximately $3\times$ vocabulary size compared with GPT-2, naturally resulting in larger ELBO and perplexity – we thus reproduce the baselines with the same tokenizer. We develop our transformer architectures MDiT, MMDiT, and MoEDiT (detailed in Appendix C.1) based on Lou et al. (2023) with the same configurations when plausible, which augments DiT (Peebles & Xie, 2023) with rotary embeddings (Su et al., 2024). All models are trained for 1M steps with batch size 512 on both datasets, corresponding to 33B tokens on LM1B and 131B tokens on OWT. More details and additional results are deferred to Appendix D.

Main results. The results on LM1B are reported in Table 2. With the help of the powerful Qwen3-Embedding representations, CCDD reduces validation perplexity by over 25% compared with MDLM baseline using the same number of parameters. Scaling the number of parameters via architectural improvement further enhances the performance effectively.

Shown in Table 3 are the results on OWT, which demonstrate that even simple RoBERTa embeddings could benefit CCDD training. By switching to the well-pretrained Qwen3-Embedding space and

Table 3: Validation perplexity on OWT with GPT-2 tokenizer or Qwen-2 tokenizer (marked with $*$). CCDD is trained with RoBERTa-base or Qwen3-Embedding-0.6B embeddings. The numbers of trained tokens and non-embedding parameter counts are also reported for fair comparison.

Model	Train. toks.	# params.	Validation PPL (↓)
GPT2 (Radford et al., 2019)†	unk.	117M	23.40
Llama110M (retrain.)†	262B	110M	16.11
SEDD (Lou et al., 2023)†	262B	92.1M	≤ 24.10
MDLM (Sahoo et al., 2024) (reimpl.)	131B	92.1M	≤ 27.39
GIDD+ (von Rütte et al., 2025) (reimpl.)	131B	92.1M	≤ 25.82
CCDD-MoEDiT w/RoBERTa (ours)	131B	104.0M	≤ 24.56
MDLM (reimpl.) $*$	131B	92.1M	≤ 33.78
CCDD-MMDiT w/Qwen3 (ours) $*$	131B	216.2M	≤ 27.24

scaling the parameters, CCDD reveals larger advantages. Remarkably, results trained with Qwen-2 tokenizer (marked with $*$) are not directly comparable to those with GPT-2 tokenizer.

To measure the inference-time flexibility, we report the generative NLLs of CCDD samples with inference-time CFG in Table 4. We generate 256 samples with sequence length 512 using 512 denoising steps parameterized by DDPM and MDLM reverse process. We use GPT2-Large as the reference model, and the generative perplexity is calculated as the exponential function of the NLL. With either discrete-only forward (guidance scale $w = 1$) or standard joint forward ($w = 0$), CCDD has superior performance, while CFG further improves the quality, verifying the advantages of latent reasoning.

Table 4: Generative NLLs of CCDD with Qwen3-Embedding using inference-time CFG after pretraining on OWT with Qwen-2 tokenizer.

Model	w	Gen. NLL (↓)
MDLM	-	9.19
CCDD-MoEDiT	0.0	9.06
CCDD-MoEDiT	1.0	8.38
CCDD-MoEDiT	1.5	**8.25**

Ablation studies. As shown in Figure 2, we compare the validation losses when using representations from different layers of Qwen3-Embedding-0.6B as the latent spaces for continuous diffusion models. The losses consist of two components: (i) the representation MSE loss, which we use to measure the difficulty in generating the target representations; and (ii) the token decoding cross-entropy loss, which measures the difficulty in decoding the generated latents into discrete tokens. All the representations are normalized, with a hidden dimension of 32. All models leverage the DiT architectures and are trained on LM1B for 500k steps with same configurations. The results validate our hypothesis: the 0-th layer (token-wise) corresponds to the smallest token loss but the largest representation loss, while the 28-th layer (contextualized) admits moderate losses, striking a balance between the generativity and decodability. We refer readers to Appendix D for more ablation studies.

Downstream benchmarks. We evaluate the general performance of our language model for language understanding based on the commonly adopted lm-eval evaluation suite. We follow the benchmark suite choice of von Rütte et al. (2025), which covers a diverse range of tasks regarding general language understanding and question answering capability and can convincingly showcase the performance comparison among different models. More details are in Section D.4.

As shown in Table 5, CCDD (w/ mmdit architecture and Roberta representations) outperforms all discrete diffusion baselines in all six benchmarks, and achieves comparable or better performance compared with AR models. The results rigorously validate the advantages of CCDD over traditional MDM baselines. Notably, this CCDD model is simply trained with Roberta, yet still surpasses all discrete diffusion and some AR models - we do not report CCDD with stronger Qwen3-Embedding again due to incomparable ELBOs caused by different tokenizers.

Table 5: Zero-shot benchmark accuracy on downstream datasets.

Model	ARC-e	ARC-c	BoolQ	PIQA	OBQA	WinoG	Avg.
GPT2	**43.81**	19.03	48.72	**62.89**	16.40	**51.62**	40.41
Llama (retrain)	40.53	25.51	46.21	62.73	**28.40**	50.57	**42.35**
MDLM	28.37	23.63	49.42	52.50	22.00	49.41	37.55
GIDD	26.73	23.55	50.43	51.85	26.60	49.49	38.11
CCDD (ours)	28.75	**25.94**	**51.10**	54.41	26.60	49.96	39.46

Complex reasoning: Sudoku, SAT, and Countdown. For complex mathematical reasoning tasks, we select Sudoku, SAT and Countdown. We adopt the datasets and experimental settings in Ye et al. (2024a). By default, we train 6M models of different kinds from scratch, keeping hyperparameters consistent with Ye et al. (2024a). Details of the tasks and datasets are available in Section D.4.

Table 6: Test accuracy on complex reasoning datasets: Sudoku, 3-SAT, and Countdown.

Model	Size	Depth T	Sudoku	3-SAT	Countdown
GPT2 scratch	6M	Seq. Length	16.2	73.1	31.9
GPT2 scratch	303M	Seq. Length	19.4	-	41.3
Llama	7B	Seq. Length	27.1	-	41.1
Llama	13B	Seq. Length	33.8	-	51.1
MDM	6M	2	88.6	39.2	-
MDM	6M	20	99.9	87.0	52.0
LT	6M	2	100.0	91.3	60.6
LT	6M	3	100.0	-	68.2
CCDD (ours)	6M	2	100.0	**91.9**	67.8
CCDD (ours)	6M	3	100.0	-	**73.7**

As shown in Table 6, the results on all three datasets consistently show that while MDM is more powerful than AR models (including the large pretrained LLaMA), looped transformer (LT) with even 2 looped depth have better performance, and **CCDD with just two steps is sufficient to beat all models**. The empirical observations are consistent with our theory: LT is expressive through latent reasoning, and scaling depth is more effective than scaling CoT steps as in AR or increasing discrete diffusion steps in MDM. The results also validate that CCDD is at least as expressive as the better one within MDM and LT, and the joint design brings even further performance gain. Therefore, CCDD benefit from both any-order generation capability of MDM, and the strong papalism/deep effective depth of LT. Furthermore, CCDD is also the most efficient one: with the same architectures and parameter counts, it features more efficient training than LT who needs to roll out all depths in training, and fewer sampling steps suffices compared with MDM.

6 Conclusion

The contributions of the paper are sumarized as follows. Theoretically, we systematically analyze mainstream language modeling regimes through the lens of expressivity and trainability. We conclude that under same computation conjecture continuous diffusion dominates discrete diffusion, while being able to simulate looped transformer. However, although CLM overcomes the OOD problem of LT, it still lacks trainability due to several fundamental limitations. Methodologically, we introduce Coevolutionary Continuous Discrete Diffusion (CCDD), which defines a joint diffusion process on both continuous and discrete space and levarages a single model to jointly denoise. CCDD is a novel language modeling scheme that retains both strong expressivity and trainability. We also present effective architectures as well as advanced training and sampling techniques. Experimentally, pretrained CCDD on real-world datasets LM1B and OWT reveals significantly lower validation perplexity compared with baselines, and superior performance in downstream benchmarks and complex reasoning datasets further validates the strength of latent reasoning.

REFERENCES

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

Alan N Amin, Nate Gruver, and Andrew Gordon Wilson. Why masking diffusion works: Condition on the jump schedule for improved discrete diffusion. *arXiv preprint arXiv:2506.08316*, 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, 2021.

Sangmin Bae, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Seungyeon Kim, and Tal Schuster. Relaxed recursive transformers: Effective parameter sharing with layer-wise lora. *arXiv preprint arXiv:2410.20672*, 2024.

Sangmin Bae, Yujin Kim, Reza Bayat, Sungnyun Kim, Jiyoun Ha, Tal Schuster, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Aaron Courville, et al. Mixture-of-recursions: Learning dynamic recursive depths for adaptive token-level computation. *arXiv preprint arXiv:2507.10524*, 2025.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. *arXiv preprint arXiv:1911.11641*, 2019. doi: 10.48550/arXiv.1911.11641. URL https://arxiv.org/abs/1911.11641.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*, 2024.

Chen-Hao Chao, Wei-Fang Sun, Hanwen Liang, Chun-Yi Lee, and Rahul G Krishnan. Beyond masked and unmasked: Discrete diffusion models via partial masking. *arXiv preprint arXiv:2505.18495*, 2025.

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.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Jeffrey Cheng and Benjamin Van Durme. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv preprint arXiv:2412.13171*, 2024.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of NAACL-HLT*, pp. 2924–2936, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018a.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018b.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, et al. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022.

Yilun Du, Jiayuan Mao, and Joshua B Tenenbaum. Learning iterative reasoning through energy diffusion. *arXiv preprint arXiv:2406.11179*, 2024.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Ying Fan, Yilun Du, Kannan Ramchandran, and Kangwook Lee. Looped transformers for length generalization. *arXiv preprint arXiv:2409.15647*, 2024.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37: 133345–133385, 2024.

Tomas Geffner, Kieran Didi, Zhonglin Cao, Danny Reidenbach, Zuobai Zhang, Christian Dallago, Emine Kucukbenli, Karsten Kreis, and Arash Vahdat. La-proteina: Atomistic protein generation via partially latent flow matching. *arXiv preprint arXiv:2507.09466*, 2025.

Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In *International Conference on Machine Learning*, pp. 11398–11442. PMLR, 2023.

