Learning Hierarchical Forward Processes for Discrete Diffusion Language Models

Inhyeok Jeong **Jeongwhan Choi** **Jaehyeon Park*** **Noseong Park†**
KAIST
{inhyeok.jeong, jeongwhan.choi, noseong}@kaist.ac.kr, jaehyeon.park@solverx.ai

Abstract

Discrete diffusion language models (DLMs) enable parallel generation but usually rely on fixed forward processes, such as masking, uniform noise, or precomputed hierarchies. We investigate whether the hierarchical forward process can instead be learned while preserving continuous-time tractability. To this end, we propose **Learnable Hierarchical Diffusion Language Model (LHDLM)**, which generalizes HDLM by replacing its fixed token-to-cluster mapping with a learnable column-stochastic map. This many-to-many hierarchy preserves the block-conditional transition structure of the underlying CTMC and admits a closed-form CT-ELBO, while recovering HDLM in the one-hot limit and masked diffusion in the single-cluster limit. We identify degenerate hierarchy maps that can arise when the same learned map is used in both forward targets and model-induced cluster predictions, and mitigate them with an information-retention regularizer. Empirically, LHDLM reduces generative perplexity by 8.3–16.1% relative to fixed HDLM at 16–128 NFEs on OpenWebText, demonstrating its practical benefit under constrained sampling budgets.

1 Introduction

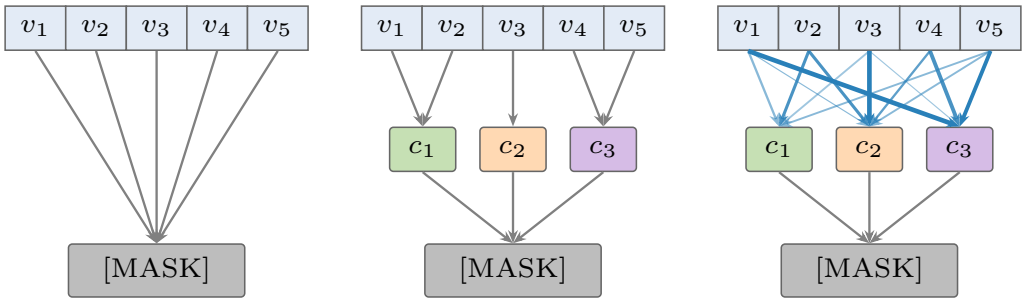
Diffusion language models (DLMs), particularly *discrete* DLMs [13, 18, 7, 20, 28], have recently emerged as a promising alternative to autoregressive language models [1, 21, 27]. By iteratively denoising corrupted text, these models enable parallel generation, bidirectional context modeling, and revision of previously generated tokens. At the core of this paradigm lies the forward diffusion process, which determines how clean text is corrupted and thereby defines the denoising targets learned by the reverse model.

Forward processes are underexplored. Despite its importance, the forward process in discrete DLMs has been much less systematically explored than reverse process parameterization, training objectives, and sampling strategies. D3PM [5] introduced a broad design space for structured discrete corruptions, but most subsequent language modeling work has relied on two computationally convenient fixed processes: uniform and absorbing-state (i.e., masked) diffusion [18, 13, 14, 4]. These choices scale efficiently to language, but they restrict the forward-process design space and leave open whether fixed corruptions are optimal for denoising.

Recent attempts in fixed forward design. Recent work has begun to revisit this design space. GIDD [23] interpolates between corruption strategies such as masking and uniform noise, while HDLM [28] introduces a semantic hierarchy that governs token-to-cluster-to-mask transitions, as illustrated in Figure 1(b). These approaches demonstrate that the structure of the forward process can

*Now at SolverX.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Masked diffusion (b) Hierarchical diffusion (c) Learnable hierarchical diffusion

Figure 1: Forward diffusion design space in discrete DLMs. *(a, b)* Prior methods use *fixed* forward processes: *(a)* All tokens corrupt to a *single* `[MASK]` state [18]; *(b)* Each token maps to exactly *one* cluster via a *fixed* assignment [28]. *(c)* Our LHDLM parameterizes a learnable, column-stochastic hierarchy map that distributes each token across multiple clusters.

substantially affect denoising difficulty and generation quality. However, the forward process remains fixed once training begins — GIDD fixes the mixing distribution and schedule, and HDLM relies on a pre-defined token-to-cluster hierarchy. This raises a natural question: *can we make the forward process data-adaptive while preserving the tractability of structured discrete diffusion?*

Learnable hierarchical forward processes. We propose **Learnable Hierarchical Diffusion Language Model (LHDLM)**, a discrete diffusion framework that preserves the hierarchy-aware CTMC structure of HDLM while replacing the fixed token-to-cluster map with a learnable hierarchy map. Rather than assigning each token to a single pre-defined cluster, LHDLM maps each token to a distribution over intermediate cluster states, allowing many-to-many token-to-cluster relationships as shown in Figure 1(c). This makes the forward process learnable while retaining tractable transition rates and a closed-form CT-ELBO. LHDLM provides a unifying view of existing models: deterministic disjoint assignments correspond to HDLM, while collapsing the intermediate hierarchy to a single absorbing state yields masked diffusion models such as MDLM [18]. Thus, masked and hierarchical diffusion occupy particular corners of a broader structured learnable forward-process family.

A new identifiability challenge. However, making the hierarchy learnable introduces a new identifiability challenge. Because the same hierarchy map determines both the forward cluster target and the model-induced cluster prediction in the ELBO, unconstrained learning can admit uninformative, collapsed mappings[3]. In these regimes, intermediate states lose their semantic role and the objective degenerates toward an MDLM-like token-prediction objective. We characterize a degeneracy in which the learned hierarchy becomes formally valid but uninformative about token identity, and introduce an information-retention regularizer that encourages it to retain token-level information while maintaining balanced use of intermediate states. § 3.5 formalizes this failure mode and the resulting regularization objective.

Contributions. Our contributions are as follows:

- We revisit the forward process in discrete DLMs as a structured learnable component, and propose LHDLM, a CTMC-based extension of HDLM that replaces the fixed token-to-cluster map with a learnable column-stochastic hierarchy map (§ 3).
- We derive a closed-form CT-ELBO for this column-stochastic hierarchy family, showing that hard HDLM is recovered when the hierarchy map is one-hot and that MDLM arises as a degenerate one-cluster special case (§ 3.4).
- We characterize degenerate solutions of the soft hierarchy objective and introduce an information-retention regularizer that encourages non-degenerate, informative hierarchy maps (§ 3.5).

[3]We use "collapsed" to denote mappings whose induced cluster variable carries little or no information about the input token.

- We evaluate LHDLM on OpenWebText, demonstrating 8.3–16.1% lower generative perplexity than fixed HDLM at 16–128 NFEs, while revealing learned soft hierarchy structures that cannot be represented by hard partitions (§§ 4.2–4.3).

2 Background

2.1 Discrete Diffusion

Discrete diffusion models define generative processes over discrete state spaces by corrupting data through a forward stochastic process and learning a reverse denoising process. Let $\mathbf{x} \in \mathcal{V}^L$ denote a clean token sequence and $\mathbf{z}_t \in \mathcal{Z}^L$ the noisy discrete state at diffusion time $t \in [0, 1]$, where the endpoints correspond to the data distribution and a simple prior distribution, respectively. In continuous-time formulations, the forward process is described as a continuous-time Markov chain (CTMC) with a possibly time-inhomogeneous transition rate matrix Q_t. The forward corruption is typically factorized across token positions, so each position evolves independently under the same family of transition dynamics.

The forward transition kernel is a central design choice in discrete diffusion. While discrete diffusion models commonly use absorbing-state or uniform corruption, early work such as D3PM [5] explored a broader space of structured transition kernels, including nearest-neighbor and band-diagonal transitions. This motivates hierarchical and learnable-forward approaches that treat the forward process itself as a modeling component.

