CoDAR: Continuous Diffusion Language Models are More Powerful Than You Think

Junzhe Shen[1], **Jieru Zhao, Ziwei He, Zhouhan Lin**[1‡]

[1] LUMIA Lab, School of Artificial Intelligence, Shanghai Jiao Tong University

lin.zhouhan@gmail.com ‡ Corresponding Author.

Abstract We study why continuous diffusion language models (DLMs) have lagged behind discrete diffusion approaches despite their appealing continuous generative dynamics. Under a controlled token-recovery study, we identify token rounding, the final projection from denoised embeddings to tokens, as a primary bottleneck. Building on these insights, we propose CoDAR (**Co**ntinuous **D**iffusion with Contextual **A**uto**R**egressive Decoder), a two-stage framework that keeps diffusion entirely continuous in an embedding space while learning a strong, context-conditional discretizer: an autoregressive Transformer decoder that cross-attends to the denoised embedding sequence and performs contextualized rounding to tokens. Experiments on LM1B and OpenWebText demonstrate that CoDAR substantially improves generation quality over latent diffusion and becomes competitive with strong discrete DLMs, while exposing a simple decoder-temperature knob to navigate the fluency–diversity trade off.

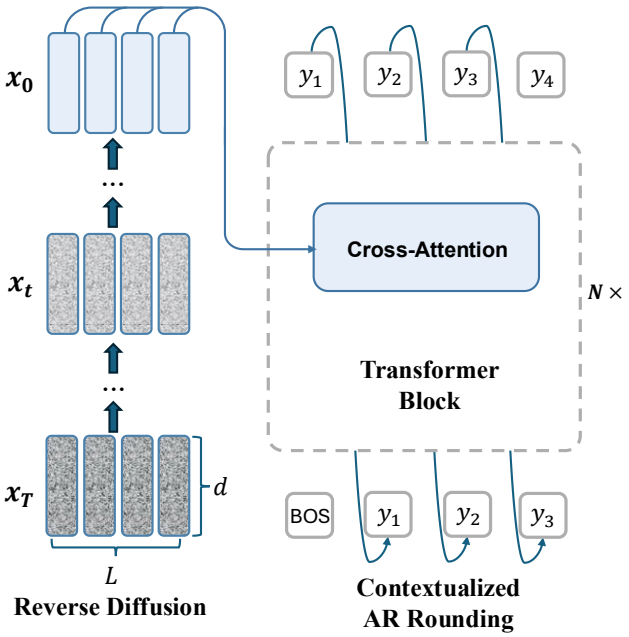

Figure 1 | Framework of CoDAR . Starting from a noisy latent sequence x_T, a reverse diffusion process progressively denoises the hidden states to x_0. $\mathbf{x}_T, \ldots, \mathbf{x}_0 \in \mathrm{R}^{L\times d}$, where L denotes the sequence length and d denotes the size of hidden states. After that, an autoregressive Transformer decoder conditions on the denoised $\mathbf{x}_0$ with cross-attention to translate $\mathbf{x}_0$ to discrete tokens $\mathbf{y}_1, \ldots, \mathbf{y}_L$.

1. Introduction

Continuous diffusion models have achieved remarkable success in domains such as image generation and structured latent spaces, where they demonstrate strong modeling capacity and sample quality. However, their adoption and scalability in natural language processing remain limited. In contrast to continuous media, language is inherently discrete, leading to fundamental challenges when applying continuous generative processes directly to text. While continuous formulations offer theoretical

advantages [Pynadath et al., 2025, Zhou et al., 2025], including stronger theoretical expressivity and smoother latent reasoning, continuous space diffusion language models (DLMs) have fallen behind discrete DLMs [He et al., 2022, Li et al., 2025].

A core difficulty lies in the misalignment between continuous diffusion and the discrete nature of language: diffusion processes typically operate over continuous spaces, whereas linguistic units such as tokens are inherently categorical. Prior works on continuous DLMs have attempted to bridge this gap using a rounding step [Gao et al., 2024, Li et al., 2022, Lin et al., 2023, Strudel et al., 2022] or formulating diffusion in the logit space [Mahabadi et al., 2024, Tae et al., 2025]. In parallel, embedding-based latent diffusion methods such as *Latent Diffusion for Language Generation* [Lovelace et al., 2023] learn diffusion in the latent space of a language autoencoder built on a pretrained encoder–decoder LM. While this provides a principled continuous space that is decodable by construction, it ties the method to encoder–decoder language models and thus narrows the choice of representations.

Beyond the continuous–discrete mismatch, a second difficulty lies in the high dimensionality of latent representations. In latent diffusion for images, recent studies document an optimization dilemma where increasing latent capacity (often via higher-dimensional latents) improves reconstruction but can hinder diffusion training and generation quality unless additional alignment or regularization is introduced [Lai et al., 2025, Yao et al., 2025, Zheng et al., 2025a]. Although language differs from vision, analogous pressures can arise when diffusing over high-dimensional embedding sequences. Due to these difficulties, research largely pivoted toward discrete modeling strategies that match the token-level structure of language [Lou et al., 2024, Sahoo et al., 2024] as the field matured.

We argue that the performance gap between continuous DLMs and discrete DLMs is not solely due to the diffusion objective itself, but rather to a *rounding bottleneck*: the challenge of mapping noisy continuous embeddings back to discrete tokens in a large space. Most existing embedding-space DLMs typically rely on a rounding operator to recover tokens [Li et al., 2022, Lin et al., 2023]. Such operators treat each position largely independently and provide limited ability to leverage linguistic context when the denoised embedding is ambiguous or off manifold. Conversely, discrete or simplex-space DLMs avoid explicit rounding but must operate directly in a categorical state space, which changes the learning dynamics and often shifts complexity into the diffusion transition design and sampling procedure [Lou et al., 2024, Sahoo et al., 2024, Tae et al., 2025].

In this work, we analyze the rounding bottleneck of continuous DLMs both theoretically and empirically, and propose CoDAR (**Co**ntinuous **D**iffusion with Contextual **A**uto**R**egressive Decoder), a novel framework that effectively resolves the two aforementioned problems by *keeping diffusion entirely continuous* while *learning a powerful, context-conditional rounding module*. Concretely, CoDAR factorizes generation into (i) a continuous diffusion process in an embedding space that is favorable for diffusion, and (ii) an autoregressive Transformer decoder that maps the generated continuous states to discrete tokens via cross-attention. This design keeps the diffusion component simple and fully continuous, and delegates the hardest part of decoding to a model class that is known to excel at sequence transduction.

