HIERARCHICAL CONTINUOUS DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion Language Models (DLMs) hold the promise of order-agnostic, parallel text generation. Recently, continuous diffusion and flow matching models have seen substantial gains, driven by carefully crafted token representations and diffusion/flow spaces. In this work, we introduce **Hierarchical Continuous Diffusion Language Models** (H-CDLMs), a simple framework that further improves continuous DLMs with minimal compute and parameter overhead. Drawing on the discrete DLM and continuous image diffusion literature on *joint diffusion*, we diffuse multiple modalities in parallel. These modalities represent tokens at different semantic granularities: in our instantiation, the tokens themselves and coarser *clusters* obtained by clustering pretrained token embeddings. We propose a general setup that allows per-modality samplers and schedules to enhance the interplay between modalities, and apply it to CoBit, yielding our H-CoBit model, which delivers large empirical gains. At dataset entropy, H-CoBit improves MAUVE and reaches a generative perplexity (GenPPL) of 49.4 on LM1B and 50.4 on OWT, improving on the baseline by 35.8 and 20.7 points and surpassing even discrete DLMs of comparable size. On GSM8K, it reaches 27.4% accuracy, outperforming prior diffusion and flow-based models.

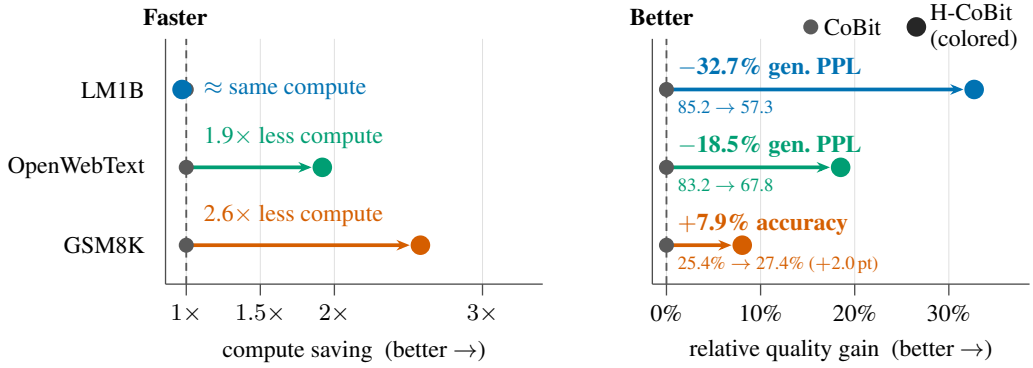


Figure 1: **Better generations for less compute.** For each dataset, **(left)** the compute H-CoBit (ours) needs to match/surpass *CoBit's best results*, relative to CoBit's own cost. Compute is each arm's measured per-denoiser-call cost times its NFE; **(right)** the quality gain, Gen-PPL is read at dataset entropy, GSM8K reports accuracy. LM1B is not budget-matched (CoBit 1M steps vs H-CoBit 600k).

1 INTRODUCTION

Recent progress in language modeling (Touvron et al., 2023) was enabled by the simple and scalable auto-regressive (AR) factorization that supports left-to-right generation via next-token prediction. This sequential nature, however, precludes parallelization and can hinder performance on non-left-to-right tasks. Diffusion Language Models (DLMs) offer a promising alternative by denoising entire sequences in parallel (Austin et al., 2021; Sahoo et al., 2024). Yet realizing their potential is difficult: diffusion (or flow) architectures lack a single canonical training recipe, and neither discrete nor continuous DLMs have fully closed the gap with AR models despite success in other modalities. A key design choice is how to represent and corrupt tokens. Discrete approaches such as Masked Diffusion Language Models (MDLM) have narrowed the gap (Sahoo et al., 2024; Austin et al., 2021;

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Sahoo et al., 2025). Continuous formulations are harder: the design space is larger, and standard Gaussian diffusion with linear schedules is poorly matched to discrete data (Shabalin et al., 2026). Recent work has nevertheless closed much of this gap through carefully chosen continuous token representations and adaptive noise schedules (Chen et al., 2026; Lee et al., 2026).

