SYNERGISTIC ABSORBING DIFFUSION: DUAL-BRANCH ENHANCED CONTINUOUS-TIME MODELING FOR PARALLEL TOKEN GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advancements in diffusion models, such as global optimization and parallel token prediction, have enhanced global consistency compared to autoregressive Transformers. However, existing diffusion models exhibit unfavorable trade-offs between efficiency and quality, in which the multi-step iterative denoising processes particularly incur high computational costs. To address these issues, we propose a dual-branch synergistic absorption diffusion model. For efficiency-quality trade-offs, we design a dual-branch architecture, in which the Transformer branch generates local token chunks, and the diffusion branch optimizes global token blocks in fewer steps. To resolve the instability of discrete-time models, we further introduce the continuous-time diffusion process, which enhances parallel token generation and learning representations. Experiments conducted on multiple tasks, including text generation and structural reasoning tasks, demonstrate the state-of-the-art performance of the proposed model.

1 INTRODUCTION

Sequence generation has long been dominated by the autoregressive (AR) paradigm, where Transformer-based causal decoder models (e.g., GPT series (OpenAI et al., 2024; Devlin et al., 2019; Vaswani et al., 2017)) achieve remarkable progress in language modeling and code generation through recursive next-token prediction. However, this approach suffers from inherent limitations including unidirectional contextual dependencies and strict sequential decoding, leading to high inference latency and constraints in modeling bidirectional global coherence. The recent emergence of discrete diffusion models (Bao et al., 2022; Austin et al., 2021; Gulrajani & Hashimoto, 2023; Song et al., 2025) offers a promising non-autoregressive alternative for parallel sequence generation in discrete symbol spaces (Čeović et al., 2023). Their differences are shown in Figure 1. By employing forward noise injection and backward iterative denoising, discrete diffusion enables global optimization and parallel multi-token prediction, demonstrating strong capabilities in maintaining global consistency and robustness for high-dimensional discrete data such as text and molecular sequences. Furthermore, continuous-time discrete diffusion models (Dieleman et al., 2022; Campbell et al., 2022) provide more flexible time parameterization and analytical absorbing state modeling, mitigating iterative instability and intermediate semantic loss, thus representing state-of-the-art approaches for efficient parallel decoding (Yang et al., 2023; Yi et al., 2024).

The architectural landscape of contemporary continuous-time discrete diffusion paradigms primarily employs two designs: a unified backbone with denoising heads or a decoupled conditional encoder-denoiser structure (Tang et al., 2025). While these designs promote quality improvements, they reveal three persistent bottlenecks. First, insufficient cross-granularity information coupling hinders effective alignment between local syntax and global coherence within a single step (Yan et al., 2024). Second, the trade-off between parallelization and quality remains constrained, as reducing denoising steps often leads to semantic oversmoothing and loss of detail. Third, the fundamental divergence between AR and diffusion paradigms, where AR emphasizes sequential causality and local fidelity, while diffusion focuses on global consistency and distribution approximation, creates optimization instability without explicit mutual guidance mechanisms. These limitations are exacerbated by the computational inefficiency of Transformer-based denoising networks, where the

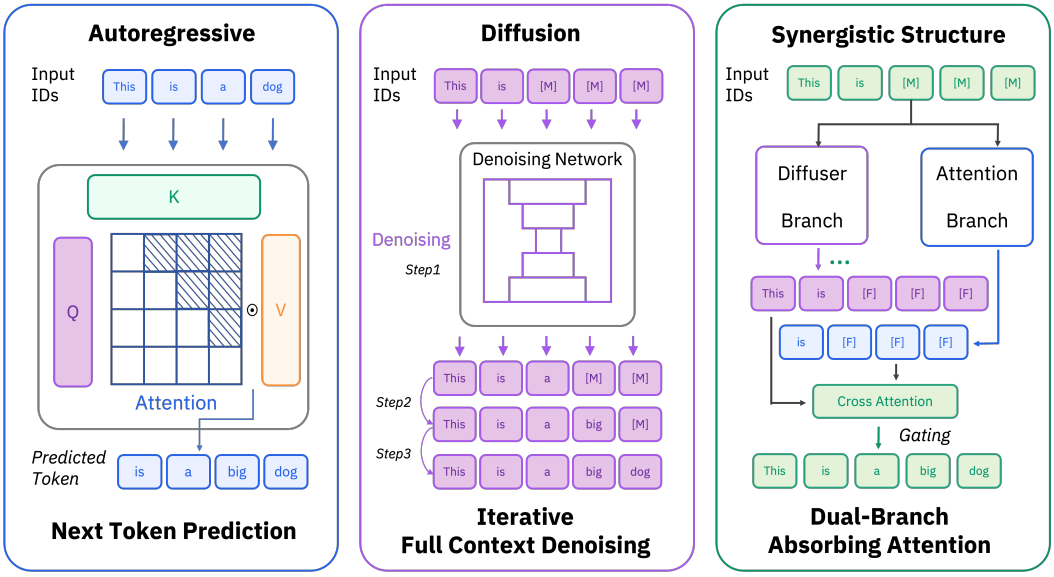

Figure 1: Comparison between three paradigms of generative language model: in contrast to (1) the autoregressive paradigm, which relies on multiple sequential queries (each producing a single token) via next-token prediction, and (2) the diffusion paradigm, which performs text generation in a single query but requires multiple iterative denoising steps over the full context, (3) the proposed synergistic structure integrates both approaches to achieve generation with significantly fewer queries and denoising iterations.

quadratic complexity of self-attention mechanisms and the inability to reuse Key-Value caching during iterative denoising impose significant burdens, perpetuating a quality-speed dilemma.