We summarize our main contributions as follows:

- We identify token rounding, especially under low dimensional hidden states, as a principal bottleneck for continuous embedding DLMs theoretically and empirically, showing why pointwise classifiers can be suboptimal and validating this with controlled token-recovery experiments.
- We propose CoDAR , a two-stage continuous diffusion language modeling framework: a continuous diffusion generator produces embedding sequences, and an autoregressive Transformer decoder performs contextual rounding back to tokens.
- We show that CoDAR improves generation quality over latent diffusion, and closes the gap

between discrete DLMs.

2. Related Work

2.1. Continuous Diffusion Language Models

Diffusion-LM [Li et al., 2022] models sequences as Gaussian vectors that are iteratively denoised into word vectors, enabling plug-and-play controllable generation. Building on the embedding-space perspective, Self-conditioned Embedding Diffusion [Strudel et al., 2022] introduces self-conditioning for diffusion over token embeddings and demonstrates competitive generation quality with potential inference efficiency benefits. Difformer [Gao et al., 2024] analyzes optimization pathologies in embedding diffusion (e.g., embedding collapse and schedule issues) and proposes anchor loss and noise rescaling to stabilize training and improve generation across tasks. GENIE [Lin et al., 2023] frames diffusion for pre-training, using an encoder plus diffusion-based decoder and a continuous paragraph denoise objective to reconstruct coherent paragraphs from corrupted inputs. LD4LG [Lovelace et al., 2023] trains diffusion in the latent space of an encoder–decoder language autoencoder, sampling compact continuous latents that are decoded by a pretrained decoder. More recent variants explore where to diffuse and how to better respect token discreteness: TESS [Mahabadi et al., 2024] performs diffusion on the logit simplex (rather than learned embeddings) with self-conditioning for fully non-autoregressive text-to-text generation. Smoothie [Shabalin et al., 2025] proposes progressively smoothing token embeddings by semantic similarity to combine semantic structure with a more natural decoding process.

2.2. Hybrid Architectures

AR-Diffusion Hybrid A growing line of work explores hybrid diffusion–autoregressive architectures that aim to combine diffusion's global refinement/parallelism with AR decoding's fluency and KV-cache-friendly generation. AR-Diffusion [Wu et al., 2023] injects causal, left-to-right structure into diffusion by using a position-dependent (dynamic) number of denoising steps so left tokens "settle" earlier and condition later ones. DGLM [Lovelace et al., 2024] uses a diffusion model to generate continuous semantic proposals (soft prompts) that steer a strong AR LM toward desired attributes. SDLM [Liu et al., 2025b] further "retrofit" pretrained AR LMs by performing diffusion within masked blocks but decoding consecutive subsequences adaptively (via Next Sequence Prediction) to maintain KV-cache compatibility and handle variable uncertainty across the sequence. Moving toward single-model synergy, TiDAR [Liu et al., 2025a] explicitly separates roles by drafting in diffusion while sampling final outputs autoregressively using structured attention masks to achieve high throughput without sacrificing AR-quality. For reasoning-centric settings, LaDiR [Kang et al., 2025] augments an existing LLM with a VAE-defined latent "thought" space and a latent diffusion model that iteratively refines blockwise reasoning trajectories, enabling more holistic revision than pure AR chain-of-thought. Planner and Executor [Berrayana et al., 2025] studies explicit collaboration where a discrete diffusion model plans and an AR model executes, showing that shifting diffusion to AR communication from text to latent space via a learned projector can markedly improve reasoning accuracy while reducing token-cost.

Continuous-Discrete Hybrid Hybrid continuous–discrete diffusion LMs model tokens and continuous representations jointly, rather than diffusing in only one space. CCDD [Zhou et al., 2025] co-evolves discrete tokens and continuous states in a single joint diffusion, aiming to preserve continuous latent expressivity while improving trainability via explicit discrete structure. CANDI [Pynadath et al., 2025] decouples discrete identity corruption from continuous geometric degradation to avoid a mismatch in effective noise regimes, enabling useful continuous gradients while retaining conditional structure.

CADD [Zheng et al., 2025b] augments mask-based discrete diffusion with a paired continuous latent that replaces the [MASK] "information void" with noisy but informative vectors that guide denoising and offer controllable diversity–precision trade-offs. Compared with these interleaved hybrid processes, our approach keeps diffusion entirely continuous in embedding space and delegates discretization to a separate contextualized decoder.

3. Theoretical Analysis

Let $Y = (Y_1, \ldots, Y_L)$ be a length-L token sequence and let $X \in \mathbb{R}^{L\times d}$ denote the denoised continuous sequence produced by the diffusion generator. Any rounding (discretization) procedure is implicitly performing posterior inference:

$$\hat{y} \in \arg\max_{y} \; p(y \mid X). \tag{1}$$

Many embedding-space DLMs implement rounding with a position-wise linear head [Gao et al., 2024, Li et al., 2022, Lin et al., 2023], which corresponds to the approximate factorization $p(y \mid X) \approx \prod_{i=1}^{L} p(y_i \mid X_i)$, treating token recovery as independent classification at each position i.

Entropy and conditional total correlation. Throughout, $H(\cdot)$ denotes Shannon entropy (in nats when log is natural). For discrete random variables,

$$H(Y \mid X) \;=\; \mathbb{E}_x\Big[-\sum_{y} p(y \mid x)\log p(y \mid x)\Big], \tag{2}$$

and similarly $H(Y_i \mid X_i)$ is the remaining uncertainty of Y_i after observing only the local vector X_i. The *conditional total correlation* (conditional TC) measures residual dependence among $(Y_1, \ldots, Y_L)$ given X:

$$\begin{aligned} \mathrm{TC}(Y \mid X) &= \mathbb{E}_x\Big[D_{\mathrm{KL}}\Big(p(y \mid x) \,\Big\|\, \prod_{i=1}^{L} p(y_i \mid x)\Big)\Big] \\ &= \sum_{i=1}^{L} H(Y_i \mid X) \;-\; H(Y \mid X) \;\geq\; 0. \end{aligned} \tag{3}$$

