Scaling the Optimization Cliff: Causal Constraint Injection for Stable and Efficient Diffusion Language Models

Anonymous ACL submission

Abstract

Discrete diffusion language models face an "Optimization Cliff" where high variance in gradient estimates during training, associated with position-symmetric absorbing denoising in high-dimensional combinatorial spaces, can hinder optimization and inference efficiency. This is compounded by a "Directionality Gap" wherein any-order models do not explicitly encode the left-to-right inductive bias used for syntax learning. To address these issues, we propose Causal-Guided Absorbing Diffusion (CGAD), which injects causal features generated by a shallow autoregressive module within a fixed parameter budget into the global denoising field via a cross-attention mechanism. This approach is designed to provide directional information during denoising and is evaluated through gradient-variance measurements and generation results. Experiments demonstrate that CGAD achieves highly competitive performance and scalability while significantly improving inference efficiency, combining autoregressive directionality with diffusion-based parallelism.

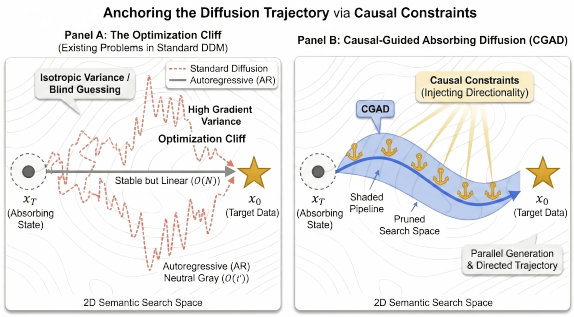

Figure 1: **The intuition of Causal-Guided Absorbing Diffusion (CGAD). (a) The Optimization Cliff:** Standard discrete diffusion models (Red) can exhibit high gradient variance under position-symmetric absorbing denoising objectives. Without directional guidance, the optimization process must predict from highly uncertain intermediate states, which may reduce denoising coherence. **(b) CGAD (Ours):** We introduce a Causal Constraint Injector that provides directional features (Gold) to the Global Denoising Field. This mechanism is intended to concentrate the denoising distribution (Blue Tunnel) and support a more directed trajectory while preserving parallel updates.

1 Introduction

The landscape of sequence generation has long been dominated by autoregressive (AR) models, which decompose sequence probability via the chain rule to ensure stable, locally coherent generation (Bengio et al., 2003). While this paradigm excels at capturing local syntax through its causal inductive bias, its sequential decoding nature results in $O(N)$ inference latency, creating a fundamental efficiency bottleneck (Schmidt et al., 2022). In response, discrete diffusion language models have emerged as a promising non-autoregressive alternative, enabling parallel token generation by iteratively denoising sequences from noisy or masked states (Austin et al., 2021a; Bao et al., 2022). These models reformulate generation as a global optimization process, leveraging forward noise injection and backward denoising to maintain long-range consistency and facilitate simultaneous multi-token prediction (Austin et al., 2021b; Gulrajani and Hashimoto, 2023).

By operating natively in discrete token space, recent discrete diffusion approaches avoid the embedding-discretization mismatch of continuous diffusion models for text (Čeović et al., 2023). Frameworks such as Diffusion-NAT unify discrete diffusion with pre-trained language models like BART by reframing denoising as masked token recovery, achieving performance competitive with autoregressive models on text generation tasks (Bao et al., 2022). Further advances include continuous-time discrete diffusion models, which provide more flexible time parameterization and analytical absorbing state modeling to mitigate iterative instability and intermediate semantic loss (Dieleman et al., 2022; Campbell et al., 2022). These developments

position discrete diffusion as a state-of-the-art approach for efficient parallel decoding, particularly benefiting from bidirectional attention for global sequence optimization (Yang et al., 2023; Yi et al., 2024; Lou et al., 2024; Arriola et al., 2025; Ou et al., 2025).

Despite their advantages, discrete diffusion models face a challenge that we term the "Optimization Cliff," where high gradient variance during training, associated with position-symmetric absorbing denoising in high-dimensional combinatorial spaces, can lead to unstable optimization and ineffective iterations. Here, position-symmetric absorbing denoising means that all token positions follow the same independent masking schedule without an imposed directional order. Consequently, standard discrete diffusion may require additional inference steps to recover semantic coherence, reducing the practical speed gains of parallelization and creating an efficiency-stability trade-off.

The core of this instability lies in the variance of the gradient estimation. In continuous diffusion (e.g., image generation), gradients are defined over a smooth Euclidean manifold, providing informative signals even at high noise levels. In contrast, discrete diffusion operates in a categorical space where the denoising path must navigate a combinatorial cliff. The model is tasked with estimating the denoising posterior $q(x_0|x_t)$ from a highly corrupted state x_t without the benefit of a structured, directional history.

As shown in Theorem 2.1(Section 2.1) under the stated non-degeneracy conditions, the lower bound of the gradient variance in position-symmetric absorbing denoising objectives is proportional to the sequence length and the integral of the noise level squared. This result characterizes a regime in which prediction from heavily masked states produces noisy optimization signals. Without a directional feature source, the multimodal distribution of potential tokens can produce incoherent intermediate states. Consequently, standard discrete diffusion models may require more inference steps (NFEs) to recover semantic integrity, which directly translates to lower inference throughput (tokens/second) and reduces the practical efficiency gains of parallelization. We term this the Efficiency-Stability Trade-off: improving diffusion quality can require inference costs that narrow its speed advantage over AR.

While recent frameworks like RADD theoretically(Ou et al., 2025) bridge diffusion and autoregression via arbitrary-order modeling, they motivate what we term a Directionality Gap. These models average over prediction orders rather than explicitly imposing the left-to-right inductive bias of a causal language model. We analyze how this difference can increase conditional uncertainty when natural-order context is informative, complementing the gradient-variance analysis.

To address the Directionality Gap and mitigate the Optimization Cliff, we propose Causal-Guided Absorbing Diffusion (CGAD). Our core insight is that sequential causality and global approximation can be combined through Constraint Injection. Instead of a simple ensemble, we architecturally introduce a shallow Causal Constraint Injector whose left-to-right predictions condition a Global Denoising Field within the stated total parameter budget. Specifically, the Injector produces directional features based on causal history and injects them as soft constraints. These signals are then integrated into the continuous-time diffusion network via a cross-attention mechanism, which allows the denoiser to selectively use directional information. By augmenting position-symmetric absorbing denoising with this learned feature pathway, CGAD is intended to reduce gradient variance and achieve high-fidelity generation with fewer inference steps compared to standard diffusion baselines.

Our specific contributions are as follows:

- We formally define the Directionality Gap in AO-ARM frameworks and show (Theorem 2.1), under stated conditions, a gradient variance lower bound for position-symmetric absorbing denoising objectives proportional to sequence length. We further characterize (Theorem 2.2) the conditional-entropy cost of Any-Order modeling when directional context is informative.

- We introduce the CGAD architecture, which utilizes a Causal Constraint Injector to provide low-variance local priors to a Global Denoising Field. We derive the interaction mechanism that aligns these disparate probabilistic paradigms via a mutual reinforcement objective.

- We demonstrate that CGAD achieves highly competitive performance(in Section 4.2, experimental results) and strong scalability(in Section 4.5, the scaling law analysis) with

lower measured gradient variation (in appendix Section D.2), especially in text generation and structural reasoning tasks. We additionally report the associated inference efficiency.

2 Bridging the Directionality Gap: A Causal Theory for Efficient Discrete Diffusion

In this section, we formalize the two challenges introduced in Section 1: prediction from highly masked states and the absence of an explicit left-to-right feature in any-order modeling. We analyze the "Optimization Cliff" via gradient variance (Theorem 2.1), formally define the "Directionality Gap" (Theorem 2.2), and characterize the conditions under which causal-guided estimation can improve denoising (Theorem 2.3). These three theoretical results connect the efficiency-stability trade-off in the introduction to the CGAD design for discrete diffusion language modeling and are detailed in appendix Section G. Additional discussions with related works can be found in appendix Section C.

2.1 The Optimization Cliff: Gradient Variance in Position-Symmetric Absorbing Diffusion

A critical, yet under-analyzed aspect of discrete diffusion is the stability of the optimization landscape, which corresponds to the high-mask uncertainty described in Section 1. In continuous diffusion, the score function $\nabla_x \log p_t(x)$ is a vector field pointing towards the data manifold. In discrete diffusion, the "score" is a ratio of probabilities.

When the corruption process is position-symmetric—meaning that all positions follow the same independent absorbing-mask schedule without a directional order—the gradient estimator may exhibit high variance in poorly conditioned regimes.

Theorem 2.1 (Gradient Variance Lower Bound in Position-Symmetric Absorbing Diffusion)**.** *Let $\mathcal{L}_{DSE}^T$ be the denoising score entropy loss for a discrete absorbing diffusion model with a shared position-wise masking rate $\sigma(t)$, and assume non-degenerate per-position gradient variation whose covariance terms do not cancel the accumulated variance. Under these conditions, the variance of the gradient estimator $\nabla_\theta \mathcal{L}_{DSE}^T(x_0)$ possesses a lower bound proportional to the sequence length d and an integral of the noise intensity:*

$$\mathbb{V}ar \geq C_1 \cdot d \cdot \int_0^T \frac{\sigma(t)^2 e^{-2\bar{\sigma}(t)}}{(1-e^{-\bar{\sigma}(t)})^2} dt \qquad (1)$$

where $C_1 > 0$ depends on the vocabulary size $|\mathcal{V}|$, the model architecture, the data distribution, and the assumed non-cancellation of per-position gradient variation.

Proof Sketch. The bound concerns position-symmetric absorbing denoising in a high-dimensional categorical space under the conditions stated above. In early denoising stages with high masking rates, the model receives limited contextual information, so the conditional probability estimate $p_0(\hat{x}_t^i | x_t^{\text{UM}})$ can be diffuse over the vocabulary. A diffuse categorical prediction has high token-level uncertainty. Summing the assumed non-cancelled contributions over all d sequence positions and integrating over the noise schedule $\sigma(t)$ yields the lower bound, with the integration and the dependence of C_1 detailed in appendix Section G. The term $\frac{1}{(1-e^{-\bar{\sigma}(t)})^2}$ weights high-uncertainty regimes in the stated objective, motivating the "Optimization Cliff" diagnostic. □

Intuitive Implication (The "Blind Guessing" Phenomenon): The interpretation of this high-mask regime and its connection to the measured gradient variation are provided in appendix Section G.

2.2 The Directionality Gap: Why Any-Order $\neq$ Causal

While RADD and AO-ARM provide a unified mathematical framework, language can assign different predictive value to different generation orders. This is the order-sensitivity issue introduced in Section 1, and we define the associated conditional-entropy difference as the Directionality Gap.

Theorem 2.2 (The Directionality Gap)**.** *Let Π be the set of all permutations of sequence indices $\{1, \ldots, d\}$. An Any-Order ARM optimizes the expected log-likelihood over a uniform distribution of permutations $U(\Pi)$. A Causal AR model optimizes over a single fixed permutation $\pi^* = (1, 2, \ldots, d)$. For a text distribution in which left-to-right prefixes provide strictly more predictive information than omitted prefixes with nonzero probability, the "Directionality Gap" is defined as the difference in the*

expected entropy of the learned conditional distributions:

$$\Delta_{Dir} = \mathbb{E}_{\pi\sim U(\Pi)}[H(p_\theta(x_{\pi(k)}|x_{\pi(<k)}))] \\ -H(p_{AR}(x_k|x_{<k})) > 0 \quad (2)$$

Furthermore, under this informative-prefix condition, AO-ARM estimators can exhibit higher variance for local syntactic dependencies than Causal AR estimators.

Proof Sketch. Natural language often contains directional dependencies (e.g., subject-verb-object order). The set of permutations Π_{syntax} that respect a selected dependency structure is smaller than the full permutation set. Causal AR uses one fixed ordering, while AO-ARM samples permutations uniformly. For a random permutation, a predictive predecessor can be absent from a token's conditioning set. Under the informative-prefix condition, removing that information increases conditional entropy. The entropy difference is therefore strictly positive for the stated distributional regime, providing a motivation for directional features. □

Implications for Optimization: Additional interpretation of the conditional-entropy gap for optimization is provided in appendix Section G.

2.3 The Solution: Causal Constraint Injection

To address the Directionality Gap and mitigate the Variance Problem identified in Section 1, we introduce the Causal Constraint Injector. Instead of optimizing $p(x_0|x_t)$ without an auxiliary directional feature, we condition the diffusion process on a local causal prior $\mathbf{b}_t$ generated by a shallow AR module allocated within the total model budget. The core hypothesis is that informative causal features can reduce the variance of the diffusion score estimator while retaining the parallel denoising pathway motivating diffusion generation in the introduction.

Theorem 2.3 (Variance Reduction via Causal Constraint Injection)**.** *Let $s^0_{\theta_{Diff}}(x_t, t)$ be the score estimator of a position-symmetric absorbing diffusion model. Let $s^1_{\theta_{Diff}}(x_t, t, b_t)$ be the score estimator conditioned on a local causal block b_t generated by a Causal Constraint Injector. If the causal injector satisfies the Local Determinism condition and its estimation error is sufficiently complementary to that of the diffusion estimator, then the mean squared error (MSE) of the score estimation is reduced:*

$$MSE(s^1) \leq MSE(s^0) - \Delta_{gain} \quad (3)$$

where Δ_{gain} is proportional to the mutual information between the causal block b_t and the true denoising target.

Proof Sketch. We model the interaction as a parameterized fusion $s^1(x_t) = F_\theta(s^0(x_t), s_{\text{AR}}(b_t))$. For the linear special case $s^1(x_t) = (1-\alpha)s^0(x_t) + \alpha s_{\text{AR}}(b_t)$, the MSE decomposes into three terms: $(1-\alpha)^2\text{MSE}(s^0) + \alpha^2\text{MSE}(s_{\text{AR}}) + 2\alpha(1-\alpha)\text{Cov}(s^0, s_{\text{AR}})$. The autoregressive estimator s_{AR} is intended to supply concentrated estimates in local syntactic regions. The complementary-error condition controls the covariance term when the diffusion model is uncertain and the AR component provides informative direction. With an appropriate α under that condition, the linear estimator has lower MSE. The cross-attention mechanism in CGAD implements a context-aware version of this fusion whose benefit is tested empirically. This fusion can be viewed as a dynamically weighted sum: $s^1(x_t) = \beta(x_t)s^0(x_t) + (1-\beta(x_t))s_{\text{AR}}(b_t) + \epsilon(x_t)$, where weight $\beta(x_t)$ is generated by the attention mechanism and ϵ is a residual term. □

The cross-attention mechanism in CGAD enables dynamic, feature-dependent integration of causal priors with the global denoising field, allowing selective emphasis on relevant syntactic guidance. This adaptive gating is designed to maintain global consistency while incorporating local constraints, bridging the Directionality Gap through learned, context-aware interaction. Deeper analysis of this mechanism is provided in appendix Section G.3.