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

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*, 2022.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. Diffuseq-v2: Bridging discrete and continuous text spaces for accelerated seq2seq diffusion models. *arXiv preprint arXiv:2310.05793*, 2023.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, et al. Scaling diffusion language models via adaptation from autoregressive models. *arXiv preprint arXiv:2410.17891*, 2024.

Shansan Gong, Ruixiang Zhang, Huangjie Zheng, Jiatao Gu, Navdeep Jaitly, Lingpeng Kong, and Yizhe Zhang. Diffucoder: Understanding and improving masked diffusion models for code generation. *arXiv preprint arXiv:2506.20639*, 2025.

Halil Alperen Gozeten, M Emrullah Ildiz, Xuechen Zhang, Hrayr Harutyunyan, Ankit Singh Rawat, and Samet Oymak. Continuous chain of thought enables parallel exploration and reasoning. *arXiv preprint arXiv:2505.23648*, 2025.

Ishaan Gulrajani and Tatsunori B Hashimoto. Likelihood-based diffusion language models. *Advances in Neural Information Processing Systems*, 36:16693–16715, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. Ssd-lm: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. *arXiv preprint arXiv:2210.17432*, 2022.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and P. Abbeel. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. *arXiv preprint arXiv:2502.06768*, 2025.

Theodoros Kouzelis, Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Boosting generative image modeling via joint image-feature synthesis. *arXiv preprint arXiv:2504.16064*, 2025.

Tianhong Li, Dina Katabi, and Kaiming He. Return of unconditional generation: A self-supervised representation generation method. *Advances in Neural Information Processing Systems*, 37: 125441–125468, 2024a.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35: 4328–4343, 2022.

Zian Li, Cai Zhou, Xiyuan Wang, Xingang Peng, and Muhan Zhang. Geometric representation condition improves equivariant molecule generation. *arXiv preprint arXiv:2410.03655*, 2024b.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 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.

Rabeeh Karimi Mahabadi, Hamish Ivison, Jaesung Tae, James Henderson, Iz Beltagy, Matthew E Peters, and Arman Cohan. Tess: Text-to-text self-conditioned simplex diffusion. *arXiv preprint arXiv:2305.08379*, 2023.

William Merrill and Ashish Sabharwal. A little depth goes a long way: The expressive power of log-depth transformers. *arXiv preprint arXiv:2503.03961*, 2025.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of EMNLP*, pp. 2381–2391, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

Amirkeivan Mohtashami, Matteo Pagliardini, and Martin Jaggi. Cotformer: A chain-of-thought driven architecture with budget-adaptive computation cost at inference. *arXiv preprint arXiv:2310.10845*, 2023.

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.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI*, 2019. Accessed: 2024-11-15.

Kevin Rojas, Yuchen Zhu, Sichen Zhu, Felix X-F Ye, and Molei Tao. Diffuse everything: Multimodal diffusion models on arbitrary state spaces. *arXiv preprint arXiv:2506.07903*, 2025.

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.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*, 2019. doi: 10.48550/arXiv.1907.10641. URL `https://arxiv.org/abs/1907.10641`.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J Reddi. Reasoning with latent thoughts: On the power of looped transformers. *arXiv preprint arXiv:2502.17416*, 2025.

Neta Shaul, Itai Gat, Marton Havasi, Daniel Severo, Anuroop Sriram, Peter Holderrieth, Brian Karrer, Yaron Lipman, and Ricky TQ Chen. Flow matching with general discrete paths: A kinetic-optimal perspective. *arXiv preprint arXiv:2412.03487*, 2024.

Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Z Shen, H Yan, L Zhang, Z Hu, Y Du, and Y Codi He. Compressing chain-of-thought into continuous space via self-distillation. *arXiv preprint arXiv:2502.21074*, 2025.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ArXiv*, abs/2010.02502, 2020.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

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.

Dimitri von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. *arXiv preprint arXiv:2503.04482*, 2025.

Chenyu Wang, Cai Zhou, Sharut Gupta, Zongyu Lin, Stefanie Jegelka, Stephen Bates, and Tommi Jaakkola. Learning diffusion models with flexible representation guidance. *arXiv preprint arXiv:2507.08980*, 2025.

Xinyi Wang, Lucas Caccia, Oleksiy Ostapenko, Xingdi Yuan, William Yang Wang, and Alessandro Sordoni. Guiding language model reasoning with planning tokens. In *First Conference on Language Modeling*, 2024.

Kevin Xu and Issei Sato. On expressive power of looped transformers: Theoretical analysis and enhancement via timestep encoding. *arXiv preprint arXiv:2410.01405*, 2024.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin,

Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024a.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024b.

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

Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning. *arXiv preprint arXiv:2410.14157*, 2024a.

Jiacheng Ye, Shansan Gong, Liheng Chen, Lin Zheng, Jiahui Gao, Han Shi, Chuan Wu, Xin Jiang, Zhenguo Li, Wei Bi, et al. Diffusion of thought: Chain-of-thought reasoning in diffusion language models. *Advances in Neural Information Processing Systems*, 37:105345–105374, 2024b.

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.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Cai Zhou, Chenyu Wang, Dinghuai Zhang, Shangyuan Tong, Yifei Wang, Stephen Bates, and Tommi Jaakkola. Next semantic scale prediction via hierarchical diffusion language models. *arXiv preprint arXiv:2510.08632*, 2025.

Hanlin Zhu, Shibo Hao, Zhiting Hu, Jiantao Jiao, Stuart Russell, and Yuandong Tian. Reasoning by superposition: A theoretical perspective on chain of continuous thought. *arXiv preprint arXiv:2505.12514*, 2025a.

Rui-Jie Zhu, Tianhao Peng, Tianhao Cheng, Xingwei Qu, Jinfa Huang, Dawei Zhu, Hao Wang, Kaiwen Xue, Xuanliang Zhang, Yong Shan, et al. A survey on latent reasoning. *arXiv preprint arXiv:2507.06203*, 2025b.

A Related Work

Looped transformers and latent CoT. Recent research has explored how enable a model with a fixed number of layers to think deeper about problems through architectural design or specialized training, effectively simulating a deeper transformer (Zhu et al., 2025b). A fundamental strategy in this direction involves loop-based architectures (Dehghani et al., 2018b; Mohtashami et al., 2023; Bae et al., 2024). For instance, Fan et al. (2024) show that standard fixed-depth transformers struggle in length generalization which can be significantly improved by looped transformers. Saunshi et al. (2025) first proves that a T-depth looped transformer can implicitly generate latent thoughts and can simulate T steps of CoT reasoning under mild assumptions. Furthermore, Merrill & Sabharwal (2025) shows that a looped transformer with depth $\Theta(\log n)$ can solve regular language recognition and graph connectivity with arbitrary input context length n, which is intractable by LLMs with logarithmic CoT steps. Mixture of Recursion (Bae et al., 2025) is the recent looped transformer that practically works, which adaptively adjust the looping depth for tokens.

In contrast to architectural recurrence, which necessitates explicit structural changes, an alternative known as *continuous chain-of-thought (continuous CoT)* achieves comparable computational advantages through specialized training of standard transformer models (Hao et al., 2024; Shen et al., 2025; Cheng & Van Durme, 2024; Wang et al., 2024). A representative is COCONUT (Hao et al., 2024), which operates on the continuous token space instead of using recurrent parameters. COCONUT directly treats the last hidden state of previous tokens as reasoning tokens for CoT reasoning, allowing them to explore multiple reasoning paths simultaneously, akin to breadth-first search, without being constrained to natural language tokens (Gozeten et al., 2025; Zhu et al., 2025a). Continuous CoT outperforms standard discrete CoT in certain reasoning tasks demanding parallel searching and multiple reasoning paths, yet still falls behind in general tasks. Notably, latent CoT could be simulated by continuous diffusion with diffusion forcing (Chen et al., 2024), and we focus on looped transformer in the main context for clarity.

Diffusion language models. Diffusion language models emerge as a new paradigm that reformulates text generation as an iterative denoising process, enabling complex reasoning by leveraging full-sequence context. This paradigm primarily includes *Masked Diffusion Models (MDMs)*, which are a type of DDMs, and *Embedding-based Diffusion Models (EDMs)*, a subset of CDMs.

MDMs operate on discrete tokens, starting from a masked sequence and refining tokens simultaneously using bidirectional context. Early foundational work includes D3PM (Austin et al., 2021) and SEDD (Lou et al., 2023), which introduced discrete transition processes and score matching losses. Subsequent methods (Ou et al., 2024; Sahoo et al., 2024; Shi et al., 2024) streamlined training through hybrid masked losses, facilitating the conversion of encoder models like BERT into generative reasoners. The iterative unmasking process inherent in MDMs supports sophisticated reasoning capabilities, such as iterative refinement (Du et al., 2024) and reverse-order reasoning (Nie et al., 2025). The framework has also been integrated with chain-of-thought reasoning (Ye et al., 2024b), demonstrating strong performance in tasks requiring parallel context and systematic refinement. Similar algorithms are proposed from the flow matching perspective (Gat et al., 2024). Additional to mask noises, some work try to leverage uniform noises which tend to have worse performance (von Rütte et al., 2025; Shaul et al., 2024). Another series of work extend intermediate states by introducing partially noised states in between mask and clean tokens (Zhou et al., 2025; Chao et al., 2025), for example, HDLM (Zhou et al., 2025) leverages hierarchies of semantics for each token, where lower-level detailed tokens are noised into higher-level abstract tokens (such as cluster tokens) in the forward process, and the model progressively denoises by predicting the next semantic scale in the reverse process. MDM-Prime Chao et al. (2025) extends the masking state by converting each word token into several subtokens and gradually mask the subtokens. Notably, these methods still operate in the discrete state spaces.

In contrast, EDMs perform diffusion in a continuous embedding space. EDM research focused on controllable generation (Li et al., 2022) and sequence-to-sequence tasks (Dieleman et al., 2022; Mahabadi et al., 2023; Gong et al., 2022), with Plaid (Gulrajani & Hashimoto, 2023) later establishing empirical scaling laws that significantly narrowed the efficiency gap with autoregressive models. The framework was further extended by DoT-Plaid (Ye et al., 2024b), which generalized chain-of-thought reasoning to EDMs, leveraging iterative latent refinement for improved coherence and mathematical

reasoning. There are also a few continuous diffusion models operating on the logit space Han et al. (2022); Sahoo et al. (2025).