Locality gap vs. dependence gap. Because X contains (weakly) more information than X_i, conditioning reduces entropy: $H(Y_i \mid X) \leq H(Y_i \mid X_i)$. This yields a useful decomposition:

$$\sum_{i=1}^{L} H(Y_i \mid X_i) \;-\; H(Y \mid X) = \underbrace{\Big(\sum_{i=1}^{L} H(Y_i \mid X) \;-\; H(Y \mid X)\Big)}_{\mathrm{TC}(Y|X)} + \underbrace{\sum_{i=1}^{L}\Big(H(Y_i \mid X_i) - H(Y_i \mid X)\Big)}_{\text{locality gap}} \geq \; \mathrm{TC}(Y \mid X). \tag{4}$$

Intuitively, $\mathrm{TC}(Y \mid X)$ captures *intrinsic sequence coupling* (syntax/semantics, long-range constraints) that remains even if the decoder sees the entire X, whereas the locality gap captures extra uncertainty introduced when a decoder is restricted to per-position evidence X_i rather than the full context X.

Proposition 1 (Optimality gap of pointwise decoding)**.** *Let $\mathcal{D}_{\mathrm{pw}}$ be the set of all decoders that factorize as $\prod_{i=1}^{L} q_i(y_i \mid X_i)$, and let $\mathcal{D}_{\mathrm{seq}}$ be the set of all conditional sequence decoders $q(y \mid X)$. Consider the expected negative log-likelihood (NLL) risk*

$$\mathcal{R}(q) = \mathbb{E}_{(X,Y)}[-\log q(Y \mid X)]. \tag{5}$$

Then,

$$\min_{q \in \mathcal{D}_{\text{pw}}} \mathcal{R}(q) \; - \; \min_{q \in \mathcal{D}_{\text{seq}}} \mathcal{R}(q) \; = \; \sum_{i=1}^{L} H(Y_i \mid X_i) \; - \; H(Y \mid X) \\ \geq \; \text{TC}(Y \mid X) \; \geq \; 0. \tag{6}$$

Proof sketch. The Bayes-optimal conditional model for $\mathcal{D}_{\text{seq}}$ is $q^*(y \mid X) = p(y \mid X)$, achieving risk $H(Y \mid X)$. For $\mathcal{D}_{\text{pw}}$, the minimizer is $q_i^*(y_i \mid X_i) = p(y_i \mid X_i)$ independently for each i, achieving $\sum_i H(Y_i \mid X_i)$. Subtracting yields equation 6. The lower bound by $\text{TC}(Y \mid X)$ follows from equation 4. Refer to Appendix A for detailed proof.

Interpretation and implications for rounding. Equation equation 6 formalizes two reasons why a linear LM head can be brittle for rounding diffusion outputs: (i) *sequence dependence:* the gap vanishes only if $Y_1, \ldots, Y_L$ are conditionally independent given X (i.e., $\text{TC}(Y \mid X) = 0$), which is rarely true for natural language; (ii) *local evidence restriction:* even if a full-context decoder could in principle exploit X to resolve ambiguity, a per-position head that only sees X_i incurs the locality gap in equation 4. In realistic diffusion sampling, X is an imperfect denoised embedding sequence: it may be off-manifold, slightly inconsistent across positions, or contain structured errors. Such imperfections typically increase both conditional dependence (larger $\text{TC}(Y \mid X)$) and locality gap, making point-wise decoding strictly suboptimal.

Why increasing d helps but does not eliminate the issue. A larger embedding dimension d increases the capacity of each X_i to encode information about Y_i. Formally,

$$H(Y_i \mid X_i) \; = \; H(Y_i) \; - \; I(Y_i; X_i), \tag{7}$$

so improving the mutual information $I(Y_i; X_i)$ (often easier when d is larger) reduces *marginal* ambiguity. However, lowering these marginal ambiguities does not imply that sequence recovery becomes pointwise: the approximation $p(y \mid X) \approx \prod_i p(y_i \mid X_i)$ would require that (i) tokens become conditionally independent given the full X, and (ii) local evidence X_i is as informative as the global context X. In general, neither holds. Moreover, even if the clean embedding manifold were perfectly invertible, diffusion-generated X can be slightly off-manifold. In that case, X may be consistent with multiple nearby token sequences. Choosing the one that is globally coherent requires reasoning over the whole sequence, not each position independently. This explains why improving per-position separability (e.g., by increasing d) can improve linear head accuracy, yet still leaves a substantial gap to contextual decoding.

To test whether the theoretical gap in equation 6 is practically significant, we run a controlled study. We train two families of token decoders to map hidden states h_i back to tokens x_i: (i) a position-wise linear classifier $p(x_i \mid h_i)$; and (ii) an autoregressive Transformer decoder that predicts x_i conditioned on previously recovered tokens and cross-attends to the full continuous sequence $H = [h_1, \ldots, h_L]$. For each position, we select the token with highest predicted probability as the predicted token, and token recovery is computed as the rate at which the predicted token matches the true token. The token recovery accuracy in shown in Figure 2.

Key observation. As shown in Figure 2, the Transformer decoder recovers tokens with high accuracy across both low- and high-dimensional representations (e.g., 0.841 at d=64 and 0.915 at d=768), while the Linear baseline performs poorly (0.154 and 0.297, respectively). This is consistent with equation 6: a linear head is constrained to $\mathcal{D}_{\text{pw}}$ and therefore cannot exploit sequence-level coupling (nonzero $\text{TC}(Y \mid X)$) nor full-context evidence (locality gap), both of which are crucial for reliable rounding when h_i are imperfect.

These results motivate our two-stage approach, where continuous diffusion generation and contextual token rounding are decoupled and separately learned:

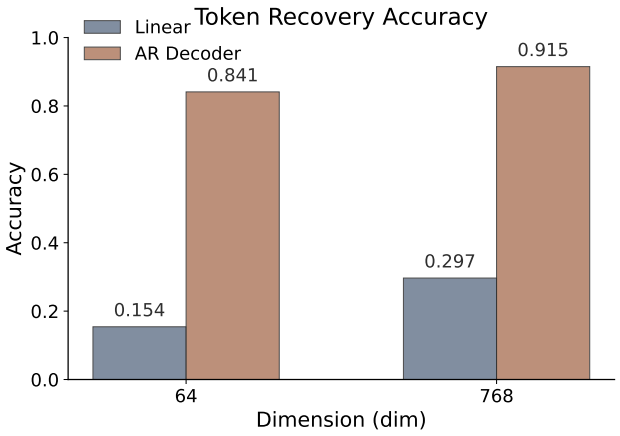

Figure 2 | Token recovery rate of point-wise linear classifier and autoregressive Transformer decoder under different sizes of hidden states.