2.2 Interpolating Forward Processes in Discrete Diffusion

Many discrete DLMs can be expressed through closed-form forward marginals that interpolate between the clean token and a noise distribution. For a single token position, let $x \in \mathcal{V}$ denote the one-hot clean token and $z_t \in \mathcal{Z}$ the noisy state at time t. MDLM [18] instantiates masked diffusion with an absorbing-state forward marginal,

$$q(z_t \mid x) = \mathrm{Cat}(z_t;\, \alpha_t x + (1 - \alpha_t)m)\,, \tag{1}$$

where m denotes the one-hot vector of the mask token.

GIDD [23] generalizes this view by specifying forward marginals in an interpolation form,

$$q(z_t \mid x) = \mathrm{Cat}(z_t;\, \alpha_t x + (1 - \alpha_t)\pi_t)\,, \tag{2}$$

where $\alpha_t \in [0, 1]$ is a monotonically decreasing noise schedule, typically with $\alpha_0 = 1$ and $\alpha_1 = 0$, and π_t is a possibly time-dependent categorical noise distribution.

This formulation subsumes common discrete diffusion processes. Absorbing-state diffusion is recovered by choosing $\pi_t = m$, while uniform diffusion is recovered by choosing π_t as the uniform distribution over the vocabulary. More generally, different choices of π_t define different forward corruption processes while preserving tractable forward marginals and variational training objectives.

2.3 Hierarchical Forward Process in Discrete Diffusion

HDLM [28] extends discrete diffusion by introducing intermediate vocabulary levels with increasing semantic abstraction. For a single token position, HDLM augments the state space from the word vocabulary $\mathcal{V}$ to

$$\mathcal{Z} = \mathcal{V} \cup \mathcal{C} \cup \{m\}, \tag{3}$$

where $\mathcal{C}$ denotes a cluster-level vocabulary and m is a special absorbing mask state. Each clean token $x \in \mathcal{V}$ is associated with a cluster token $c = \Gamma x \in \mathcal{C}$ through a predefined surjective mapping matrix $\Gamma \in \{0, 1\}^{|\mathcal{C}| \times |\mathcal{V}|}$.

The forward process is a continuous-time Markov chain over this expanded state space, in which each token position independently follows the hierarchy

$$x \to c \to m. \tag{4}$$

Thus, tokens are progressively corrupted from fine-grained word states to coarser semantic states and eventually to the mask state according to a predefined noise schedule. Since the forward process is

applied independently across positions, different tokens in the same sequence may occupy different hierarchy levels at the same diffusion time.

In the reverse process, the denoising model predicts finer-grained tokens from coarser states, yielding a next-semantic-scale prediction process from mask to cluster-level semantics and finally to word-level tokens.

2.4 Entropy and Mutual Information: Information-Theoretic Measures for Learned Hierarchies

We briefly review the information-theoretic quantities used to analyze intermediate representations in our model. For a discrete random variable $\mathcal{C}$ with distribution $p(c)$, the entropy

$$H(\mathcal{C}) = -\sum_{c} p(c) \log p(c) \tag{5}$$

measures the uncertainty, or effective usage, of its states. For two random variables X and C, the mutual information

$$I(\mathcal{C}; X) = H(\mathcal{C}) - H(\mathcal{C} \mid X) \tag{6}$$

measures how much observing $\mathcal{C}$ reveals about X. Equivalently, $I(\mathcal{C}; X) = 0$ if and only if $\mathcal{C}$ is independent of X on the data support.

This decomposition separates two properties of a discrete intermediate representation. The marginal entropy $H(\mathcal{C})$ is high when the available states are broadly used, while the conditional entropy $H(\mathcal{C} \mid X)$ is low when the state is predictive of the input. Thus, a representation can fail in two complementary ways: it may use only a small subset of states, or it may use many states while remaining uninformative about the input.

This perspective is related to the information bottleneck principle [22, 2], where a representation is constrained to pass only limited information about the input. In our setting, any finite intermediate vocabulary already imposes such a capacity constraint, since

$$I(\mathcal{C}; X) \le H(\mathcal{C}) \le \log |\mathcal{C}|. \tag{7}$$

We use these quantities to analyze when an intermediate diffusion state carries useful token-level information and to formulate an information-retention regularizer designed to prevent input-independent collapse, as detailed in §3.5.

3 Learnable Hierarchical Diffusion Language Model

We introduce **Learnable Hierarchical Diffusion Language Model (LHDLM)**, a discrete diffusion framework that makes the forward process learnable while preserving a structured continuous-time Markov chain (CTMC) formulation. The key component is a *learnable hierarchy map*, which replaces the fixed token-to-cluster map of HDLM with a distribution over intermediate semantic states. We first define this module, then describe the induced forward and reverse processes, and finally derive the training objective and regularization.

3.1 Cluster Matrix Γ_ϕ

Let $\mathcal{V}$ denote the token vocabulary, $\mathcal{C}$ the set of intermediate semantic states, and m the absorbing mask state. LHDLM operates on the augmented state space

$$\mathcal{S} = \mathcal{V} \sqcup \mathcal{C} \sqcup \{m\}. \tag{8}$$

For each clean token $x \in \mathcal{V}$, the learnable hierarchy map defines a learnable distribution over intermediate states,

$$\Gamma_\phi(\cdot \mid x) \in \Delta^{|\mathcal{C}|-1}. \tag{9}$$

Equivalently, $\Gamma_\phi \in \mathbb{R}^{|\mathcal{C}| \times |\mathcal{V}|}$ is column-stochastic:

$$\Gamma_\phi(c \mid x) \ge 0, \qquad \sum_{c \in \mathcal{C}} \Gamma_\phi(c \mid x) = 1. \tag{10}$$

Unlike the deterministic and disjoint token-to-cluster assignment used in HDLM, Γ_ϕ allows each token to distribute probability mass across multiple intermediate states.

In implementation, Γ_ϕ can be parameterized by any differentiable module. For example, given a token representation e_x and cluster representations $\{p_c\}_{c\in\mathcal{C}}$, one may use

$$\Gamma_\phi(c \mid x) = \frac{\exp\left(p_c^\top e_x/\tau_\Gamma\right)}{\sum_{c'\in\mathcal{C}} \exp\left(p_{c'}^\top e_x/\tau_\Gamma\right)}, \tag{11}$$

where τ_Γ controls the sharpness of the learned hierarchy. We denote by $\mathbf{c}_\phi(x)$ the corresponding distribution embedded in the augmented state space $\mathcal{S}$, with probability mass placed only on cluster states. In our implementation, we use cluster embedding parameterization with $\tau_\Gamma = 0.1$. The token representation e_x is taken from the model vocabulary embedding, and the cluster representations $\{p_c\}_{c\in\mathcal{C}}$ are learnable prototypes initialized from the K-means centroids used by HDLM. The embeddings are optimized jointly with the denoising model, so Γ_ϕ remains column-stochastic by construction while allowing the hierarchy to deviate from the initial fixed clustering.

3.2 Learnable Forward Process

LHDLM preserves the hierarchy-aware marginal structure of HDLM while replacing its fixed token-to-cluster map with a learnable column-stochastic hierarchy map Γ_ϕ. Let $\mathbf{x}$ denote the one-hot representation of token x embedded in $\mathcal{S}$, and let $\mathbf{m}$ denote the one-hot absorbing mask state. The forward marginal at diffusion time $t \in (0, 1)$ is

$$q_{t,\phi}(z_t \mid x) = \mathrm{Cat}(z_t;\, \alpha_t \mathbf{x} + \beta_{t,c}\mathbf{c}_\phi(x) + \beta_{t,m}\mathbf{m}), \tag{12}$$

where $\alpha_t + \beta_{t,c} + \beta_{t,m} = 1$.

LHDLM is not restricted to a specific schedule. In our experiments, we adopt the noise schedule with $\alpha_t = 1 - t$ from HDLM [28], which yields

$$\beta_{t,c} = -(1-t)\ln(1-t), \quad \beta_{t,m} = t + (1-t)\ln(1-t). \tag{13}$$

This schedule controls how the probability mass moves from token states to intermediate cluster states and eventually to the mask state. Because Γ_ϕ is column-stochastic, the forward marginal in Eq. (12) induces a time-inhomogeneous CTMC with the same block structure as HDLM. Thus, LHDLM preserves the Markovian hierarchy $x \to c \to m$ while replacing the fixed token-to-cluster transition with learnable rates.