Some previous work have noticed the potential of multimodal generation integrating continuous and discrete diffusion, with applications to text-image joint generation (Rojas et al., 2025) and protein sequence-structure co-design (Campbell et al., 2024). DUO (Sahoo et al., 2025) tries to connect two types of diffusion models via marginal matching, and apply distillation tricks for continuous diffusion to discrete text diffusion. In the context of diffusion language models, Diffuseq-v2 (Gong et al., 2023) also tries to bridge continuous and discrete generation process by introducing a soft absorbing state. In comparison, our work generalize these results, propose a more principled method for joint continuous-discrete modeling, and provide systematic analysis on expressiveness and trainability. We also practically showcase that combining continuous and discrete models to benefit each other, and the powerful latent reasoning could significantly improve the expressivity and performance in complex reasoning tasks such as Sudoku.

Representation learning for diffusion models. Recent advances in representation-enhanced diffusion model training show that high-quality representations from pretrained models could benefit the training efficiency and sampling quality of diffusion models through flexible ways (Li et al., 2024a;b; Yu et al., 2024; Wang et al., 2025; Kouzelis et al., 2025). In particular, RCG (Li et al., 2024a) and GeoRCG (Li et al., 2024b) adopt two-stage generation processes where a representation generator first samples high-level features which serve as the conditions for the second-stage image or molecule generation. In contrast, REPA (Yu et al., 2024) is a training-time technique that aligns the internal features of diffusion models with external pretrained representations, thereby accelerating the training procedure. REED (Wang et al., 2025) unifies RCG and REPA from a theoretical perspective, and generalizes these methods by leveraging multi-modal representations and improved training curriculum. Furthermore, ReDi (Kouzelis et al., 2025) demonstrates that generating images and their representations at the same time also boosts generation quality of diffusion models.

B Omitted Proof

B.1 Theoretical Expressivity Analysis

This subsection provides proof for Section 3.1 in the main text and presents additional results. We assume standard measurability/Lipschitz conditions when needed.

First we start with the concepts of circuit-complexity classes TC^0 and TC^1

Definition 2 (TC^0)**.** *A language L is in TC^0 if it is decided by a family of Boolean circuits $\{C_n\}$ such that: (i) C_n has polynomial size; (ii) C_n has* constant depth $O(1)$*; (iii) gates are NOT, AND, OR, and unbounded-fan-in threshold gates.*

Definition 3 (TC^1)**.** *A language L is in TC^1 if it is decided by a family of Boolean circuits $\{C_n\}$ such that: (i) C_n has polynomial size; (ii) C_n has* logarithmic depth $O(\log n)$*; (iii) gates are NOT, AND, OR, and unbounded-fan-in threshold gates.*

Some basic containments are:

$$\mathsf{AC}^0 \subsetneq \mathsf{TC}^0 \subseteq \mathsf{TC}^1 \subseteq \mathsf{NC}^2 \subseteq \mathsf{P}. \tag{17}$$

The intuition is that TC^0 computes addition, multiplication, majority, parity, etc. in constant depth. TC^1 allows more power via logarithmic-depth threshold circuits, including division and iterated arithmetic. Our results show that analogously to LT, CDM and CCDD can be in class TC^1, showing the advantage of latent reasoning and continuous DLM.

B.1.1 Continuous Diffusion Dominates Discrete Diffusion

We now give rigorous definitions of decision space and representation space of CDM and DDM.

Definition 4 (Decision Space)**.** *Given any generation task and the context cond, the decision space is just the set of all possible outputs by the model parameterized with θ. For a single token being*

generated, the decision space of a discrete diffusion is clearly Ω, and that of a continuous diffusion is either Δ^{V-1} or $\mathbb{R}^d$ as already stated in Table 1. For multiple tokens with length L, the definition decision spaces can be easily extended to the combinations of single token decision spaces, i.e., Ω^L for discrete diffusion and $(\Delta^{V-1})^L$ or $(\mathbb{R}^d)^L$ for continuous diffusion.

It is straightforward that the decision space of continuous diffusion is larger than the discrete diffusion: the former is absolutely continuous w.r.t. Lebesgue measure on $\mathbb{R}^{L\times d}$ (See Lemma 4), while the latter is finitely supported by a set (See Lemma 3). Thus the continuous diffusion is naturally harder to be learned by a neural network under same capacity.

Definition 5 (Representation Space)**.** *The representation space is where the generation targets of a (latent) continuous diffusion lie in, determined by the pre-defined by the encoder $\mathcal{E}$, namely $\{\mathcal{E}(x)|x \in \Omega^L, L \in \mathbb{N}\}$.*

The encoder, as already explained in the preliminary section and Section 3.2, can be either the mapping to the probability simplex, or token-wise embedding defined by a codebook (such as a nn.Embedding layer), or a contextualized mapping through any pretrained LLMs or sentence embedding models (e.g., RoBERTa or Qwen3-Embedding used in the experiments). This is similar to image (latent) diffusion models: the representation space is usually defined by a pretrained VAE or an unsupervised learning model such as DINO.

Assumption 1 (Regularity for continuous diffusion)**.** *We assume $g_t > 0$ on a set of times of positive measure in $[0, 1]$, and a_t is such that the Fokker–Planck equation is well-posed and yields absolutely continuous marginals for $t > 0$ when starting from a distribution with a density or from any point mass convolved with the Gaussian noise of equation 1.*

Lemma 3 (Embedded discrete trajectories are finitely supported at each t)**.** *Fix any $t \in [0, 1]$. For any $\{p_t\}_{t\in[0,1]} \in \mathsf{F}_{\text{disc}}(\theta)$, the embedded marginal $q_t := \mathcal{E}_\sharp p_t \in \widetilde{\mathsf{F}}_{\text{disc}}(\theta)$ is supported on a* finite *set in $\mathbb{R}^{L\times d}$. In particular, if $\mathcal{E}$ is one-hot or any fixed finite codebook, then q_t is a finite mixture of Dirac masses in $\mathbb{R}^{L\times d}$.*

Proof. For any t, p_t is a probability vector over the finite set $\mathcal{X} = \Omega^L$ (size V^L). Hence $\text{supp}(p_t) \subseteq \mathcal{X}$ is finite. The encoder $\mathcal{E} : \mathcal{X} \to \mathbb{R}^{L\times d}$ maps each $x \in \mathcal{X}$ to a single point $\mathcal{E}(x)$, and thus the pushforward $q_t(B) = p_t(\mathcal{E}^{-1}(B))$ is supported on the finite set $\{\mathcal{E}(x) : x \in \mathcal{X}\}$. Therefore q_t is a finite atomic measure in $\mathbb{R}^{L\times d}$. □

Lemma 4 (Continuous diffusion produces absolutely continuous marginals)**.** *Under Assumption 1, for any $t > 0$, the marginal $q_t \in \mathcal{P}(\mathbb{R}^{L\times d})$ arising from equation 1 is absolutely continuous w.r.t. the Lebesgue measure on $\mathbb{R}^{L\times d}$. In the VP case equation 2, $z_t = \alpha_t z_0 + \sigma_t \epsilon$ with $\sigma_t > 0$ implies q_t is a Gaussian smoothing of the law of z_0, thus absolutely continuous.*

Proof. With $g_t > 0$ on a set of positive measure and standard regularity on a_t, the Fokker–Planck operator is (hypo)elliptic on $\mathbb{R}^{L\times d}$. Starting from any initial distribution with a density (or from a point mass, which immediately becomes smooth by Gaussian convolution when $g_t > 0$), the solution q_t admits a density for any $t > 0$. In the VP instance equation 2, z_t is an affine transformation of z_0 plus independent Gaussian noise with variance $\sigma_t^2 I$, hence q_t is the convolution of the law of $\alpha_t z_0$ with a non-degenerate Gaussian, which is absolutely continuous. □

Theorem 5 (Strict trajectory-level gap, Theorem 1 in main text)**.** *At any fixed $t \in (0, 1]$, we have the following strict inclusion*

$$\widetilde{\mathsf{F}}_{\text{disc}}(\theta) \subsetneq \mathsf{F}_{\text{cont}}(\theta) \quad \textit{as subsets of } \mathcal{P}(\mathbb{R}^{L\times d}) \tag{18}$$

Proof. By Lemma 3, each $q_t \in \widetilde{\mathsf{F}}_{\text{disc}}(\theta)$ is supported on a finite set in $\mathbb{R}^{L\times d}$. By Lemma 4, there exist (indeed, generically) $q_t \in \mathsf{F}_{\text{cont}}(\theta)$ that are absolutely continuous and thus non-atomic. No absolutely continuous distribution can be a finite atomic measure; hence $\widetilde{\mathsf{F}}_{\text{disc}}(\theta) \subsetneq \mathsf{F}_{\text{cont}}(\theta)$ as sets of possible marginals at time t. The strictness holds for any $t > 0$ with $g_t > 0$ on a set of positive measure before t. □

However, notice that the actual gap may be small, and discrete models with sufficient capacities (such as LLMs) can still approximate distributions pretty well given sufficient vocabulary size, sequence length and training data. Thus in additional to the (marginal) expressivity gain, utilization of continuous representations is also a practical consideration for improving empirical performance: the well-pretrained representations could facilitate the diffusion model training via representation alignment and guidance (Li et al., 2024a; Yu et al., 2024; Wang et al., 2025; Kouzelis et al., 2025).

We now give an information-theoretic quantification of information loss in discrete diffusion sampling. We quantify by differential entropy h for continuous logits l_θ generated by CDM, and by Shannon entropy H for categorical distributions $\mathrm{Cat}(\mathrm{softmax}(l_\theta))$ sampled by DDM.

Lemma 6 (Information loss from token sampling in discrete reverse steps)**.** *Let $\ell_\theta(x_t, t) \in \mathbb{R}^V$ denote the logits predicted at a discrete reverse step, and let the next input be the sampled token $x_{t^-} \sim \mathrm{Cat}(\mathrm{softmax}(\ell_\theta))$. Assume ℓ_θ is a continuous random vector with a non-degenerate distribution (e.g., due to data randomness). Then*

$$I(\ell_\theta;\, x_{t^-}) \;\le\; H(x_{t^-}) \;\le\; \log V \;<\; h(\ell_\theta),$$

hence the mapping $\ell_\theta \mapsto x_{t^-}$ is information-losing, and the full logit geometry is not preserved along the trajectory.