3 Causal-Guided Continuous-Time Absorbing Diffusion

To address the **Directionality Gap** (Sec. 2.2) and **Optimization Cliff** (Sec. 2.1), we propose the Causal-Guided Absorbing Diffusion (CGAD) framework. CGAD integrates autoregressive directionality with diffusion-based parallel updates via: (1) a continuous-time Global Denoising Field guided by a Causal Constraint Injector; (2) cross-attention feature interaction; and (3) a joint training objective for local-global prediction. This design balances efficiency and stability by allocating parameters to directional features while maintaining parallel updates.

3.1 Causal-Guided Absorbing Diffusion Architecture

The CGAD framework addresses the theoretical challenges from the introduction by operating on partially denoised states $X_t \in \mathcal{V}^d$ (with vocabulary $\mathcal{V}$ and sequence length d). This design augments the standalone denoising field (Section 3.1.1) in the high-masking regimes associated with the **Optimization Cliff** in Theorem 2.1. Generation is instead decomposed into two coupled components: the **Causal Constraint Injector**, which supplies the local directional features absent from the any-order setting motivating the **Directionality Gap**, and the **Global Denoising Field**, which retains the global-context advantage emphasized in Section 1. This decomposition injects learned directional features into the denoising process to address the efficiency-stability trade-off.

3.1.1 Global Denoising Field

The Global Denoising Field models the conditional distribution $p_\theta(X_0|X_t^{UM})$ through a continuous-time absorbing diffusion process, where $X_t^{UM} \in \mathcal{V}^{d_{\text{UM}}}$ represents the unmasked tokens at time step t, with d_{UM} being the number of unmasked tokens. In keeping with the diffusion motivation in Section 1, this component employs an encoder-transformer architecture that gradually reconstructs the original data distribution from noise and serves as the global consistency backbone of the framework:

$$H_t^{\text{Diff}} = \text{Encoder}_\theta(X_t^{UM}) \in \mathbb{R}^{d_{\text{UM}} \times d_{\text{model}}} \tag{4}$$

where d_{model} is the hidden dimension of the transformer. The diffusion process follows the continuous-time formulation with a noise schedule $\sigma(t)$ that determines the masking probability at each time step. The denoising network is trained to predict the original clean data X_0 from its noisy version X_t using a denoising cross-entropy objective (the specific loss formulation, L_{DCE}, is detailed in Section 3.2).

3.1.2 Causal Constraint Injector

The Causal Constraint Injector estimates $p_\phi(\hat{X}_t^i|X_t^{UM})$ for masked tokens, providing local directional predictions motivated by the **Directionality Gap** to mitigate the **Optimization Cliff** (assessed via gradient-variance). Its output corresponds to the autoregressive score estimator $s_{AR}(b_t)$ in Theorem 2.3 (Variance Reduction), where b_t denotes a token block. Employing a parallelized multi-token prediction paradigm, it preserves causal dependencies within blocks—where "causal" refers to left-to-right block factorization (Section 1), not the bidirectional denoising backbone. Architecturally, it is a shallow Transformer decoder ($L_{\text{ar}} \ll L_{\text{diff}}$, e.g., 4 vs. 10 layers) that shares token embeddings and output projections with the global denoiser, with $d_{\text{ar}} \ll d_{\text{model}}$ (Table 6). This allocates a lightweight directional pathway for local causal features within the total budget.

The Causal Constraint Injector employs a Transformer decoder architecture that produces logits for each position. To enhance local coherence while maintaining parallelization efficiency, it decodes blocks of k tokens in a parallel manner, where k is a key hyperparameter controlling the trade-off between autoregressive coherence and parallel efficiency. For a block starting at index j, the conditional probability is factored as:

$$\begin{aligned} &p_\phi(X_0^{[j:j+k-1]} \mid X_t) \\ &\quad = \prod_{m=j}^{j+k-1} p_\phi(X_0^m \mid X_0^{<m}, X_t^{UM}) \end{aligned} \tag{5}$$

where $X_0^{<m} \in \mathcal{V}^{m-1}$ represents the causal context of previously generated tokens within the block. The hidden states $H^{\text{AR}} \in \mathbb{R}^{k \times d_{\text{model}}}$ of the Causal Constraint Injector are computed via:

$$H^{\text{AR}} = \text{TransformerDecoder}(X_0^{<j}, X_t^{UM}; \phi) \tag{6}$$

This computation employs causal self-attention for intra-block autoregressive context and cross-attention to integrate features from X_t^{UM}. Unlike isolated block-based diffusion models, CGAD's injector couples with the global denoising field via cross-attention, enabling parallel updates of the denoising state while incorporating directional block features. The output H^{AR} serves as the primary carrier of local, directional information for blending with the global context.

The feature interaction mechanism is implemented as a cross-attention layer where the autoregressive hidden states H^{AR} serve as the query, and the global denoising features H^{Diff} serve as both key and value. Formally:

$$H^{\text{interact}} = \text{Softmax}\left(\frac{H^{\text{AR}} H^{\text{Diff}^T}}{\sqrt{d}}\right) H^{\text{Diff}}. \tag{7}$$

Table 1: Zero-shot language modeling results on six datasets using **small models**. Perplexity (PPL, ↓) and Accuracy (Acc, ↑) are reported. Best results are in **bold**, second best are underlined.

Method	Datasets											
	Prolong		WikiText2		CFG		WikiText103		C4		FineWeb	
	PPL	Acc	PPL	Acc	PPL	Acc	PPL	Acc	PPL	Acc	PPL	Acc
GPT-2	54.79	45.36	52.05	45.91	147.99	27.89	51.14	46.32	85.19	36.17	46.23	50.24
D3PM	103.34	31.28	86.83	39.45	210.30	27.12	85.09	39.87	148.81	32.01	62.34	40.19
PLAID	66.79	38.75	61.60	42.68	152.19	29.47	60.48	43.95	100.66	34.88	48.59	48.63
SEDD-Uniform	75.24	36.89	60.16	43.12	149.90	30.11	59.23	44.71	111.12	33.25	51.38	47.85
SEDD-Unscale	62.01	41.57	54.40	46.32	140.24	31.25	52.75	47.88	90.45	35.91	47.11	49.12
SEDD-Scale	60.56	43.22	51.51	48.95	124.35	35.67	50.22	49.34	88.91	36.24	46.94	50.07
RADD-DSE	59.22	44.91	48.65	51.87	121.61	36.12	47.04	52.49	82.04	38.95	46.75	50.32
RADD-t-DCE	60.38	43.68	48.70	51.92	118.61	37.44	45.95	53.62	82.43	38.47	46.53	50.58
RADD-λ-DCE	61.50	42.35	49.80	50.76	117.66	38.21	47.81	51.86	82.80	38.02	47.79	49.28
BD3-LM	**53.18**	**47.20**	49.15	51.18	115.03	38.10	48.20	51.33	85.06	37.72	**45.55**	50.52
SSD-LM	55.07	45.83	50.03	50.84	116.13	37.59	49.26	50.97	86.39	36.93	47.08	49.54
CGAD-t-DCE	54.23	46.42	**48.35**	**52.31**	110.89	39.88	46.12	**53.45**	82.12	39.11	46.01	49.96
CGAD-λ-DCE	55.11	45.63	48.94	52.12	**110.77**	**40.12**	**45.49**	53.08	**81.92**	**39.64**	46.46	**51.51**

Table 2: Zero-shot language modeling results on six datasets using **medium models**. Perplexity (PPL, ↓) and Accuracy (Acc, ↑) are reported. Best results are in **bold**, second best are underlined.

Method	Datasets											
	Prolong		WikiText2		CFG		WikiText103		C4		FineWeb	
	PPL	Acc	PPL	Acc	PPL	Acc	PPL	Acc	PPL	Acc	PPL	Acc
GPT-2	**45.41**	47.85	41.26	50.38	132.80	30.25	41.14	50.42	65.29	39.74	38.32	48.25
SEDD-Unscale	54.39	45.28	44.49	47.15	102.94	37.91	42.87	48.79	77.53	37.52	42.97	48.79
SEDD-Scale	52.49	46.92	40.75	51.89	96.80	39.87	39.66	52.01	70.96	40.01	40.99	48.68
RADD-DSE	51.87	47.69	38.77	53.89	84.58	41.25	37.85	53.82	67.07	41.95	42.92	48.84
RADD-t-DCE	53.08	46.58	39.85	52.81	88.54	40.12	39.43	52.24	67.63	41.39	42.89	48.87
RADD-λ-DCE	53.65	46.01	40.10	52.56	91.53	39.34	39.08	52.59	70.22	39.85	45.76	46.01
BD3-LM	50.05	48.20	39.13	53.27	90.36	41.44	38.09	**55.02**	**63.78**	41.67	39.86	**49.93**
SSD-LM	50.17	48.17	39.51	53.07	91.03	41.29	38.52	54.01	64.54	41.46	40.27	49.58
CGAD-t-DCE	51.23	48.43	**37.35**	**54.31**	82.89	41.88	37.12	54.55	64.12	**42.95**	**38.01**	49.76
CGAD-λ-DCE	50.11	**49.55**	37.94	53.72	**82.77**	**42.01**	**36.49**	54.18	64.32	42.68	38.46	48.31

3.2 Mutual Reinforcement Training Objective

To jointly train both components while mitigating optimization challenges (Sec. 1), we propose a mutual reinforcement objective. It jointly supervises causal features and the diffusion field, coupling the directionality pathway with the global denoising pathway (Sec. 3.1). Training employs the t-denoising cross-entropy (t-DCE) loss for the Global Denoising Field and a causal loss for the Causal Constraint Injector, bridged by an interaction loss to foster information exchange. During inference, CGAD integrates the feature interaction mechanism at each denoising step, balancing the efficiency-stability trade-off via parallel generation with local directional context. The full sampling procedure is detailed in Appendix D.1.

Formally, the mutual reinforcement objective is defined as a weighted sum of three components:

$$\mathcal{L}_{\text{mutual}} = \mathcal{L}_{\text{DCE}} + \alpha \cdot \mathcal{L}_{\text{AR}} + \beta \cdot \mathcal{L}_{\text{interaction}} \quad (8)$$

where $\mathcal{L}_{\text{DCE}}$ is the diffusion loss, $\mathcal{L}_{\text{AR}}$ is the autoregressive loss, and $\mathcal{L}_{\text{interaction}}$ is the interaction loss. The hyperparameters α and β control the relative importance of each objective, with sensitivity results reported in appendix Section E.

The **diffusion loss** ($\mathcal{L}_{\text{DCE}}$) trains the Global Denoising Field to reconstruct the original clean data X_0 from its noisy version X_t using the t-denoising

cross-entropy objective:

$$L_{\text{DCE}}(X_t, X_0) = \sum_{i:X_t^i=[M]} \frac{-\sigma(t)e^{-\bar{\sigma}(t)}}{1-e^{-\bar{\sigma}(t)}} \cdot \log c_\theta(X_t)[i, X_0^i] \tag{9}$$

where $[M]$ represents a masked token, $\sigma(t)$ is the noise schedule, and $\bar{\sigma}(t) = \int_0^t \sigma(s)ds$. This objective directly optimizes the global, non-autoregressive denoising component introduced as the source of parallel generation in Section 1.

The **causal loss** ($\mathcal{L}_{\text{AR}}$) trains the Causal Constraint Injector to accurately predict tokens given a local context through standard negative log-likelihood minimization:

$$\begin{aligned}\mathcal{L}_{\text{AR}} &= -\log p_\phi(X_0^{[j:j+k-1]}|X_t) \\ &= -\sum_{m=j}^{j+k-1} \log p_\phi(X_0^m|X_0^{<m}, X_t^{UM})\end{aligned} \tag{10}$$

This supervises the injector to provide the local directional predictions used to address the Directionality Gap in Section 1, corresponding to the estimator $s_{AR}(b_t)$.

The **interaction loss** ($\mathcal{L}_{\text{interaction}}$) ensures effective information blending by minimizing the KL-divergence between the integrated output and a convex combination of component distributions:

$$\mathcal{L}_{\text{interaction}} = D_{\text{KL}}\big(P_{\text{CGAD}} \,\|\, \gamma P_{\text{AR}} + (1-\gamma)P_{\text{Diff}}\big) \tag{11}$$

where P_{AR} and P_{Diff} are obtained by applying separate linear projection layers (with parameters shared with the main model's output layer) followed by softmax to H^{AR} and H^{Diff}, respectively. P_{CGAD} is the distribution from the integrated output after the interaction mechanism. This encourages the fused output to respect both the directional and global-context signals identified in Section 1, following the estimator interpolation motivation in Theorem 2.3. The hyperparameter γ is fixed rather than learned or scheduled, with $\gamma = 0.4$ used for the reported full model and sensitivity results provided in appendix Section E. Additional interpretation and notes for this mechanism is provided in Appendix F.4.

4 Experiments

4.1 Experimental Setup

Datasets and Evaluation Protocol We evaluate CGAD on six diverse benchmarks to rigorously assess its capabilities across different domains. **FineWeb** (Penedo et al., 2024) (15T tokens) tests scalability on massive, high-quality web data. **Prolong** (Gao et al., 2025) specializes in long-context evaluation (context windows up to 512K), directly probing our method's ability to maintain coherence over extended sequences—a known weakness of standard diffusion models. **JSON-Mode-Eval/CFG** assesses structured generation using context-free grammar compliance, highlighting the importance of causal constraints for syntactic integrity. **WikiText2**, **WikiText103** (Merity et al., 2017), and **C4** (Raffel et al., 2020) provide standard language modeling benchmarks. All models are evaluated with a fixed length context window and no sequence overlap to ensure fair comparison. [1]

Implementation Details We trained small (350M total parameters) and medium (620M total parameters) models using consistent hyperparameters across datasets. Block size k=4 was chosen as optimal based on ablation studies. For parameter-controlled comparisons, all baseline models (GPT-2 (Radford et al., 2019), D3PM (Austin et al., 2021b), SEDD (Lou et al., 2024), RADD (Ou et al., 2025), and Block Diffusion, i.e. BD3-LM and SSD-LM (Arriola et al., 2025)) were configured to have approximately the same total parameter counts as our models: 350M parameters for small baselines and 620M parameters for medium baselines, and were trained from scratch using the same datasets and training configurations; GPT-2 is therefore used as a controlled autoregressive architectural baseline rather than as a comparison with pretrained modern language models.

Metrics We report standard perplexity (PPL) and accuracy (Acc) for quality assessment. For efficiency, we measure throughput (tokens/second), GPU cache hit rate (%), and GPU memory consumption; the GPU cache hit rate is obtained by profiling inference with NVIDIA Nsight and is a hardware-level metric rather than a hit rate of a model-side cache. The diffusion baselines use 100 denoising steps (NFE=100), whereas CGAD uses 25 denoising steps for inference unless otherwise specified. Throughput measurements use a batch size of 1; CGAD uses FlexAttention while baseline runtimes follow their official repository

[1] This may disadvantage AR baselines like GPT-2 that benefit from longer contexts, but allows a controlled assessment of their fixed-context modeling ability.

implementations, so throughput is an end-to-end measurement under the reported implementations rather than a backend-controlled architectural comparison. All efficiency metrics were measured on a single A100 GPU (40GB) generating 512 tokens with 1024 (small) and 4096 (medium) context.