3.3 Reverse Process

The reverse process follows the Bayesian denoising parameterization used in CTMC-based discrete diffusion models. Given a noisy state z_t, the denoising network predicts a distribution over clean tokens:

$$\mathbf{x}_\theta(z_t, t) \in \Delta^{|\mathcal{V}|-1}. \tag{14}$$

The model-induced distribution over intermediate semantic states is obtained by applying the same learnable hierarchy map:

$$\Gamma_\phi \mathbf{x}_\theta(z_t, t). \tag{15}$$

Therefore, cluster-level predictions are not produced by an auxiliary classifier head; they are derived self-consistently from token predictions through the learned hierarchy.

For any distribution $\mathbf{x}_\theta$ over clean tokens, define the model-induced forward marginal

$$q_{t,\phi}(\cdot \mid \mathbf{x}_\theta) := \sum_{x\in\mathcal{V}} \mathbf{x}_\theta[x]\, q_{t,\phi}(\cdot \mid x). \tag{16}$$

The reverse transition rates are then given by the posterior induced by the forward CTMC:

$$\widehat{Q}_{t,\theta,\phi}(z_t, z_s) = Q_{t,\phi}(z_s, z_t)\frac{q_{t,\phi}(z_s \mid \mathbf{x}_\theta)}{q_{t,\phi}(z_t \mid \mathbf{x}_\theta)}, \qquad z_s \neq z_t, \tag{17}$$

with diagonal entries chosen so that the reverse rates sum to zero. This construction keeps the reverse process tied to the learnable forward process.

3.4 Training Objective

We train LHDLM by minimizing a regularized training objective consisting of the negative continuous-time ELBO and an information-retention regularizer on the learned hierarchy. For a clean token x, sampled noisy state z_t, and model prediction $\mathbf{x}_\theta$, we can derive CT-ELBO:

$$\begin{aligned}\log p(x) \geq &- \mathbb{E}_{t,z_t\sim q_{t,\phi}(\cdot|x)}\Big[\delta_{z_t,c}\, w_{t,c}\, \mathrm{CE}\big(x, \tilde{x}_{\theta,\phi}(z_t)\big) + \delta_{z_t,m}\, w_{t,m}\, \mathrm{CE}\big(\Gamma_\phi x, \Gamma_\phi x_\theta\big)\Big] \\ &+ \mathbb{E}_{t,z_t\sim q_{t,\phi}(\cdot|x)}\big[\delta_{z_t,m}\, w_{t,m} H(\Gamma_\phi x)\big],\end{aligned} \tag{18}$$

where

$$\tilde{x}_{\theta,\phi}(z_t) = \frac{x_\theta \odot (\Gamma_\phi^\top z_t^{(c)})}{z_t^{(c)\top}(\Gamma_\phi x_\theta)}, \qquad w_{t,c} = \frac{-\alpha_t'}{\beta_{t,c}}, \qquad w_{t,m} = \frac{\beta_{t,m}'}{\beta_{t,m}} \tag{19}$$

The full objective is the regularized negative expectation of Eq. (18) over data, diffusion time, and forward samples:

$$\begin{aligned}\mathcal{L}(\theta,\phi) = &\mathbb{E}_{x,t,z_t\sim q_{t,\phi}(\cdot|x)}\left[\delta_{z_t,c}\, w_{t,c}\, \mathrm{CE}\big(x, \tilde{x}_{\theta,\phi}(z_t)\big) + \delta_{z_t,m}\, w_{t,m}\, \mathrm{CE}\big(\Gamma_\phi x, \Gamma_\phi x_\theta\big)\right] \\ &- \mathbb{E}_{x,t,z_t\sim q_{t,\phi}(\cdot|x)}\big[\delta_{z_t,m}\, w_{t,m} H(\Gamma_\phi x)\big] + \mathcal{R}_{\mathrm{reg}}(\phi)\end{aligned} \tag{20}$$

For sequences, we aggregate the denoising loss over token positions, including padding tokens during training, while excluding them from ELBO-based evaluation.

When Γ_ϕ is deterministic and one-hot, the cluster-state term reduces to the HDLM within-cluster token prediction loss, and the mask-state KL reduces to the HDLM cluster-level prediction loss. Consequently, Eq. (18) recovers the HDLM CT-ELBO. Furthermore, when $|C| = 1$ and the sole cluster state is identified with the mask state, the forward process reduces to masked diffusion with $\alpha_t = 1 - t$, corresponding to the log-linear noise schedule used in MDLM. See Appendix B for detailed derivations and reductions to existing diffusion objectives.

Optimizing the learnable forward process. Unlike hard HDLM, the forward distribution in LHDLM depends on the learnable hierarchy map Γ_ϕ. Consequently, the training objective depends on ϕ in two ways: directly through the closed-form CT-ELBO terms, and indirectly through the sampled forward state $z_t \sim q_{t,\phi}(z_t \mid x)$. The former can be optimized by standard backpropagation, whereas the latter involves gradients through discrete samples. We therefore estimate the sampling-path gradient for ϕ with a score-function estimator:

$$\nabla_\phi \mathbb{E}_{z_t\sim q_{t,\phi}(\cdot|x)}[\ell(z_t;\theta,\phi)] = \mathbb{E}_{z_t\sim q_{t,\phi}}\left[\nabla_\phi \ell(z_t;\theta,\phi) + \ell(z_t;\theta,\phi)\nabla_\phi \log q_{t,\phi}(z_t \mid x)\right]. \tag{21}$$

For the sampling-path gradient, only the cluster-state sampling branch depends on Γ_ϕ. We use a REINFORCE estimator with a detached baseline for this term.

3.5 Degeneracy and Information-Retention Regularization

Making the hierarchy learnable introduces a self-referential identifiability problem. The same mapping Γ_ϕ defines both the forward cluster target $\Gamma_\phi(\cdot \mid x)$ and the model-induced cluster prediction $\Gamma_\phi \mathbf{x}_\theta$. As a result, the unregularized objective admits input-independent collapsed configurations.

To see this, suppose there exists a distribution $u \in \Delta^{|\mathcal{C}|-1}$ such that

$$\Gamma_\phi(\cdot \mid x) = u \qquad \text{for all } x \in \mathcal{V}. \tag{22}$$

Then the intermediate state carries no information about the clean token. Moreover, for any model prediction $\mathbf{x}_\theta$, column-stochasticity gives $\Gamma_\phi \mathbf{x}_\theta = u$. Therefore, the mask-state KL becomes $\mathrm{KL}(u\|u) = 0$. For a cluster-state input c_k, the cluster-state log-ratio also collapses to

$$\log \frac{x_\theta[x]\,\Gamma_\phi(k \mid x)}{(\Gamma_\phi \mathbf{x}_\theta)_k} = \log \frac{x_\theta[x]\,u_k}{u_k} = \log x_\theta[x]. \tag{23}$$

Thus, the hierarchy no longer contributes semantic information, and the objective degenerates to an MDLM-like token-prediction objective. Rank-one collapse and the globally uniform map $\Gamma_\phi(\cdot \mid x) = \mathbf{1}/|\mathcal{C}|$ for all x are two extreme cases of this input-independent family.

To discourage convergence to this collapsed manifold, we add an explicit information-retention regularizer. Let X denote a clean token drawn from the empirical token distribution, and let C be sampled from the learned hierarchy,

$$C \sim \Gamma_\phi(\cdot \mid X). \tag{24}$$

The marginal cluster usage is

$$\bar{p}_\phi(c) = \mathbb{E}_X[\Gamma_\phi(c \mid X)]. \tag{25}$$

Then

$$I_\phi(C; X) = H_\phi(C) - H_\phi(C \mid X) = H(\bar{p}_\phi) - \mathbb{E}_X\left[H(\Gamma_\phi(\cdot \mid X))\right]. \tag{26}$$

The finite intermediate vocabulary already imposes a capacity bottleneck. Motivated by this information-bottleneck perspective, we formulate the regularizer as

$$\mathcal{R}_{\text{reg}}(\phi) = \lambda_{\text{bal}}\left(\log|\mathcal{C}| - H(\bar{p}_\phi)\right) + \lambda_{\text{cond}}\mathbb{E}_X\left[H(\Gamma_\phi(\cdot \mid X))\right]. \tag{27}$$

The marginal entropy term discourages mass concentration by encouraging broad use of the intermediate vocabulary, while the conditional entropy term discourages globally diffuse input-independent assignments by favoring informative token-to-cluster distributions. Importantly, this term is a soft pressure rather than a hard constraint: selective high-entropy assignments may still be preferred when they improve the CT-ELBO.