- a continuous diffusion model generates a sequence of token embeddings in $\mathbb{R}^{L\times d}$;
- an autoregressive Transformer decoder with cross-attention maps the generated embeddings back to discrete tokens.

This design addresses the previously noted trainability bottleneck by explicitly learning a dedicated embedding-to-token decoder, while keeping the diffusion process entirely continuous and allowing free choice of embeddings. In particular, diffusion no longer needs to land exactly on token embeddings at every position; it only needs to generate continuous states that are *decodable under context*, allowing the decoder to leverage linguistic regularities to resolve residual dependence and local ambiguity predicted by equation 4.

4. Continuous Diffusion with Contextual AutoRegressive Decoder

Let $\mathbf{y} = (y_1, \ldots, y_L)$ be a token sequence with vocabulary size $|V|$. A frozen embedding model $E : V^L \to \mathbb{R}^{L\times d}$ maps tokens to continuous representations $\mathbf{x}_0 = E(\mathbf{y})$. We denote the diffusion state at time $t \in [0, 1]$ as $\mathbf{x}_t \in \mathbb{R}^{L\times d}$. Our denoiser $f_\theta(\mathbf{x}_t, t)$ operates purely in the continuous space, while a separate autoregressive decoder $p_\phi(\mathbf{y} \mid \hat{\mathbf{x}}_0)$ performs discrete realization. Unless stated otherwise, $E(\cdot)$ is fixed and only (θ, ϕ) are optimized. The framework of CoDAR is shown in Figure 1.

4.1. Continuous Diffusion for Embedding Generation

Given a tokenized sequence $\mathbf{y} = (y_1, \ldots, y_L)$, we obtain continuous embeddings

$$\mathbf{x}_0 = E(\mathbf{y}) \in \mathbb{R}^{L\times d},$$

where $E(\cdot)$ can be an arbitrary pretrained text embedding model.

We define a variance-preserving (VP) continuous diffusion process[Song et al., 2020] on embeddings by corrupting $\mathbf{x}_0$ with Gaussian noise:

$$\mathbf{x}_t = \alpha(t)\mathbf{x}_0 + \sigma(t)\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

where $t \sim \mathcal{U}(0, 1)$, and $\alpha(t), \sigma(t)$ are a noise schedule satisfying $\alpha(t)^2 + \sigma(t)^2 = 1$. We use cosine schedule proposed by Nichol and Dhariwal [2021].

We use the velocity parameterization [Salimans and Ho, 2022] because it interpolates between predicting noise and predicting data, and has been shown to improve stability—especially when sampling with few steps—relative to direct ϵ-prediction. Concretely, we train the denoiser to predict the *velocity*:

$$\mathbf{v}_t \triangleq \alpha(t)\boldsymbol{\epsilon} - \sigma(t)\mathbf{x}_0,$$

We parameterize the denoiser as

$$\hat{\mathbf{v}}_\theta = f_\theta(\mathbf{x}_t, t),$$

and recover estimates of $\mathbf{x}_0$ and $\boldsymbol{\epsilon}$ via

$$\hat{\mathbf{x}}_0 = \alpha(t)\mathbf{x}_t - \sigma(t)\hat{\mathbf{v}}_\theta, \qquad \hat{\boldsymbol{\epsilon}} = \sigma(t)\mathbf{x}_t + \alpha(t)\hat{\mathbf{v}}_\theta.$$

Under the VP formulation with v-prediction, the denoiser's objective is to align its output $\hat{\mathbf{v}}_\theta$ with the ground-truth velocity $\mathbf{v}_t = \alpha(t)\boldsymbol{\epsilon} - \sigma(t)\mathbf{x}_0$. We optimize the following velocity prediction loss:

$$\mathcal{L}_{\text{diff}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1),\,\mathbf{x}_0,\,\boldsymbol{\epsilon}\sim\mathcal{N}(0,I)}\left[w(t)\cdot\|f_\theta(\mathbf{x}_t,t)-\mathbf{v}_t\|_2^2\right],$$

where $w(t)$ is a time-dependent weighting function that can improve optimization stability (for example, constant weighting or SNR-based schemes used in image diffusion literature[Ho et al., 2020, Karras et al., 2022, Nichol and Dhariwal, 2021]).

4.2. Contextualized Rounding with AR Decoder

To map the generated continuous embeddings back to text, we employ a Transformer decoder p_ϕ that uses cross-attention over the denoised embedding sequence. Formally, the conditional likelihood of a token sequence $\mathbf{y} = (y_1, \ldots, y_L)$ given the recovered embeddings is factorized as

$$p_\phi(\mathbf{y} \mid \hat{\mathbf{x}}_0) = \prod_{i=1}^{L} p_\phi(y_i \mid y_{<i}, \hat{\mathbf{x}_0}),$$

This two-stage strategy first runs diffusion in a continuous latent space, and then decodes the result into discrete text. This is similar to the latent diffusion for language setup in LD4LG [Lovelace et al., 2023], which combines a latent model with a encoder-decoder language model to produce text from latent vectors. Unlike LD4LG, we do not need to rely on an encoder–decoder language model and can use off-the-shelf, state-of-the-art text embedding models instead.

Under teacher forcing, the decoder is trained to reconstruct the ground-truth tokens from the (ideally denoised) continuous embeddings with the standard cross-entropy objective:

$$\mathcal{L}(\phi;\theta) = \mathbb{E}\left[-\sum_{i=1}^{L}\log p_\phi(y_i \mid y_{<i}, \mathbf{x}_0)\right].$$

To improve robustness and generalization to the imperfect outputs produced by the diffusion model, we follow the noise-augmentation strategy from Representation Autoencoder [Zheng et al., 2025a] training: we add a small Gaussian perturbation $\mathbf{n} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ to the recovered embeddings during decoder training. This encourages the decoder to tolerate and correct slight deviations from clean embeddings

The resulting noise-augmented decoding loss becomes

$$\mathcal{L}_{\text{dec}}(\phi;\theta) = \mathbb{E}\left[-\sum_{i=1}^{L}\log p_\phi(y_i \mid y_{<i}, \mathbf{x}_0 + \mathbf{n})\right],$$

which trains the decoder to be resilient to small perturbations in its conditioning embeddings and better aligned with the outputs of the diffusion denoiser.