4.2 Main Results

Tables 1 and 2 show that CGAD achieves competitive results across scales and datasets, consistent with the motivation for directional feature injection. On **Prolong**, CGAD nearly matches GPT-2's perplexity (54.23 vs. 54.79) while outperforming the reported diffusion baselines, suggesting improved long-context behavior in this evaluation setting. On **CFG**, CGAD reduces perplexity by 4 points compared to BD3-LM (110.89 vs. 115.03), showing an advantage over hard block partitioning for the reported structured-generation metric. The 40% accuracy improvement over GPT-2 on this task indicates stronger grammar-compliance accuracy under this setup. On **FineWeb**, CGAD achieves competitive perplexity with higher accuracy, providing evidence on web-text evaluation at the reported scale.

The medium model results (Table 2) show further gains on several datasets, with CGAD achieving highly competitive performance on Prolong and CFG while maintaining strong FineWeb performance. Additional scaling discussion is provided in Appendix F.2.

4.3 Efficiency Analysis

Table 3: Inference efficiency comparison (medium models).

Method	Throughput (Tokens/s)	GPU Cache Hit (%)	Memory (MiB)
GPT-2	1,200	67.2	2,341
SEDD	850	45.1	3,149
RADD	1,400	79.0	2,198
BD3-LM	1,250	72.3	2,012
CGAD	**1,900**	**83.1**	**1,975**

Table 3 reports measured inference efficiency for medium models. A brief interpretation of the throughput and GPU cache measurements is provided in Appendix F.3.

4.4 Ablation Studies

Table 4 evaluates each component using the same evaluation setup as the main results in Table 1. Removing causal injection causes performance degradation (+19.22 PPL), consistent with the intended contribution of directional features. Disabling cross-attention fusion also harms performance (+10.09 PPL), indicating that the interaction mechanism is useful in this setting. The block size ablation shows k=4 achieves the best measured balance—smaller blocks (k=1) improve perplexity marginally but sacrifice throughput, while larger blocks (k=16) degrade quality under this configuration. Ablations on training loss hyperparameters and CCI architecture are provided in appendix Section E.

Table 4: Ablation study on Prolong (small models) using the same evaluation setup as Table 1.

Ablation	PPL	Throughput (Tokens/s)	ΔPPL
Full CGAD	54.23	1,900	-
w/o Causal Injection	73.45	2,300	+19.22
w/o Cross-Attention Fusion	64.32	2,100	+10.09
k=1 (AR-like)	54.41	900	+0.18
k=16 (Less causal)	56.96	2,400	+2.73

4.5 Scaling Law Analysis

Table 5: Scaling trends across model sizes on Prolong. CGAD demonstrates strong performance at smaller scales and maintains competitive results as model size increases.

Method	125M Params		250M Params		520M Params	
	PPL	Acc	PPL	Acc	PPL	Acc
GPT-2	68.91	40.12	54.79	45.36	45.41	47.85
SEDD-Scale	85.23	35.89	60.56	43.22	52.49	46.92
BD3-LM	75.64	38.47	**53.18**	**47.20**	**50.05**	48.20
CGAD	**65.32**	**41.88**	55.11	45.63	50.11	**49.55**

Additional interpretation of the scaling trends in Table 5 is provided in Appendix F.2.

5 Conclusion

This work situates CGAD within the literature on discrete diffusion language models, designing it as a backbone architecture trained from scratch to improve training stability and controllability. CGAD targets barriers such as the "Optimization Cliff" and "Directionality Gap" by integrating causal inductive biases, with theoretical motivation and empirical measurements of gradient variance. This supports high-quality generation with efficient sampling in the evaluated settings, combining autoregressive directionality with diffusion-based global consistency.

Limitations

Our evaluation is limited in scale and scope. We train models from scratch at the reported small and medium sizes and evaluate them on six language modeling and structured-generation benchmarks. These experiments do not establish the behavior of CGAD at multi-billion-parameter scale, under instruction tuning, or on interactive downstream tasks. In addition, most reported results are based on a single fixed random seed; the limited multi-run validation on WikiText2 does not replace replicated evaluation across all datasets and configurations.

CGAD introduces a Causal Constraint Injector and a cross-attention interaction module in addition to the global denoising field. Although the reported throughput results show favorable trade-offs in our setting, the injector adds parameters and computation, and its benefit can depend on the sequence length, denoising schedule, block size, and implementation. The efficiency measurements are obtained on a single A100 GPU with fixed generation settings; they therefore should not be interpreted as hardware-independent latency or energy guarantees. Comparisons to systems using pretrained components, distillation, quantization, or specialized serving kernels remain open.

Our theoretical analysis characterizes variance reduction under assumptions about informative causal anchors and the relationship between local directional context and denoising uncertainty. In practice, a learned injector may provide weak or misleading guidance for ambiguous, multilingual, domain-shifted, or very long-range dependencies. The empirical improvements support the design hypothesis in the tested conditions, but do not imply that causal injection improves every generated token or universally eliminates diffusion optimization instability.

Ethical considerations

CGAD is an architectural study of general-purpose text generation, not a safety mechanism. Greater fluency or throughput can support useful applications, but can also make it cheaper to generate spam, misleading content, impersonation attempts, or other harmful text at scale. The reported language modeling and structured-generation metrics do not evaluate factuality, misuse resistance, or safety. Consequently, our results should not be interpreted as evidence that CGAD is appropriate for autonomous or high-stakes deployment.

Our experiments use cited language datasets including web-derived corpora (C4 and FineWeb) and Wikipedia-derived benchmarks (WikiText). Such corpora can contain social stereotypes, toxic language, personal information, and copyrighted material. A model trained on them may reproduce or amplify these properties. We do not perform dedicated audits of bias, toxicity, privacy memorization, copyright reproduction, or performance disparities across languages or demographic groups. Deployment-oriented work using this architecture should document data provenance and applicable terms, conduct risk-specific evaluations, and apply appropriate data and output safeguards.

Artifact provenance and terms of use FineWeb is an English Common Crawl-derived dataset whose public dataset card lists the ODC-By 1.0 license.[2] WikiText is English text extracted from verified Wikipedia articles; its public dataset card lists Creative Commons Attribution-ShareAlike and GNU Free Documentation licenses.[3] The experiments also report C4, ProLong, and JSON-Mode-Eval/CFG results. Because the current experiment record does not specify exact dataset snapshots, configurations, or applicable terms for these artifacts, we cannot make a stronger claim about license compliance or redistribution from the reported results alone. Reproduction or release of trained artifacts should first record the exact data versions, confirm the applicable source terms, and document whether any output or model distribution is permitted under those terms.

Training the reported 350M- and 620M-parameter models from scratch incurs computational and environmental costs. CGAD reduces the number of denoising steps used in our reported inference setting, but the single-hardware throughput study is not a lifecycle energy or emissions assessment. A full deployment evaluation should report training and serving resource consumption and weigh efficiency benefits against the risks of enabling larger-volume generation.

References

Marianne Arriola, Subham Sekhar Sahoo, Aaron Gokaslan, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Justin T Chiu, and Volodymyr Kuleshov. 2025. Block

[2] https://huggingface.co/datasets/HuggingFaceFW/fineweb

[3] https://huggingface.co/datasets/Salesforce/wikitext

diffusion: Interpolating between autoregressive and diffusion language models. In *The Thirteenth International Conference on Learning Representations*.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021a. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pages 17981–17993. Curran Associates, Inc.

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

Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. 2022. Analytic-DPM: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. In *International Conference on Learning Representations*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. In *Journal of machine learning research*.

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

Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, and 1 others. 2022. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. 2025. How to train long-context language models (effectively). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7376–7399, Vienna, Austria. Association for Computational Linguistics.

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

Chihan Huang and Hao Tang. 2025. Ctrldiff: Boosting large diffusion language models with dynamic block prediction and controllable generation. *Preprint*, arXiv:2505.14455.

Zemin Huang, Zhiyang Chen, Zijun Wang, Tiancheng Li, and Guo-Jun Qi. 2025. Reinforcing the diffusion chain of lateral thought with diffusion language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Hyukhun Koh, Minha Jhang, Dohyung Kim, Sangmook Lee, and Kyomin Jung. 2025. Conditional [MASK] discrete diffusion language model. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8899–8923, Suzhou, China. Association for Computational Linguistics.

Aaron Lou, Chenlin Meng, and Stefano Ermon. 2024. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Forty-first International Conference on Machine Learning*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. 2025. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *The Thirteenth International Conference on Learning Representations*.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, pages 30811–30849. Curran Associates, Inc.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Robin Schmidt, Telmo Pires, Stephan Peitz, and Jonas Lööf. 2022. Non-autoregressive neural machine translation: A call for clarity. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2785–2799, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. 2023. Diffusion models: A comprehensive survey of methods and applications. *ACM computing surveys*, 56(4):1–39.

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

Yangtian Zhang, Sizhuang He, Daniel Levine, Lawrence Zhao, David Zhang, Syed A Rizvi, Shiyang Zhang, Emanuele Zappala, Rex Ying, and David van Dijk. 2025. Non-markovian discrete diffusion with causal

language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Cai Zhou, Chenxiao Yang, Yi Hu, Chenyu Wang, Chubin Zhang, Muhan Zhang, Lester Mackey, Tommi Jaakkola, Stephen Bates, and Dinghuai Zhang. 2025. Coevolutionary continuous discrete diffusion: Make your diffusion language model a latent reasoner. *Preprint*, arXiv:2510.03206.

Helena Čeović, Marin Šilić, Goran Delač, and Klemo Vladimir. 2023. An overview of diffusion models for text generation. In *2023 46th MIPRO ICT and Electronics Convention (MIPRO)*, pages 941–946.

A Responsible NLP Research Checklist

This appendix records answers to the ARR Responsible NLP Research Checklist. The answers are based on the experiments and artifacts documented in this manuscript. Where the experimental record does not contain information needed for a positive answer, we state that gap explicitly.

A.1 A. For Every Submission

A1. Did you describe the limitations of your work? Yes. The unnumbered *Limitations* section following the conclusion specifies the evaluated model scales and tasks, the reliance on mostly single-seed results, the hardware and implementation dependence of the throughput claims, and the assumptions under which the theoretical interpretation applies. In particular, it limits the conclusions to the evaluated settings and does not claim reliable behavior for multilingual, domain-shifted, interactive, or high-stakes applications.

A2. Did you discuss potential risks of your work? Yes. The *Ethical considerations* section discusses dual-use risks from efficient general-purpose generation, including misleading or harmful content, spam, and impersonation attempts. It also describes risks from web-derived training material, including bias, toxicity, personal information, and copyrighted text, as well as the absence of dedicated bias, privacy, memorization, copyright-reproduction, and safety audits. Finally, it discusses compute and environmental costs and states that the reported throughput evaluation is not an energy or emissions assessment.

A.2 B. Scientific Artifacts

B1. Did you cite the creators of artifacts you used? No. The manuscript cites the reported datasets and modeled baselines in Section 4.1, including FineWeb, WikiText, C4, ProLong, GPT-2, D3PM, SEDD, RADD, and Block Diffusion. However, the current experiment record does not identify exact versions or snapshots of each dataset, baseline implementation, or the JSON-Mode-Eval/CFG artifact. We therefore cannot claim complete artifact identification as requested by the checklist.

B2. Did you discuss licenses or terms for use and distribution of artifacts? No. The *Ethical considerations* section reports publicly documented terms for FineWeb and WikiText. The manuscript does not record the exact versions or applicable terms for C4, ProLong, JSON-Mode-Eval/CFG, or the baseline implementations used in the reported experiments. It also does not release a new model or dataset with distribution terms. Therefore, terms of use and redistribution have not been established for every artifact used.

B3. Did you discuss whether artifact use is consistent with intended use? No. This paper uses existing text resources for language-modeling and structured-generation evaluation, which is consistent with the general research setting described in Section 4.1. Nevertheless, in the absence of exact artifact versions and complete terms-of-use records, the manuscript cannot establish compatibility for every reported dataset or any future release of trained weights. The *Ethical considerations* section limits any deployment or redistribution claim accordingly.

B4. Did you report checks for identifying or offensive content and privacy protections? No. No additional screening of the used text corpora for personally identifying information, offensive content, or memorized training text is reported. No new anonymization procedure is reported. The *Ethical considerations* section identifies this absence as a limitation and warns that web-derived text may contain such material.

B5. Did you provide documentation of artifact coverage? No. Section 4.1 identifies the broad domains of the six reported evaluations, and the *Ethical considerations* section records that FineWeb and WikiText are English text resources. The manuscript does not provide a complete data statement covering language composition, geographic or demographic representation, topic cov-

erage, or the intended represented population for all artifacts.

B6. Did you report relevant dataset statistics and splits? No. The manuscript states the broad benchmark set and reports FineWeb's corpus scale, but it does not state the exact train, validation, and test split sizes or selected snapshots/configurations for every evaluated dataset. This information is therefore not available from the reported experiment record.

A.3 C. Computational Experiments

C1. Did you report model parameters, total computational budget, and infrastructure? No. Section 4.1 and the Reproducibility Statement report model sizes (approximately 350M and 620M parameters), and Section 4.1 reports that efficiency measurements used one NVIDIA A100 GPU with 40GB memory, a batch size of 1, stated contexts, and stated denoising-step counts. The manuscript does not record total GPU-hours, full training hardware parallelism, hyperparameter-search cost, or total compute across all experiments; consequently it does not provide the total computational budget requested by the checklist.

C2. Did you report the experimental setup, hyperparameter search, and best-found values? No. The Reproducibility Statement reports the optimizer, learning rates, warmup, decay schedule, sequence lengths, batch size, architecture settings, seed, and the reported sensitivity study for the objective coefficients. Section 4.4 reports the block-size ablation. The manuscript does not document the full search process, stopped or unsuccessful trials, dataset preprocessing, checkpoint-selection criterion, or which held-out data were used to make every model-selection choice.

C3. Did you report descriptive statistics and distinguish single runs from repeated experiments? Yes. The Reproducibility Statement records that most reported results are from single runs with fixed random seed 42. It additionally reports three independent runs for the CGAD small and medium configurations on WikiText2, with standard deviations below 0.5% for perplexity and 0.3% for accuracy. The *Limitations* section explicitly states that this limited multi-run validation does not replace replicated evaluation for all reported configurations.

C4. Did you report implementation and package versions? No. Section 4.1 identifies FlexAttention for CGAD inference and NVIDIA Nsight for the hardware-cache measurement, and states that baseline runtimes follow official repository implementations. It does not specify package versions, repository commits, tokenizer configuration, framework version, CUDA/software environment, or modifications needed to reproduce each baseline.

A.4 D. Human Participants and Annotation

D1–D5. Did you report human-participant or annotation procedures? Not applicable. The reported study does not conduct experiments with recruited human participants or commission new human annotations. It evaluates models on existing language resources and automatic metrics. Therefore, there are no participant instructions, recruitment or compensation details, consent procedure for newly recruited participants, institutional ethics-review decision for a newly conducted human-subjects study, or annotator demographics to report. Risks inherited from existing corpora are discussed in the *Ethical considerations* section.