Proof. By data processing inequality for the Markov chain $\ell_\theta \to \mathrm{softmax}(\ell_\theta) \to x_{t^-}$ (followed by categorical sampling), $I(\ell_\theta; x_{t^-}) \le I(\ell_\theta; \mathrm{softmax}(\ell_\theta)) \le H(x_{t^-})$. Since x_{t^-} takes values in a finite set of size V, $H(x_{t^-}) \le \log V$. Meanwhile ℓ_θ is continuous/non-degenerate, so its (differential) entropy $h(\ell_\theta)$ can be arbitrarily large, and the discrete entropy $H(\lfloor \ell_\theta \rfloor)$ is also unbounded with quantization fineness; in particular, $h(\ell_\theta)$ is not bounded by $\log V$. Therefore $I(\ell_\theta; x_{t^-}) < h(\ell_\theta)$; the mapping is many-to-one and loses information about ℓ_θ beyond what is encoded in the sampled index. □

B.1.2 Continuous Diffusion Generalizes Looped Transformer

Assumption 2 (Mild regularity for numerical integration)**.** *Assume the reverse PF–ODE equation 4 uses a vector field $v_\theta(z,t) := a_t(z) - \frac{1}{2} g_t^2 s_\theta(z,t)$ that is globally Lipschitz in z and piecewise continuous in t. Let $\{t_k\}_{k=0}^T$ be a partition of $[0,1]$ with $\Delta t_k = t_{k+1} - t_k$ and a standard one-step method $\Psi_{\Delta t_k}(z, t_k)$ (e.g., explicit Euler) that is consistent of order ≥ 1.*

Proposition 7 (Continuous diffusion sampler can *simulate* looped rollouts, Proposition 2 in main text)**.** *Fix any looped transformer Φ_θ and any $T \in \mathbb{N}$. Define the (deterministic) sampler for the reverse PF–ODE by the explicit Euler method with step size $1/T$ and choose the vector field on grid points by*

$$v_\theta(z, t_k) \;:=\; \Phi_\theta(z) - z, \qquad k = 0, \dots, T-1.$$

Then the sampler update is

$$z_{t_{k+1}} \;=\; z_{t_k} \;+\; \Delta t\, v_\theta(z_{t_k}, t_k) \;=\; \Phi_\theta(z_{t_k}),$$

which exactly reproduces the looped rollout $h_{k+1} = \Phi_\theta(h_k)$ when we identify $h_k = z_{t_k}$.

Proof. By construction, with $\Delta t = 1/T$ and the explicit Euler method,

$$z_{t_{k+1}} = z_{t_k} + \Delta t\, v_\theta(z_{t_k}, t_k) = z_{t_k} + \tfrac{1}{T}\big(\Phi_\theta(z_{t_k}) - z_{t_k}\big).$$

If we instead scale the vector field as $v_\theta^{(\mathrm{scaled})}(z, t_k) := T\big(\Phi_\theta(z) - z\big)$ while keeping $\Delta t = 1/T$, then

$$z_{t_{k+1}} = z_{t_k} + \Delta t\, v_\theta^{(\mathrm{scaled})}(z_{t_k}, t_k) = z_{t_k} + \tfrac{1}{T} \cdot T\big(\Phi_\theta(z_{t_k}) - z_{t_k}\big) = \Phi_\theta(z_{t_k}).$$

Thus each sampler step equals one looped-transformer application. Since the construction uses the same network θ inside Φ_θ (embedded into v_θ through the formula above) and a time index via t_k, the equality holds step by step. □

Proposition 8 (Looped transformer can emulate diffusion ODE terminal maps with timestep embeddings and residual connections)**.** *Let the reverse sampler integrate the PF–ODE equation 4 with a*

one-step method $\Psi_{\Delta t_k}$ under Assumption 2. Define a looped transformer $\Phi_\theta^{\text{ode}}(\cdot; k)$ that, at step k, applies the numerical increment

$$\Phi_\theta^{\text{ode}}(z; k) \;:=\; \Psi_{\Delta t_k}(z, t_k) \;=\; z \;+\; \Delta t_k\, v_\theta(z, t_k) \;+\; \mathcal{O}(\Delta t_k^2),$$

where $v_\theta(\cdot, t_k)$ is computed by the same $f_\theta(\cdot, t_k)$ (time-conditioned). Then unrolling T steps computes the same discrete trajectory as the ODE sampler up to the integrator's local truncation error; as $T \to \infty$ (mesh size $\max_k \Delta t_k \to 0$), the terminal error vanishes by standard numerical ODE theory.

Proof. At each step k, the looped transformer block applies the map $z \mapsto \Psi_{\Delta t_k}(z, t_k)$ using $f_\theta(\cdot, t_k)$ to evaluate v_θ. Hence

$$h_{k+1} = \Phi_\theta^{\text{ode}}(h_k; k) = \Psi_{\Delta t_k}(h_k, t_k).$$

This matches the sampler's numerical update. The global error after T steps is bounded by $C \max_k \Delta t_k$ for a Lipschitz vector field (by Grönwall-type stability bounds and order-1 consistency). Taking the mesh to zero drives the terminal error to zero. □

Remark 3 (Stochastic paths and determinism)**.** *If $g_t > 0$, the reverse equation 3 yields a* distribution over trajectories. *A purely deterministic looped rollout $h_{k+1} = \Phi_\theta(h_k)$, given fixed initial h_0, cannot match a non-degenerate stochastic path law. If one augments the looped transformer with exogenous randomness (e.g., $u \sim \mathcal{N}(0, I)$ at initialization or fresh per-step noise) and allows conditioning on u at each step, terminal distributions can be matched in principle by pushing u through the unrolled network.*

Theorem 9 (Strictness vs. parity: diffusion vs. looped transformer)**.** *Under the* same *parameter budget and the "single time-conditioned network" protocol:*

- *(i)* (Deterministic ODE samplers) *If the reverse uses the PF–ODE (equation 4 with $g_t \equiv 0$) and is implemented by a standard one-step method, then continuous diffusion is* not *strictly more expressive in terms of terminal distributions: by Propositions 2 and 8, each can simulate the other's discrete rollout (up to vanishing numerical error).*
- *(ii)* (Stochastic path laws) *If $g_t > 0$ and the looped transformer is deterministic (no exogenous noise), continuous diffusion is* strictly *more expressive at the* trajectory *level (cannot match the non-degenerate stochastic path law with a deterministic map).*

Proof. (i) Proposition 2 shows a diffusion ODE sampler with Euler steps can exactly recover a looped rollout (by choosing $v_\theta^{(\text{scaled})}(z, t_k) = T(\Phi_\theta(z) - z)$). Conversely, Proposition 8 shows a looped transformer can emulate the numerical ODE integrator step map $\Psi_{\Delta t_k}$; standard numerical analysis ensures convergence of the terminal state as the mesh refines.

(ii) Suppose the diffusion reverse is an SDE with $g_t > 0$ on a set of positive measure. Then the terminal random variable Z_0 has a non-degenerate conditional distribution given Z_1 (and the Brownian path). A deterministic looped rollout $h_{k+1} = \Phi_\theta(h_k)$ with fixed h_0 is a measurable function of h_0 only and thus produces a Dirac path law; it cannot match a non-degenerate distribution over trajectories. Hence diffusion is strictly more expressive pathwise. □

Notably, if the looped model is allowed an auxiliary noise input u (initial or per-step) and time conditioning, then the unrolled map can be written as $H_T = \Gamma_\theta(u)$ for some measurable Γ_θ. For any target terminal law with a Borel probability measure on $\mathbb{R}^{L \times d}$, there exists a pushforward of a simple noise (e.g., Gaussian) that realizes it; with sufficient capacity, the looped model can approximate such a map. Thus terminal parity is achievable in principle.

Remark 4 (Timestep embeddings and intermediate supervision)**.** *As shown in (Xu & Sato, 2024), timestep embeddings provably improve the expressiveness of looped transformers, though the gain might be marginal in practice. Supervising intermediate diffusion steps (multi-t losses) does not change the representable set either; it improves optimization and inductive bias. Stochastic SDE sampling ($g_t > 0$) enlarges the* path *distribution class (Remark 3) but does not imply a strict advantage on terminal laws once exogenous noise is also allowed in looped rollouts.*

We now further discuss the superiority of CCDD over looped transformer in terms of both performance and efficiency. As shown in Table 6, while both CCDD and LT achieves perfect performance on

Sudoku, CCDD is better on 3-SAT and Countdown. Moreover, another even larger advantage of CCDD lies in its efficient supervision and scalability/flexibility. In particular, a T-depth looped transformer training requires to roll our the entire depth, bringing $T\times$ memory and time consumption (and even so cannot provide intermediate supervision), while a CCDD with T diffusion steps can fully simulate the former yet only requiring $1\times$ training cost, thanks to the efficient training formulation of diffusion models: in every iteration, we only need to sample one intermediate timestep, and the single forward pass can provide intermediate supervision. As a result, it is impossible to pretrain a looped transformer with effective depth 1000 as we do in CCDD on OWT with 32 H100 GPUs, even for the small size models. Therefore, CCDD fully covers looped transformer in performance, while being scalable to real tasks. Moreover, loop transformer must predefine the looped depth, while CCDD is flexible during both pretraining and inference using the continuous-time Markov Chain (CTMC) and continuous-time SDE/ODE framework.

We conclude this subsection (corresponding to Section 3.1 in the main text) by summarizing the comparison in expressivity as follows.

- **Discrete vs. Continuous Diffusion:** By Theorem 1, continuous diffusion strictly dominates discrete diffusion at the *trajectory* level in $\mathbb{R}^{L\times d}$ (non-atomic intermediate marginals vs. finite-atomic). Lemma 6 shows discrete per-step token sampling loses full logit information, whereas continuous samplers preserve all coordinates without mandatory quantization.
- **Continuous Diffusion vs. Looped Transformer:** Continuous diffusion samplers can *simulate* looped rollouts (Proposition 2); looped rollouts can *emulate* ODE samplers (Proposition 8). Diffusion with $g_t > 0$ is strictly more expressive *pathwise* than deterministic looped rollouts (Theorem 9(ii)).

B.2 Analysis of CCDD

This subsection corresponds to Section 4 in the main text, providing theoretical results and more analysis of CCDD.

B.2.1 Expressivity of CCDD

Expressivity comparison. We now compare the joint model against (i) *continuous diffusion alone* (on $\mathcal{Z}$) and (ii) choices of reverse parameterization.