4.3. Inference

At inference time, generation proceeds in two stages. First, we sample a continuous embedding sequence by running the learned reverse diffusion process: starting from Gaussian noise $\mathbf{x}_{t=1} \sim \mathcal{N}(0, I)$, we iteratively apply the denoiser $f_\theta(\mathbf{x}_t, t)$ (with a chosen numerical solver) to progressively remove noise and obtain a final denoised embedding sequence $\mathbf{x}_0 \in \mathbb{R}^{L \times d}$. Second, we translate these hidden states into tokens using the autoregressive Transformer decoder p_ϕ conditioning on $\hat{\mathbf{x}}_0$ via cross-attention. The decoder samples $y_i \sim p_\phi(y_i \mid y_{<i}, \hat{\mathbf{x}}_0)$ until an end-of-sequence token is produced (or a length limit is reached). In this way, diffusion handles global continuous-sequence generation, while the AR decoder performs discrete token realization from the generated hidden states.

5. Experiment

Table 1 | Unconditional generation on OpenWebText. We compare CoDAR to discrete baselines (MDLM and SEDD) using 1,000 generated samples per method, reporting generative perplexity and an n-gram–based diversity score. We also include three diagnostic reference points: training set text, decoded ground-truth embeddings ("recovered training set"), and decoded Gaussian noise, to contextualize the metrics. We use 250 sampling steps for all runs while varying the decoder temperature for CoDAR .

Model	Gen. PPL(↓)	Diversity(↑)
Training set	16.75	0.2191
Recovered training set	25.07	0.2282
Decoded noise	14.24	0.0380
MDLM*	123.73	0.4784
SEDD	129.57	0.4742
CoDAR ($T = 0.00$)	47.71	0.1660
CoDAR ($T = 0.25$)	50.68	0.1937
CoDAR ($T = 0.50$)	66.31	0.2670
CoDAR ($T = 0.75$)	109.80	0.3718
CoDAR ($T = 1.00$)	164.90	0.4842

5.1. Setup

We evaluate the language modeling capability of our model via unconditional text generation, focusing on the quality of generated samples. We compare CoDAR against a latent diffusion language model (LD4LG [Lovelace et al., 2023]) and two strong discrete diffusion baselines, MDLM [Sahoo et al., 2024] and SEDD [Lou et al., 2024]. For all comparisons, we report generative perplexity (Gen. PPL) as a proxy for fluency (lower is better), and an n-gram–based diversity metric as a proxy for lexical variety (higher is better) following previous work [Lovelace et al., 2023].

To compare against the latent diffusion baseline LD4LG, we train CoDAR and LD4LG on the One

Billion Word Benchmark (LM1B) [Chelba et al., 2013] using the standard data split. We train models using sequence length $L = 128$ with sentence packing. We train both models for 250k steps. For LD4LG, we use the BART variant and follow the baseline's standard autoencoder setup and training procedure as reported in the original work.

For the comparison with discrete baselines, we train all models on the OpenWebText [Gokaslan and Cohen, 2019] dataset. Since OpenWebText is substantially larger and more diverse than LM1B, it provides a stronger stress test for unconditional generation quality. We train CoDAR for 250k steps using context length $L = 512$ with sequence packing. For a fair comparison, we retrain MDLM and SEDD following prior practice [Lou et al., 2024, Sahoo et al., 2024, 2025] using standard training iterations (1M steps) but matching our context length of 512. We use the Qwen2 tokenizer [Zhang et al., 2025] for CoDAR and for baselines when possible. LD4LG cannot use the Qwen2 tokenizer because its BART-based autoencoder is tied to the BART vocabulary. MDLM does not work well with Qwen2 tokenizer (resulting in very high generative perplexity), which is consistent with observations in Zhou et al. [2025]. We therefore use the GPT-2 tokenizer for MDLM (marked as * in Table 1 and Table 4).

Evaluation metrics. For each method, we generate 1,000 unconditional samples and compute generative perplexity (Gen. PPL) computed by GPT2-large [Radford et al., 2019] to measure fluency. To quantify diversity, we use the metric used in LD4LG, defined as the product of distinct n-gram ratios for $n \in \{2, 3, 4\}$:

$$\text{Div} = \prod_{n=2}^{4} \frac{|\text{unique } n\text{-grams}(\{\mathbf{w}_i\})|}{|\text{total } n\text{-grams}(\{\mathbf{w}_i\})|},$$

where $\{\mathbf{w}_i\}$ is a set of generated samples.

To contextualize the generative perplexity (Gen. PPL) and diversity scores, we report three reference points.

- Training set: Text drawn directly from the *training set*, serving as a data-distribution anchor under our evaluation protocol.
- Recovered training set: obtained by decoding ground-truth text embeddings, reflecting reconstruction error induced by the decoder.
- Decoded noise: Decoding pure Gaussian noise with our contextualized decoder, which exhibit low Gen. PPL but extremely low diversity, indicative of a degenerate mode of highly repetitive text.

These probes show that (i) the decoder is capable of producing fluent text, but (ii) a meaningful generative model must match *both* fluency and diversity, not fluency alone.

Training Details For the diffusion model, we use the same model architecture as MDLM[Sahoo et al., 2024]. For the contextualized autoregressive decoder, we use the same model architecture as GPT2-small [Radford et al., 2019] but with additional cross attention. The decoder is trained for 1 epoch on OpenWebText. We choose Qwen3-Embedding [Zhang et al., 2025] as CoDAR 's pretrained embedding as it supports custom dimensions for the final embedding. We set the dimension of embeddings to 64 unless otherwise specified. All embeddings are normalized following Rombach et al. [2022].

Table 2 | Unconditional generation on LM1b.