When $\lambda_{\text{bal}} = \lambda_{\text{cond}} = \lambda$, this becomes

$$\mathcal{R}_{\text{reg}}(\phi) = \lambda\left(\log|\mathcal{C}| - I_\phi(C; X)\right), \tag{28}$$

so minimizing the regularizer maximizes the information retained by the learned hierarchy up to the fixed cluster budget.

4 Experiments

In this section, we evaluate LHDLM on language modeling benchmarks through validation perplexity and generative perplexity evaluated across sampling budgets. We further examine the role of information-retention regularization and the token-to-cluster assignment structures learned by the soft hierarchy.

4.1 Experimental Setup

Datasets. We train our models primarily on OpenWebText (OWT) [9] and report perplexity on the validation set. We also train models on LM1B [8] and report test perplexity on its standard held-out benchmark in Appendix E.

Table 1: Validation and generative perplexity on OpenWebText. † denotes results reported in the corresponding works; [1] denotes results reported in [28]. The suffix 64 in HDLM-64 and LHDLM-64 indicates the number of clusters ($|C| = 64$).

Model	Train. toks.	Valid. PPL (↓)	Gen. PPL (↓)
GPT2 [16]†	unk.	23.40	-
Llama110M (retrain.)†	262B	16.11	-
SEDD [13]†	262B	≤24.10	-
MDLM [18][1]	131B	≤27.39	163.7
GIDD+ [23][1]	131B	≤25.82	170.2
HDLM-64 [28][1]	131B	≤23.36	144.2
HDLM-64 (reimpl.)	131B	≤26.05	**136.2**
LHDLM-64 ($\lambda = 0.1$)	131B	≤26.88	155.61
LHDLM-64 ($\lambda = 0.1$)	262B	≤26.09	-
LHDLM-64 ($\lambda = 0.5$)	131B	≤27.39	153.26
LHDLM-64 ($\lambda = 1.0$)	131B	≤27.72	226.56

Setting. In our experiment, we follow the architecture [15] and scale settings used in prior work [13, 18, 23, 28]. Our primary experiments use $|C| = 64$, with additional results for $|C| = 128$ provided in Appendix E. We adopt pretrained token embeddings following [28]. More detailed settings are described in Appendix D.

Evaluation metrics. We report validation perplexity upper bounds and generative perplexity. Following prior work, we measure generative perplexity using GPT-2 Large [16] at six sampling budgets (16, 32, 64, 128, 256, and 512 NFEs). For the HDLM-64 and LHDLM-64 ($\lambda = 0.1$) comparison, we evaluate three matched training seeds.

4.2 Main Results

Table 1 shows that LHDLM-64 ($\lambda = 0.1$) achieves broadly comparable validation and generative perplexity to fixed HDLM on OWT. Under constrained sampling budgets, however, LHDLM consistently outperforms HDLM at 16–128 NFEs across three matched training seeds, as shown in Figure 2. This advantage diminishes as the sampling budget increases and does not persist at the 512-NFE endpoint. These results highlight the practical benefit of learning the hierarchy for low-NFE generation, rather than a uniform improvement across all sampling budgets.

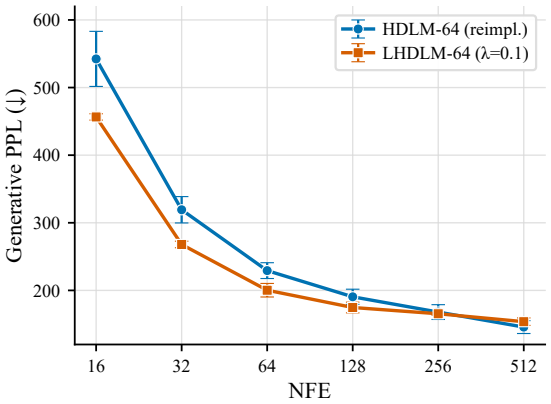


Figure 2: Generative perplexity on OWT across NFEs. Curves and error bars indicate the mean and standard deviation across three matched training seeds. LHDLM-64($\lambda = 0.1$) consistently achieves lower generative perplexity at 16–128 NFEs.

4.3 Regularization and Learned Hierarchy Analyses

Table 2 presents the regularizer ablation results, with each configuration trained on 13.1B tokens. The results reveal distinct failure modes when the two components of the information-retention regularizer are applied separately. Using only the marginal-entropy term produces nearly uniform cluster usage but also nearly uniform token assignments, resulting in negligible mutual information between tokens and clusters. In contrast, using only the conditional-entropy term produces sharp assignments but concentrates most of the probability mass on a single cluster. Combining both terms avoids these degenerate solutions, maintaining broad cluster utilization and substantial mutual information between tokens and clusters.

A near-binary soft hierarchy. At $\lambda = 0.1$, Γ_ϕ converges to a near-binary configuration over the vocabulary. Approximately 59.4% of tokens receive sharp single-cluster assignments, while 40.5% receive essentially uniform assignments over all 64 clusters. Only about 0.1% of tokens lie in between. Thus, the learned hierarchy does not simply produce diffuse assignments everywhere; instead, it distinguishes tokens with strong cluster preferences from those that remain largely cluster-agnostic.

Function/content split. The uniformly assigned tokens exhibit a recognizable linguistic pattern. Table 3 shows that these tokens are predominantly function-like or context-general, including sentence-initial capitalized words, punctuation, digits, single-character tokens, and common short subwords. In contrast, sharply assigned tokens are more often associated with topic-bearing content words.

Table 2: Ablation study of the marginal and conditional entropy regularizers. Removing either component leads to a distinct degenerate assignment pattern.

Setting	λ_{bal}	λ_{cond}	$H(C)/\log\|C\|$	$H(C \mid X)/\log\|C\|$	$I(C;X)/\log\|C\|$	Eff. clusters	Valid. PPL (↓)
Both	0.1	0.1	0.962	0.366	**0.5963**	**54.58**	33.61
Marginal only	0.1	0	0.9999	0.9999	0.0000	63.98	32.86
Conditional only	0	0.1	0.070	0.0014	0.068	1.34	37.23

This suggests that the learned hierarchy captures a distinction between tokens with specific semantic cluster preferences and those that are comparatively cluster-agnostic.

Soft assignments versus hard partitions. A deterministic hierarchy assigns every token to exactly one cluster and therefore cannot directly represent diffuse token-to-cluster memberships. To examine the functional role of these soft assignments, we project the learned Γ_ϕ to its hard-argmax partition using $\arg\max_k \Gamma_\phi[k, v]$ and recompute the ELBO. At $\lambda = 0.1$, validation PPL increases from 26.88 to 40.19 under this projection, indicating that the learned soft assignments are important to the evaluated model. At $\lambda = 0.5$, where the hierarchy is already nearly one-hot, the same projection produces little degradation.

Interpretation. The cluster-state term of the CT-ELBO favors assignments that provide useful information for token prediction. For semantically specific tokens, sharp cluster assignments may support more informative intermediate states, whereas broadly distributed tokens may benefit from remaining cluster-agnostic. The conditional-entropy term of the information-retention regularizer penalizes diffuse per-token assignments, and the sharper hierarchy observed at $\lambda = 0.5$ is consistent with stronger regularization pressure.

Table 3: Linguistic categorization of the 200 most uniformly assigned tokens in the LHDLM-64 ($\lambda = 0.1$) trained on OWT. Proportions are approximate.