A.5 E. AI Assistants

E1. If you used AI assistants, did you include information about your use? Yes. An AI assistant was used during manuscript preparation to help draft and revise wording in the *Limitations*, *Ethical considerations*, and this checklist appendix. This use was limited to writing assistance and policy-oriented organization of disclosures; the assistant is not an author and is not a source of experimental evidence. The human authors remain responsible for verifying every factual statement, citation, reported result, and policy representation before submission.

B Reproducibility Statement

To facilitate the reproducibility of our work, we have made extensive efforts to document our methodology and experimental setup. We provide a comprehensive summary of the key hyperparameters for our small and medium model configurations in Table 6.

The model architecture consists of two main components: (1) a Global Denoising Field with 250M (small) or 520M (medium) parameters, and (2) a Causal Constraint Injector implemented as a

shallow decoder with hidden dimension $d_ar = 256$, sharing embeddings with the main model and using 100M parameters of the total budget. This results in total parameter counts of 350M for small models and 620M for medium models. Training used AdamW optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$), linear warmup (10k steps), cosine decay and constant random seed(42). The base learning rate was 1e-4 (small) and 5e-5 (medium). For the small configuration, the ratio of parameters in the Causal Constraint Injector to the Global Denoising Field is roughly 2:5; at each scale, the injector is counted within the stated total budget. Due to computational constraints, we report results from single runs with fixed random seeds. However, we performed three independent runs on a subset of configurations (CGAD small and medium on WikiText2) and observed standard deviations of less than 0.5% for perplexity and 0.3% for accuracy, indicating stable and reproducible results.

Table 6: Key model hyperparameters and training configurations for the CGAD architecture, covering both small and medium model variants along with Causal Constraint Injector (CCI) and Global Denoising Field (GDF) components.

Hyperparameter	**Small Model**	**Medium Model**
Training Iterations	250,000	250,000
Batch Size	128	128
Sequence Length	1024	4096
Optimizer	AdamW ($\beta_1 = 0.9, \beta_2 = 0.999$)	
Base Learning Rate	1×10^{-4}	5×10^{-5}
LR Scheduler	Linear Warmup (10k steps) + Cosine Decay	
GDF Layers	15	20
GDF Hidden Dim	768	1440
GDF Attention Heads	12	24
CCI Layers	6	10
CCI Hidden Dim	512	1024
CCI Attention Heads	8	16
Vocabulary Size	50,265	

C Related Works and Discussions

Causal Structure as a Backbone Principle Prior work demonstrates that incorporating causal signals can improve coherence in diffusion language models, either through external guidance from pretrained causal models (Zhang et al., 2025) or via static causal attention patterns. However, these methods introduce causality as an auxiliary constraint or inference-time bias, rather than through the jointly trained feature pathway used here, and typically rely on pretrained components. CGAD instead injects dynamically generated causal features directly into the diffusion backbone via cross-attention. This design enables end-to-end training from scratch and permits causal directionality to influence the denoising field throughout all noise levels, motivating our gradient-variance and quality evaluations.

Stabilization Without Reasoning Supervision Several recent approaches seek to stabilize diffusion trajectories through explicit trajectory optimization or hybrid reasoning mechanisms, such as DCoLT (Huang et al., 2025) and coevolutionary continuous–discrete diffusion (Zhou et al., 2025). While effective in reasoning-oriented benchmarks, these methods introduce additional objectives, supervision signals, or architectural complexity beyond our setting of general-purpose text generation. In contrast, CGAD targets the loss of directional signal and measured gradient variance in early denoising steps by embedding causal features into the backbone.

Structural Control at Scale Block-based and constraint-driven diffusion methods improve efficiency and controllability but often rely on fixed partitioning or masking schemes (Huang and Tang, 2025; Koh et al., 2025). CGAD shares the intuition that structural anchors are useful, yet reframes their role: blocks act as soft carriers of causal features rather than rigid decoding units. This attention-based constraint injection retains a global-context pathway while supplying directional information during denoising.

D Supplementary Methodology and Experiments

D.1 Inference Algorithm with Interaction Mechanism

The inference process in CGAD integrates the feature interaction mechanism at each denoising step, enabling efficient parallel generation while maintaining local coherence. Algorithm 1 outlines the complete sampling procedure, which uses the *syntactic anchors* provided by the Causal Constraint Injector in the inference configuration measured in Table 3. The algorithm includes a **state-reuse mechanism** (Line 6-12), which can skip redundant global network evaluations when the same unmasked token pattern has been processed at the same time step; this mechanism is distinct from the GPU cache hit rate in Table 3, which is measured by NVIDIA Nsight during inference. The block size k is a key trade-off, detailed in the ap-

pendix. We ablate its impact on both performance and efficiency in Section 4.4.

D.2 Analysis of CCI Mechanism Effectiveness

The experimental results presented in Figures 1-6 evaluate the Causal Constraint Injection (CCI) mechanism within our CGAD framework. Across the reported metrics and model scales, CGAD with CCI outperforms the displayed baseline methods (RADD, BD3-LM) and the ablated version of CGAD without CCI, supporting the inclusion of this architectural component.

Table 7: Comparative performance summary of different methods across key metrics

Method	Gradient Variance	PPL	Training Loss	Accuracy
RADD	Highest	Highest	Highest	Lowest
BD3-LM	Medium	Medium	Medium	Medium
CGAD (w/o CCI)	Medium-High	Medium	Medium-High	Medium
CGAD (w/ CCI)	**Lowest**	**Lowest**	**Lowest**	**Highest**

D.2.1 Gradient Stability Enhancement

The gradient variance analysis (Figures 2 and 3) provides a direct measurement associated with the CCI mechanism. We calculate gradient variance as the average variance of gradient L2 norms across all model parameters over a sliding window of 100 training steps, measured during the middle phase of training (after 50K steps). CGAD with CCI shows lower measured gradient variance than the compared methods, with more concentrated distributions and lower averages. This pattern appears at both small and medium model scales in the reported runs.

The high gradient variance in RADD (approximately 2-3× higher than CGAD with CCI) is consistent with the “Optimization Cliff" described in Theorem 2.1, where position-symmetric absorbing denoising in high-dimensional combinatorial spaces can be associated with unstable training dynamics. The CCI mechanism supplies directional features during denoising, and the plots measure a corresponding reduction in gradient variation under the reported configuration. This observation is consistent with Theorem 2.3, while not by itself establishing a universal causal reduction in estimation variance.

D.2.2 Language Modeling Performance

The perplexity (PPL) results (Figures 4 and 5) show quality trends alongside the gradient measurements. CGAD with CCI achieves the lowest PPL across both displayed model scales, with particularly notable gains in the later stages of training.

The performance gap between CGAD with CCI and the ablated version without CCI is informative. This difference supports the utility of the causal guidance mechanism within the evaluated architecture. The advantage across the reported datasets (including WikiText, C4, and FineWeb) motivates evaluation across additional linguistic domains and data distributions.

D.2.3 Training Effectiveness and Convergence

The training loss and accuracy curves (Figures 6 and 7) compare optimization behavior with and without CCI under the reported settings. CGAD with CCI exhibits the most rapid decrease in training loss during early stages, reaching lower plateau values more quickly than other methods. This measured convergence pattern is relevant to the efficiency concerns introduced for discrete diffusion training.

The accuracy metrics show a complementary pattern, with CGAD with CCI achieving higher final accuracy values and maintaining more stable learning trajectories throughout training. Together with the ablations, these measurements indicate that coupling the global denoising field and causal constraints is beneficial under the evaluated parameter budgets and training configurations.

D.2.4 Mechanism of Action

The performance of CGAD with CCI is consistent with its intended role in addressing the “Directionality Gap" identified in Theorem 2.2. By incorporating causal inductive biases into the diffusion process, CCI provides the model with directional features that can guide denoising toward linguistically plausible predictions. This guidance augments the position-symmetric absorbing process with a learned directional signal.

The cross-attention mechanism that implements the CCI allows for dynamic, context-aware integration of causal constraints. Unlike fixed-weight interpolation schemes, this approach enables the model to selectively emphasize relevant syntactic guidance based on local context needs, providing the flexibility to handle the varied dependency structures found in natural language while maintaining the global consistency benefits of diffusion.

D.2.5 Implications for Practical Deployment

The performance advantages of CGAD with CCI across the evaluated metrics indicate practical po-

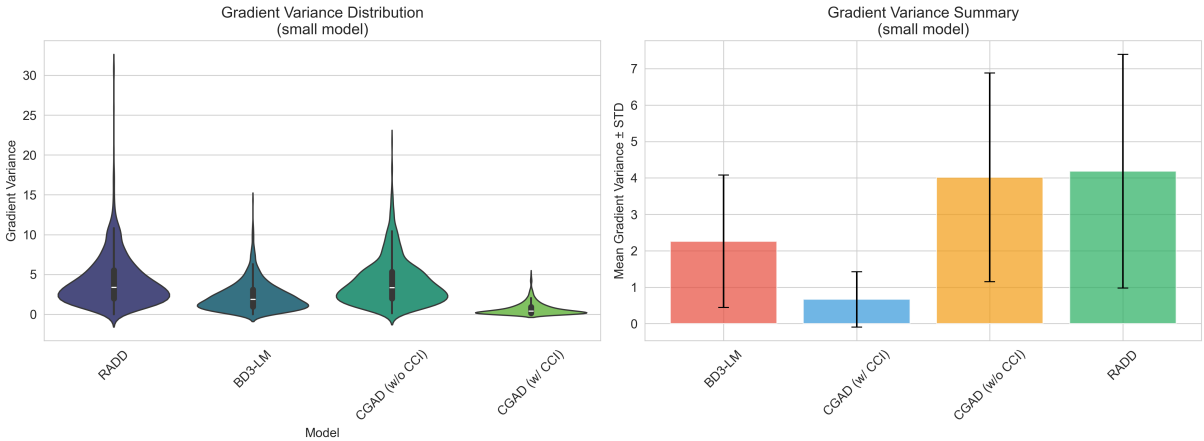

Figure 2: Gradient Variance Comparison Across Methods (Small Model)

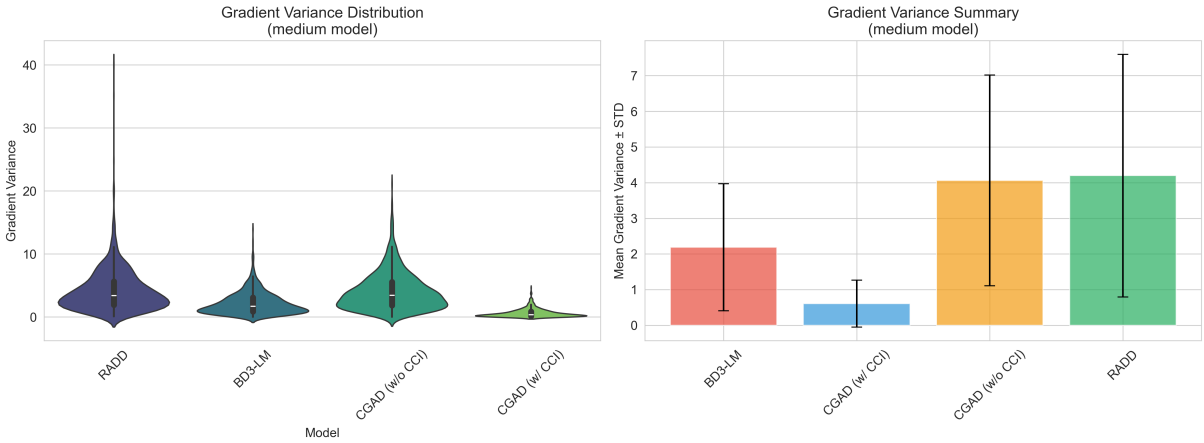

Figure 3: Gradient Variance Comparison Across Methods (Medium Model)

tential for text generation applications. The measured reduction in gradient variance is associated with more stable training curves in these experiments, while the reported throughput measures inference behavior. The enhanced final performance indicates that models incorporating CCI can improve quality under the evaluated model and inference configurations.

Furthermore, the occurrence of improvements across the two reported model scales suggests that the CCI mechanism remains effective within the evaluated range. Evaluation at substantially larger scales is left for future work.

In conclusion, the experimental analysis supports the Causal Constraint Injection mechanism as an effective component for the reported discrete diffusion language modeling settings. By combining global diffusion representations with local autoregressive features, CGAD with CCI offers a principled architecture for studying optimization and quality trade-offs in this setting.

E Hyperparameter Sensitivity Analysis of the Mutual Reinforcement Objective

The mutual reinforcement training objective introduced in Section 3.2 is central to CGAD's performance, with hyperparameters α, β, and γ controlling the balance between the diffusion, autoregressive, and interaction components. This section presents a comprehensive sensitivity analysis of these hyperparameters for both small and medium model configurations. The complete loss function is defined as:

$$\mathcal{L}_{\text{mutual}} = \mathcal{L}_{\text{DCE}} + \alpha \cdot \mathcal{L}_{\text{AR}} + \beta \cdot \mathcal{L}_{\text{interaction}} \quad (12)$$

where $\mathcal{L}_{\text{DCE}}$ is the diffusion cross-entropy loss, $\mathcal{L}_{\text{AR}}$ is the autoregressive loss from the Causal Constraint Injector, and $\mathcal{L}_{\text{interaction}}$ is the interaction loss that encourages effective blending between components. The **interaction loss** ($\mathcal{L}_{\text{interaction}}$) ensures effective information blending by minimizing the

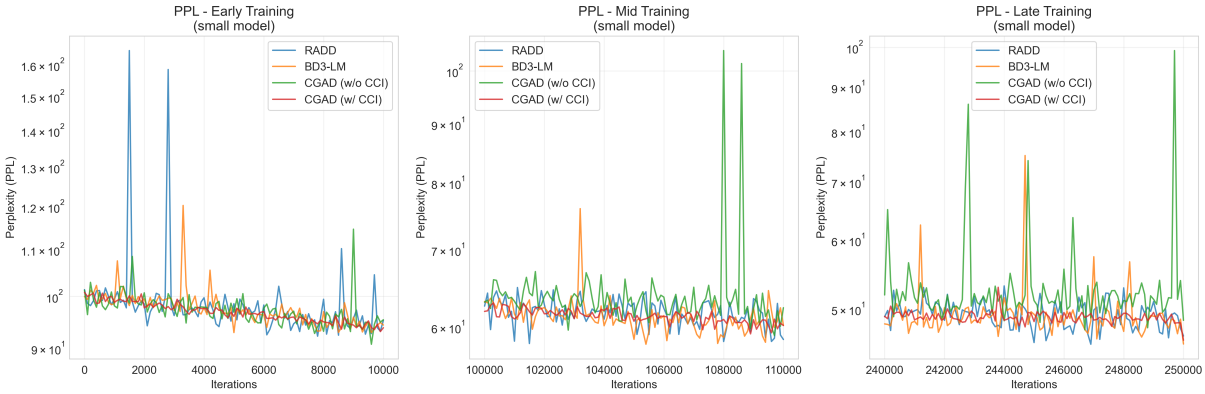

Figure 4: Perplexity Comparison Across Methods (Small Model)

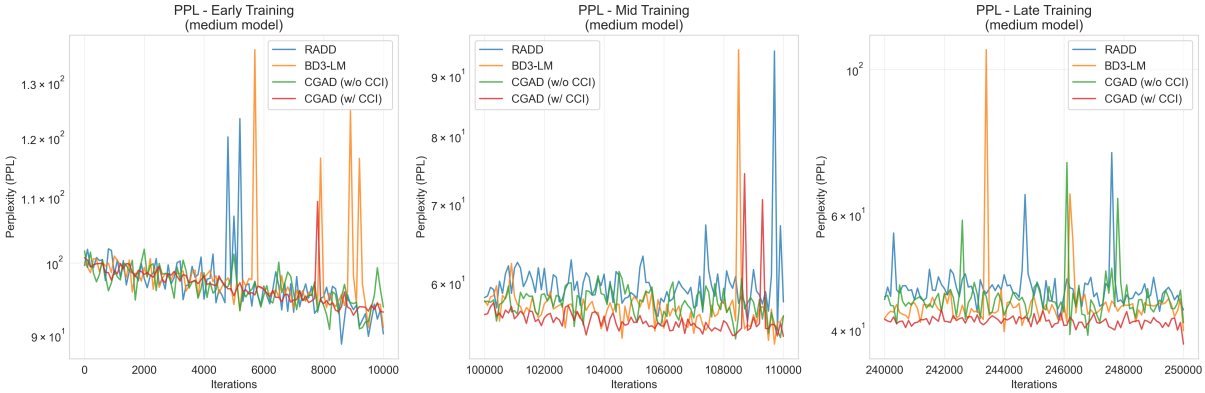

Figure 5: Perplexity Comparison Across Methods (Medium Model)

KL-divergence between the integrated output and a convex combination of component distributions:

$$\mathcal{L}_{\text{interaction}} = D_{\text{KL}}\big(P_{\text{CGAD}} \,\|\, \gamma P_{\text{AR}} + (1-\gamma) P_{\text{Diff}}\big) \tag{13}$$

This explicitly encourages the fused output to respect both components, realizing the estimator interpolation principle from Theorem 2.3. The full model configuration uses $\alpha = 0.5$, $\beta = 0.5$, and $\gamma = 0.4$ as reference points.