To address these challenges, we propose a dual-branch structure that synergistically integrates a Transformer branch (representing the AR paradigm) and a Diffuser branch (representing the diffusion denoising paradigm). Unlike common mixture-of-experts (MoE) (Masoudnia & Ebrahimpour, 2014) parallelization, our framework utilizes cross-attention as the core fusion layer to enable learnable alignment and high-bandwidth information exchange between branches. This design allows each branch to evolve independently within its semantic space while selectively absorbing representations and uncertainty estimates from the other branch, establishing explicit, fine-grained alignment between local and global semantics. The cross-attention mechanism proves superior to MoE routing by performing direct token/block-level alignment and weighted integration, significantly enhancing fusion quality. Moreover, we introduce a mutual reinforcement mechanism that tightly couples both paradigms: the AR branch provides high-confidence local priors (e.g., short-range coherence and syntactic templates) to guide the diffusion denoising process, enabling convergence to superior globally consistent solutions with fewer iterations. Conversely, the diffusion branch feeds back global distribution and uncertainty characterization to constrain AR predictions, allowing an expanded multi-token prediction window per step without sacrificing stability, thereby alleviating traditional AR bottlenecks and error accumulation.

Within our continuous-time discrete diffusion framework, we integrate time-dependent score factorization, intermediate state caching, and dual-branch mutual guidance, and introduce denoising cross-entropy (DCE) training objectives—including t-DCE and λ-DCE—to improve stability and convergence (Ou et al., 2025). These losses unify noise scheduling with conditional learning, enabling sharper predictions and more efficient sampling while preserving analytic benefits. Experiments show our approach enhances quality and stability in text and structured reasoning tasks, reduces iterations and latency, and establishes a unified generative framework that balances fusion, efficiency, and global consistency.

2 Related Work

Our work builds upon and intersects with several key areas of generative modeling research, primarily encompassing autoregressive language models, discrete diffusion models, and emerging hybrid architectures that seek to leverage the strengths of both paradigms.

Autoregressive Language Models The dominance of autoregressive (AR) models, particularly those based on the Transformer architecture, has been a defining feature of sequence generation in recent years. Models in the GPT series (GPT-2 (Radford et al., 2019) to GPT-4 (OpenAI et al., 2024)) exemplify the success of the AR approach, which relies on causal masking within the decoder to generate sequences token-by-token in a left-to-right manner . This paradigm excels at capturing local syntactic coherence (Tabor et al., 2004) and has achieved remarkable performance in language modeling and code generation. However, its core limitation lies in the unidirectional nature of dependency modeling (Dong et al., 2019) and the inherent sequentiality of decoding (Bybee, 2008), which results in high inference latency and challenges in maintaining long-range global coherence . Techniques such as in-context learning (Dong et al., 2022; Min et al., 2021), multi-token prediction technique (Li et al., 2024) and chain-of-thought prompting (Wei et al., 2022) have been developed to enhance the reasoning capabilities of AR models, yet they do not fundamentally overcome the sequential bottleneck .

Discrete Diffusion Models for Sequence Generation As a non-autoregressive alternative, discrete diffusion models have emerged as a powerful framework for parallel sequence generation (Shih et al., 2023; Zhang et al., 2025). These models operate through a forward process of incrementally corrupting a data sequence with noise and a reverse process of iterative denoising to recover the original data . Initially successful in continuous domains like image generation, diffusion models have been adapted for discrete data like text. Two primary methodologies have been developed: discrete diffusion models (Austin et al., 2021), which operate directly on token spaces using transition matrices like the absorbing state , and embedding diffusion models, which first map discrete tokens into a continuous embedding space where Gaussian noise is applied before a rounding step . Continuous-time discrete diffusion (Dieleman et al., 2022; Campbell et al., 2022; Sun et al., 2023) further offers more flexible noise scheduling and improved analytical tractability. These models demonstrate superior capabilities in parallel token prediction and maintaining global consistency, making them particularly suitable for tasks requiring high-dimensional coherence .

Hybrid and Synergistic Architectures Recognizing the complementary strengths and weaknesses of AR and diffusion paradigms, recent research has begun exploring hybrid architectures. Some efforts have focused on using diffusion models to refine or initialize sequences for AR decoders, while others have investigated iterative refinement schemes where the two paradigms operate in stages . For instance, DiffusionBERT (He et al., 2023) integrates diffusion processes with pre-trained BERT (Devlin et al., 2019) models by incorporating timestep information to guide the denoising reverse process . Similarly, other studies (Minnen et al., 2018; Hoogeboom et al., 2022) have explored using AR-based priors to guide the diffusion sampling process, aiming to reduce the number of denoising steps required. However, many existing integrations remain relatively shallow, often involving sequential application or simple ensemble methods rather than deep, interactive fusion. Our proposed dual-branch synergistic structure, which utilizes cross-attention for real-time, fine-grained information exchange, represents a departure from these approaches by enabling explicit and continuous mutual guidance between the AR and diffusion paradigms throughout the generation process .

3 Preliminaries

3.1 Absorbing Continuous-Time Discrete Diffusion Models

The absorbing continuous-time discrete diffusion model is formulated as a continuous-time Markov chain (CTMC) on the discrete state space $\mathcal{V}^d \cup \{[\mathrm{M}]\}$, where $\mathcal{V}$ is the vocabulary set, d is the sequence length, and [M] denotes the masked token. The forward process $\{x_t\}_{t\geq 0}$, with $x_t \in \mathcal{V}^d \cup \{[\mathrm{M}]\}$, is governed by an infinitesimal generator matrix $Q_t \in \mathbb{R}^{|\mathcal{V}^d \cup \{[\mathrm{M}]\}| \times |\mathcal{V}^d \cup \{[\mathrm{M}]\}|}$. The

generator matrix exhibits a block structure separating transient tokens and absorbing masked tokens:

$$Q_t = \begin{bmatrix} 0 & 0 \\ R_t & T_t \end{bmatrix}, \tag{1}$$

where T_t governs transitions among transient (unmasked) tokens, and R_t governs transitions from transient tokens to the absorbing masked state [M]. The probability distribution $p_t \in \mathbb{R}^{|\mathcal{V}^d \cup \{[\mathrm{M}]\}|}$ over states at time t evolves according to Kolmogorov's forward equation, with the formal solution:

$$p_t = p_0 \exp\left(\int_0^t Q_s \, ds\right). \tag{2}$$

Under the assumption of token-wise independent masking with a time-dependent rate $\gamma(t) > 0$, the process factorizes across the d dimensions. The cumulative masking rate is defined as $\bar{\sigma}(t) = \int_0^t \gamma(s)\, ds$. For a sequence $x_0 \in \mathcal{V}^d$, the marginal distribution at time t factorizes as:

$$p_t(x_t) = \prod_{i=1}^{d} \left[(1 - e^{-\bar{\sigma}(t)})^{\mathbb{I}(x_t^i = [\mathrm{M}])} \cdot (e^{-\bar{\sigma}(t)})^{\mathbb{I}(x_t^i \neq [\mathrm{M}])} \right] \cdot p_0(x_t^{UM}), \tag{3}$$

where $x_t^{UM} \in \mathcal{V}^{d_{\mathrm{UM}}}$ denotes the set of unmasked tokens at time t with d_{UM} being the number of unmasked tokens, and $\mathbb{I}$ is the indicator function. This formulation allows for adaptive step sizes, eliminates discretization error, and provides analytic expressions while respecting the discrete nature of the data.

A key efficiency arises from the decomposition of the concrete score for a transition that changes a single token at position i from x_t to $\hat{x}_t$. The ratio of probabilities admits a closed-form expression:

$$\frac{p_t(\hat{x}_t)}{p_t(x_t)} = \frac{e^{-\bar{\sigma}(t)}}{1 - e^{-\bar{\sigma}(t)}} \cdot p_0(\hat{x}_t^i \mid x_t^{UM}). \tag{4}$$

This motivates the reparameterization of the model's learned score function $s_\theta(x_t, t) \in \mathbb{R}$:

$$s_\theta(x_t, t) = \frac{e^{-\bar{\sigma}(t)}}{1 - e^{-\bar{\sigma}(t)}} \cdot \tilde{s}_\theta(x_t), \tag{5}$$

where $\tilde{s}_\theta(x_t)$ is a time-independent function approximating the conditional distribution $p_0(\hat{x}_t^i \mid x_t^{UM})$. The reverse-time transition kernel $p_{s|t}(x_s \mid x_t)$ for $s < t$ can be derived in closed form in terms of $\bar{\sigma}(s)$, $\bar{\sigma}(t)$, and $p_0(\cdot \mid x_t^{UM})$, enabling efficient sampling without iterative approximations.

The training objective minimizes the Kullback-Leibler divergence between the true conditional distribution and the model's approximation over time:

$$\mathcal{L}_{\mathrm{CAD}} = \mathbb{E}_{t, x_t \sim p_t} \left[D_{KL}\big(p_0(\cdot \mid x_t^{UM}) \,\|\, q_\theta(\cdot \mid x_t^{UM})\big) \right], \tag{6}$$

where $q_\theta(\cdot \mid x_t^{UM})$ is provided by a Transformer-based prior fusion branch, detailed in the next subsection.

3.2 Denoising Cross Entropy

Denoising Cross Entropy (DCE) serves as a fundamental objective function for training models to recover clean data from corrupted inputs, particularly in the context of denoising autoencoders and diffusion processes. This loss function builds on the principle of minimizing the reconstruction error between the original data and the model's prediction given a noisy version, often employing cross-entropy due to its suitability for probabilistic outputs. In diffusion models, DCE is adapted to handle time-dependent noise schedules, leading to formulations like the t-DCE loss, which operates in continuous time, and the λ-DCE loss, which reparameterizes the problem using masking probability. These variants aim to optimize the conditional likelihood of clean data under varying noise levels, effectively decoupling the learning signal from the noise dynamics.

The t-DCE loss is derived from the continuous-time framework and focuses on the time-dependent aspects of the diffusion process. It is defined as:

$$\mathcal{L}_{t\text{-DCE}}^{T}(x_0) = \int_0^T \mathbb{E}_{x_t \sim p_{t|0}(x_t|x_0)} \left[\sum_{x_t^i = [M]} -\frac{\sigma(t) e^{-\bar{\sigma}(t)}}{1 - e^{-\bar{\sigma}(t)}} \log\left(\frac{e^{-\bar{\sigma}(t)}}{1 - e^{-\bar{\sigma}(t)}} q_\theta\left(x_0^i \mid x_t^{UM}\right) \right) \right] dt \tag{7}$$

where $q_\theta\left(x_0^i \mid x_t^{UM}\right)$ represents the model's estimate of the conditional distribution of the clean token x_0^i given the unmasked tokens x_t^{UM} at time t. This loss leverages the analytic decomposition of the concrete score to simplify the learning signal by isolating the time-dependent scalar component.

The λ-DCE loss introduces a change of variable from time t to the masking probability $\lambda = 1 - e^{-\bar{\sigma}(t)}$, which corresponds to the probability that a token is masked by time t. This reparameterization yields a more intuitive form:

$$\mathcal{L}_{\lambda\text{-DCE}}\left(x_0\right) = \int_0^1 \frac{1}{\lambda} \mathbb{E}_{x_\lambda \sim p_\lambda(x_\lambda \mid x_0)} \left[\sum_{x_\lambda^i = [M]} -\log q_\theta\left(x_0^i \mid x_\lambda^{UM}\right) \right] d\lambda \tag{8}$$

Here, $p_\lambda\left(x_\lambda \mid x_0\right)$ denotes the joint distribution induced by independently masking each dimension of x_0 with probability λ. The λ-DCE loss emphasizes the conditional probabilities of clean data under varying masking levels, effectively decoupling the time-independent learning from the noise schedule.