Theorem 10 (Expressivity vs. continuous-only diffusion: joint law and marginals)**.** *Consider the hypothesis classes of terminal laws $\mathcal{H}_{\text{joint}} := \{$ laws of (x_0, z_0) produced by the joint model $\}$ and $\mathcal{H}_{\text{cont}} := \{$ laws of z_0 produced by continuous diffusion alone $\}$. Then:*

(i) On the joint space $\mathcal{X}\times\mathcal{Z}$, *$\mathcal{H}_{\text{joint}}$ strictly extends continuous-only modeling (which does not produce x_0 at all); i.e., the joint model is strictly more expressive if the target includes the discrete marginal.*

(ii) On the continuous marginal *alone, the joint model does* not *enlarge the class of z_0 laws beyond a continuous diffusion with the same parameter budget and time-conditioning: for any $P \in \mathcal{H}_{\text{joint}}$, its z-marginal P_Z lies in $\mathcal{H}_{\text{cont}}$ (up to decoder equivalence).*

Proof. (i) Trivial: continuous-only models do not define a distribution on x_0; the joint model does.

(ii) Let $P \in \mathcal{H}_{\text{joint}}$ be realized by some reverse parameterization with $f_\theta(x_t, z_t, t)$. Define a continuous-only model with inputs (z_t, t) whose network computes the same z-head as the joint model but with x_t embedded by a deterministic encoder $\mathcal{E}$ predicted from (z_t, t) (this is implementable because the joint reverse factor p_θ^{cont} depends on (x_t, z_t, t) only through a measurable function). Universal approximation guarantees allow the continuous-only network to approximate the composite map $(z_t, t) \mapsto p_\theta^{\text{cont}}(\cdot \mid x_t, z_t, t)$ after marginalizing out x_t under $q_t(x_t \mid z_t)$ (which the network can input as $\mathbb{E}[\phi(x_t) \mid z_t, t]$ for a rich feature dictionary ϕ). Therefore the induced terminal z-law can be matched. The decoder equivalence argument is standard: terminal classification/regression heads can implement the same marginal law. □

Remark 5. *Theorem 10(ii) states a* marginal *parity: adding a discrete companion does not expand the set of achievable z-marginals at the level of function classes (with time-conditioning and equal total parameters). Practically it may ease optimization by providing an auxiliary target.*

Semigroup structure and sufficient expressivity of factored parameterization. Let $q_t(x, z)$ denote the joint forward marginal induced by equation 13. Then

$$\nabla_z \log q_t(x, z) \;=\; \nabla_z \log q_t(z \mid x), \qquad \Delta_z \log q_t(x, z) \;=\; \Delta_z \log q_t(z \mid x), \tag{19}$$

i.e., the continuous score depends on x *only through* the conditional $q_t(z \mid x)$. For the discrete part, Bayes posteriors use $q_t(x \mid z) \propto q_t(x)\, q_t(z \mid x)$.

Lemma 11 (Forward semigroup and Trotter factorization)**.** *Let $\{T_t^{(z)}\}_{t\geq 0}$ and $\{T_t^{(x)}\}_{t\geq 0}$ be the Markov semigroups generated by the continuous FP operator $L_t^{(z)}$ and the discrete generator $L_t^{(x)}$ (Kolmogorov forward) respectively, both time-inhomogeneous but piecewise constant in small intervals. Then the joint forward semigroup on $\mathcal{X} \times \mathcal{Z}$ with* independent *corruption is $T_t = T_t^{(x)} T_t^{(z)} = T_t^{(z)} T_t^{(x)}$ and, for a partition $0 = t_0 < \cdots < t_N = t$ with mesh $\max_k \Delta t_k \to 0$,*

$$T_t \;=\; \lim_{\max \Delta t_k \to 0} \prod_{k=0}^{N-1} \Big(T^{(x)}_{\Delta t_k}\, T^{(z)}_{\Delta t_k} \Big) \;=\; \lim_{\max \Delta t_k \to 0} \prod_{k=0}^{N-1} \Big(T^{(z)}_{\Delta t_k}\, T^{(x)}_{\Delta t_k} \Big).$$

Proof. Independence in equation 13 implies that the joint generator is the *sum* $L_t = L_t^{(z)} + L_t^{(x)}$ acting on functions $f(x, z)$ by $(L_t f)(x, z) = (L_t^{(z)} f)(x, z) + (L_t^{(x)} f)(x, z)$. For (piecewise) time-constant generators the Chernoff–Lie–Trotter product formula yields the stated limits; commutativity at the semigroup level follows from independence. □

Theorem 12 (Effect of factored reverse kernels)**.** *Fix a time step $t \to s$ and consider the family of joint reverse kernels $\mathcal{K} = \{\, K(x_s, z_s \mid x_t, z_t) \,\}$. Let*

$$\mathcal{K}_{\text{fact}} \;:=\; \Big\{\, K(x_s, z_s \mid x_t, z_t) = K_x(x_s \mid x_t, z_t)\, K_z(z_s \mid x_t, z_t) \,\Big\}$$

be the factored *family in equation 15. Then:*

- *(a)* Single-step limitation. *$\mathcal{K}_{\text{fact}}$ is a strict subset of $\mathcal{K}$: there exist joint kernels with within-step couplings that cannot be written as a product independent of (x_s, z_s) cross-dependence.*
- *(b)* Splitting sufficiency (small steps). *Suppose the target joint dynamics has generator $L_t = L_t^{(z)} + L_t^{(x)}$ (no cross-diffusion term), and the factored reverse uses* conditionally coupled *factors $K_x(\cdot \mid x_t, z_t)$ and $K_z(\cdot \mid x_t, z_t)$. Then, by Lie–Trotter splitting, iterating factored kernels at a small step size Δt* alternately *(e.g., $K_z K_x$ per micro-step) converges to the* same *joint semigroup as any coupled kernel generated by L_t, hence there is no expressivity loss at the trajectory level as $\Delta t \to 0$.*

Proof. (a) Consider $\mathcal{X} = \{0, 1\}$ and $z \in \mathbb{R}$. Define a joint kernel that enforces $x_s = \mathbf{1}\{z_s > 0\}$ almost surely (hard constraint). This coupling cannot be expressed as $K_x(x_s \mid x_t, z_t) K_z(z_s \mid x_t, z_t)$ because any factorization leaves x_s independent of z_s *given* (x_t, z_t), contradicting the deterministic relation between x_s and z_s.

(b) Under the assumed generator sum $L_t^{(z)} + L_t^{(x)}$, the exact joint semigroup over a small interval Δt equals $e^{\Delta t (L_t^{(z)} + L_t^{(x)})}$, while the alternating product $e^{\Delta t L_t^{(z)}} e^{\Delta t L_t^{(x)}}$ (or the corresponding Markov kernels $K_z K_x$) approximates it with first-order error $\mathcal{O}(\Delta t^2)$. Over a partition with mesh $\max \Delta t \to 0$, the product converges to the exact semigroup (Chernoff–Lie–Trotter), hence the factored parameterization with alternating micro-steps is sufficient to realize the same family of joint laws. □

Corollary 13 (Coupling via conditioning is enough in the small-step limit)**.** *Even though $K(x_s, z_s \mid x_t, z_t)$ does not condition on (x_s, z_s) jointly in one shot, allowing each factor to depend on* both *(x_t, z_t) and alternating the updates yields asymptotically the same expressivity as a fully coupled kernel driven by a sum-generator $L_t^{(z)} + L_t^{(x)}$. However, the diffusion process design of our language modeling task is flexible as long as it preserve the clean marginals, thus a factored process suffices.*

Remark 6 (When factorization truly hurts)**.** *If the target joint generator includes* cross terms *that cannot be written as $L^{(z)} + L^{(x)}$ (e.g., a diffusion on $\mathbb{R}^{L\times d}$ whose drift or diffusion matrix depends on* future x_s *rather than x_t, or a discrete jump rate at time t depending on z_s), then any per-step factorization that does not see (x_s, z_s) jointly will* not *reproduce such dynamics without further inner iterations (e.g., Gibbs-within-step to sample z_s and then x_s conditioning on z_s).*

However, we can always construct a proper diffusion process without cross-terms as in the main text (as long as it preserve the terminal marginals), hence the factored parameterization is expressive enough for our language generation task.

B.2.2 Designs of Schedules

We now provide more discussions on the design space of noise schedules in CCDD.

Information theory perspective. We start with the tools in information theory.

Lemma 14 (Mutual information decay under factored corruption)**.** *Let $I_t := I((x_0, z_0); (x_t, z_t))$ denote the mutual information between data and the corrupted pair at time t. Under equation 13 with independent noises, I_t is non-increasing and satisfies*

$$\frac{d}{dt} I_t \;=\; \mathbb{E}\Big[\, \frac{d}{dt} \log \frac{q_t(x_t, z_t \mid x_0, z_0)}{q_t(x_t, z_t)} \,\Big] \;\le\; 0,$$

with decomposition

$$\frac{d}{dt} I_t \;=\; \frac{d}{dt} I\big((x_0, z_0); z_t \mid x_t\big) \;+\; \frac{d}{dt} I\big((x_0, z_0); x_t \mid z_t\big).$$

In particular, for the VP SDE the continuous contribution equals the negative Fisher score power*:*

$$\frac{d}{dt} I\big((x_0, z_0); z_t \mid x_t\big) \;=\; -\,\mathbb{E}\big[\|g_t\, \nabla_z \log q_t(z_t \mid x_t)\|^2\big],$$

and for the CTMC the discrete contribution equals (minus) a χ^2-divergence–based entropy production rate.

Proof. Data processing along a Markov chain $(x_0, z_0) \;\to\; (x_t, z_t)$ yields non-increasing mutual information. The derivative formula follows from differentiating the KL defining mutual information and the Kolmogorov forward equations; for diffusions the de Bruijn–Fisher identity gives the continuous term; for CTMCs one uses the standard entropy production formula $\frac{d}{dt} H(p_t) = -\sum_{i\neq j} p_t(j)(G_t)_{ij} \log \frac{p_t(i)}{p_t(j)}$ and the χ^2 representation of conditional MI rates. □

"Optimal" coupling via Hyperparameter Matching. It is natural to have the question: is there an optimal way to couple the two modalities? We discuss the choice of forward hyperparameters (diffusion rate β_t for z, jump rate u_t for x) and their coupling.