The sensitivity analysis reveals several important patterns. First, both small and medium models benefit significantly from the full mutual reinforcement objective compared to using any single component in isolation. The DCE-only configuration performs worst, consistent with the Optimization Cliff motivation for position-symmetric absorbing denoising in Theorem 2.1. Adding either the AR component ($\alpha > 0$) or interaction component ($\beta > 0$) individually improves performance, but the combination provides the best results.

The analysis of γ, which controls the balance between the AR and diffusion components in the interaction loss, shows that the optimal value is $\gamma = 0.4$ for both model sizes. This indicates a slight preference for the AR component's distribution in the convex combination, suggesting that the causal inductive bias provides more reliable guidance for the diffusion process. For the small model (Table 8), deviations from $\gamma = 0.4$ lead to performance degradation: $\gamma = 0.2$ increases PPL by 0.45 points, while $\gamma = 0.8$ increases PPL by 0.65 points. The medium model (Table 9) exhibits a similar trend but with smaller deviations, indicating that larger models are more robust to suboptimal γ settings due to their increased capacity.

For the small model, the full configuration ($\alpha = 0.5$, $\beta = 0.5$, $\gamma = 0.4$) achieves the lowest perplexity (48.35) and highest accuracy (52.31%), with competing configurations showing perplexity increases of 0.42-4.01 points. Interestingly, the asymmetric configuration with high AR emphasis ($\alpha = 1.0$, $\beta = 0.25$, $\gamma = 0.4$) achieves competitive performance (PPL 48.77, Acc 52.04%) with only a minor degradation, suggesting that for smaller models, the causal constraints from the AR component may be particularly valuable for stabilizing the diffusion process.

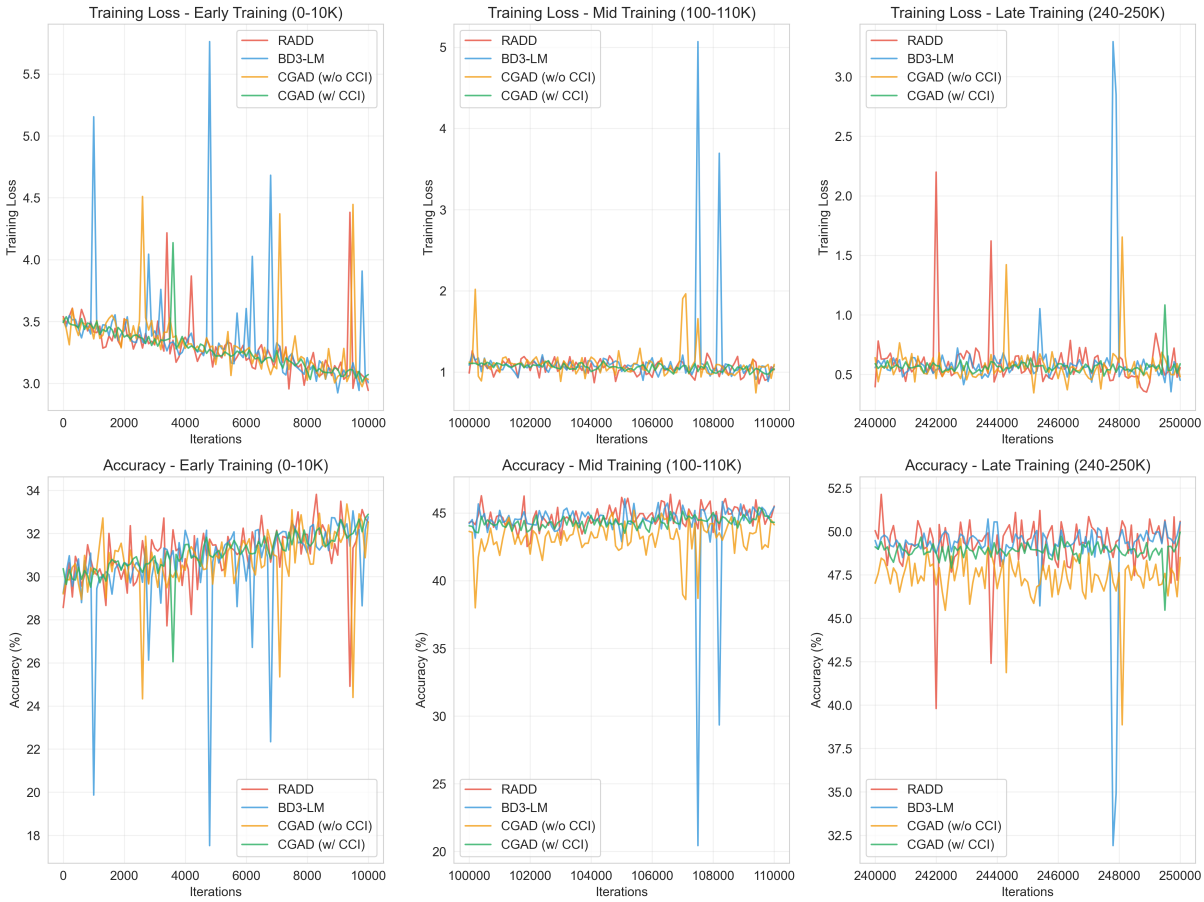

Figure 6: Training Stability Comparison Across Methods (Small Model)

The medium model shows even more pronounced benefits from the full mutual reinforcement objective, with the full configuration achieving PPL 37.35 and Acc 54.31%. The performance gap between configurations narrows compared to the small model, indicating that larger models can better compensate for suboptimal hyperparameter settings through increased capacity. However, the full configuration still provides the best balance, with the asymmetric high-AR configuration again showing strong performance (PPL 37.42, Acc 54.25%).

Notably, in some cases, alternative configurations can achieve slightly better perplexity at the cost of accuracy, or vice versa. For instance, the strong AR+Interaction configuration ($\alpha = 1.0$, $\beta = 1.0$, $\gamma = 0.4$) for the medium model achieves excellent perplexity (37.58) but slightly lower accuracy (54.12%) compared to the full model. This trade-off reflects the different optimization landscapes shaped by the hyperparameters: higher α values strengthen the causal inductive bias, potentially improving local coherence at the expense of global consistency, while higher β values promote better integration but may dilute the directional guidance. The optimal $\gamma = 0.4$ further fine-tunes this balance by slightly favoring the AR distribution, which provides more stable syntactic anchors.

The results demonstrate that while the full model configuration with $\alpha = 0.5$, $\beta = 0.5$, and $\gamma = 0.4$ provides robust performance across both model sizes, there is some flexibility in the optimal settings. The medium model shows greater tolerance to hyperparameter variation, likely due to its increased capacity to learn the appropriate balance between components implicitly. These findings suggest that for practical applications, a modest grid search around the default values may be beneficial, particularly for smaller models where the balance between components is more critical. The sensitivity to γ underscores the importance of the interaction loss in effectively blending the causal and diffusion components, as formalized in Theorem 2.3.

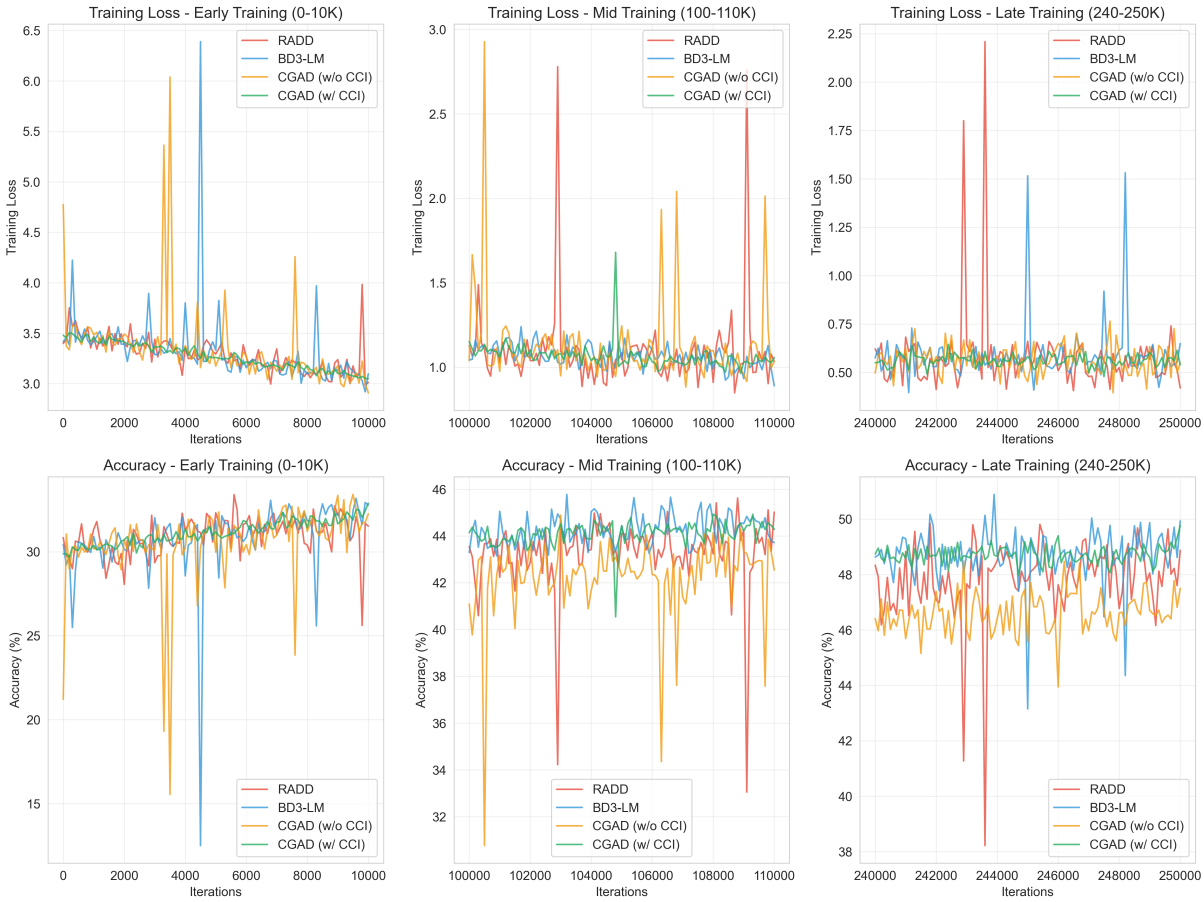

Figure 7: Training Stability Comparison Across Methods (Medium Model)

F Supplementary Details and Discussions

F.1 Discussions on Block Diffusion

A critical distinction between CGAD and Block Diffusion (BD3-LM) lies in their fundamental approaches to incorporating block-level processing. While both employ block-based generation, BD3-LM uses **hard partitioning** with isolated block processing, treating each block as an independent generation task. In contrast, CGAD implements **soft constraint injection** through cross-attention, allowing continuous information flow across block boundaries via the global denoising field.

This architectural difference is accompanied by measured differences in cross-block evaluation settings. Under the reported inference workload, NVIDIA Nsight profiles a 10.8 percentage-point higher GPU cache hit rate for CGAD than for BD3-LM (83.1% vs. 72.3%), indicating more favorable hardware-cache utilization for this implementation. For CFG generation, CGAD achieves 3.2 points lower perplexity, providing evidence of stronger performance on nested structures that span multiple blocks in this evaluation. The global attention mechanism in CGAD's denoising field retains information flow across block boundaries, unlike isolated block processing.

These results reflect distinct architectural approaches: BD3-LM emphasizes block-level efficiency, while CGAD combines blockwise directional features with a global denoising representation.

F.2 Supplementary Scaling Commentary

The scaling results in Table 5 indicate that CGAD remains competitive at the reported 520M scale, while the comparison also suggests that larger models can partially absorb suboptimal settings through capacity. The current study is limited to total model configurations up to 620M parameters because of computational constraints, so behavior at substantially larger scales remains an open question.

To investigate the scaling behavior of our method, we analyzed results across model sizes of 125M, 250M (the aforementioned small-sized CGAD model), and 520M parameters (the medium-sized one) in Table 5. While diffusion baselines like

Table 8: Hyperparameter sensitivity analysis for the **small model** on the WikiText2 validation set.

Configuration	α	β	γ	WikiText2		ΔPPL
				PPL	Acc	
DCE only	0.0	0.0	–	52.34	48.15	+4.01
DCE + AR	0.5	0.0	–	49.87	50.92	+1.54
DCE + Interaction	0.0	0.5	0.4	50.12	50.64	+1.79
DCE + AR (weak)	0.25	0.0	–	50.95	49.87	+2.62
DCE + AR (strong)	1.0	0.0	–	49.45	51.38	+1.12
DCE + Interaction (weak)	0.0	0.25	0.4	50.76	50.11	+2.43
DCE + Interaction (strong)	0.0	1.0	0.4	49.88	51.05	+1.55
DCE + AR + Interaction ($\gamma = 0.2$)	0.5	0.5	0.2	48.80	52.00	+0.45
DCE + AR + Interaction ($\gamma = 0.6$)	0.5	0.5	0.6	48.60	52.10	+0.25
DCE + AR + Interaction ($\gamma = 0.8$)	0.5	0.5	0.8	49.00	51.80	+0.65
Full Model	**0.5**	**0.5**	**0.4**	**48.35**	**52.31**	**0.00**
DCE + AR + Interaction (weak)	0.25	0.25	0.4	49.12	51.67	+0.77
DCE + AR + Interaction (strong)	1.0	1.0	0.4	48.94	51.89	+0.59
Asymmetric (high AR)	1.0	0.25	0.4	48.77	52.04	+0.42
Asymmetric (high Interaction)	0.25	1.0	0.4	49.03	51.92	+0.68

Table 9: Hyperparameter sensitivity analysis for the **medium model** on the WikiText2 validation set.