Category	Share	Examples
Sentence-initial capitalized	$\sim 25\%$	The, This, He, She, It, If, When, As, For, In
Punctuation and symbols	$\sim 25\%$	:, _, -,), (, [, ", &, . . ., /
Digits and number-related	$\sim 10\%$	0–9, 10, 11, 12, 15, 23, 25, 30
Single-character tokens	$\sim 15\%$	A, M, P, S, T, a, b, c, d, e, g
Common short subwords	$\sim 15\%$	an, er, es, ic, in, is, it, or, re, st, th
High-frequency content words	$\sim 10\%$	night, summer, season, city, players, September, British

5 Related Work

Discrete diffusion and conventional forward processes. D3PM [5] established a general framework for discrete diffusion using Markov transition operators Q_t. Despite this generality, most discrete diffusion language models adopt restricted forward processes, notably absorbing-state (masked) diffusion or uniform diffusion, for analytical and practical convenience [13, 18, 20]. These processes typically use predefined corruption structures, with much of the subsequent research focusing on reverse-process modeling and sampling.

Forward-process design in discrete diffusion. Several works revisit the forward process of discrete DLMs by modifying transition dynamics or introducing intermediate states. GenMD4 [20] learns token-dependent masking rates, while GIDD [23] explores hybrid masking and uniform noise through predefined mixing schedules. HDLM [28] and TDLM [26] enrich intermediate states with fixed semantic hierarchies from offline clustering or vocabulary trees, while PRIME [7] introduces partially masked token states for finer-grained denoising. These approaches demonstrate the importance of forward-process design, but do not jointly learn the mapping from word tokens to intermediate semantic states. In contrast, LHDLM learns the token-to-intermediate transition mapping jointly with the denoising model. Its column-stochastic Γ_ϕ controls the distribution of transition rates across semantic cluster states, allowing the effective connectivity of the forward process to adapt while preserving the hierarchical CTMC structure.

Learning the forward diffusion process. FLDD [6] investigates end-to-end learning of the discrete forward process to better align the forward-induced reverse targets with a factorized generative model, particularly for few-step generation. It adopts a flexible non-Markovian formulation with learnable forward marginals and posteriors. LHDLM takes a complementary, structured approach: it learns the word-to-cluster transition mapping within the Markovian, block-structured hierarchy of HDLM. This constrained design preserves valid CTMC transition rates and a closed-form hierarchy-aware CT-ELBO, while enabling direct comparisons with fixed HDLM.

Relation to reverse-process and sampling improvements. A complementary line of work improves denoising or sampling dynamics through remasking [25], schedule-conditioned diffusion [3], and inference-time scaling [19]. Soft-Masked Diffusion Language Models [11] instead incorporate previous denoising predictions into dynamically blended mask embeddings. These approaches enhance how intermediate information is utilized during generation, whereas LHDLM learns the semantic transition structure of the forward process itself.

6 Conclusion

In this work, we introduce the Learnable Hierarchical Diffusion Language Model (LHDLM), a discrete diffusion framework that learns the column-stochastic token-to-cluster map within a hierarchy-aware CTMC. LHDLM extends the fixed hierarchical forward process into a learnable component while preserving tractable transition dynamics and a closed-form CT-ELBO. We identify degenerate solutions in which the learned hierarchy becomes uninformative about token identity and introduce an information-retention regularizer to discourage these solutions. Component-wise ablations demonstrate that its marginal- and conditional-entropy terms address complementary collapse modes.

Empirically, LHDLM achieves lower generative perplexity than fixed HDLM under constrained sampling budgets, although this advantage does not persist at the highest evaluated budget. Beyond generation performance, the learned soft hierarchy exhibits distinct assignment structures, separating sharply assigned content tokens from cluster-agnostic function-like tokens. The substantial degradation observed when projecting the learned soft map onto a hard partition further highlights the functional importance of soft assignments.

Our findings demonstrate the potential of learning hierarchical forward processes within a tractable discrete diffusion framework, while highlighting the importance of regularization and the dependence of generation performance on sampling budgets. Future work may explore improved regularization and inductive biases, larger model scales, multiple hierarchy levels, and alternative parameterizations of the hierarchy map.

Acknowledgments and Disclosure of Funding

This work was partly supported by the Institute for Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korean government (MSIT) (No. RS-2025-25442149, LG AI STAR Talent Development Program for Leading Large-Scale Generative AI Models in the Physical AI Domain, 1%; No. RS-2024-00457882, AI Research Hub Project, 1%), National Research Foundation (NRF) of Korea grant funded by the Ministry of Science and ICT (MSIT) of Korea (No. RS-2022-NR068758, 1%) and the KICT Strategic Research Program (SRP) (project no. 20260437-001, Development of a Construction Site-Specific Vertical AI Model and Sustainable Platform, 97%) funded by the Ministry of Science and ICT. Jeongwhan Choi was additionally supported by the KAIST Jang Young Sil Fellow Program.

References

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

[2] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.

[3] Alan N. Amin, Nate Gruver, and Andrew Gordon Wilson. Why masking diffusion works: Condition on the jump schedule for improved discrete diffusion, 2025.

[4] Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[5] 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.

[6] Grigory Bartosh, Teodora Pandeva, Sushrut Karmalkar, and Javier Zazo. Forward-learned discrete diffusion: Learning how to noise to denoise faster. In *The Fourteenth International Conference on Learning Representations*, 2026.

[7] Chen-Hao Chao, Wei-Fang Sun, Hanwen Liang, Chun-Yi Lee, and Rahul G Krishnan. Beyond masked and unmasked: Discrete diffusion models via partial masking. *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[8] 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.

[9] Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`, 2019.

[10] Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuan-Jing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4521–4534, 2023.

[11] Michael Hersche, Samuel Moor-Smith, Thomas Hofmann, and Abbas Rahimi. Soft-masked diffusion language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

[12] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. Diffusion-lm improves controllable text generation. *ArXiv*, abs/2205.14217, 2022.

[13] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, pages 32819–32848, 2024.

[14] 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.

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

[16] 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.

[17] Litu Rout, Constantine Caramanis, and Sanjay Shakkottai. Anchored Diffusion Language Model. In *The Thirty-Ninth Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[18] 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.

[19] Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dallatorre, Bernardo P. de Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models, 2025.

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

[21] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[22] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *ArXiv*, physics/0004057, 2000.

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

[24] Alex Wang and Kyunghyun Cho. BERT has a mouth, and it must speak: BERT as a Markov random field language model. In Antoine Bosselut, Asli Celikyilmaz, Marjan Ghazvininejad, Srinivasan Iyer, Urvashi Khandelwal, Hannah Rashkin, and Thomas Wolf, editors, *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[25] Guanghan Wang, Yair Schiff, Subham Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. *arXiv preprint arXiv:2503.00307*, 2025.

[26] Zihao Wu, Haoming Yang, Juncheng Dong, and Vahid Tarokh. Rethinking token prediction: Tree-structured diffusion language model, 2026.

[27] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[28] Cai Zhou, Chenyu Wang, Dinghuai Zhang, Shangyuan Tong, Yifei Wang, Stephen Bates, and Tommi Jaakkola. Next semantic scale prediction via hierarchical diffusion language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 41496–41531. Curran Associates, Inc., 2025.

A Detailed proof of CT-ELBO with column-stochastic hierarchy map Γ_ϕ

Building on the hierarchical CTMC formulation and CT-ELBO derivation of [28], we derive the LHDLM objective by relaxing their fixed, deterministic token-to-cluster mapping to a learnable, column-stochastic map Γ_ϕ. Throughout this appendix, the derivation is pointwise in ϕ: for any fixed column-stochastic Γ_ϕ, the resulting forward process defines a valid hierarchy-aware CTMC.