Our work builds on two independent recent innovations. First, the parallel diffusion of multiple modalities has yielded improvements in the image generation literature. Kouzelis et al. (2025); Baade et al. (2026) showed that diffusing both DINOv2 features and pixels with a single network improve performance. Second, Zhou et al. (2025) have shown that predicting higher level representations of tokens before the tokens themselves in a discrete diffusion setup improves generation quality.

Motivated by these results, we introduce *Hierarchical Continuous Diffusion Language Models* (H-CDLMs), which augment continuous diffusion language models with auxiliary modalities that represent tokens at coarser semantic granularities. In the simplest instantiation, which we study in this work, we use two modalities: the original token sequence and the corresponding sequence of cluster IDs, obtained by clustering pretrained token embeddings. We embed the cluster IDs into a continuous space and *jointly* diffuse and denoise both modalities with a single model. At inference, the samplers and schedules of the two modalities can be tuned independently to optimize their interaction.

The cluster modality is especially well suited to continuous text diffusion. Token embeddings encode rich linguistic and semantic structure: tokens that play similar roles tend to occupy nearby regions of the embedding space, and hence the same cluster. Under continuous noise, recovering the exact embedding of a token can be difficult, but identifying the region in which it lies is often still feasible. Cluster IDs supply exactly this coarse-grained signal. By indicating which region of the embedding space a token belongs to, they narrow the set of plausible tokens without committing to a specific one (Figure 2). The cluster modality thus acts as a semantic anchor that steers the denoising trajectory toward the target token. Crucially, because the two modalities are denoised in parallel, information flows bidirectionally across semantic granularities: coarse cluster predictions can guide token recovery under high noise, while token-level predictions simultaneously refine the cluster estimates.

H-CDLM adds little parameter or computational overhead when the number of clusters is small relative to the vocabulary, and can be applied directly to existing continuous DLMs. We apply it to CoBit, a state-of-the-art continuous DLM that uses bit representations to reduce output-head size. The resulting model, H-CoBit, delivers large gains in both generation quality and efficiency.

Our contributions are:

1. We introduce H-CDLM, a framework for jointly diffusing hierarchical textual modalities in continuous DLMs, with independent samplers and noise schedules per-modality.
2. We instantiate the framework on CoBit using two modalities constructed from pretrained token embeddings, yielding H-CoBit.
3. We show that H-CoBit delivers large gains across benchmarks (Figure 1). At entropy matched to that of the data, it reduces generative perplexity (GenPPL) from 85.2 to 49.4 on LM1B and from 71.1 to 50.4 on OpenWebText (OWT), improvements of 35.8 and 20.7 points over CoBit that surpass even discrete DLMs, while also improving MAUVE on both datasets. It matches CoBit's performance at a substantially lower budget, and it reaches 27.4% accuracy on GSM8K, outperforming all prior continuous diffusion and flow language models. A matched random-cluster control fails to recover these gains, confirming that they stem from meaningful cluster semantics rather than added capacity or auxiliary structure.

More broadly, H-CDLMs offer a path toward language models that denoise text at multiple levels of abstraction at once, from tokens to progressively coarser clusters and potentially to entire sentences. We view our two-level instantiation as a first step in this direction.

2 Related work

Discrete Diffusion Models. Discrete Diffusion Models operate directly in the discrete space of tokens. They rest on a Markov chain which defines the corruption mechanism. D3PM (Austin et al., 2021) proposes a general framework for such processes. Corruption mechanisms range from masking to uniform token transitions. MDLM (Sahoo et al., 2024) first demonstrated strong generation quality for text by simplifying D3PM on masked diffusion, while Duo (Sahoo et al., 2025) showed that

uniform discrete architectures can be equally good, and drew a connection to Gaussian diffusion of one hot vectors. HDLM (Zhou et al., 2025) extends MDLM by building a hierarchical vocabulary of tokens, allowing for a fine grained unmasking process. TDLM (Wu et al., 2026) drops HDLM's full vocabulary prediction by only predicting the next scale semantics along a hierarchical tree diffusion process – thus considerably increasing throughput. Our work can be seen as a continuous analog of HDLM (and TDLM), although we diffuse all modalities jointly rather than unmasking clusters before tokens. Another line of work augments masked discrete diffusion with a paired continuous diffusion process: tokens are corrupted by both masking and Gaussian noise, and a single model learns to reverse both processes jointly (Pynadath et al., 2026a; Zhou et al., 2026; Zheng et al., 2025).