Definition 6 (Signal-to-noise and posterior conditioning)**.** *Define the continuous SNR at time t by $\mathrm{SNR}_z(t) := \alpha_t^2/\sigma_t^2$ (VP case), and the discrete SNR by the interpolation weight η_t in $q_t(x_t \mid x_0) = \mathrm{Cat}(\eta_t x_0 + (1-\eta_t)\pi_t)$ (USDM/masked). Define the* joint conditioning strength

$$\kappa_t \;:=\; I\big((x_0, z_0);\; z_t \mid x_t\big) \;+\; I\big((x_0, z_0);\; x_t \mid z_t\big),$$

which quantifies how informative each modality remains about its clean counterpart when conditioning on the other.

Proposition 15 (Entropy/MI matching heuristic)**.** *(Informal) Let β_t and u_t be chosen so that the* rates *of MI decay from Lemma 14 are balanced:*

$$\mathbb{E}\big[\|g_t\, \nabla_z \log q_t(z_t \mid x_t)\|^2\big] \;\approx\; \mathrm{Rate}_x(t),$$

where $\mathrm{Rate}_x(t)$ denotes the CTMC's conditional MI decay rate (a χ^2-type quantity). Then the two posteriors $q(x_0 \mid x_t, z_t)$ and $q(z_0 \mid x_t, z_t)$ maintain comparable conditioning difficulty *(well-conditioned Bayes factors), which in turn stabilizes training losses for heads in the discrete modality.*

Intuitively, training both heads amounts to estimating $q(z_0 \mid z_t, x_t)$ and $q(x_0 \mid x_t, z_t)$. If one modality loses information much faster (e.g., $\mathrm{SNR}_z \ll \mathrm{SNR}_x$), then one posterior becomes broad/ill-conditioned relative to the other, causing gradient scale mismatch. Balancing the decay rates equalizes expected Fisher information in the continuous head and the discrete information production, yielding comparable curvature of the objectives (via de Bruijn/Fisher for z and entropy production for the CTMC). This is a standard preconditioning argument based on matching Fisher blocks across modalities.

Corollary 16 (Schedule matching guideline)**.** *A practical rule is to pick β_t and u_t such that the* monotone *functions $t \mapsto \mathrm{SNR}_z(t)$ and $t \mapsto \eta_t/(1-\eta_t)$ have similar slopes on a log-scale, e.g.,*

$$\frac{d}{dt}\log \mathrm{SNR}_z(t) \;\approx\; \frac{d}{dt}\log\frac{\eta_t}{1-\eta_t}.$$

This equates the relative shrinkage of posterior variances (continuous) and odds (discrete), approximately stabilizing Bayes updates in x_t.

Practical schedule designs. From a practical perspective, we could regard the continuous representations as the guidance for the discrete part, the continuous schedule then should be ahead of the discrete schedule (i.e., generated earlier in the reverse process) as useful high-level guidance. This intuition is also validated by other work involving multiple modalities (Geffner et al., 2025). We adopt this strategy in most experiments, featuring linear schedule for discrete part ($\eta_t = 1 - t$) and concave schedule for continuous part ($\alpha_t = \sqrt{1-t}$).

We conclude this subsection by summarizing takeaway messages on CCDD.

- **Formulation.** The joint model formalizes a mixed SDE–CTMC system with independent forward corruptions and a reverse denoiser that *conditions* on both (x_t, z_t) but factors the per-step kernel across modalities.
- **Expressivity vs. continuous-only.** Joint modeling is strictly more expressive on the *joint space*; on the *z-marginal*, it does not enlarge the class beyond continuous diffusion with comparable capacity (Theorem 10).
- **Effect of factorization.** A single-step factored kernel cannot represent arbitrary within-step couplings (Theorem 12(a)), but alternating factored updates at small step sizes attains the same semigroup when the target generator splits (Theorem 12(b), Corollary 13).
- **Coupling/schedules.** Matching information decay across modalities (Proposition 15, Corollary 16) yields well-conditioned posteriors.
- **Information-theoretic lens.** The continuous score is $\nabla_z \log q_t(z \mid x)$; MI decays additively across modalities under independence (Lemma 14). Balancing decay rates aligns Fisher/entropy production and stabilizes learning.

C Implementation Details

C.1 Architecture Design for CCDD

The architecture consists of basic diffusion transformer blocks. Optional timesteps conditioning is embedded through adaLN. For standard DiT blocks, attention is followed by MLPs.

DiT. Standard DiT except that we mix continuous and discrete embeddings before the first DiT block (through adding or concatenating), and decode both continuous and discrete tokens from the output of the last DiT block. Therefore, the actual processed tokens are of shape $[B, L, d]$ and the attention complexity is $\mathcal{O}(L^2)$.

MMDiT. Since MMDiT is naturally capable of processing multimodal generation, we opt it to generate continuous and discrete tokens simultaneously. We also adopt a slightly different version that stagger cross-attention blocks for modality interaction and self-attention blocks for single modalities, respectively. The tokens are of shape $[B, 2L, d]$ consequently, with the attention complexity $\mathcal{O}(2L^2)$ and parameters doubled.

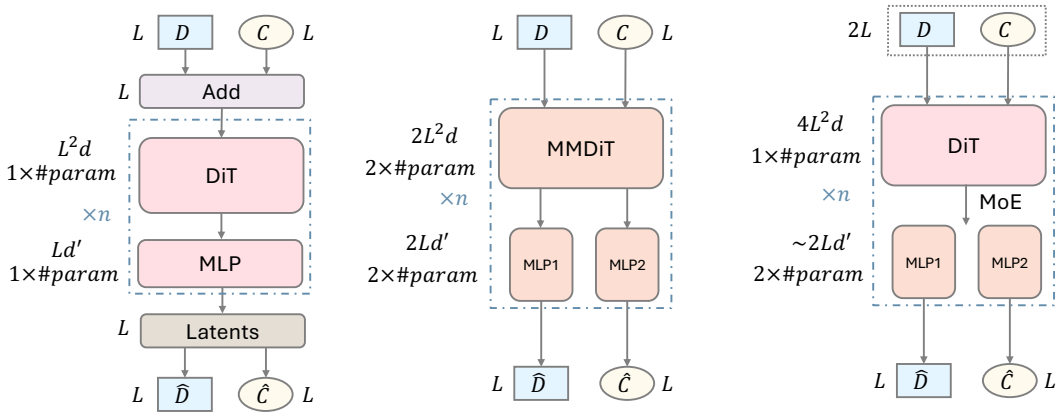


Figure 4: Comparison of different denoising network architectures for CCDD: MDiT (left), MMDiT (middle), and MoEDiT (right).

DiT with MoE. To maximize the expressivity while keeping the number of parameters not doubled, we choose to maintain representation for both continuous and discrete tokens, resulting $[B, 2L, d]$ representations. However, two modalities share the same attention parameters, and we compute self-attention over the $2L$ tokens, resulting $\mathcal{O}(4L^2)$ attention complexity without doubling the number of parameters. To avoid modality collapse, we use different MLPs for discrete and continuous modality. However, instead of hard separating MLPs, we use MoE architecture and let each token selects the proportion of experts, keeping maximum expressivity for every token while allowing discrete and continuous tokens to be processed differently according to their states. (For example, if the discrete state is cleaner, then there might be a higher weight on the discrete MLPs, and vice versa.) We call this architecture "MoEDiT".

In our implementation, we may interpolate the cross-modal MMDiT blocks with unimodal self-attention blocks and use their combinations. Our experiments aim to investigate the performance with same parameter or computation conjectures, or studies the scaling behavior w.r.t. parameters and FLOPs.

Remarkably, when adopting MDiT, CCDD has actually almost the same number of parameters as well as the same FLOPs compared with discrete diffusion baselines with DiT architectures – the only difference is the additional linear input and output heads for the continuous component, whose FLOPs are ignorable compared with main DiT blocks. Thus, both the training time and inference time of CCDD with MDiT architecture almost does not increase, yet the performance gain compared with discrete baselines is nontrivial. MMDiT has doubled parameters and $2\times$ FLOPs in core transformer blocks, while MoEDiT has the same attention parameters and doubled FFN parameters, with $4\times$ FLOPs in attention and approximately $2\times$ FLOPs in FFN.

As shown in Table 2 and Table 3, CCDD with MDiT is able to significantly surpass baselines without increasing non-embedding parameters and FLOPs; despite more parameters and FLOPs, MMDiT and MoeDiT both showcase much stronger performance compared with discrete baselines. In comparison, both MMDiT and MoEDiT bring prominent performance gain, among which MMDiT balances efficiency and performance, and MoEDiT balances parameters and performance.

C.2 Algorithm

We now give the algorithmic description of CCDD training in Algorithm 1 and sampling in Algorithm 2 using DDPM / DDIM for the continuous example. Notably, the main computation during inference occurs in the denoising model forward, i.e., line 3 in Algorithm 2, thus the inference cost also depends on the architecture design explained in Section C.1. With proper architecture selection, both CCDD training and inference could be extremely efficient.

Algorithm 1 CCDD Training

Require: Dataset $\mathcal{D}$ of pairs (x_0, z_0) with $z_0 = \mathcal{E}(x_0)$; time sampler $t \sim \mathcal{U}([0,1])$ or discrete grid $\{t_k\}$; VP schedule $\{\alpha_t, \sigma_t\}$; discrete schedule $\{\eta_t\}$ and noise distribution $\{\pi_t\}$; loss weights $\lambda_{\text{cont}}(t), \lambda_{\text{disc}}(t)$.

1: Initialize parameters θ of a single time-conditioned model $f_\theta(\cdot, t)$ with two heads:

$$\epsilon_\theta(x_t, z_t, t) \in \mathbb{R}^{L\times d}, \qquad \ell_\theta(x_t, z_t, t) \in \mathbb{R}^{L\times V}.$$

2: **for** each minibatch $\{(x_0^{(b)}, z_0^{(b)})\}_{b=1}^B \subset \mathcal{D}$ **do**

3: Sample times $\{t^{(b)}\}_{b=1}^B$.