Model	Gen.PPL(↓)	Diversity(↑)
LD4LG	167.47	0.5797
CoDAR	104.76	0.3264

5.2. Main Results

On OpenWebText unconditional generation (Table 1), CoDAR spans a smooth fluency–diversity frontier by varying the decoder temperature while keeping the model and procedure fixed. At low temperatures, CoDAR is markedly more fluent than discrete baselines: Gen. PPL drops to 47.71 (T=0.00) and 50.68 (T=0.25), yet diversity remains non-trivial (0.1660–0.1937) and far from the collapse of decoded noise (diversity 0.0380). As temperature increases, diversity rises monotonically (0.2670 → 0.3718 → 0.4842 for T=0.50, 0.75, 1.00) with a corresponding increase in Gen. PPL (66.31 → 109.80 → 164.90), forming a clear Pareto trade-off. Crucially, at T=1.00 CoDAR reaches diversity 0.4842, matching or slightly exceeding MDLM (0.4784) and SEDD (0.4742), showing that CoDAR can operate in the same diversity regime as strong discrete counterparts, while offering substantially better fluency when the operating point favors it. On LM1B unconditional generation, we reuse the decoder trained with OpenWebText and the decoder's temperature is set to 1. The results are shown in Table 2, CoDAR significantly outperforms LD4LG in terms of fluency while maintaining nontrivial diversity.

5.3. Ablations

5.3.1. Sampler and Sampling Steps

As the diffusion process of CoDAR is fully continuous, we can leverage higher-order numerical solvers to improve few-step sampling. We compare standard ancestral sampling to DPM-Solver [Lu et al., 2022] while varying the number of sampling steps from 250 down to 25. We fix the decoder temperature to 1.0 for all runs. The results are shown in Table 3.

Across all step budgets, DPM-Solver consistently yields better fluency (lower Gen. PPL) than ancestral sampling while maintaining high diversity. For example, at 250 steps, DPM-Solver reduces Gen. PPL from 164.90 (ancestral) to 147.53, with comparable diversity. The advantage is even more pronounced in the low-step regime: at 100 steps, DPM-Solver achieves 154.83 Gen. PPL versus 185.91 for ancestral sampling, while keeping diversity near 0.495. Even at 25 steps, DPM-Solver preserves strong diversity and slightly improves Gen. PPL (212.32 vs. 214.86). Overall, these results show that advanced solvers make CoDAR substantially more effective for fast generation, improving sample quality without sacrificing diversity.

Table 4 | Few-step unconditional generation on OpenWebText. We compare our model sampled with DPM-Solver (decoder temperature $T = 1$) against discrete diffusion baselines (MDLM, SEDD) under matched step budgets.

Sampling Steps	Model	Gen.PPL(↓)	Div.(↑)
25	MDLM*	232.78	0.5287
	SEDD	221.63	0.5171
	CoDAR	212.32	0.4929
50	MDLM*	165.71	0.5046
	SEDD	164.24	0.492
	CoDAR	178.82	0.4942
100	MDLM*	137.62	0.4877
	SEDD	143.19	0.481
	CoDAR	154.83	0.4947
250	MDLM*	123.73	0.4784
	SEDD	131.96	0.4742
	CoDAR	147.53	0.488

Table 3 | Solver ablation for CoDAR . We compare ancestral sampling to DPM-Solver for unconditional generation across sampling budgets with decoder temperature set to 1.0.

Solver	Sampling Steps	Gen.PPL(↓)	Div.(↑)
Ancestral	25	214.86	0.4251
	50	206.54	0.4734
	100	185.91	0.4757
	250	164.89	0.4842
DPM-Solver	25	212.32	0.4929
	50	178.82	0.4942
	100	154.83	0.4947
	250	147.53	0.488

Now we rival discrete baselines (MDLM/SEDD) in few-step generation by combining CoDAR with DPM-Solver. Table 4 reports Gen. PPL (↓) and diversity (↑) at matched sampling budgets (25/50/100/250 steps), with decoder temperature fixed to T=1 for CoDAR .

At the most aggressive budget of 25 steps, CoDAR achieves the best fluency among all methods, while retaining strong diversity. As the step budget increases, MDLM/SEDD become more fluent, but CoDAR remains competitive and stays in the same diversity regime: at 50–250 steps, CoDAR yields diversity around 0.49, comparable to the baselines. For instance, at 250 steps, CoDAR attains 0.488 diversity versus 0.478/0.474 for MDLM/SEDD, though with higher Gen. PPL. At intermediate budgets (100 steps), CoDAR reaches 154.83 Gen. PPL with 0.4947 diversity, surpassing the baselines' diversity (0.4877/0.4892) while trailing in fluency.

Table 6 | Ablation on choice of decoder

Decoder	Gen.PPL(↓)	Div.(↑)
Linear	153.44	0.1238
Transformer Decoder	164.90	0.4842

Overall, these results highlight two takeaways: (i) thanks to the continuous formulation, CoDAR can exploit advanced solvers to enable high-quality fast sampling, and (ii) in the few-step regime (especially 25 steps), CoDAR is already on par with or better than discrete diffusion baselines in fluency, while maintaining comparable diversity.

Table 5 | Ablation on dimension of hidden states.

Hidden Size	Solver	Gen.PPL(↓)	Div.(↑)
768	Ancestral	523.07	0.5764
	DPM-Solver	546.10	0.6212
256	Ancestral	294.42	0.5056
	DPM-Solver	300.01	0.5475
64	Ancestral	164.90	0.4842
	DPM-Solver	147.53	0.488

5.3.2. Hidden State Dimension

We study the effect of the hidden state dimension on the overall generation quality. Specifically, we vary $d \in \{64, 256, 768\}$ while keeping the decoder temperature to 1. As shown in Table 5, increasing the hidden dimension does not translate into better text quality. While larger hidden states generally provide the decoder with higher representational capacity, they hinder the diffusion process. This leads to a degradation in overall text quality: the generative perplexity increases substantially from 164.90 at $d = 64$ to 294.42 at $d = 256$, and further to 523.07 at $d = 768$. Interestingly, even the DPM-Solver, which is designed to handle diffusion more efficiently, struggles with the increased complexity brought by higher hidden dimensions. As the hidden state dimension grows, the DPM-Solver's generative perplexity surpasses that of the ancestral sampler, indicating that the solver cannot mitigate the added difficulty of a larger state space.