HDLM with a hard hierarchy map. HDLM defines an augmented state space

$$\mathcal{S} = \mathcal{V} \sqcup \mathcal{C} \sqcup \{m\}, \tag{29}$$

where $\mathcal{V}$ is the word vocabulary, $\mathcal{C}$ is the cluster vocabulary, and m is the absorbing mask state. In standard HDLM, each word token $x \in \mathcal{V}$ is mapped to exactly one cluster by a deterministic column-one-hot matrix

$$\Gamma \in \{0,1\}^{|\mathcal{C}|\times|\mathcal{V}|}, \qquad \Gamma x \in \mathcal{C}. \tag{30}$$

The one-intermediate-level HDLM forward marginal is

$$q_t(z_t \mid x) = \mathrm{Cat}(z_t;\, \alpha_t x + \beta_{t,c}\Gamma x + \beta_{t,m} m)\,, \qquad \alpha_t + \beta_{t,c} + \beta_{t,m} = 1. \tag{31}$$

Under the hierarchy-aware CTMC and the Bayesian reverse parameterization used in HDLM, the CT-ELBO can be written as

$$\begin{aligned}\log p(x) \geq -\mathbb{E}_{t,z_t\sim q_t(\cdot|x)}\Bigg[&\delta_{z_t,c}\, w_{t,c}\, \mathrm{CE}\Bigg(x,\, \frac{x_\theta \odot (\Gamma^\top \Gamma x)}{x_\theta^\top \Gamma^\top \Gamma x}\Bigg) \\ &+ \delta_{z_t,m}\, w_{t,m}\, \mathrm{CE}(\Gamma x, \Gamma x_\theta)\Bigg],\end{aligned} \tag{32}$$

where

$$w_{t,c} = \frac{-\alpha_t'}{\beta_{t,c}}, \qquad w_{t,m} = \frac{\beta_{t,m}'}{\beta_{t,m}}. \tag{33}$$

The first term applies when the observed noisy state is a cluster token and restricts word prediction to the corresponding hard cluster fiber. The second term applies when the observed state is the mask token and asks the model to predict the correct hard cluster.

Relaxing the hard map. LHDLM replaces the deterministic map Γ with a learnable column-stochastic hierarchy map

$$\Gamma_\phi \in \mathbb{R}^{|\mathcal{C}|\times|\mathcal{V}|}, \qquad \Gamma_\phi(c \mid x) \geq 0, \qquad \sum_{c\in\mathcal{C}} \Gamma_\phi(c \mid x) = 1. \tag{34}$$

For a clean token x, $\Gamma_\phi x$ is now a distribution over cluster states rather than a single cluster token. The generalized forward marginal is

$$q_{t,\phi}(z_t \mid x) = \mathrm{Cat}(z_t;\, \alpha_t x + \beta_{t,c}\Gamma_\phi x + \beta_{t,m} m)\,. \tag{35}$$

Column-stochasticity is exactly the condition needed for the word-to-cluster transition rates to sum correctly, so the same block-structured CTMC construction as HDLM remains valid after replacing Γ by Γ_ϕ.

For compactness, define

$$\gamma_\phi(x) := \Gamma_\phi x, \qquad \mu_{\theta,\phi}(z_t) := \Gamma_\phi x_\theta(z_t, t). \tag{36}$$

Here $\gamma_\phi(x)$ is the forward cluster target induced by the clean token, while $\mu_{\theta,\phi}$ is the model-induced cluster distribution obtained by pushing the word prediction x_θ through the same hierarchy map.

CT-ELBO integrand. Following the HDLM CTMC derivation, the nonzero forward transitions are from word states to cluster states and from cluster states to the mask state. We evaluate the CT-ELBO integrand by cases.

Case 1: $z_t = x$. There is no non-diagonal transition into a word state. As in HDLM, this case contributes only a diagonal term, and the resulting bias cancels in expectation with the corresponding cluster and mask diagonal terms. Thus it does not contribute to the final denoising loss.

Case 2: $z_t = c_k$**.** Suppose the observed cluster state is c_k. The only non-diagonal transition into c_k from the clean-data path is from the word state x. Using Eq. (35),

$$q_{t,\phi}(c_k \mid x) = \beta_{t,c}\Gamma_\phi(k \mid x), \qquad q_{t,\phi}(c_k \mid x_\theta) = \beta_{t,c}(\Gamma_\phi x_\theta)_k. \tag{37}$$

The cluster-state log ratio therefore becomes

$$\log \frac{q_{t,\phi}(x \mid x_\theta)\, q_{t,\phi}(c_k \mid x)}{q_{t,\phi}(c_k \mid x_\theta)\, q_{t,\phi}(x \mid x)} = \log \frac{x_\theta[x]\, \Gamma_\phi(k \mid x)}{(\Gamma_\phi x_\theta)_k}. \tag{38}$$

The transition-rate prefactor is the same as in HDLM:

$$w_{t,c} = \frac{-\alpha_t'}{\beta_{t,c}}. \tag{39}$$

Unlike hard HDLM, however, the observed cluster state c_k is not necessarily the unique cluster of x. Therefore the cluster-state term must condition on the actually observed cluster token $z_t^{(c)}$. Let $z_t^{(c)} \in \{0,1\}^{|\mathcal{C}|}$ denote the cluster part of z_t. Then Eq. (38) can be written as

$$\log \frac{x_\theta[x]\, x^\top \Gamma_\phi^\top z_t^{(c)}}{z_t^{(c)\top} \Gamma_\phi x_\theta}. \tag{40}$$

Equivalently, define the cluster-conditioned word prediction

$$\tilde{x}_{\theta,\phi}(z_t) = \frac{x_\theta \odot (\Gamma_\phi^\top z_t^{(c)})}{z_t^{(c)\top}(\Gamma_\phi x_\theta)}. \tag{41}$$

Then the cluster-state contribution is

$$-w_{t,c}\, \mathrm{CE}\big(x, \tilde{x}_{\theta,\phi}(z_t)\big). \tag{42}$$

Case 3: $z_t = m$**.** When the observed state is the mask token, all cluster states can transition into m. The corresponding log term sums over the soft forward cluster distribution:

$$\begin{aligned} &\sum_{k=1}^{|\mathcal{C}|} \Gamma_\phi(k \mid x) \log \frac{(\Gamma_\phi x_\theta)_k}{\Gamma_\phi(k \mid x)} \\ &\qquad = -\mathrm{KL}\big(\Gamma_\phi x \,\|\, \Gamma_\phi x_\theta\big). \end{aligned} \tag{43}$$

Thus the mask-state contribution is naturally a negative KL term:

$$-w_{t,m}\, \mathrm{KL}\big(\Gamma_\phi x \,\|\, \Gamma_\phi x_\theta\big), \qquad w_{t,m} = \frac{\beta_{t,m}'}{\beta_{t,m}}. \tag{44}$$

Writing this KL in cross-entropy form gives

$$-\mathrm{KL}\big(\Gamma_\phi x \,\|\, \Gamma_\phi x_\theta\big) = -\mathrm{CE}\big(\Gamma_\phi x, \Gamma_\phi x_\theta\big) + H(\Gamma_\phi x). \tag{45}$$

This is the source of the additional entropy term that does not appear in hard HDLM, where Γx is one-hot and therefore has zero entropy.

Soft CT-ELBO. Combining the cluster-state and mask-state cases, and using the same zero-mean diagonal cancellation as in HDLM, we obtain

$$\begin{aligned} \log p(x) \geq \mathbb{E}_{t, z_t \sim q_{t,\phi}(\cdot|x)} \Bigg[&\delta_{z_t,c}\, w_{t,c} \log \frac{x_\theta[x]\, x^\top \Gamma_\phi^\top z_t^{(c)}}{z_t^{(c)\top} \Gamma_\phi x_\theta} \\ &- \delta_{z_t,m}\, w_{t,m}\, \mathrm{KL}\big(\Gamma_\phi x \,\|\, \Gamma_\phi x_\theta\big) \Bigg]. \end{aligned} \tag{46}$$