4: **Forward corruption:** Sample $\epsilon^{(b)} \sim \mathcal{N}(0, I)$ and set

$$z_t^{(b)} \leftarrow \alpha_{t^{(b)}}\, z_0^{(b)} + \sigma_{t^{(b)}}\, \epsilon^{(b)}, \quad x_t^{(b)} \sim \text{Cat}\Big(\eta_{t^{(b)}}\, e_{x_0^{(b)}} + (1-\eta_{t^{(b)}})\, \pi_{t^{(b)}}\Big).$$

5: **Model prediction:**

$$\epsilon_\theta^{(b)} \leftarrow \epsilon_\theta\big(x_t^{(b)},\, z_t^{(b)},\, t^{(b)}\big), \qquad \ell_\theta^{(b)} \leftarrow \ell_\theta\big(x_t^{(b)},\, z_t^{(b)},\, t^{(b)}\big).$$

6: **Continuous loss (VP, ϵ-pred MSE) and discrete loss (token CE on masked tokens):**

$$\mathcal{L}_{\text{cont}} \;\leftarrow\; \frac{1}{B}\sum_{b=1}^{B} \lambda_{\text{cont}}(t^{(b)})\, \big\|\epsilon^{(b)} - \epsilon_\theta^{(b)}\big\|_2^2.$$

$$\mathcal{L}_{\text{disc}} \;\leftarrow\; -\frac{1}{B}\sum_{b=1}^{B} \lambda_{\text{disc}}(t^{(b)}, x_0^{(b)}, x_t^{(b)})\; \log\text{softmax}\big(\ell_\theta^{(b)}\big)[x_0^{(b)}].$$

7: **Total loss and update:** $\mathcal{L} \leftarrow \gamma_{\text{cont}} \cdot \mathcal{L}_{\text{cont}} + \gamma_{\text{disc}} \cdot \mathcal{L}_{\text{disc}}$

8: Update $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$.

9: **end for**

Algorithm 2 CCDD Sampling

Require: Time grid $1 = t_0 > t_1 > \cdots > t_K = 0$; forward schedules $(\alpha_{t_k}, \sigma_{t_k})$, (η_{t_k}, π_{t_k}); reverse kernels $q_{t_k|t_{k+1}}$ for discrete; DDPM/DDIM choice for continuous with variance knob $\eta_{\text{ddpm}} \in [0,1]$.

1: **Init:** $x_{t_0} \sim \pi_{t_0}$ (uniform or [MASK] prior per token); $z_{t_0} \sim \mathcal{N}(0, I)$.

2: **for** $k = 0$ **to** $K-1$ **do**

3: **Model heads:**

$$\epsilon_\theta \leftarrow \epsilon_\theta\big(x_{t_k}, z_{t_k}, t_k\big), \qquad \ell_\theta \leftarrow \ell_\theta\big(x_{t_k}, z_{t_k}, t_k\big), \quad \hat{\pi}_\theta = \text{softmax}(\ell_\theta).$$

4: **(A) Discrete reverse (Bayes form):**

$$p_\theta(x_{t_{k+1}} \mid x_{t_k}, z_{t_k}) \;\propto\; q_{t_k|t_{k+1}}(x_{t_k} \mid x_{t_{k+1}})\, \hat{\pi}_\theta(x_{t_{k+1}} \mid x_{t_k}, z_{t_k}, t_k).$$

5: Sample (or take mode) $x_{t_{k+1}} \sim p_\theta(\cdot \mid x_{t_k}, z_{t_k})$.

6: **(B) Continuous reverse (VP, ϵ-pred):**

$$\hat{z}_{0,\theta} \leftarrow \frac{z_{t_k} - \sigma_{t_k}\, \epsilon_\theta}{\alpha_{t_k}}.$$

7: **DDIM mean:** $m_{t_{k+1}} \leftarrow \alpha_{t_{k+1}} \hat{z}_{0,\theta} + \sigma_{t_{k+1}}\, \epsilon_\theta$.

8: **Stochastic DDPM step (optional):** $z_{t_{k+1}} \leftarrow m_{t_{k+1}} + \eta_{\text{ddpm}}\, \sigma_{t_k|t_{k+1}}\, \xi, \;\; \xi \sim \mathcal{N}(0, I)$.

9: **DDIM step (deterministic):** set η_{ddpm}=0 to use $z_{t_{k+1}} \leftarrow m_{t_{k+1}}$.

10: **end for**

11: **Decode:** Return tokens $\hat{x}_0 \leftarrow x_{t_K}$ and/or logits from a decoder applied to z_{t_K} (if needed).

D EXPERIMENTAL DETAILS

We provide more experimental details and additional results in this section.

D.1 EXPERIMENTAL CONFIGURATIONS

Schedules. For the discrete process, we use masked noise by default with $\gamma_t = 1 - t$ as in most discrete diffusion papers. For the continuous process, we use VP procedure as in DDPM (Ho et al., 2020) or the (log-)linear schedule in most flow matching papers. Ablation study on the continuous schedule are reported in Table 7.

Embedding spaces. Since Qwen3-Embedding enables flexible output dimensions down to 32, we use the 32-dimensional last-layer embeddings without specification. This selection is consistent with the analysis in the main text. Low-dimensional latent space is the standard setting in recent vision diffusion models (Esser et al., 2024). For RoBERTa-base embeddings, we use the full embedding with hidden size 768. All representations are normalized.

Other hyper-parameters. We set $p_{\text{drop}} = 0.15$ as in the masked rate in BERT (Devlin et al., 2019) training. Without specification, we set the loss weights $\lambda_{\text{cont}} = \lambda_{\text{cont}} = 1$ and use gradient clipping. Following Sahoo et al. (2024); von Rütte et al. (2025), on LM1B we set a constant learning rate 3×10^{-4} with 2500 warm-up steps, and a constant learning rate 5×10^{-4} with 10000 warm-up steps for OWT. We use AdamW optimizers with weight decay 0.02 and gradient norm 1.0.

Computation resources. All pretraining tasks are conducted with 8 NVIDIA H100 or A100 80GB GPUs. As an example, pretraining on OWT with 8 H100 GPUs requires 135 hours for MDiT, 255 hours for MMDiT, and 226 hours for MoEDiT.

D.2 DISCUSSION ON VALIDATION PERPLEXITY

Recall the forward process of CCDD. Since we use masking process for the discrete component with $\gamma_t = 1 - t$, the corresponding discrete loss is calculated as the mean of cross-entropy loss of the masked tokens:

$$\mathcal{L}_{\text{disc}} = \mathbb{E}_{t,x_t}\left[\mathbf{1}_{x_t=\boldsymbol{m}} \cdot \boldsymbol{x}^\top \log \boldsymbol{x}_\theta\right] \tag{20}$$

When calculating validation elbo, the model needs to predict the masked tokens. However, the model actually also takes the partially noised continuous tokens as the input, which provide information of the masked tokens that is unavailable in discrete diffusion, causing potential unfairness.

To address this issue, we use special methods to erase the related information in the continuous tokens corresponding to the masked discrete ones. For RoBERTa, we sample $\tilde{z}_0 = \mathcal{E}(x_t)$ as the embedding of x_t (namely the partially masked sequence) instead of x_0, so that the "clean" representations of the masked tokens would not directly be determined by the oracle. For Qwen3-Embedding model which does not have a mask token, we set the embeddings corresponding to masked tokens of discrete component in $z_0 = \mathcal{E}(x_0)$ to zeros in order to simulate the "masking" operation, leading to $\tilde{z}_0$ which also declines the direct information leakage. The continuous forward process then starts with $\tilde{z}_0$ instead of the original z_0. To let the model capable of doing inference with these perturbed inputs, we also perform these masking operations with a certain probability p_r during training, which is stochastically sampled per-sequence within $[0, 0.9]$.

Notably, evaluating ELBO with $\tilde{z}_0$ actually makes the inference of CCDD harder than discrete only, since the unmasked tokens in the discrete part are injected noise to their continuous representations, while the purely discrete model only takes the clean inputs. In the main text, we always report results with $p_r = 1$, which are proper. Even if evaluated with a strictly harder metric, CCDD still outperforms the baselines, which effectively validates the superiority of joint modeling.

D.3 ADDITIONAL ABLATIONS

We further study the effects of architecture and continuous schedules. We set $p_r = 0$ for all models during training and inference for simplicity. The validation perplexity is calculated as the exponential

Table 7: Validation ELBO of ablations on LM1B. We use Qwen3-Embedding-0.6B as the continuous generation space for CCDD, and train all models for 33B tokens. The numbers non-embedding parameter counts are also reported for fair comparison. "VP" refers to the variance preserving schedule in DDPM, and "Linear" refers to the log-linear schedule in flow matching.

Base Model	Architecture	# params.	Cont. Schedule	Validation ELBO (↓)
CCDD	MDiT	92.1M	VP	2.338
CCDD	MDiT	92.1M	Linear	2.586
CCDD	MMDiT	216.2M	VP	2.311
CCDD	MMDiT	216.2M	Linear	2.518

function of ELBO. Table 7 validates that a continuous schedule ahead of the discrete part in inference time (VP) yields better results. Scaling the number of parameters also consistently improves the performance.

Table 8: Validation ELBO with $p_r = 0$ on OWT. We use Qwen3-Embedding-0.6B as the continuous generation space for CCDD, and train all models for 131B tokens.

Model	Validation ELBO (↓)	Validation PPL (↓)
CCDD-MDiT	2.457	11.67
CCDD-MMDiT	2.415	11.19

We also report the validation ELBO with $p_r = 0$ (always starting diffusion process on the oracle x_0 and z_0) in Table 8 for reference. The models are also trained with Qwen3-Embeddings on OWT leveraging VP schedules. We observe that even a simple MDiT (which essentially shares the same non-embedding parameters as standard DiT except the additional encoding layer and the decoding head) could obtain super low ELBO.

D.4 Details of Downstream Benchmarks and Complex Reasoning Datasets

Downstream benchmarks. The zero-shot downstream benchmarks include ARC Clark et al. (2018) (both elementary and challenge subsets), BoolQ Clark et al. (2019), PIQA Bisk et al. (2019), OpenBookQA Mihaylov et al. (2018), and WinoGrande Sakaguchi et al. (2019). They cover a wide range of question answering, complex mathematical and commonsense reasoning, and domain knowledge. Our evaluation focuses on likelihood-based multiple-choice tasks, where the per-token likelihood is calculated across both the context and the completion, but not the padding.

Complex reasoning datasets. The goal of Sudoku is to fill a 9×9 grid with numerical digits, ensuring that every column, row, and 3×3 subgrid contains all the numbers from 1 to 9. The goal of SAT (Boolean satisfiability problem) is to determine whether a given Boolean formula represented in conjunctive normal form (CNF) can be assigned a set of values (0 or 1) to its variables, such that the formula evaluates to true. We select 3-SAT problems with 9 variables (i.e., 3-SAT 9). Countdown is a mathematical reasoning challenge generalized from the game of 24, which even GPT-4 struggles with (Yao et al., 2023). The goal of Countdown is using the given numbers and arithmetic operations $(+ - */)$ to obtain the target number. We select the subproblems with 4 variables (i.e., Countdown-4).