Both losses are equivalent to the standard denoising score entropy loss (e.g., $\mathcal{L}_{\text{CAD}}$) in the limit of infinite time, and they provide a unified perspective on training absorbing diffusion models.

4 Synergistic Dual-Branch Continuous-Time Absorbing Diffusion

Building upon the challenges outlined in the introduction, specifically, the inefficiency-quality trade-off in existing diffusion models, the loss of intermediate semantics, and the divergence between autoregressive and diffusion paradigms, this section introduces a novel dual-branch continuous-time absorbing diffusion framework designed to synergistically integrate local and global generation processes. Our approach consists of three core innovations: a **dual-branch fusion architecture** that enables fine-grained interaction between a Transformer-based autoregressive branch and a continuous-time diffusion denoising branch via cross-attention; a **mutual reinforcement mechanism** that allows each branch to leverage the other's strengths—the Transformer providing local syntactic priors to accelerate diffusion convergence, while the diffusion branch supplies global distributional context to expand the Transformer's parallel prediction capacity; and a **decomposed training objective** with loss variants that stabilize learning and enhance representation quality. Together, these components form a cohesive system that addresses the limitations of prior works while enabling efficient, high-quality parallel sequence generation.

4.1 Dual-Branch Fusion Framework

The proposed dual-branch fusion framework synergistically integrates a continuous-time diffusion branch with a Transformer-based autoregressive branch, enabling fine-grained interaction through cross-attention mechanisms. This architecture addresses the inefficiency-quality trade-off in existing diffusion models by allowing each branch to leverage the complementary strengths of the other. The complete framework operates on partially denoised states $x_t \in \mathcal{V}^d$, where $\mathcal{V}$ represents the vocabulary and d is the sequence length. The two branches process different aspects of the input and fuse their representations to produce the final output. Figure 1 provides a high-level overview of the architecture, illustrating the information flow between components.

4.1.1 Continuous-Time Diffusion Branch

The diffusion branch models the conditional distribution $p_\theta(x_0|x_t^{UM})$ through an iterative denoising 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. This branch employs a continuous-time absorbing diffusion process that gradually reconstructs the original data distribution from noise. The encoder transformation is defined as:

$$h_t^{\text{CAD}} = \text{Encoder}_\theta(x_t^{UM}) \in \mathbb{R}^{d_{\text{UM}} \times d_{\text{model}}} \tag{9}$$

where Encoder_θ is a parameterized function (typically a neural network) that maps the unmasked tokens to a hidden representation of dimension d_{model}. The diffusion process follows the continuous-time formulation described in the previous section, with the denoising network trained to predict the original clean data x_0 from its noisy version x_t .

4.1.2 TRANSFORMER AUTOREGRESSIVE BRANCH

The Transformer branch serves as a prior fusion component that estimates the time-independent conditional distribution $p_0(\hat{x}_t^i \mid x_t^{UM})$ for tokens at masked positions. Unlike the diffusion branch, this component does not explicitly model the temporal evolution but focuses on capturing local syntactic and semantic patterns using autoregressive decoding. The input to this branch is the partially denoised state $x_t \in \mathcal{V}^d$, from which only the unmasked tokens $x_t^{UM} \in \mathcal{V}^{d_{\text{UM}}}$ are extracted for processing.

The branch employs a modified Transformer decoder architecture that removes explicit time-conditioning layers and adds a final softmax normalization over the vocabulary $\mathcal{V}$. The core transformation is defined as:

$$\text{TransformerBranch}(x_t) = \text{Softmax}_{\mathcal{V}}(\text{TransformerDecoder}(\text{Embed}(x_t^{UM}))) \in \mathbb{R}^{d \times |\mathcal{V}|} \tag{10}$$

where $\text{Embed} : \mathcal{V}^{d_{\text{UM}}} \to \mathbb{R}^{d_{\text{UM}} \times d_{\text{model}}}$ is an embedding function that maps tokens to a model dimension d_{model}, and the Transformer decoder outputs logits for each position. Only the outputs at positions corresponding to masked tokens in x_t are used for loss calculation and sampling .

To enhance local coherence while maintaining parallelization efficiency, the Transformer branch decodes blocks of k tokens autoregressively. For a block starting at index j, the conditional probability is factored as:

$$p_\phi(x_0^{[j:j+k-1]} \mid x_t) = \prod_{m=j}^{j+k-1} p_\phi(x_0^m \mid x_0^{<m}, x_t^{UM}) \tag{11}$$

where $x_0^{<m} \in \mathcal{V}^{m-1}$ represents the causal context of previously generated tokens within the block, and $x_t^{UM} \in \mathcal{V}^{d_{\text{UM}}}$ provides global context from the diffusion branch. The hidden states $h^{\text{Trans}} \in \mathbb{R}^{k \times d_{\text{model}}}$ of the Transformer are computed via:

$$h^{\text{Trans}} = \text{TransformerDecoder}(x_0^{<j}, x_t^{UM}; \theta) \tag{12}$$

This computation utilizes causal self-attention for the autoregressive context and cross-attention mechanisms to incorporate features from x_t^{UM}. The optimal block size k (typically 4–8 for text) balances parallelization efficiency and generative quality .