Continuous Diffusion and Flow Language Models. Continuous Diffusion or Flow Language Models corrupt continuous representations of tokens, and train a denoiser network to reconstruct clean token embedding sequences from noisy ones. Recently, LangFlow (Chen et al., 2026) and FLM/FMLM (Lee et al., 2026) have shown that, when carefully designed, such processes can yield similar performance to their discrete diffusion counterparts. FLM embeds tokens as one hot vectors and directly diffuses in the one hot vector space, while LangFlow exploits trained embeddings. Both models however train their denoiser to predict tokens directly and then pull the token's representation backwards into the diffusion process, entailing a large vocabulary sized output head and low throughput. ELF (Hu et al., 2026) shows that continuous DLMs can be effective with *minimal adaptation* to the discrete nature of the data, and especially without predicting actual tokens. This allows a smaller output head whose size depends on the hidden dimension rather than vocabulary. S-FLM (Deschenaux & Gulcehre, 2026) leverages Flow Matching on the hypersphere to improved token representations, with strong performance on downstream tasks. CoBit (Batzolis et al., 2026) exploits a bit representation of tokens, previously introduced by Chen et al. (2023), alongside a carefully designed parametrization and stochastic sampling. This further reduces output head size and yields the strongest results known for continuous DLMs.

Multimodal joint Diffusion. Multimodal joint diffusion diffuses multiple representations of a data sample in parallel, via a single denoiser network. For images, this can consist in diffusing pixels and DinoV2 (Oquab et al., 2024) features jointly, and keeping only the generated pixels at the end of the sampling process (Kouzelis et al., 2025; Baade et al., 2026). This augmentation yields generation quality gains (Kouzelis et al., 2025), especially when coupled with an ordered scheduler which enhances the interplay between high and low level data features through the sampling process (Baade et al., 2026). This framework rests on the idea that diffusing high and low level features jointly provides a form of guidance to the denoiser. It has not, to the best of our knowledge, been extended to continuous DLMs yet. Bao et al. (2023) jointly diffuse image and text pairs, enabling unconditional text generation, but they do not evaluate this specific task.

3 Background

In this section, we present a general setting for continuous diffusion and flow models for text. Our setting covers existing continuous flow and diffusion language models (Chen et al., 2023; Lee et al., 2026; Chen et al., 2026; Batzolis et al., 2026; Hu et al., 2026; Deschenaux & Gulcehre, 2026), although they differ in their token representation, corruption parametrization, training objective and sampler. The only exception is $\mathcal{S}$-FLM (Deschenaux & Gulcehre, 2026), whose corruption is a geodesic interpolation on the hypersphere. Details on how this setup applies lie in Appendix B.

3.1 Continuous Diffusion Of Discrete Tokens

We consider text sequences composed of L tokens over a vocabulary $\mathcal{V}$. We aim to approximate the distribution $p_{\text{data}} : \mathcal{V}^L \to [0, 1]$ of a sequence $y \in \mathcal{V}^L$ via the learned model $p_\theta(y)$.

Continuous diffusion first requires a continuous representation in a space E^d of the vocabulary via the encoder and decoder functions $\mathcal{E} : y \to x$ and $\mathcal{D} : x \to y$, verifying $\mathcal{D}(\mathcal{E}(y)) = y$, with d the embedding dimension. Generally, we set $E = \mathbb{R}$, although recent work also explores hyperspherical embeddings, that is $E = \mathbb{S}$ (Deschenaux & Gulcehre, 2026). We write $x_0 = \mathcal{E}(y) \in \mathbb{R}^{L\times d}$, applying $\mathcal{E}$ position-wise. The diffusion process lives in $E^{L\times d}$, following the Gaussian corruption process

$$x_\sigma = x_0 + \sigma\epsilon, \qquad \epsilon \sim \mathcal{N}(\epsilon; 0, I_{Ld}), \qquad \sigma \in [\sigma_{\min}, \sigma_{\max}]. \tag{1}$$