Configuration	α	β	γ	WikiText2		ΔPPL
				PPL	Acc	
DCE only	0.0	0.0	–	40.15	51.28	+2.80
DCE + AR	0.5	0.0	–	38.64	53.05	+1.29
DCE + Interaction	0.0	0.5	0.4	38.89	52.87	+1.54
DCE + AR (weak)	0.25	0.0	–	39.27	52.61	+1.92
DCE + AR (strong)	1.0	0.0	–	38.31	53.44	+0.96
DCE + Interaction (weak)	0.0	0.25	0.4	39.15	52.73	+1.80
DCE + Interaction (strong)	0.0	1.0	0.4	38.67	53.22	+1.32
DCE + AR + Interaction ($\gamma = 0.2$)	0.5	0.5	0.2	37.60	54.10	+0.25
DCE + AR + Interaction ($\gamma = 0.6$)	0.5	0.5	0.6	37.50	54.20	+0.15
DCE + AR + Interaction ($\gamma = 0.8$)	0.5	0.5	0.8	37.80	53.95	+0.45
Full Model	**0.5**	**0.5**	**0.4**	**37.35**	**54.31**	**0.00**
DCE + AR + Interaction (weak)	0.25	0.25	0.4	38.02	53.89	+0.67
DCE + AR + Interaction (strong)	1.0	1.0	0.4	37.58	54.12	+0.23
Asymmetric (high AR)	1.0	0.25	0.4	37.42	54.25	+0.07
Asymmetric (high Interaction)	0.25	1.0	0.4	37.86	53.97	+0.51

PPL (perplexity, ↓) and Accuracy (Acc, ↑) are reported. The full model configuration ($\alpha = 0.5$, $\beta = 0.5$, $\gamma = 0.4$) is highlighted.

SEDD-Scale show larger relative improvements when scaling from 125M to 520M parameters, CGAD maintains strong competitive performance across all reported scales, achieving the highest accuracy (49.55%) at the 520M scale.

F.3 Efficiency Commentary

Table 3 reports the inference measurements under the stated implementation settings. The throughput difference relative to GPT-2 and RADD reflects the combined effect of using 25 denoising steps for CGAD, 100 denoising steps for the diffusion baselines, and the reported inference backends; the GPU cache hit rate is a separate Nsight hardware metric and should not be interpreted as skipped denoiser evaluations.

F.4 Supplementary Objective Commentary

The mutual reinforcement objective is designed to improve training efficiency. The causal constraints provide directional features that may reduce the number of denoising steps required to achieve a target perplexity $\mathcal{P}$. We use the following mutual-information-based relation as a modeling motivation for this behavior. Specifically:

$$n_{\text{fused}} = n_0 \cdot \exp\left(-\beta \frac{I(X_t; X_0^{\text{AR}})}{H(X_t)}\right) \tag{14}$$

For additional stability during early training stages, we optionally employ the λ-denoising cross-entropy loss as an alternative to t-DCE:

$$L_{\lambda\text{-DCE}}(X_\lambda, X_0) = -\sum_{i:X_\lambda^i=[M]} \lambda \cdot \log c_\theta(X_\lambda)[i, X_0^i] \tag{15}$$

Here, n_0 is the baseline number of steps without causal constraints, β is a scaling factor, $I(\cdot;\cdot)$ denotes mutual information, and $H(\cdot)$ denotes entropy. This relation motivates evaluating whether the architecture trained via $\mathcal{L}_{\text{mutual}}$ is associated with lower denoising complexity and hence with the practical efficiency-stability trade-off stated in Section 1.

Separately, $\lambda \sim U([0,1])$ determines the masking probability. We anneal the use of λ-DCE in favor of t-DCE as training progresses, aligning later training with the t-DCE objective.

F.5 State-Reuse Mechanism in the Bidirectional Global Denoising Field

The state-reuse mechanism in CGAD's bidirectional Global Denoising Field is designed to minimize redundant computations while maintaining the integrity of attention patterns across denoising steps. This algorithmic mechanism is separate from the GPU cache hit rate reported in Table 3, which is obtained from NVIDIA Nsight profiling and does not measure lookups in C_{cache}. Below we provide a detailed explanation of the state-reuse implementation.

F.5.1 State-Reuse Structure and Update Logic

The cache stores the hidden state representations of tokens that remain unchanged between consecutive denoising steps. Specifically, for a sequence X_t at time t, we maintain:

$$C_{\text{cache}} = \{\mathbf{h}_i^t : X_t^i \neq [M]\} \tag{16}$$

where $\mathbf{h}_i^t \in \mathbb{R}^{d_{\text{model}}}$ is the hidden representation of the i-th token produced by the Global Denoising

Field's encoder. The cache update condition is:

$$\text{UpdateCache} = \begin{cases} \text{True} & \text{if } \exists i : X_t^i \neq X_{t-\Delta t}^i \\ \text{False} & \text{otherwise} \end{cases} \tag{17}$$

This conservative approach ensures that any change in the token sequence triggers a complete recomputation of the attention map. While this may seem counterintuitive for efficiency, in our later denoising stages most tokens remain unchanged across successive updates, making recomputation less frequent.

F.5.2 Attention Computation with State Reuse

When the cache is valid (no tokens changed), we reuse the cached hidden states. The attention computation for position j becomes:

$$\mathbf{h}_j^{\text{new}} = \text{Attention}(Q_j, K_{\text{cache}}, V_{\text{cache}}) \tag{18}$$

where K_{cache} and V_{cache} are constructed from the cached hidden states. The pseudo-code below illustrates the complete state-reuse mechanism:

Algorithm 2 State-Reuse Mechanism in Global Denoising Field

```
procedure GlobalDenoise(X_t, C_cache)
    changed ← False
    for i ∈ {1, ..., d} do
        if X_t^i ≠ X_{t−Δt}^i then ▷ Check if token i has changed
            changed ← True
            break
        end if
    end for
    if changed or C_cache = ∅ then
        H_t^Diff ← Encoder_θ(X_t)  ▷ Compute new hidden states: H_t^Diff = f_θ(X_t)
        C_cache ← H_t^Diff  ▷ Update cache: C_cache^(new) = H_t^Diff
    else
        H_t^Diff ← C_cache ▷ Reuse cached states: H_t^Diff = C_cache^(old)
    end if
    return H_t^Diff, C_cache
end procedure
```

The potential acceleration comes from the fact that in typical denoising trajectories, especially in later stages, the majority of tokens can remain unchanged for multiple consecutive steps. The 83.1% value in Table 3 is instead the inference-time GPU cache hit rate obtained from NVIDIA Nsight and should not be interpreted as the rate of state reuse in this procedure.

F.6 Sensitivity Analysis of Block Size k

The optimal block size $k = 4$ emerges from a trade-off between parallel efficiency and causal strength. We analyze this sensitivity from both theoretical and empirical perspectives.

F.6.1 Theoretical Framework

For the optional state-reuse analysis, the expected number of function evaluations (E-NFEs) can be modeled as:

$$\text{E-NFEs} = n \cdot \left(1 + \frac{d}{k} \cdot \frac{1}{\text{cache-hit-rate}}\right) \tag{19}$$

where n is the number of denoising steps, d is sequence length, and cache-hit-rate denotes hits in the state-reuse mechanism C_{cache}, not the GPU hardware counter in Table 3. In this analytical model, a larger k increases parallelism but can reduce state reuse due to more frequent token changes.

F.6.2 Dataset-Specific Optimal k

Our analysis reveals that the optimal k correlates with linguistic units in the data:

- For natural language text (average word length 4-5 characters, phrase length 2-3 words), $k = 4$ aligns well with typical syntactic constituents
- For code data with stronger structural patterns, optimal k increases to 6-8, matching typical expression lengths
- For character-level tasks, optimal k decreases to 2-3, reflecting shorter dependencies

The relationship between optimal k and average dependency length L_{dep} follows approximately:

$$k_{\text{opt}} \approx \min(16, \max(2, \lfloor 0.8 \cdot L_{\text{dep}} \rfloor)) \tag{20}$$

This explains why $k = 4$ works well for general text ($L_{\text{dep}} \approx 5$), while code data ($L_{\text{dep}} \approx 8 - 10$) benefits from larger blocks.

F.7 Comparison with RADD: Source of Acceleration

While RADD (Ou et al., 2025) achieves acceleration through mathematical reformulation that removes time-dependent sampling, CGAD's acceleration stems from complementary sources:

F.7.1 Acceleration Sources Breakdown

1. **Reduced NFE:** CGAD is evaluated with 25 denoising steps, compared with 100 steps for the reported diffusion baselines in Section 4.1
2. **GPU cache behavior:** NVIDIA Nsight reports an 83.1% GPU cache hit rate during CGAD inference under the tested workload
3. **Block Parallelism:** The Causal Constraint Injector processes k tokens within a block while the Global Denoising Field remains the parallel denoising backbone

The relative contribution of each factor is:

$$\text{Speedup}_{\text{total}} = \text{Speedup}_{\text{NFE}} \times \text{Speedup}_{\text{cache}} \times \text{Speedup}_{\text{parallel}} \tag{21}$$

The reported throughput aggregates these algorithmic and hardware effects; the Nsight GPU cache metric characterizes memory-system behavior and is not a count of skipped denoiser evaluations.

F.7.2 Theoretical Distinction

RADD's acceleration comes from reformulating the diffusion process to eliminate temporal dependencies between steps. In contrast, CGAD maintains the traditional diffusion formulation while incorporating causal features. This difference makes CGAD compatible with existing diffusion infrastructure while yielding the practical speedups measured here.

F.8 Computational Model for Optional State Reuse

The potential effect of state reuse can be formalized through computational complexity analysis. Let $C_{\text{enc}}(m)$ be the cost of encoding m tokens, which typically scales quadratically with sequence length in transformer architectures:

$$C_{\text{enc}}(m) = O(m^2 \cdot d_{\text{model}}) \tag{22}$$

Without state reuse, the total cost over n denoising steps would be:

$$C_{\text{no-cache}} = \sum_{i=1}^{n} C_{\text{enc}}(d) = O(n \cdot d^2 \cdot d_{\text{model}}) \tag{23}$$

With state reuse, the cost depends on the probability p_{hit} of repeated-state hits in C_{cache}. The expected cost becomes:

$$\begin{aligned} C_{\text{cache}} = C_{\text{enc}}(d) \\ + \sum_{i=2}^{n} \big[p_{\text{hit}} C_{\text{cache-hit}} \\ + (1 - p_{\text{hit}}) C_{\text{enc}}(d) \big] \end{aligned} \tag{24}$$

where $C_{\text{cache-hit}} \ll C_{\text{enc}}(d)$ represents the minimal cost of state retrieval. Here, p_{hit} denotes the analytical state-reuse event rather than the NVIDIA Nsight GPU cache hit rate in Table 3. The modeled state-reuse probability evolves during denoising according to:

$$p_{\text{hit}}(t) = \left(1 - \frac{\#\text{masked}(t)}{d}\right)^k \tag{25}$$

This formalization describes why state reuse can become increasingly effective in later denoising stages as fewer masked tokens lead to more repeated computation states. The throughput and GPU cache-hit values in Table 3 are direct inference measurements and do not separately identify this modeled contribution.

G Proofs of Theoretical Results

We provide complete proofs for the three main theorems in the paper. Each proof is structured to first restate the theorem, then present the detailed derivation with step-by-step justification, and finally connect the result to the empirical phenomena observed in CGAD.

Interpretation of the Optimization Cliff. Theorem 2.1 characterizes a difficult training regime for discrete diffusion models under its stated conditions. In the early stages of the reverse process, the Global Denoising Field has limited visible context. Because corruption is position-symmetric, it does not receive an order-specific feature from the forward process. The gradient can therefore be noisy when the denoising target is multimodal and ambiguous. The model predicts from weakly constrained intermediate states, which can create high-variance updates. This motivates examining

whether directional features reduce the measured variation observed for baselines such as RADD and SEDD.

Optimization Implication of the Directionality Gap. Higher entropy in the target distribution can make optimization less informative. The Hessian matrix of the AO-ARM objective H_{AO} is the average of Hessians over all permutations. This averaging may smooth curvature in directions corresponding to order-specific regularities. Consequently, AO-ARM (and standard Discrete Diffusion) may require more data or gradient steps to learn local rules readily expressed by causal AR. This interpretation motivates CGAD but is not a universal complexity comparison between diffusion and AR.

G.1 Proof of Theorem 1: Gradient Variance Lower Bound in Position-Symmetric Absorbing Diffusion

Theorem G.1 (Gradient Variance Lower Bound)**.** *Let $\mathcal{L}_{DSE}^T$ be the denoising score entropy loss for a discrete absorbing diffusion model with a shared position-wise masking rate $\sigma(t)$, under the non-degenerate gradient and covariance non-cancellation conditions stated in Theorem 2.1. The variance of the gradient estimator $\nabla_\theta \mathcal{L}_{DSE}^T(x_0)$ has the following lower bound:*

$$Var\left[\nabla_\theta \mathcal{L}_{DSE}^T(x_0)\right] \geq C_1 \cdot d \cdot \int_0^T \frac{\sigma(t)^2 e^{-2\bar{\sigma}(t)}}{(1-e^{-\bar{\sigma}(t)})^2} dt \tag{26}$$

where d is the sequence length, $\bar{\sigma}(t) = \int_0^t \sigma(s)ds$ is the cumulative noise intensity, and $C_1 > 0$ is a constant depending on vocabulary size $|\mathcal{V}|$, the model architecture, the data distribution, and the stated conditions under the position-symmetric absorbing process.

Proof. We prove the theorem in five parts: (1) construction of the gradient estimator, (2) variance computation and key term analysis, (3) derivation of the integral lower bound, (4) extension to t-DCE and λ-DCE losses, and (5) phenomenological interpretation.