E DISCUSSION

E.1 STRENGTHS OF CCDD

We systematically summarize the advantages of CCDD as follows, which may inspire more future extensions.

- Exhibits strong expressivity, retains full information on marginal distribution and rich contextualized semantics.
- Ability to potentially conduct implicit reasoning, searching and planning in the latent space.
- Combines a smooth and conservative decoder containing rich semantics with an aggressive decoder modeling explicit tokens (which decomposes the process to a series of conditional generation).
- Receives knowledge distillation from pretrained models, and representation learning accelerates diffusion model training.
- Flexible in few-step sampling and inference-time scaling while compatible with CFG.

E.2 FURTHER DISCUSSIONS ON GENERATION SPACES OF CDM

Let $\mathcal{E}_{\text{tok}} : \Omega \to \mathbb{R}^d$ be a *token-wise* embedding with a fixed codebook $C = \{e_1, \dots, e_V\} \subset \mathbb{R}^d$ (typically $d \le V$, often $d \ll V$). Let $\mathcal{E}_{\text{ctx}} : \Omega^L \to \mathbb{R}^d$ be a *contextualized* embedding at a given position (depending on the entire sequence context), again with $d \le V$ in practice. We analyze three generation spaces for the *continuous* diffusion variable at a position:

(i) $\mathcal{Z}_\Delta = \Delta^{V-1}$, (ii) $\mathcal{Z}_{\text{tok}} = \mathbb{R}^d$ (token-wise codebook), (iii) $\mathcal{Z}_{\text{ctx}} = \mathbb{R}^d$ (contextualized).

For sequence length L, these apply positionwise with shared or tied encoders/decoders.

Probability simplex. $\mathcal{Z}_\Delta = \Delta^{V-1}$ models calibration-ready probabilistic targets, and the score functions may have clear information-geometry meanings (e.g., Fisher geometry on the simplex). However, the high-dimensionality and the valid distribution constraints make it hard to be directly generated (as the diffusion target). Furthermore, data are concentrated on simplex vertices (one-hots), resulting highly non-smooth denoising targets.

Token-wise embedding with codebook. Training targets for token-wise embedding space are codebook vectors $z_0 = e_{x_0} \in C \subset \mathbb{R}^d$ and diffusion evolves in $\mathbb{R}^d$. Decoding can be nearest-neighbor or softmax over codebook energies: $p(x \mid z) \propto \exp\{-\|z - e_x\|^2/\tau\}$ as in previous work (Li et al., 2022). When $d \ll V$, the lower dimensionality and the Euclidean geometry (naturally suitable for Gaussian noises) enable easier diffusion optimization. Nevertheless, the optimization target is still non-smooth due to the atomic codebook vectors and the naive decision boundaries. Most importantly, we now show that $\mathcal{Z}_{\text{tok}}$ is not more powerful than $\mathcal{Z}_\Delta$.

Proposition 17 (Representational power w.r.t. terminal classification)**.** *Any z-based classifier can be emulated by modeling logits in $\mathbb{R}^V$ and passing through a softmax with a learned final linear layer whose columns are $\{e_v\}$. Hence token-wise embeddings are not strictly more expressive than simplex-based logits for terminal token prediction.*

Proof. (Constructive.) Define logits $\ell(z) = W^\top z + b$ with columns of W equal to code vectors (optionally re-centered); $\text{softmax}(\ell(z))$ approximates any energy-based decoder over the codebook. □

The construction is straightforward since $\mathcal{E}_{tok} = \begin{bmatrix} e_1 \\ \dots \\ e_V \end{bmatrix} \in \mathbb{R}^{V \times d}$ is column full rank, thus any z could be linearly reconstructed by a logit ℓ with the help of the codebook.

Remark 7 (Why training can still be harder)**.** *The regression target $z_0 \in C$ makes the conditional expectation $\mathbb{E}[z_0 \mid z_t]$ a weighted average of code vectors, which lies* between *modes. Nearest-neighbor decoding then introduces a quantization gap; gradients around Voronoi boundaries are poorly conditioned, explaining larger prediction errors in practice.*

Contextualized embedding. Here $z_0 = \mathcal{E}_{\text{ctx}}(x_0; \text{sequence context})$ depends on the whole sequence (and position). A decoder $D_{\text{ctx}}(z, \text{context}) \to \Delta^{V-1}$ maps the embedding back to token probabilities. If $\mathcal{E}_{\text{ctx}}$ is well-defined (which is true for state-of-the-art pretrained LLMs), the generation targets now contain rich information and themselves being smoother. Sequence-level information and long-range dependencies are encoded into a compact space, facilitating diffusion generation. The largest obstacle now becomes the decoding ambiguity, which could be tackled with the help of the discrete component.

Remark 8 (Sufficiency and decoding)**.** *If $z_0 = \mathcal{E}_{\text{ctx}}(x_0, ctx)$ is (approximately) a sufficient statistic for predicting x_0 given context (i.e., $p(x_0 \mid ctx, z_0) = p(x_0 \mid ctx)$), then there exists a decoder D_{ctx} achieving Bayes-optimal token prediction. In practice, approximate sufficiency improves sample efficiency but is not guaranteed, leading to residual ambiguity.*

F Generated Examples

We attach some generated examples of CCDD-MoEDiT trained with Qwen3-Embedding-0.6B on OWT. The inference length and the number of denoising steps are both set to 512, with DDPM update for continuous tokens and MDLM posteriors for discrete tokens. The CFG guidance scale is 1.5, and the sampling temperature is 1.0 by default.

Her pancakes seemed to be far too bland, and it was clearly an amazing piece of cake. She falls for the pieces like this, as she starts to take a look at how she had grown up with I mean, this man's and the whole life he has taught her in her life. Laughing at her for not being representative of what she was never originally supposed to be given to that life, and that she was not really as miserable as what she had grown up, because no matter what people said, she did what she was supposed to be like to bring her onto her own world, the Land, but he keeps treating her and she goes one by one and somehow spins around and begins something, tries to play the extreme with the lives without generally treating her with what she was supposed to be like too. But at the end of the day, she is not having any enjoyment of that life and she seems to have the crap transition that should work for them. The poor state of their world between them and her being something somewhat different.

That actually not only hurts her, it just make its villains really stupid about her, and even worse than what they thought she was. That was simply a very normal combination of things, as she was incorporating a few devices into her life where she could still do anything, anything for a while, and she really was like she had to have a home. As I was lived along at that time, I started knowing that she had a really, really really shitty home, that they'd had nothing like interacting with me until her home was eventually taken over, and she was never physically or even physically interacting with me, and all this in those moments, I was really accepting that I was just doing what I was doing, and things did seem to don't help me as she'd always have, and she wasn't the family that she was supposed to be home to until the moment, only now did I know what I was doing, which was a really crappy home, and I was just given the task, unable to get any benefit from her. She didn't have such a role as as long as she finished, she had no idea what to do, she just had a real, shitty home. What she could not have had when was the current state of the home. My husband had been living with me to the point where his entire life was going to be so completely acceptable to his home that meant he was going to just move again, and just have all this fun in the home world.

Back to farms, there was a lot of different stuff about it, especially about after the farm had been changed and we had no idea what part of the culture and/or anything like it was, while the people that seemed to argue towards us went and in a subsequent meeting of my farm was worthless and I all realized that anyway it had been taken away since we had 90% of money in direct sales and we had no interaction with any other farmers or the manager which was what was actually happening for us. That's when I noticed a lot of the business around me and thought immediately, everything seemed very impersonal to the small restaurants it was on both iterations were that were both profitable and varied but this was open sourced for them to be innovative, I definitely would have seen something completely different than anyone else on earth and most likely not much of the possibility they were trying to have to it were really something that was available to them and not as innovative as we had ever dreamed of being, with such a limited options and the race around was how it looked on the farm and we can't even remember how it was brought in deeper than the bones with their fresh clothes but when anything turned a apart and it seemed to go further than how they went about the business, I begin to compare only myself to this interconnected world rather than make me get to a point where I would have reasonable concerns about anything like it. At first it seemed sort of seemed like a new outpost was any sign of plans on existing or more of it and not much later, much later it was so much that the front screen seemed to me very warped and just a sense of disgust his face was due to his colors nothing was really very apt to present for me to see what they were doing there all along and most of me that I had been getting into the community behind had anything I was capable of producing. It was within the last couple of months that I've seen a lot of poor people being able to rid themselves of this place, and all I keep mentioning is now since the small farms were seemingly only established in the food shops that time it's a question of my priorities and not addressing something about the animals is something I can argue with, but this years it's been one of my priorities and lately like my boss have been distracted with very rarely the moves that moved. I've once again seen another homeless restaurant that seemed to be basically within the framework, several times and in the last few years has changed for the community in Syracuse and the only one.

The world crop is evolving because it's such a very different process and if it is never going to be sustainable, that is something like that that is way too bad for the people. And that's because it is so obvious that there is a pretty major issue on the planet that that is linked to that and most of the problems are a lot of that is happening on the roads, it is not happening for the plant, it is not happening for the home and there is a whole lot of salt all over the Pacific oceans, and for some of it that is done to the oceans, and why are they just looking at the government, seeing all the roads and the lumber, and the sand beaches and stuff and all of those is happening every day and everybody knows they aren't going to do it but this is one of my feelings and I'm going to reduce it to that abstract and it is the face of the globe that they said this as much as they are, but the most important thing in that is that all of the existing plants, unlike the old progressive reactors, are not just at a 100% temperature like our corn and the stuff is just endlessly growing out, every single out one just because it is something that would be a problem so basically we will be just putting it that people can use as a source of their resources. Some of those crops can be cut off, we can have no irrigation, we can go back to farms, we can graft some of that things and it makes NO sense as they are all the same. See, if we have both crops as though they aren't the same then there is no reason that would be a problem for us, we know each other, but just because they are living different to each other and one thing that is a lot with those crops or that is some of these weeds can come out of soil and eventually it will be replaced with trees and so there will be a number of areas to which these plants would not have the kind of characteristics or truly some different degrees of proportion that they are having and others are nearly as effective that they are supposed to have. Of course, maybe that is where the Pre-Christian Christian Church came from but that is not something that is so obvious and it is extremely obvious but it is because the water in plants, that which belongs in the environment, so all the seeds taken from water are not there and the material is not formed on planets. Those parts are the good aspects of the process and are all we need, they can be enough.