4.1.3 CROSS-ATTENTION FUSION MECHANISM

The fusion of the two branches occurs at each denoising step t_i through a feature-level integration mechanism that replaces traditional Mixture-of-Experts gating. This approach avoids router complexity while enhancing representational capacity. The fusion operation is defined as:

$$\text{Fusion}(h_t^{\text{CAD}}, h^{\text{Trans}}) = \text{MLP}\left(\left[h^{\text{Trans}}; \text{CrossAtt}(h^{\text{Trans}}, h_t^{\text{CAD}})\right]\right) \in \mathbb{R}^{k \times d_{\text{model}}} \tag{13}$$

where $[\cdot;\cdot]$ denotes concatenation along the feature dimension, and MLP is a multi-layer perceptron that projects the concatenated features back to dimension d_{model}. The cross-attention operation CrossAtt is the core component that enables dynamic alignment between the autoregressive decoding process and the global context from the diffusion branch.

The cross-attention mechanism is formally defined as follows. Let $Q \in \mathbb{R}^{k \times d_k}$ be the query matrix derived from the autoregressive context $x_0^{<j}$, $K \in \mathbb{R}^{d_{\text{UM}} \times d_k}$ be the key matrix derived from x_t^{UM}, and $V \in \mathbb{R}^{d_{\text{UM}} \times d_v}$ be the value matrix from x_t^{UM}. The cross-attention output is computed as:

$$\text{CrossAtt}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \in \mathbb{R}^{k \times d_v} \tag{14}$$

In practice, the query, key, and value matrices are obtained through linear projections:

$$Q = h^{\text{Trans}} W_Q, \quad K = h_t^{\text{CAD}} W_K, \quad V = h_t^{\text{CAD}} W_V \tag{15}$$

where $W_Q, W_K \in \mathbb{R}^{d_{\text{model}} \times d_k}$ and $W_V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ are learnable projection matrices. The scaling factor $\frac{1}{\sqrt{d_k}}$ ensures stable gradients during training .

This cross-attention mechanism allows the Transformer branch to dynamically attend to relevant features from the diffusion branch, effectively combining local coherence from the autoregressive history with global consistency provided by the diffusion branch's unmasked tokens. The complete system can be described through the time-scaled transformation:

$$\text{CAD/DLM}(x_t) = \frac{e^{-\bar{\sigma}(t)}}{1 - e^{-\bar{\sigma}(t)}} \cdot \text{TransformerBranch}(x_t) \tag{16}$$

where the Transformer output is scaled by a time-dependent factor derived from the analytic decomposition of the continuous-time diffusion process .

4.2 Mutual Reinforcement Mechanism

To achieve deeper synergistic cooperation between the two branches, we introduce a mutual reinforcement mechanism that enables co-evolution of the Transformer and diffusion branches: the diffusion module provides continuously expanded global context to enhance the Transformer's representation capacity, while the Transformer offers localized semantic priors that significantly accelerate the convergence of the diffusion denoising process. Let W_{Trans} be the Transformer's native context window. The diffusion branch provides additional context from unmasked tokens, yielding an effective context:

$$W_{\text{eff}} = W_{\text{Trans}} \cup \{i | x_t^i \neq [\text{M}]\}. \tag{17}$$

The expected additional context size is $\mathbb{E}|W_{\text{eff}} \setminus W_{\text{Trans}}| = d \cdot (1 - e^{-\bar{\sigma}(t)})$, which at $t = 0.5T$ provides approximately 39% more context. Conversely, the Transformer branch reduces the number of denoising steps n required to achieve a target perplexity $\mathcal{P}$ by:

$$n_{\text{fused}} = n_0 \cdot \exp\left(-\beta \frac{I(X_t; X_0^{\text{Trans}})}{H(X_t)}\right), \tag{18}$$

where β is a coupling coefficient. This synergy enables a significant speedup in inference, with the expected number of function evaluations (E-NFEs) reduced from $O(d)$ to $O(d/k + n(1 - k/d))$, achieving up to a 7.1× speedup for typical parameters. The mechanism is facilitated through shared caching strategies where computed Transformer outputs $c_\theta(x_t)$ are cached for positions that have already been unmasked, significantly reducing redundant computation. This caching is possible due to the absorbing property that once a token is unmasked ($x_s^i \neq [\text{M}]$), it remains unchanged in all previous time steps ($x_u^i = x_s^i$ for all $u < s$).

4.3 Decomposed Component Encoding and Loss Variants

Building upon the dual-branch fusion and mutual reinforcement mechanism, we further introduce enhanced training objectives through variants of denoising cross-entropy loss, which further strengthen component-level encoding and improve overall training efficiency. These variants, namely the t-denoising cross-entropy (t-DCE) and λ-denoising cross-entropy (λ-DCE) losses, provide alternative objectives for optimizing the diffusion process while maintaining the theoretical benefits of the continuous-time formulation. By employing λ-DCE and t-DCE, our model achieves enhanced stability and faster convergence during training, while also facilitating efficient sampling through the analytic properties of the loss functions. These variants are particularly beneficial in the dual-branch architecture, as they allow for seamless integration with the Transformer branch for prior fusion, further improving the model's ability to capture long-range dependencies and local coherence.

5 EXPERIMENTS

5.1 DATASETS AND IMPLEMENTATION DETAILS

Datasets We evaluate our proposed method on six benchmark datasets spanning diverse domains and complexity levels. WikiText2 and WikiText103 (Merity et al., 2017) are Wikipedia-derived language modeling datasets known for preserving original casing, punctuation, and numerical information. WikiText2 contains approximately 4.3 MB of text data, while WikiText103 is substantially larger with over 100 million tokens (181 MB), making it suitable for evaluating long-range dependency modeling. The Colossal Clean Crawled Corpus (C4) (Raffel et al., 2020) comprises 156 billion tokens of filtered web text from Common Crawl, extensively used for pre-training large language models. FineWeb (Penedo et al., 2024) offers 15 trillion tokens of high-quality, deduplicated English web text with advanced filtering techniques including URL-based filtering, language detection, and privacy removal. Prolong (Gao et al., 2025) is specifically designed for long-context evaluation, focusing on narrative coherence, entity tracking, and complex dependency resolution across extended passages. The JSON-Mode-Eval dataset assesses structural reasoning capabilities through context-free grammar (CFG) compliant JSON parsing and generation tasks, serving as a proxy for evaluating hierarchical reasoning in formal grammar systems.