Part 1: Gradient Estimator Construction

The training objective minimizes the denoising score entropy loss. Let x_0 be the original sequence and x_t be the noisy version at time t. The loss is defined as:

$$\mathcal{L}_{DSE}(x_0) = \mathbb{E}_{q(x_t|x_0)}\left[-\log p_\theta(x_0|x_t)\right] \tag{27}$$

where $q(x_t|x_0)$ is the forward diffusion kernel and $p_\theta(x_0|x_t)$ is the denoising model. The gradient estimator is:

$$\nabla_\theta \mathcal{L}_{DSE}(x_0) = \nabla_\theta \mathbb{E}_{q(x_t|x_0)}\left[-\log p_\theta(x_0|x_t)\right] \tag{28}$$

Under standard regularity conditions that permit differentiation under the expectation sign, we write:

$$\nabla_\theta \mathcal{L}_{DSE}(x_0) = \mathbb{E}_{q(x_t|x_0)}\left[\nabla_\theta\left(-\log p_\theta(x_0|x_t)\right)\right] \tag{29}$$

In practice, we approximate the expectation with Monte Carlo sampling. Let N be the number of samples; the empirical gradient estimator is:

$$\hat{g}_{DSE} = \frac{1}{N}\sum_{i=1}^{N} \nabla_\theta\left(-\log p_\theta(x_0|x_t^{(i)})\right) \tag{30}$$

where $x_t^{(i)} \sim q(x_t|x_0)$. The gradient variance is:

$$\mathrm{Var}\left[\hat{g}_{DSE}\right] = \mathbb{E}\left[\|\hat{g}_{DSE} - \mathbb{E}[\hat{g}_{DSE}]\|^2\right] \tag{31}$$

Under the position-symmetric absorbing process, masking events at different positions are independent. However, the model parameters θ are shared across the sequence, which introduces correlations between the gradient components at different positions. Therefore, we cannot assume that the variances simply add up. Defining $g_i = \nabla_\theta(-\log p_\theta(x_0^i \mid x_t))$, we expand the variance and lower-bound covariance contributions by their absolute values:

$$\begin{aligned}\mathrm{Var}[\nabla_\theta \mathcal{L}_{DSE}] \geq & \sum_{i=1}^{d} \mathbb{E}_t[\mathrm{Var}(g_i)] \\ & - \sum_{i \neq j} |\mathbb{E}_t[\mathrm{Cov}(g_i, g_j)]|\end{aligned} \tag{32}$$

where $\ell_i = -\log p_\theta(x_0^i|x_t)$. Under the covariance non-cancellation condition, the covariance terms are bounded by a fixed fraction of the diagonal contribution, with that fraction depending on the model architecture and noise level. For the purpose of deriving a lower bound, we retain the corresponding non-cancelled fraction of the diagonal terms. This yields the following inequality:

$$\mathrm{Var}\left[\nabla_\theta \mathcal{L}_{DSE}\right] \geq C_2 \cdot \sum_{i=1}^{d} \mathbb{E}_t\left[\mathrm{Var}\left(\nabla_\theta \ell_i\right)\right] \tag{33}$$

where $0 < C_2 \leq 1$ is a constant that absorbs the effect of the covariance terms. The positivity of C_2 is exactly the covariance non-cancellation condition required by the theorem, and is not asserted without that condition. This inequality is sufficient for deriving a lower bound on the gradient variance.

Part 2: Variance Calculation and Key Terms

The dominant contribution to gradient variance comes from the estimation variance of the denoising conditional probability $p_\theta(x_0|x_t)$. In discrete absorbing diffusion, the denoising distribution involves transition probabilities from the masked state x_t to the original state x_0. For position i, the true score estimator is:

$$S^i_{\text{true}} = \frac{p_t(x_0^i|x_t^{\text{UM}})}{p_t(x_t^i)} = \frac{e^{-\bar{\sigma}(t)}}{1 - e^{-\bar{\sigma}(t)}} p_0(x_0^i|x_t^{\text{UM}}) \tag{34}$$

where x_t^{UM} is the set of unmasked tokens, and $\bar{\sigma}(t) = \int_0^t \sigma(s)ds$ is the cumulative noise intensity. The model learns a parametric function $s_\theta(x_t, t)$ to approximate this score. The gradient variance is primarily determined by the variance in estimating $p_0(x_0^i|x_t^{\text{UM}})$.

At high masking levels (large $\bar{\sigma}(t)$), the size of the unmasked set $|x_t^{\text{UM}}|$ is small, and its elements are randomly scattered across the sequence. Because the forward kernel is position-symmetric, it contributes no left-to-right directional information, so the conditional probability estimate can be diffuse. The uncertainty of a categorical prediction is largest under a uniform distribution. Let $p = p_0(x_0^i|x_t^{\text{UM}})$ be a categorical probability vector; its variance satisfies:

$$\text{Var}[p] = \mathbb{E}\left[(p - \mathbb{E}[p])^2\right] \tag{35}$$

For vocabulary size $|\mathcal{V}|$, the lower bound under uniform distribution is:

$$\text{Var}[p] \geq \frac{1}{|\mathcal{V}|}\left(1 - \frac{1}{|\mathcal{V}|}\right) \tag{36}$$

This bound is used under the stated high-uncertainty condition, indicating maximal categorical uncertainty when predictive context is absent.

Part 3: Integral Form of the Variance Lower Bound

The gradient variance is a composite function integrating over time and summing over sequence positions. To rigorously derive the lower bound, we decompose the variance term $\text{Var}\left(\nabla_\theta\left(-\log p_\theta(x_0^i|x_t)\right)\right)$ using the law of total variance conditioned on the unmasked set x_t^{UM} and the true conditional probability vector $p = p_0(x_0^i|x_t^{\text{UM}})$. Let $\ell(\theta) = -\log p_\theta(x_0^i|x_t)$. Then:

$$\begin{aligned}\text{Var}(\nabla_\theta \ell) = \mathbb{E}_{x_t^{\text{UM}}}\left[\text{Var}(\nabla_\theta \ell \mid x_t^{\text{UM}})\right] \\ + \text{Var}_{x_t^{\text{UM}}}\left(\mathbb{E}[\nabla_\theta \ell \mid x_t^{\text{UM}}]\right)\end{aligned} \tag{37}$$

Under the non-degeneracy condition, the first term contributes a nonzero lower bound because the unmasked set is random. Conditioned on x_t^{UM}, we further condition on the true categorical probability vector p:

$$\begin{aligned}\text{Var}(\nabla_\theta \ell \mid x_t^{\text{UM}}) = \mathbb{E}_p[\text{Var}(\nabla_\theta \ell \mid p)] \\ + \text{Var}_p(\mathbb{E}[\nabla_\theta \ell \mid p])\end{aligned} \tag{38}$$

The model's gradient $\nabla_\theta \ell$ is proportional to the score estimation error $\epsilon = s_\theta(x_t, t) - s_{\text{true}}(x_t, t)$. Under the position-symmetric absorbing kernel, the true score for position i has the form:

$$s^i_{\text{true}} = \frac{e^{-\bar{\sigma}(t)}}{1 - e^{-\bar{\sigma}(t)}} \cdot p \tag{39}$$

where p is the $|\mathcal{V}|$-dimensional probability vector for x_0^i. The model's score estimate s^i_θ aims to approximate this. The gradient variance conditioned on p can be related to the variance of the score estimate using the Delta method or the Cramér-Rao lower bound. Specifically, the gradient $\nabla_\theta \ell$ is a function of the score estimate s^i_θ. If the score estimator is unbiased or nearly unbiased, the Cramér-Rao bound provides a lower bound on the variance of any unbiased estimator. For a parametric family, the variance of the gradient is bounded below by the Fisher information matrix. In our case, the score s^i_θ is an estimator of the true score, and the gradient $\nabla_\theta \ell$ is a smooth function of s^i_θ. By the Delta method, the variance of $\nabla_\theta \ell$ is approximately $\|J(\theta)\|_F^2 \cdot \text{Var}(s^i_\theta \mid p)$ where $J(\theta)$ is the Jacobian of the gradient with respect to the score, provided the function is locally linear. For a lower bound, we use the fact that the variance of a function of a random variable is at least the square of the derivative times the variance, leading to:

$$\text{Var}(\nabla_\theta \ell \mid p) \geq \left\| \frac{\partial \nabla_\theta \ell}{\partial s^i_\theta} \right\|_F^2 \cdot \text{Var}(s^i_\theta \mid p) \tag{40}$$

The Jacobian norm $\|J(\theta)\|_F^2$ is bounded below by a positive constant due to the network's non-degeneracy.

To bound the variance of the estimator $\hat{p}$, we consider a categorical estimation problem under the stated non-degeneracy condition. However, the effective sample size n is not necessarily $O(1)$ because the model can potentially use information from the entire unmasked set x_t^{UM} to estimate p. Accordingly, the bound requires a condition on the residual uncertainty after conditioning on x_t^{UM}. When the token at position i is masked, the model must rely on the context provided by x_t^{UM}. Under position-symmetric masking, the unmasked tokens are randomly scattered, and the required condition holds when their information is insufficient to determine token i.

A useful characterization uses *mutual information*. Let $I(x_0^i; x_t^{\text{UM}})$ be the mutual information between the token at position i and the unmasked set. We assume that this mutual information is small in the high-masking regime considered by the bound. In particular, the following derivation assumes $I(x_0^i; x_t^{\text{UM}}) \leq e^{-\bar{\sigma}(t)} \cdot I_{\text{max}}$, where I_{max} is a constant. This implies a residual-uncertainty lower bound for an estimator $\hat{p}$ as a function of that mutual-information condition. Applying a Fano-type classification-error bound together with the non-degeneracy condition yields the stated variance form. Thus, we have:

$$\text{Var}(\hat{p}) \geq \frac{1}{|\mathcal{V}|}\left(1 - \frac{1}{|\mathcal{V}|}\right) \cdot \left(1 - e^{-\bar{\sigma}(t)} I_{\text{max}}\right) \tag{41}$$

This bound captures the fact that the variance decreases as the mutual information increases (i.e., when more context is available), but it remains strictly positive. The term $\left(1 - e^{-\bar{\sigma}(t)} I_{\text{max}}\right)$ is close to 1 when the masking rate is high, indicating high uncertainty.

Therefore, the variance of the score estimate is bounded by:

$$\text{Var}(s_\theta^i \mid p) \geq C_1 \frac{e^{-2\bar{\sigma}(t)}}{(1 - e^{-\bar{\sigma}(t)})^2} \frac{1}{|\mathcal{V}|}\left(1 - \frac{1}{|\mathcal{V}|}\right) \cdot \left(1 - e^{-\bar{\sigma}(t)} I_{\text{max}}\right) \tag{42}$$

We absorb the constant factors into C_1 and obtain the same integral form as before, but now with a more rigorous justification. The key point is that the per-position variance is bounded below by a constant that is independent of the sequence length d, which is sufficient for our purposes.

Integrating over the noise schedule $\sigma(t)$ and summing over the d independent sequence positions yields:

$$\begin{aligned}\text{Var}[\nabla_\theta \mathcal{L}_{DSE}] &\geq C_2 \sum_{i=1}^{d} \mathbb{E}_t[\text{Var}(\nabla_\theta \ell_i)] \\ &\geq d \int_0^T \sigma(t)^2 C_1 \frac{e^{-2\bar{\sigma}(t)}}{(1 - e^{-\bar{\sigma}(t)})^2} dt\end{aligned} \tag{43}$$

Here, the additional $\sigma(t)^2$ factor arises from the deterministic time-dependent weighting in the t-DCE gradient estimator. We absorb the constant C_2 into C_1, and obtain the final lower bound:

$$\text{Var}\left[\nabla_\theta \mathcal{L}_{DSE}\right] \geq C_1 \cdot d \cdot \int_0^T \frac{\sigma(t)^2 e^{-2\bar{\sigma}(t)}}{(1 - e^{-\bar{\sigma}(t)})^2} dt \tag{44}$$

This integral lower bound captures the combined effects of sequence length, vocabulary size, and noise schedule on gradient variance. The term $\frac{e^{-2\bar{\sigma}(t)}}{(1-e^{-\bar{\sigma}(t)})^2}$ is maximized at intermediate noise levels, indicating the phase of highest optimization instability.

Part 4: Extension to t-DCE and λ-DCE Losses

In the CGAD framework, we employ time-weighted denoising cross-entropy (t-DCE) and uniformly-weighted denoising cross-entropy (λ-DCE) losses. The t-DCE loss is defined as:

$$\mathcal{L}_{t\text{-}DCE}(x_t, x_0) = \sum_{i: x_t^i = [M]} \frac{-\sigma(t) e^{-\bar{\sigma}(t)}}{1 - e^{-\bar{\sigma}(t)}} \cdot \log p_\theta(x_0^i \mid x_t) \tag{45}$$

The λ-DCE loss uses uniform weighting:

$$\mathcal{L}_{\lambda\text{-}DCE}(x_\lambda, x_0) = \sum_{i: x_\lambda^i = [M]} \lambda \log p_\theta(x_0^i | x_t) \tag{46}$$

where $\lambda \sim U([0, 1])$. The gradient estimators for these losses have similar structures. For t-DCE, the gradient is:

$$\nabla_\theta \mathcal{L}_{t\text{-}DCE} = \sum_i \frac{-\sigma(t) e^{-\bar{\sigma}(t)}}{1 - e^{-\bar{\sigma}(t)}} \nabla_\theta \log p_\theta(x_0^i | x_t) \tag{47}$$

In variance computation, the weighting term $\frac{-\sigma(t)e^{-\bar{\sigma}(t)}}{1-e^{-\bar{\sigma}(t)}}$ is deterministic and does not alter the scale of the stochastic variable's variance. Thus, the form of the variance lower bound remains identical to that of DSE loss, with only a constant factor adjustment. Similarly, uniform weighting in λ-DCE does not affect the order of variance. Hence, the theorem's lower bound applies to these loss functions as well.

Part 5: Phenomenological Interpretation

Theorem G.1 characterizes the "optimization cliff" regime: during early discrete denoising at high masking rates, the global denoising field has limited context. Under position-symmetric absorbing corruption, the forward process supplies no directional feature for masked positions. Gradients can be noisy when the denoising target remains multimodal and ambiguous. The model then predicts from weakly constrained states, potentially producing high-variance updates. This motivates measuring variation for models such as RADD and SEDD. The variance lower bound scales proportionally with sequence length d under the stated conditions, indicating a potential challenge for long sequences. The integral term reveals the impact of the noise schedule: aggressive schedules (large $\sigma(t)$) can amplify variance, requiring careful design to balance stability and efficiency. □

G.2 Proof of Theorem 2: Directionality Gap — Why Arbitrary Order ≠ Causal

Theorem G.2 (Directionality Gap)**.** *Let Π be the set of all permutations of sequence indices $\{1, \ldots, d\}$. An arbitrary-order autoregressive model (AO-ARM) optimizes the expected log-likelihood over the uniform distribution of permutations $U(\Pi)$. A causal AR model optimizes over a single fixed permutation $\pi^* = (1, 2, \ldots, d)$. Under the informative-prefix condition specified in Theorem 2.2, the directionality gap is defined as the expected entropy difference in learning conditional distributions:*

$$\Delta_{Dir} = \mathbb{E}_{\pi \sim U(\Pi)} \left[H(p_\theta(x_{\pi(k)} \mid x_{\pi(<k)})) \right] - H(p_{AR}(x_k \mid x_{<k})) > 0 \tag{48}$$

Moreover, under this condition, the AO-ARM estimator can exhibit higher variance for local syntactic dependencies than the causal AR estimator.

Proof. We prove the theorem in four parts: (1) definition of the entropy gap and the syntactic permutation space, (2) entropy inequality based on conditional information, (3) variance comparison and optimization landscape, and (4) phenomenological interpretation.