Equivalently, using Eq. (41) and Eq. (45), this becomes

$$\begin{aligned} \log p(x) \geq &- \mathbb{E}_{t, z_t \sim q_{t,\phi}(\cdot|x)} \Big[\delta_{z_t,c}\, w_{t,c}\, \mathrm{CE}\big(x, \tilde{x}_{\theta,\phi}(z_t)\big) \\ &\qquad + \delta_{z_t,m}\, w_{t,m}\, \mathrm{CE}\big(\Gamma_\phi x, \Gamma_\phi x_\theta\big) \Big] \\ &+ \mathbb{E}_{t, z_t \sim q_{t,\phi}(\cdot|x)} \big[\delta_{z_t,m}\, w_{t,m} H(\Gamma_\phi x) \big], \end{aligned} \tag{47}$$

where

$$\tilde{x}_{\theta,\phi}(z_t) = \frac{x_\theta \odot (\Gamma_\phi^\top z_t^{(c)})}{z_t^{(c)\top}(\Gamma_\phi x_\theta)}, \qquad w_{t,c} = \frac{-\alpha_t'}{\beta_{t,c}}, \qquad w_{t,m} = \frac{\beta_{t,m}'}{\beta_{t,m}}. \tag{48}$$

Reduction to HDLM. If each column of Γ_ϕ is one-hot, then $\Gamma_\phi x$ is a deterministic cluster token. On the event $z_t \in \mathcal{C}$, the only possible cluster state is $z_t^{(c)} = \Gamma_\phi x$. Therefore,

$$\Gamma_\phi^\top z_t^{(c)} = \Gamma_\phi^\top \Gamma_\phi x, \tag{49}$$

and Eq. (41) reduces to the HDLM within-cluster word prediction distribution

$$\frac{x_\theta \odot (\Gamma_\phi^\top \Gamma_\phi x)}{x_\theta^\top \Gamma_\phi^\top \Gamma_\phi x}. \tag{50}$$

Moreover, $H(\Gamma_\phi x) = 0$ because $\Gamma_\phi x$ is one-hot. Thus Eq. (47) reduces exactly to the HDLM CT-ELBO in Eq. (32).

B LHDLM preserves CTMC with block-conditional transition

The forward marginals in Eq. (12) admit a valid time-inhomogeneous CTMC generator because Γ_ϕ is column-stochastic. Using the convention that $Q_{t,\phi}(a, b)$ denotes the transition rate from state a to state b, the nonzero transition rates are

$$Q_{t,\phi}(x, c) = -\frac{\alpha_t'}{\alpha_t}\Gamma_\phi(c \mid x), \qquad x \in \mathcal{V},\ c \in \mathcal{C}, \tag{51}$$

$$Q_{t,\phi}(c, m) = -\frac{\alpha_t' + \beta_{t,c}'}{\beta_{t,c}}, \qquad c \in \mathcal{C}, \tag{52}$$

with diagonal entries chosen so that each row sums to zero. Thus, LHDLM preserves the block-structured CTMC of HDLM while making the token-to-cluster transition rates learnable.

Reduction to Existing Diffusion Objectives The proposed formulation contains hard HDLM as a limiting case. When Γ_ϕ is constrained to be a deterministic column-one-hot assignment, $\Gamma_\phi x$ becomes the hard cluster state of HDLM, and the soft CT-ELBO reduces to the HDLM CT-ELBO. In the degenerate hierarchy where the cluster vocabulary has a single state, which is identified with the mask state, the intermediate hierarchy carries no nontrivial semantic information, recovering masked diffusion models such as MDLM. Thus, LHDLM can be viewed as a learnable generalization of hierarchical diffusion, with masked diffusion corresponding to the collapsed single-cluster hierarchy.

C Degeneracy and Identifiability

Making the hierarchy learnable enlarges HDLM from deterministic column-one-hot maps to the full column-stochastic family. This flexibility introduces an identifiability problem: the same Γ_ϕ appears in the forward target $\Gamma_\phi(\cdot \mid x)$ and in the model-induced cluster prediction $\Gamma_\phi \mathbf{x}_\theta$. Consequently, the soft CT-ELBO admits collapsed configurations in which the intermediate channel becomes input-uninformative.

Collapsed input-independent maps. Consider the input-independent family

$$\mathcal{M}_{\deg} := \left\{ \Gamma_\phi : \Gamma_\phi(\cdot \mid x) = u \text{ for all } x \in \mathcal{V},\ u \in \Delta^{|\mathcal{C}|-1} \right\}. \tag{53}$$

For any $\Gamma_\phi \in \mathcal{M}_{\deg}$, the cluster variable $C \sim \Gamma_\phi(\cdot \mid X)$ is independent of the clean token X, so $I_\phi(C; X) = 0$. Moreover, by column-stochasticity, $\Gamma_\phi \mathbf{x}_\theta = u$ for any token distribution $\mathbf{x}_\theta$. Thus, the mask-state term vanishes,

$$\mathrm{KL}(\Gamma_\phi(\cdot \mid x) \,\|\, \Gamma_\phi \mathbf{x}_\theta) = \mathrm{KL}(u \| u) = 0, \tag{54}$$

and the cluster-state loss at $z_t = c_k$ reduces to

$$-\log \frac{\mathbf{x}_\theta[x]\,(\Gamma_\phi)_{k,x}}{(\Gamma_\phi \mathbf{x}_\theta)_k} = -\log \frac{\mathbf{x}_\theta[x]\,u_k}{u_k} = -\log \mathbf{x}_\theta[x], \tag{55}$$

for every k in the support of u. Hence, on $\mathcal{M}_{\deg}$, the hierarchy carries no semantic information and the objective locally degenerates to an MDLM-like word-prediction objective. Rank-one collapse, $\Gamma_\phi(k \mid x) = \mathbb{1}[k = k_0]$, and the globally uniform map, $\Gamma_\phi(k \mid x) = 1/|\mathcal{C}|$, are two extreme instances.

Information-retention regularization. The finite cluster vocabulary already imposes a bottleneck of capacity at most $\log |\mathcal{C}|$ nats, but the collapsed family above uses none of this capacity. We therefore add an information-retention regularizer. Let $X \sim p_{\mathrm{data}}$ and $C \sim \Gamma_\phi(\cdot \mid X)$. Define the marginal cluster usage

$$\bar{p}_\phi(c) = \mathbb{E}_X[\Gamma_\phi(c \mid X)]. \tag{56}$$

Then

$$I_\phi(C; X) = H(\bar{p}_\phi) - \mathbb{E}_X\left[H(\Gamma_\phi(\cdot \mid X))\right]. \tag{57}$$

The two terms correspond to complementary collapse modes: rank-one collapse destroys marginal entropy, while the globally uniform input-independent map maximizes conditional entropy.

We use

$$\mathcal{R}_{\mathrm{reg}}(\phi) = \lambda_{\mathrm{bal}}\left(\log |\mathcal{C}| - H(\bar{p}_\phi)\right) + \lambda_{\mathrm{cond}}\mathbb{E}_X\left[H(\Gamma_\phi(\cdot \mid X))\right]. \tag{58}$$

The first term encourages broad use of the intermediate vocabulary and penalizes mass concentration; the second discourages globally diffuse input-independent assignments by favoring informative token-to-cluster distributions. When $\lambda_{\mathrm{bal}} = \lambda_{\mathrm{cond}} = \lambda$,

$$\mathcal{R}_{\mathrm{reg}}(\phi) = \lambda\left(\log |\mathcal{C}| - I_\phi(C; X)\right), \tag{59}$$

so minimizing the regularizer maximizes the information retained by the learned hierarchy up to the fixed cluster budget.

The final objective is

$$\mathcal{L}(\theta, \phi) = \mathcal{L}_{\mathrm{NELBO}}(\theta, \phi) + \mathcal{R}_{\mathrm{reg}}(\phi). \tag{60}$$

This regularizer is a soft identifiability pressure rather than a hard deterministic constraint: selective high-entropy assignments may still be preferred when they improve the CT-ELBO, but globally input-independent collapsed configurations are penalized.

D Additional Experiment Details

D.1 Training Setup

Training settings. All OWT experiments follow the training setting commonly adopted in recent work on discrete DLMs. For OWT, models are trained with a sequence length of 512 and a batch size of 512. The standard training budget is 500k optimization steps (approximately 131B tokens), while extended-budget experiments use 1M steps (approximately 262B tokens). For LM1B, models are trained on sequences of length 128 with a batch size of 256 for 1M optimization steps, corresponding to approximately 33B tokens. Training is performed on 4–8 NVIDIA A100/B200 GPUs with 80–192GB memory using mixed-precision computation in `bf16`.