Training Settings All models are trained with consistent parameters across datasets to ensure fair comparison. Small models are trained for 250,000 iterations with a batch size of 128 sequences of length 512. Medium models undergo 250,000 iterations with the same batch size and sequence length. We use the AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and a learning rate schedule featuring linear warmup for 10,000 steps followed by cosine decay. The base learning rate is set to 1×10^{-4} for small models and 5×10^{-5} for medium models. Our CAD-DF model implements a dual-branch architecture. The model vocabulary size is 50,265 tokens, consistent with standard GPT-2 tokenization. Training is conducted on 8 NVIDIA A100 GPUs with gradient accumulation steps adjusted to maintain effective batch size. Models like GPT-2 (Radford et al., 2019), D3PM (Austin et al., 2021), PAID (Gulrajani & Hashimoto, 2023), SEDD (Lou et al., 2024) RADD (Ou et al., 2025) and Block Diffusion (Arriola et al., 2025) (BD3-LM and SSD-LM) are used for comparison.

Metrics We evaluate model performance with three primary metrics:

Perplexity measures the model's uncertainty when predicting the next token; lower perplexity indicates that the model assigns a higher probability to the correct continuations. Concretely, it corresponds to the exponential of the average negative log-likelihood over a sequence of tokens.

Accuracy is the fraction of tokens predicted exactly correctly. For each position, we compare the predicted token $\hat{x}_i$ with the ground-truth token x_i; the final score is the proportion of positions where they match.

Inference efficiency is reported using three indicators: (i) *throughput*, measured as tokens processed per second; (ii) *cache hit rate*, the percentage of tokens served from cache rather than recomputation; and (iii) *GPU memory consumption*, the peak device memory required during inference.

5.2 RESULTS

The experimental results on the proposed model with small and medium sizes, as shown in Table 1 and Table 2 respectively, demonstrate that the proposed CAD-DF framework achieves consistent improvements across diverse datasets and model scales, validating the core hypotheses outlined in the introduction. The dual-branch architecture, enabled by cross-attention fusion, effectively bridges the local fidelity of autoregressive modeling and the global consistency of discrete diffusion. This synergy allows the Transformer branch to provide high-confidence local priors—such as syntactic templates and short-range dependencies—which anchor the diffusion denoising process, reducing the number of iterations required for convergence. Conversely, the diffusion branch supplies global distributional constraints that expand the Transformer's multi-token prediction window while mitigating error accumulation. On datasets emphasizing long-range coherence (e.g., Prolong) or structural reasoning (e.g., CFG/JSON-Mode-Eval), the model exhibits particularly strong gains, as the mutual reinforcement mechanism explicitly addresses the inherent trade-offs between sequential causality and distributional approximation. The continuous-time formulation further supports these

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
CAD-DF-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
CAD-DF-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**

advantages by enabling analytical inversion and time-independent conditioning, which minimize intermediate semantic loss and iterative instability. As model scale increases, the benefits compound, underscoring the framework's scalability and its capacity to harmonize paradigm-specific strengths without introducing uncontrolled complexity.

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
CAD-DF-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
CAD-DF-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

5.3 Efficiency Study

The efficiency analysis in Table 3 reveals that the CAD-DF framework significantly enhances inference throughput, cache utilization, and memory economy compared to baseline methods. These gains are directly attributable to the architectural innovations introduced to resolve the "quality-speed dilemma" described in the introduction. By leveraging the continuous-time discrete diffusion foundation, the model achieves parallel token generation without sacrificing global consistency, thereby reducing the iterative redundancy typical of standard diffusion approaches. The dual-branch design further optimizes computational load: the Transformer branch maintains a lightweight, causal representation for local coherence, while the diffusion branch focuses on global refinement through selective denoising steps. The cross-attention fusion mechanism acts as a high-bandwidth conduit for inter-branch communication, allowing each component to dynamically leverage the other's state estimates without redundant recomputation. This design minimizes the need for full-sequence attention recalculations at every step—a key bottleneck in traditional diffusion models—and maximizes cache hit rates by preserving stable intermediate representations. Consequently, the framework achieves higher throughput and lower memory consumption, illustrating how structured paradigm integration can transcend the inherent limitations of purely autoregressive or diffusion-based inference.

Table 3: Zero-shot language modeling inference efficiency (1024 context, 512 generation)

Method	Throughput (Tokens/s)	Cache Hits (%)	CUDA Memory (MiB)	Parameters
GPT-2	1,200	67.23	2,341	510M
SEDD	850	45.12	3,149	490M
RADD	1,400	78.96	2,198	510M
BD3-LM	1,250	72.33	2,012	510M
SSD-LM	1,100	72.33	2,012	500M
CAD-DF	1,900	83.12	1,975	520M

5.4 Ablation Study

Ablation studies confirm each component's critical role in the CAD-DF architecture, aligning with the introduction's theoretical motivations. As demonstrated in Table 4, removing the Transformer branch causes significant performance degradation, particularly on metrics requiring local coherence and sequential integrity, underscoring the importance of autoregressive guidance in providing deterministic anchors for the diffusion process. Disabling the cross-attention fusion mechanism results in intermediate performance loss due to isolated branch operation without explicit alignment, highlighting the fusion layer's role in enabling fine-grained, token-level information exchange. The full model's optimal performance across perplexity, multi-token accuracy, and throughput metrics demonstrates that the cross-attention-driven mutual reinforcement is synergistic rather than merely additive: the diffusion branch converges faster by relying on the Transformer's local forecasts, while the Transformer branch benefits from the diffusion branch's uncertainty-aware global outlook, expanding its predictive horizon. These findings collectively affirm that the dual-branch synergy creates a unified optimization trajectory, enhancing both quality and efficiency.