5.3.3. Choice of Decoder

We study the effect of decoder architecture by comparing a linear head and a Transformer decoder (Table 6). As noted in Section 3, linear heads perform poorly in token recovery; we examine whether this limitation extends to text generation. Although the linear decoder attains a slightly lower Gen.PPL (153.44 vs. 164.90), it suffers from extremely low diversity (0.1238), indicating severe repetition and mode collapse, which substantially degrades text quality in practice. In contrast, the Transformer decoder achieves much higher diversity (0.4842) while maintaining competitive perplexity, highlighting the importance of contextual modeling for high quality decoding. Overall, linear decoders are inadequate for generation due to their limited expressive capacity.

6. Conclusion

In this work, we argue that the performance gap between continuous and discrete diffusion language models arises primarily from a decoding rounding rather than from limitations of continuous diffusion itself. Through theoretical analysis and controlled token-recovery experiments, we show that rounding is inherently sequence dependent and that pointwise linear heads are provably suboptimal for mapping continuous representations back to tokens. Building on this insight, we propose CoDAR , a two-stage framework that performs diffusion entirely in an embedding space while using a context aware autoregressive Transformer decoder to realize discrete tokens. Experiments on LM1B and OpenWebText demonstrate that CoDAR substantially improves over latent diffusion baselines and becomes competitive with strong discrete DLMs, while exposing a simple decoder temperature mechanism to smoothly trade off fluency and diversity. Together, these results suggest that continuous diffusion and discrete language modeling are complementary rather than competing, and that treating rounding as a contextual problem unlocks much of the unrealized potential of continuous diffusion language models.

References

Lina Berrayana, Ahmed Heakl, Muhammad Abdullah Sohail, Thomas Hofmann, Salman Khan, and Wei Chen. Planner and executor: Collaboration between discrete diffusion and autoregressive models in reasoning, 2025. URL https://arxiv.org/abs/2510.15244.

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

Zhujin Gao, Junliang Guo, Xu Tan, Yongxin Zhu, Fang Zhang, Jiang Bian, and Linli Xu. Empowering diffusion models on the embedding space for text generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4664–4683, 2024.

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

Zhengfu He, Tianxiang Sun, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models. *arXiv preprint arXiv:2211.15029*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Haoqiang Kang, Yizhe Zhang, Nikki Lijing Kuang, Nicklas Majamaki, Navdeep Jaitly, Yi-An Ma, and Lianhui Qin. Ladir: Latent diffusion enhances llms for text reasoning, 2025. URL https://arxiv.org/abs/2510.04573.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Bolin Lai, Xudong Wang, Saketh Rambhatla, James M Rehg, Zsolt Kira, Rohit Girdhar, and Ishan Misra. Toward diffusible high-dimensional latent spaces: A frequency perspective. *arXiv preprint arXiv:2511.22249*, 2025.

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

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. Diffusion-LM improves controllable text generation. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=3s9IrEsjLyk.

Zhenghao Lin, Yeyun Gong, Yelong Shen, Tong Wu, Zhihao Fan, Chen Lin, Nan Duan, and Weizhu Chen. Text generation with diffusion language models: A pre-training approach with continuous paragraph denoise. In *International Conference on Machine Learning*, pages 21051–21064. PMLR, 2023.

Jingyu Liu, Xin Dong, Zhifan Ye, Rishabh Mehta, Yonggan Fu, Vartika Singh, Jan Kautz, Ce Zhang, and Pavlo Molchanov. Tidar: Think in diffusion, talk in autoregression, 2025a. URL https://arxiv.org/abs/2511.08923.

Yangzhou Liu, Yue Cao, Hao Li, Gen Luo, Zhe Chen, Weiyun Wang, Xiaobo Liang, Biqing Qi, Lijun Wu, Changyao Tian, et al. Sequential diffusion language models. *arXiv preprint arXiv:2509.24007*, 2025b.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*, pages 32819–32848. PMLR, 2024.

Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Q Weinberger. Latent diffusion for language generation. *Advances in Neural Information Processing Systems*, 36:56998–57025, 2023.

Justin Lovelace, Varsha Kishore, Yiwei Chen, and Kilian Weinberger. Diffusion guided language modeling. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14936–14952, 2024.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Rabeeh Karimi Mahabadi, Hamish Ivison, Jaesung Tae, James Henderson, Iz Beltagy, Matthew E Peters, and Arman Cohan. Tess: Text-to-text self-conditioned simplex diffusion. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2347–2361, 2024.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8162–8171. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/nichol21a.html.

Patrick Pynadath, Jiaxin Shi, and Ruqi Zhang. Candi: Hybrid discrete-continuous diffusion models. *arXiv preprint arXiv:2510.22510*, 2025.

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

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

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

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin T Chiu, and Volodymyr Kuleshov. The diffusion duality. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=9P9Y8FOSOk.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Alexander Shabalin, Viacheslav Meshchaninov, and Dmitry Vetrov. Smoothie: Smoothing diffusion on token embeddings for text generation. *arXiv preprint arXiv:2505.18853*, 2025.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Robin Strudel, Corentin Tallec, Florent Altché, Yilun Du, Yaroslav Ganin, Arthur Mensch, Will Grathwohl, Nikolay Savinov, Sander Dieleman, Laurent Sifre, et al. Self-conditioned embedding diffusion for text generation. *arXiv preprint arXiv:2211.04236*, 2022.

Jaesung Tae, Hamish Ivison, Sachin Kumar, and Arman Cohan. Tess 2: A large-scale generalist diffusion language model. *arXiv preprint arXiv:2502.13917*, 2025.

Tong Wu, Zhihao Fan, Xiao Liu, Hai-Tao Zheng, Yeyun Gong, Jian Jiao, Juntao Li, Jian Guo, Nan Duan, Weizhu Chen, et al. Ar-diffusion: Auto-regressive diffusion model for text generation. *Advances in Neural Information Processing Systems*, 36:39957–39974, 2023.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15703–15712, 2025.

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

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders, 2025a. URL https://arxiv.org/abs/2510.11690.

Huangjie Zheng, Shansan Gong, Ruixiang Zhang, Tianrong Chen, Jiatao Gu, Mingyuan Zhou, Navdeep Jaitly, and Yizhe Zhang. Continuously augmented discrete diffusion model for categorical generative modeling. *arXiv preprint arXiv:2510.01329*, 2025b.