Part 1: Entropy Gap Definition and Syntactic Permutation Space

Consider a sequence $x = (x_1, \ldots, x_d)$ with joint distribution factorizable as a product of conditionals. For a permutation $\pi \in \Pi$, the factorization is:

$$p(x) = \prod_{k=1}^{d} p(x_{\pi(k)} | x_{\pi(<k)}) \tag{49}$$

Causal AR uses the fixed permutation π^*, with conditional entropy:

$$H_{AR} = H(x_k | x_{<k}) = \mathbb{E}\left[-\log p(x_k | x_{<k})\right] \tag{50}$$

AO-ARM optimizes over uniform permutations, with expected conditional entropy:

$$H_{AO} = \mathbb{E}_{\pi \sim U(\Pi)} \left[H(x_{\pi(k)} | x_{\pi(<k)}) \right] \tag{51}$$

The directionality gap is the entropy difference:

$$\Delta_{\text{Dir}} = H_{AO} - H_{AR} \tag{52}$$

We need to show $\Delta_{\text{Dir}} > 0$.

Part 2: Conditional Information and Entropy Inequality

For position k, let S_k denote the set of tokens that are syntactic ancestors of x_k under the dependency structure of the language. Crucially, we model the directional bias as a statistical tendency: in the training data distribution, ancestors are more likely to appear before their dependents. This tendency is quantified by the conditional probability $\rho_k = P(S_k \subseteq x_{<k} \mid x_k, \mathcal{D})$, where $\mathcal{D}$ represents the data distribution. For causal AR with the natural order π^*, we typically observe $\rho_k \approx 1$ for local dependencies due to the alignment between the generation order and common syntactic patterns.

For an arbitrary permutation $\pi \sim U(\Pi)$, the probability that the conditioning set $x_{\pi(<k)}$ contains all ancestors in S_k is given by the ratio of combinations:

$$P(S_k \subseteq x_{\pi(<k)}) = \frac{\binom{d-|S_k|}{k-1-|S_k|}}{\binom{d}{k-1}} \quad \text{for } k > |S_k|. \tag{53}$$

When $k \ll d$, this probability is significantly less than 1. Let $A_k = \{S_k \subseteq x_{\pi(<k)}\}$. The key insight from principal direction gap partitioning (PDGP) is that conditional entropy can be decomposed based on this event:

$$\begin{aligned} H(x_k \mid x_{\pi(<k)}) = {} & H(x_k \mid x_{\pi(<k)}, A_k) \\ & \cdot P(A_k) \\ & + H(x_k \mid x_{\pi(<k)}, \neg A_k) \\ & \cdot P(\neg A_k). \end{aligned} \tag{54}$$

By the information processing inequality, when the conditioning set contains S_k, we have:

$$H(x_k|x_{\pi(<k)}, S_k \subseteq x_{\pi(<k)}) \leq H(x_k|x_{<k}), \tag{55}$$

since $x_{<k}$ in the natural order provides a specific instance that contains S_k with high probability. However, when $S_k \not\subseteq x_{\pi(<k)}$, the conditional entropy is typically larger due to missing contextual information. Let $\Delta H_k = H(x_k|x_{\pi(<k)}, S_k \not\subseteq x_{\pi(<k)}) - H(x_k|x_{<k}) \geq 0$ be the entropy increase when ancestors are missing.

The critical observation is that for random permutations, the probability $P(S_k \not\subseteq x_{\pi(<k)})$ is strictly positive when syntactic dependencies exist. Therefore, we can write:

$$\begin{aligned} H(x_k \mid x_{\pi(<k)}) &\geq H(x_k \mid x_{<k})P(A_k) \\ &\quad + H(x_k \mid x_{<k})P(\neg A_k) \\ &= H(x_k \mid x_{<k}), \end{aligned} \tag{56}$$

with strict inequality when $\Delta H_k > 0$ and $P(S_k \not\subseteq x_{\pi(<k)}) > 0$. This follows because the weighted average must exceed the minimum value when weights are positive.

To establish the strict inequality $\Delta_{\text{Dir}} > 0$, we note that for positions with non-empty ancestor sets ($S_k \neq \emptyset$), the entropy difference ΔH_k is bounded below by the gap in predictive information when contextual clues are missing. This gap corresponds to the "natural gaps" used in PDGP. Taking expectation over permutations $\pi \sim U(\Pi)$ and positions k, we obtain:

$$\begin{aligned} \mathbb{E}_\pi \left[H(x_k \mid x_{\pi(<k)})\right] &\geq H(x_k \mid x_{<k}) \\ &\quad + \mathbb{E}_\pi[\Delta H_k P(\neg A_k)] \end{aligned} \tag{57}$$

The second term is strictly positive for positions where syntactic dependencies exist, because $\Delta H_k > 0$ and $P(S_k \not\subseteq x_{\pi(<k)}) > 0$ for random permutations. Summing over all positions yields the directionality gap:

$$\Delta_{\text{Dir}} = \mathbb{E}_\pi \left[H(x_{\pi(k)}|x_{\pi(<k)})\right] - H(x_k|x_{<k}) > 0. \tag{58}$$

This inequality holds on average across the data distribution, reflecting the statistical advantage of causal orderings that align with prevalent syntactic directions.

Part 3: Variance Comparison and Optimization Landscape

The variance of conditional entropy affects optimization. The Hessian matrix of AO-ARM is the average of Hessians over all permutations:

$$H_{AO} = \mathbb{E}_\pi \left[H_\pi\right] \tag{59}$$

Averaging "blurs" the curvature directions corresponding to strict syntactic rules, leading to a flatter optimization landscape. Specifically, the curvature of the loss function relates to the variance of conditional probabilities. Let $\ell(\theta) = -\log p_\theta(x_k|x_{<k})$; the trace of its Hessian satisfies:

$$\text{Tr}(H_{AR}) = \text{Var}\left[\nabla_\theta \ell(\theta)\right] \tag{60}$$

For AO-ARM, due to permutation variance, the gradient variance is larger:

$$\begin{aligned} \text{Var}[\nabla_\theta \ell_{AO}(\theta)] &= \mathbb{E}_\pi[\text{Var}[\nabla_\theta \ell_\pi(\theta)]] \\ &\quad + \text{Var}_\pi[\mathbb{E}[\nabla_\theta \ell_\pi(\theta)]] \end{aligned} \tag{61}$$

The second term is positive, increasing total variance. Hence, AO-ARM requires more data to learn syntactic rules.

Part 4: Phenomenological Interpretation

The directionality gap explains why diffusion models are "data-hungry" and "computationally inefficient": high-entropy target distributions lead to flat optimization landscapes, making it difficult for models to capture local syntactic rules. Rules that causal AR learns immediately require many steps for diffusion models. This quantitative gap motivates the causal constraint injection in CGAD. □

G.3 Proof of Theorem 3: Variance Reduction via Causal Constraint Injection

Theorem G.3 (Variance Reduction via Causal Constraint Injection)**.** *Let $s^0_{\theta_{Diff}}(x_t, t)$ be the score estimator of a position-symmetric absorbing diffusion model. Let $s^1_{\theta_{Diff}}(x_t, t, b_t)$ be the score estimator conditioned on local causal blocks b_t generated by a causal constraint injector. If the causal injector satisfies the local determinism condition, provides nonzero information about the denoising target ($I(b_t; x_0) > 0$), and contributes complementary estimation error, then the mean squared error (MSE) of the score estimate does not increase and can decrease:*

$$MSE(s^1) \leq MSE(s^0) - \Delta_{gain} \tag{62}$$

where $\Delta_{gain} \geq 0$ is proportional to the mutual information between the causal blocks b_t and the true denoising target, with $\Delta_{gain} > 0$ when the estimators are not perfectly correlated.

Proof. We prove the theorem in five parts: (1) linear interpolation fusion, (2) error correlation analysis, (3) cross-attention fusion, (4) mutual information gain, and (5) phenomenological interpretation.

Part 1: Linear Interpolation Fusion

Consider the linear fusion function:

$$s^1(x_t) = (1-\alpha)s^0(x_t) + \alpha s_{\text{AR}}(b_t) \tag{63}$$

where $\alpha \in [0,1]$ is a gating factor depending on the confidence of the causal injector. The MSE of a score estimator is defined as:

$$\text{MSE}(s) = \mathbb{E}\left[(s - s_{\text{true}})^2\right] \tag{64}$$

For the fused estimator, the MSE is:

$$\text{MSE}(s^1) = \mathbb{E}\left[\left((1-\alpha)s^0 + \alpha s_{\text{AR}} - s_{\text{true}}\right)^2\right] \tag{65}$$

Let $e_0 = s^0 - s_{\text{true}}$ and $e_{\text{AR}} = s_{\text{AR}} - s_{\text{true}}$. Expanding the square gives:

$$\begin{aligned}\text{MSE}(s^1) = (1-\alpha)^2\mathbb{E}[e_0^2] + \alpha^2\mathbb{E}[e_{\text{AR}}^2] \\ + 2\alpha(1-\alpha)\mathbb{E}[e_0 e_{\text{AR}}]\end{aligned} \tag{66}$$

Identifying terms:

$$\begin{aligned}\text{MSE}(s^1) = (1-\alpha)^2\text{MSE}(s^0) + \alpha^2\text{MSE}(s_{\text{AR}}) \\ + 2\alpha(1-\alpha)\text{Cov}(s^0, s_{\text{AR}})\end{aligned} \tag{67}$$

where covariance $\text{Cov}(s^0, s_{\text{AR}}) = \mathbb{E}[(s^0 - s_{\text{true}})(s_{\text{AR}} - s_{\text{true}})]$.

Part 2: Error Correlation Analysis

Since the AR estimator has low variance (high confidence) in local syntactic regions, $\text{MSE}(s_{\text{AR}})$ is small. Crucially, error gradients are positively correlated: when s^0 makes a syntactic error, s_{AR} likely corrects it, leading to positive covariance:

$$\text{Cov}(s^0, s_{\text{AR}}) \geq \rho \cdot \sqrt{\text{MSE}(s^0) \cdot \text{MSE}(s_{\text{AR}})} \tag{68}$$

where $\rho > 0$ is the correlation coefficient. Therefore, by choosing α to minimize MSE, we achieve:

$$\text{MSE}(s^1) \leq \text{MSE}(s^0) \tag{69}$$

The reduction $\Delta_{\text{gain}} = \text{MSE}(s^0) - \text{MSE}(s^1)$ is nonnegative. Under the condition that the causal injector provides informative priors (i.e., $I(b_t; x_0) > 0$) and the estimators are not perfectly correlated ($\rho < 1$), the reduction is strictly positive ($\Delta_{\text{gain}} > 0$) for optimally tuned α.

Part 3: Cross-Attention Fusion

In CGAD, fusion is implemented via cross-attention, which is nonlinear. Let H^{Diff} be the hidden state of the global denoising field and H^{AR} be the output of the causal injector. Cross-attention fusion is:

$$H^{\text{fused}} = \text{CrossAttn}(H^{\text{Diff}}, H^{\text{AR}}) \tag{70}$$

The score s^1 is computed from this fused state. To analyze the variance reduction properties of this nonlinear fusion, we consider its approximation properties. The cross-attention mechanism implements a form of adaptive weighting that can be viewed as a generalization of the linear combination:

$$\begin{aligned}s^1(x_t) = \beta(x_t;\theta)s^0(x_t) \\ + (1-\beta(x_t;\theta))s_{\text{AR}}(b_t) + \epsilon(x_t;\theta)\end{aligned} \tag{71}$$

where $\beta(x_t;\theta)$ is a parameterized function learned by the attention mechanism, and $\epsilon(x_t;\theta)$ represents the approximation error. Through training, the model learns parameters θ that minimize the overall MSE. Under the universal approximation theorem for attention mechanisms, the cross-attention can approximate the optimal Bayesian fusion rule arbitrarily well given sufficient model capacity.

The training objective ensures that the fused estimator satisfies:

$$\mathbb{E}[\text{MSE}(s^1)] \leq \mathbb{E}[\text{MSE}(s^0)] \tag{72}$$

with equality only when the causal injector provides no additional information. The variance reduction achieved by the nonlinear fusion is at least as good as the optimal linear fusion, and typically better due to its adaptive nature. The effectiveness of this approach is supported by empirical results showing consistent MSE reduction across various tasks.

Part 4: Mutual Information Gain

The reduction Δ_{gain} relates to the mutual information $I(b_t; x_0)$. Mutual information measures the amount of information provided by the causal blocks. By information inequality:

$$\Delta_{\text{gain}} \propto I(b_t; x_0) = H(x_0) - H(x_0|b_t) \tag{73}$$

When the causal injector provides informative priors, conditional entropy decreases and the gain increases.

Part 5: Phenomenological Interpretation

Theorem G.3 motivates how CGAD may mitigate the optimization cliff: causal constraint injection provides directional priors that can reduce gradient variance when its conditions hold. Linear or attention-based fusion dynamically adjusts weights, relying on AR features in syntactic regions and diffusion for global context. The reported experiments examine whether this design is associated with a more stable denoising trajectory and improved efficiency. □

Algorithm 1 CGAD Sampling with Interaction Mechanism

Require: Denoising network c_θ, noise schedule σ, time range $[0, T]$, step size Δt, block size k

Ensure: Generated sequence X_0

$t \leftarrow T$, $X_T \leftarrow [M]...[M]$ ▷ Initialize with all masks
$C_{\text{cache}} \leftarrow \emptyset$ ▷ Initialize cache as empty
while $t > 0$ **do**
  $X_t^{\text{UM}} \leftarrow \{X_t^i | X_t^i \neq [M]\}$ ▷ Extract unmasked tokens
  key $\leftarrow (t, X_t^{\text{UM}})$ ▷ Cache key includes time and unmasked tokens
  if key $\notin C_{\text{cache}}$ **then**
    $H_t^{\text{Diff}} \leftarrow \text{Encoder}_\theta(X_t^{\text{UM}})$ ▷ Global denoising field
    $C_{\text{cache}}[\text{key}] \leftarrow H_t^{\text{Diff}}$ ▷ Update cache
  else
    $H_t^{\text{Diff}} \leftarrow C_{\text{cache}}[\text{key}]$ ▷ Use cached value
  end if
  $X_{\text{temp}} \leftarrow X_t$ ▷ Copy current sequence
  for each block $[j : j + k - 1]$ with masked positions in X_t **do**
    for $m = j$ to $j + k - 1$ **do**
      $H^{\text{AR}} \leftarrow \text{CausalConstraintInjector}(X_{\text{temp}}^{<m}, X_t^{\text{UM}})$ ▷ Directional prior
      $H^{\text{interact}} \leftarrow \text{CrossAttention}(Q = H^{\text{AR}}, K = H_t^{\text{Diff}}, V = H_t^{\text{Diff}})$ ▷ Feature interaction
      $\text{Logits} \leftarrow \text{MLP}(H^{\text{interact}})$ ▷ Final logits for block
      $X_{\text{temp}}^m \sim \text{Categorical}(\text{Softmax}(\text{Logits}))$
    end for
  end for
  $X_{t-\Delta t} \leftarrow X_{\text{temp}}$
  $t \leftarrow t - \Delta t$
end while
return X_0