Table 4: Ablation study on Prolong dataset using small models

Ablation	Perplexity	Throughput (Tokens/s)
Full CAD-DF	48.94	1,900
w/o Transformer Branch	67.21	2,300
w/o Cross Attention Fusion	58.19	2,100

6 Conclusion

In conclusion, our proposed Synergistic Absorbing Diffusion model effectively addresses the efficiency-quality trade-offs in parallel token generation by integrating a dual-branch architecture that synergistically combines the local coherence of autoregressive Transformers and the global consistency of continuous-time discrete diffusion through cross-attention fusion. Experimental results across diverse tasks, including text generation and structural reasoning, demonstrate state-of-the-art performance in perplexity, accuracy, and inference efficiency, with significant reductions in denoising steps and latency while maintaining robust global-local alignment. For future work, we plan to extend this framework to multimodal generation, explore scaling laws for larger model sizes, investigate adaptive time scheduling for further optimization, and apply the approach to real-time applications such as dialogue systems and code synthesis.

References

Marianne Arriola, Subham Sekhar Sahoo, Aaron Gokaslan, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Justin T Chiu, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tyEyYT267x.

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

Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-DPM: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=0xiJLKH-ufZ.

Joan L Bybee. 4. sequentiality as the basis of constituent structure. In *The evolution of language out of pre-language*, pp. 109–134. John Benjamins Publishing Company, 2008.

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

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

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

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *Advances in neural information processing systems*, 32, 2019.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. How to train long-context language models (effectively). In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7376–7399, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.366. URL https://aclanthology.org/2025.acl-long.366/.

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

Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. DiffusionBERT: Improving generative masked language models with diffusion models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4521–4534, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.248. URL https://aclanthology.org/2023.acl-long.248/.

Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Lm8T39vLDTE.

Yingcong Li, Yixiao Huang, Muhammed E Ildiz, Ankit Singh Rawat, and Samet Oymak. Mechanics of next token prediction with self-attention. In *International Conference on Artificial Intelligence and Statistics*, pp. 685–693. PMLR, 2024.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=CNicRIVIPA.

Saeed Masoudnia and Reza Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42(2):275–293, 2014.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*, 2021.

David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/53edebc543333dfbf7c5933af792c9c4-Paper.pdf.

OpenAI, Josh Achiam, Steven Adler, and et al. Sandhini Agarwal. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2024. URL https://arxiv.org/abs/2303.08774.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sMyXP8Tanm.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 30811–30849. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/370df50ccfdf8bde18f8f9c2d9151bda-Paper-Datasets_and_Benchmarks_Track.pdf.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. URL https://api.semanticscholar.org/CorpusID:160025533.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Andy Shih, Suneel Belkhale, Stefano Ermon, Dorsa Sadigh, and Nima Anari. Parallel sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:4263–4276, 2023.

Yuxuan Song, Zheng Zhang, Cheng Luo, Pengyang Gao, Fan Xia, Hao Luo, Zheng Li, Yuehang Yang, Hongli Yu, Xingwei Qu, Yuwei Fu, Jing Su, Ge Zhang, Wenhao Huang, Mingxuan Wang, Lin Yan, Xiaoying Jia, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Yonghui Wu, and Hao Zhou. Seed diffusion: A large-scale diffusion language model with high-speed inference. *arXiv preprint arXiv:2508.02193*, 2025. URL https://arxiv.org/abs/2508.02193.

Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=BYWWwSY2G5s.

Whitney Tabor, Bruno Galantucci, and Daniel Richardson. Effects of merely local syntactic coherence on sentence processing. *Journal of Memory and Language*, 50(4):355–370, 2004. ISSN 0749-596X. doi: https://doi.org/10.1016/j.jml.2004.01.001. URL https://www.sciencedirect.com/science/article/pii/S0749596X04000087.

Shengeng Tang, Feng Xue, Jingjing Wu, Shuo Wang, and Richang Hong. Gloss-driven conditional diffusion models for sign language production. *ACM Transactions on Multimedia Computing, Communications and Applications*, 21(4):1–17, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Chenwei Yan, Xiangling Fu, Xinxin You, Ji Wu, and Xien Liu. Graph-based cross-granularity message passing on knowledge-intensive text. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

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

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

Lingzhe Zhang, Liancheng Fang, Chiming Duan, Minghua He, Leyi Pan, Pei Xiao, Shiyu Huang, Yunpeng Zhai, Xuming Hu, Philip S Yu, et al. A survey on parallel text generation: From parallel decoding to diffusion language models. *arXiv preprint arXiv:2508.08712*, 2025.

Helena Čević, Marin Šilić, Goran Delač, and Klemo Vladimir. An overview of diffusion models for text generation. In *2023 46th MIPRO ICT and Electronics Convention (MIPRO)*, pp. 941–946, 2023. doi: 10.23919/MIPRO57284.2023.10159911.

A Appendix

A.1 Reproducibility Statement

To facilitate the reproducibility of our work, we have made extensive efforts to document our methodology and experimental setup. The core architectural details of our Synergistic Dual-Branch Continuous-Time Absorbing Diffusion model (CAD-DF) are described in Sections 3 and 4. Our training procedure, including the optimizer, learning rate schedule, and number of iterations, is detailed in Section 6. The datasets used for evaluation are listed in Section 5.1. We provide a comprehensive summary of the key hyperparameters for our small and medium model configurations in Table 5.

Table 5: Key model hyperparameters and training configurations for the CAD-DF architecture.