Cai Zhou, Chenxiao Yang, Yi Hu, Chenyu Wang, Chubin Zhang, Muhan Zhang, Lester Mackey, Tommi Jaakkola, Stephen Bates, and Dinghuai Zhang. Coevolutionary continuous discrete diffusion: Make your diffusion language model a latent reasoner, 2025. URL https://arxiv.org/abs/2510.03206.

A. Proof

Detailed proof of Proposition 1. Let $p(x, y)$ denote the true joint distribution of (X, Y), and let Δ be the probability simplex over $\mathcal{V}^L$ (all length-L token sequences). We assume each candidate decoder $q(\cdot \mid x)$ is a valid conditional distribution in Δ for p-a.e. x.

Proof. We prove the two minimizers and then the lower bound by conditional total correlation.

Step 1: Bayes-optimal decoder over $\mathcal{D}_{\text{seq}}$. Fix any conditional distribution $q(y \mid x)$. The expected NLL risk can be rewritten by conditioning on X:

$$\mathcal{R}(q) = \mathbb{E}_X\Big[\mathbb{E}_{Y|X}\big[-\log q(Y \mid X)\big]\Big] = \mathbb{E}_X\Big[\sum_y p(y \mid X)\big(-\log q(y \mid X)\big)\Big]. \tag{8}$$

For each fixed value x, define the *cross-entropy* between $p(\cdot \mid x)$ and $q(\cdot \mid x)$:

$$H\big(p(\cdot \mid x), q(\cdot \mid x)\big) := -\sum_y p(y \mid x) \log q(y \mid x). \tag{9}$$

A standard identity (cross-entropy decomposition) states that for any two distributions P, Q on the same space,

$$H(P, Q) = H(P) + D_{\text{KL}}(P \| Q), \tag{10}$$

hence, applying it pointwise at each x yields

$$-\sum_y p(y \mid x) \log q(y \mid x) = H\big(p(\cdot \mid x)\big) + D_{\text{KL}}\big(p(\cdot \mid x) \,\|\, q(\cdot \mid x)\big). \tag{11}$$

Taking expectation over X gives

$$\mathcal{R}(q) = \underbrace{\mathbb{E}_X\big[H(p(\cdot \mid X))\big]}_{H(Y|X)} + \mathbb{E}_X\Big[D_{\text{KL}}\big(p(\cdot \mid X) \,\|\, q(\cdot \mid X)\big)\Big]. \tag{12}$$

Since KL divergence is nonnegative and equals 0 iff the two distributions agree almost surely, the second term in equation 12 is minimized (to 0) by choosing

$$q^*(y \mid x) = p(y \mid x) \quad \text{for } p\text{-a.e. } x. \tag{13}$$

Therefore,

$$\min_{q \in \mathcal{D}_{\text{seq}}} \mathcal{R}(q) = H(Y \mid X). \tag{14}$$

(These facts follow from the cross-entropy/KL identity and nonnegativity of KL; see, e.g., standard information theory references. [])

Step 2: Bayes-optimal decoder over $\mathcal{D}_{\text{pw}}$. Now restrict to pointwise-factorized decoders

$$q(y \mid x) = \prod_{i=1}^{L} q_i(y_i \mid x_i), \qquad x = (x_1, \ldots, x_L). \tag{15}$$

Then the log-likelihood separates:

$$-\log q(Y \mid X) = -\sum_{i=1}^{L} \log q_i(Y_i \mid X_i), \tag{16}$$

and by linearity of expectation,

$$\mathcal{R}(q) = \sum_{i=1}^{L} \mathbb{E}\big[-\log q_i(Y_i \mid X_i)\big]. \tag{17}$$

For each position i, the inner expectation depends only on the joint law of (X_i, Y_i). Repeating the same cross-entropy decomposition as in Step 1 but for the conditional distribution $p(y_i \mid x_i)$ gives, for any $q_i(\cdot \mid x_i)$,

$$\mathbb{E}\big[-\log q_i(Y_i \mid X_i)\big] = H(Y_i \mid X_i) + \mathbb{E}_{X_i}\Big[D_{\mathrm{KL}}\big(p(\cdot \mid X_i) \,\|\, q_i(\cdot \mid X_i)\big)\Big], \tag{18}$$

where $p(\cdot \mid X_i)$ abbreviates $p(Y_i = \cdot \mid X_i)$. The KL term is minimized to 0 iff

$$q_i^*(y_i \mid x_i) = p(y_i \mid x_i) \quad \text{for } p\text{-a.e. } x_i. \tag{19}$$

Plugging into equation 17 yields

$$\min_{q \in \mathcal{D}_{\mathrm{pw}}} \mathcal{R}(q) = \sum_{i=1}^{L} H(Y_i \mid X_i). \tag{20}$$

Step 3: The exact optimality gap. Subtracting the two minima obtained above gives the equality:

$$\min_{q \in \mathcal{D}_{\mathrm{pw}}} \mathcal{R}(q) \;-\; \min_{q \in \mathcal{D}_{\mathrm{seq}}} \mathcal{R}(q) = \sum_{i=1}^{L} H(Y_i \mid X_i) \;-\; H(Y \mid X). \tag{21}$$

Step 4: Lower bound by conditional total correlation and nonnegativity. By the definition of conditional total correlation,

$$\mathrm{TC}(Y \mid X) = \mathbb{E}_X\left[D_{\mathrm{KL}}\left(p(Y \mid X) \,\middle\|\, \prod_{i=1}^{L} p(Y_i \mid X)\right)\right] = \sum_{i=1}^{L} H(Y_i \mid X) - H(Y \mid X) \;\geq\; 0, \tag{22}$$

where nonnegativity follows from $\mathrm{KL} \geq 0$. Moreover, since X contains at least as much information as X_i, conditioning reduces entropy:

$$H(Y_i \mid X) \;\leq\; H(Y_i \mid X_i) \qquad \forall i. \tag{23}$$

Therefore,

$$\sum_{i=1}^{L} H(Y_i \mid X_i) - H(Y \mid X) \;\geq\; \sum_{i=1}^{L} H(Y_i \mid X) - H(Y \mid X) = \mathrm{TC}(Y \mid X) \;\geq\; 0, \tag{24}$$

which proves Proposition 1. □