Model configuration. We instantiate all models using a DiT-style transformer backbone [15] and apply the GPT-2 tokenizer [16]. Our experiments are carried out at the SMALL scale, which consists of a 12-layer transformer with 12 attention heads and a hidden dimension of 768. This configuration results in approximately 92M non-embedding parameters. With $d_{\mathrm{hid}} = 768$ and $|C| = 64$, the cluster prototype matrix contains only $768 \times 64 = 49{,}152$ parameters, adding about 5×10^4 parameters in total.

Optimization details. Optimization is performed with the Adam optimizer using $\beta = (0.9, 0.99)$ and $\epsilon = 10^{-9}$. We adopt an initial learning rate of 5×10^{-4}, which is linearly warmed up during the first 10k steps and subsequently decayed using a cosine schedule. A weight decay of 0.02 is applied, and gradients are clipped to a maximum norm of 1.0.

Stability considerations. To ensure stable optimization, time-dependent loss weights associated with token- and cluster-level denoising are clipped during training. The maximum clipping thresholds are set to 2.0 or 10.0 following [28]. These weights are only clipped for computing the training objective. All denominators appearing in loss and ELBO weighting terms are lower-bounded by 10^{-4}.

Sequence handling. For sequences exceeding the maximum context length, a random contiguous window of 512 tokens is sampled during training. Shorter sequences are padded to the maximum length. Padding tokens contribute to the training loss but are excluded from ELBO-based evaluation metrics.

D.2 Diagnostics and Recipe Robustness

We provide additional diagnostics to characterize the learned hierarchy and examine its sensitivity to regularization and training choices. These analyses complement the generation results by highlighting properties that are not fully captured by validation PPL alone.

Soft assignments and hard projection. The learned hierarchy exhibits qualitatively distinct assignment patterns depending on the IB regularization strength. At $\lambda = 0.5$, Γ_ϕ is nearly one-hot, and hard-argmax projection has little effect on validation PPL. At $\lambda = 0.1$, however, the learned hierarchy distinguishes sharply assigned, cluster-specific tokens from tokens with diffuse memberships across clusters. The latter include many function words, punctuation marks, and common subwords. Hard-argmax projection substantially degrades validation PPL without retraining the denoiser, indicating that this learned configuration relies on genuinely soft assignments rather than merely approximating a hard partition.

Over-compression. Stronger IB regularization does not necessarily improve generation quality. At $\lambda = 1.0$, the learned hierarchy exhibits more concentrated cluster usage and substantially worse generative PPL than at moderate regularization, despite a relatively small difference in validation PPL. This illustrates that an ELBO-derived validation metric alone may not adequately reflect the generative consequences of the learned forward process.

Recipe robustness. We also examine the sensitivity of joint forward–reverse learning to the `force_transitting_within` option. Enabling this option during training at low IB regularization leads to rapid cluster collapse, whereas the default configuration maintains a non-degenerate learned hierarchy. In contrast, changing the option only at inference produces little difference in measured generation quality for the same checkpoint. These observations distinguish training-time stability from inference-time sensitivity and motivate our default training recipe.

Low-NFE generation. Importantly, the limitations of validation PPL as a comparative metric do not imply that learning the hierarchy offers no practical benefit. In matched multi-seed experiments, LHDLM consistently outperforms fixed HDLM under constrained denoising budgets. This suggests that the learned forward process provides an advantage in low-NFE generation, even when improvements in validation PPL or high-NFE generation are not observed.

Summary. Together, these diagnostics demonstrate that LHDLM learns nontrivial hierarchy structures whose behavior depends on regularization and training choices. While soft assignments introduce representational flexibility beyond fixed hard hierarchies, their practical benefit is most evident in low-NFE generation rather than uniformly improved language modeling metrics. These findings highlight the importance of evaluating learnable forward processes through both structural diagnostics and generation efficiency.

Table 4: Likelihood-based zero-shot accuracy (%) on six downstream benchmarks. Results are grouped by training-token budget. **Bold** indicates the best performance within each training budget.

Model	Train toks.	ARC-e	ARC-c	BoolQ	PIQA	OBQA	WinoG.	Avg. ↑
HDLM-64 (reimpl.)	131B	**26.17**	**23.72**	**49.97**	50.44	20.00	**51.38**	36.95
LHDLM-64 ($\lambda = 0.1$)	131B	26.05	22.78	49.76	**52.29**	**23.60**	50.91	**37.56**
HDLM-64 (reimpl.)	262B	25.79	23.55	49.91	50.98	22.00	50.75	37.16
LHDLM-64 ($\lambda = 0.1$)	262B	**26.77**	23.55	50.83	51.90	**24.20**	50.36	37.93
HDLM-128 (reimpl.)	262B	26.56	**24.66**	**51.99**	50.82	22.80	47.20	37.34
LHDLM-128 ($\lambda = 0.1$)	262B	26.47	**24.66**	50.12	**51.96**	23.60	**52.80**	**38.27**

E Additional Experimental Results

We provide additional results on zero-shot downstream benchmarks and LM1B test perplexity. We first evaluate the models trained under the experimental settings described in the main text on six zero-shot downstream tasks. As shown in Table 4, LHDLM consistently achieves modest improvements in average accuracy over its matched HDLM counterparts across different training budgets and cluster vocabulary sizes, although performance varies across individual tasks. We additionally report test perplexity on LM1B in Table 5. Under the same training budget, LHDLM-64 achieves comparable test perplexity to our HDLM-64.

Table 5: Test perplexity on LM1B. † reported in [28].

Model	Train. Toks.	Test PPL (↓)
AR Transformer	33B	22.32
BERT-Mouth [24]	-	$\leq$142.89
D3PM (absorb) [5]	-	$\leq$76.90
Diffusion-LM [12]	-	$\leq$118.62
DiffusionBert [10]	-	$\leq$63.78
SEDD [13]	33B	$\leq$32.79
MDLM [18]	33B	$\leq$27.04
HDLM-64 † [28]	33B	$\leq$26.95
ADLM [17]	33B	$\leq$**26.40**
HDLM-64(reimpl.) [28]	33B	$\leq$28.02
LHDLM-64(ours)	33B	$\leq$27.91

F Impact Statement

This paper presents research aimed at advancing the field of discrete DLMs. While improvements in language modeling capabilities can accelerate both beneficial and detrimental applications of text generation, we do not foresee unique risks arising specifically from learnable forward diffusion processes beyond those already present in existing language models. We believe the primary impact of this research lies in its methodological implications, offering a new perspective on the design space of diffusion-based generative models.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The findings in Section 4 and Appendix support and are consistent with the main claims articulated in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We explicitly address the limitations of our work in Section 6, where we discuss key assumptions and potential weaknesses of our approach.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: This paper provides detailed proof of CT-ELBO in Section A.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed experimental setup in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The experiments are based on publicly available datasets, but the code is not publicly available at submission time. We plan to release the code and reproduction instructions upon publication.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Comprehensive information about the model configuration, optimizer and hyperparameters used in each experiment, is available in Appendix D.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We conducted three matched-seed runs for HDLM and LHDLM ($\lambda = 0.1$) to assess the reproducibility of our main results, reporting mean performance and variability across runs. Due to the substantial computational resources and long runtimes required, all other experiments were conducted with a single run, and we do not report statistical significance estimates.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide details on hardware and exeperiments in Appendix D.1. We conduct all experiments on 4xA100s(80GB), 8xA100s(80GB), 4xB200s(192GB) or 8xB200s(192GB). Experiments are done in 30 hours in 8xB200s.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We affirm that our work fully complies with the NeurIPS Code of Ethics, including requirements related to anonymity, responsible data use, and ethical experimental design.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: In Section F, we discuss the potential societal benefits and risks of our work. We also commit to continuously monitoring its broader impacts and promoting reproducibility by releasing code and evaluation protocols.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper poses no such risks. This models are trained on public datasets.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All external assets used in this work are publicly available, and we provide appropriate citations.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: We provide no new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.