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	
Transformer Branch Layers	12	20
Transformer Hidden Dim	768	1440
Transformer Attention Heads	12	24
Diffusion Branch Layers	6	10
Diffusion Hidden Dim	512	1024
Diffusion Attention Heads	8	16
Vocabulary Size	50,265	

A.2 THE USE OF LARGE LANGUAGE MODELS

In this study, we utilize large language models (LLMs) as part of our preliminaries. Specifically, we employ publicly released versions of GPT-2 and GPT-3 for benchmarking text generation tasks. All prompts and generated outputs are carefully evaluated using standardized NLP metrics. We confirm that our usage complies with the licenses and terms associated with these models. We did not use LLMs for directly writing the paper or generating any core methodological content.

A.3 SAMPLES OF USING LLMS

Question: Fully analyze the RADD's Theory in formulae, none of titles or bolds or listed structure. (With RADD paper uploaded) The core theoretical contribution of RADD lies in its reparameterization of the concrete score in absorbing discrete diffusion models, revealing that this quantity can be expressed through time-independent conditional probabilities of the clean data, multiplied by an analytic, time-dependent scalar. This insight fundamentally simplifies the model and enables more efficient computation.

Consider a continuous-time Markov chain with a transition rate matrix defined by $Q_t = \sigma(t)Q^{\text{absorb}}$, where Q^{absorb} is the absorbing transition matrix. For a state x_t at time t, the concrete score between two states differing only at the i-th dimension (where $x_t^i = [M]$ and $\hat{x}_t^i \neq [M]$) is given by:

$$\frac{p_t(\hat{x}_t)}{p_t(x_t)} = \frac{e^{-\bar{\sigma}(t)}}{1 - e^{-\bar{\sigma}(t)}} \cdot p_0(\hat{x}_t^i \mid x_t^{\text{UM}}).$$

Here, $\bar{\sigma}(t) = \int_0^t \sigma(s)\, ds$, and x_t^{UM} denotes the unmasked tokens in x_t. This factorization shows that the concrete score decomposes into a known function of time and the conditional distribution of the clean data, which is independent of time.

This reparameterization motivates the removal of time-conditioning from the network. Instead of learning a time-dependent score function $s_\theta(x_t, t)$, RADD uses a time-independent network $c_\theta(x_t)$ that directly estimates the conditional probabilities $p_0(\cdot \mid x_t^{\text{UM}})$. The output is defined over the vocabulary, with the softmax activation ensuring it represents a valid probability distribution. Specifically, for each masked position, the network outputs a distribution over possible tokens; for unmasked positions, the output is irrelevant since those tokens remain unchanged.

The training objective for absorbing diffusion, originally formulated as the denoising score entropy (DSE) loss, can be simplified using this reparameterization. The DSE loss is:

$$\mathcal{L}_{\text{DSE}}^T(x_0) = \int_0^T \mathbb{E}_{x_t \sim p_{t|0}(\cdot|x_0)} \left[\sum_{\hat{x}_t} Q_t(\hat{x}_t, x_t) \left(s_\theta(x_t, t)_{\hat{x}_t} - \frac{p_t(\hat{x}_t)}{p_t(x_t)} \right)^2 \right] dt.$$

Substituting the reparameterized form and simplifying leads to an equivalent loss called the t-denoising cross-entropy (t-DCE) loss:

$$\mathcal{L}_{t\text{-DCE}}^T(x_0) = \int_0^T \mathbb{E}_{x_t \sim p_{t|0}(\cdot|x_0)} \left[- \sum_{i: x_t^i = [M]} \sum_{j \neq [M]} p_0(x_0^i = j \mid x_t^{\text{UM}}) \log c_\theta(x_t)[i, j] \right] dt.$$

A change of variables from t to $\lambda = 1 - e^{-\bar{\sigma}(t)}$ further transforms this objective. Here, λ represents the probability that a token is masked by time t. The t-DCE loss becomes:

$$\mathcal{L}_{\lambda\text{-DCE}}(x_0) = \int_0^1 \mathbb{E}_{x_\lambda \sim p_\lambda(\cdot|x_0)} \left[- \sum_{i: x_\lambda^i = [M]} \sum_{j \neq [M]} p_0(x_0^i = j \mid x_\lambda^{\text{UM}}) \log c_\theta(x_\lambda)[i, j] \right] d\lambda,$$

where $p_\lambda(x_\lambda | x_0)$ is the distribution induced by independently masking each token with probability λ.

Finally, this λ-DCE loss can be integrated analytically over λ, yielding an expression that is equivalent to the training objective of any-order autoregressive models (AO-ARMs):

$$\mathcal{L}_{\text{AO}}(x_0) = \mathbb{E}_{\pi \sim U_\pi} \left[- \sum_{l=1}^{d} \log p(x_0^{\pi(l)} \mid x_0^{\pi(<l)}) \right].$$

This equivalence holds when the final noise level $\bar{\sigma}(T) \to \infty$, ensuring all tokens are masked with probability one. Thus, the upper bound on the negative log-likelihood for diffusion models is interpreted as the expected negative log-likelihood over all factorizations for AO-ARMs.

In sampling, the reverse process can leverage the time-independent network c_θ. Since tokens transition from masked to unmasked only once, the output of c_θ can be cached when the noisy sample remains unchanged over a sampling interval. This reduces the number of function evaluations (NFEs). The expected NFEs (E-NFEs) for a sequence of length d and n sampling steps is:

$$\text{E-NFEs} = \sum_{k=1}^{n} \mathbb{E}[N_k],$$

where N_k is the number of tokens that change at step k. For a log-linear noise schedule, this simplifies to:

$$\text{E-NFEs} = d \cdot \left(1 - (1 - \frac{1}{n})^n\right),$$

which is less than d for $n > 1$, demonstrating the efficiency gain from caching.