Note that, without loss of generality, we adopt the variance-exploding (EDM) convention (Karras et al., 2022). Flow matching formulations use general parametrizations $x_t = \alpha_t x_0 + \sigma_t \epsilon$ (Chen et al., 2026; Lee et al., 2026), and reduce to equation 1 by rescaling (see explicit correspondence in Appendix B). We consider a denoiser that predicts the clean data embeddings $D_\theta(x_\sigma, \sigma) \approx \mathbb{E}[x_0|x_\sigma] \in E^{L\times d}$.

3.2 Training

Due to the conditional expectation formulation of $D_\theta(x_\sigma, \sigma)$, the natural training objective of this model is solving the regression problem $D = \operatorname{argmin}_{\hat{D}} \mathcal{L}_{MSE}(\hat{D})$, with

$$\mathcal{L}_{\text{MSE}} = \mathbb{E}_{x_0,\, \epsilon,\, \sigma\sim p_{\text{train}}}\Big[\frac{w(\sigma)}{L}\sum_{i=1}^{L}\big\|D_\theta(x_\sigma,\sigma)_i - x_{0,i}\big\|^2\Big], \tag{2}$$

where p_{train} is the training noise density and $w(\sigma)$ a weighting function; both control how learning effort is allocated across noise levels (Karras et al., 2022). Recent work replaces or augments the regression loss with other training objectives, including cross entropy loss (Chen et al., 2026; Lee et al., 2026; Deschenaux & Gulcehre, 2026). We refer to Appendix B for further details.

3.3 Sampling

We sample from this general model by backward integration of the probability-flow ODE induced by the forward process in equation 1, along

$$\frac{dx}{d\sigma} = -\sigma\,\nabla_x \log p_\sigma(x) = \frac{x - D_\theta(x,\sigma)}{\sigma}, \tag{3}$$

using standard solvers such as Heun or Euler on a discrete noise grid $\sigma_{\max} = \sigma_0 > \sigma_1 > \cdots > \sigma_N = \sigma_{\min}$. The final step consists in decoding position-wise via $\mathcal{D}$. The underlying discrete nature of the data requires two main adjustments that decisively improve generation quality:

Information-uniform noise schedules. For text, the conditional information about token identities is not released uniformly with noise, it is concentrated in a narrow band of noise. Following recent continuous DLMs (Chen et al., 2026; Batzolis et al., 2026; Lee et al., 2026; Deschenaux & Gulcehre, 2026), we adopt information-uniform noise schedules to focus sampling on noise regions where information about the tokens changes the most.

Stochastic churn. A second lever trades the deterministic ODE for a stochastic sampler (Karras et al., 2022): at step i, the state is churned back up to $\hat{\sigma}_i = (1+\gamma)\sigma_i$ before the solver integrates down to σ_{i+1}: $\hat{x}_i = x_i + s_{\text{noise}}\sqrt{\hat{\sigma}_i^2 - \sigma_i^2}\,\epsilon$, $\epsilon \sim \mathcal{N}(\epsilon; 0, I_{Ld})$. The per-step churn rate $\gamma = S_{\text{churn}}/N \geq 0$ sets how much noise is re-injected, while $s_{\text{noise}} \geq 1$ inflates its variance (Karras et al., 2022). Unlike γ, s_{noise} is a property of the sampler rather than of the parametrization, hence model-agnostic. Churn is essential to CoBit's generation quality (Batzolis et al., 2026); Section 4.3 makes γ *per-modality*. Noise truncation (Deschenaux & Gulcehre, 2026) is a further such lever; we discuss how these intertwine with our method in Sections 4.3 and Appendix B.4.

4 Method

In this section, we present our hierarchical framework for continuous DLMs. We define a multimodal diffusion process (Section 4.1) and detail our hierarchical modalities for text (Section 4.2). We then show how information-uniform noise scheduling extends Per-modality (Section 4.3).

4.1 Hierarchical continuous diffusion

Following the multimodal formulation of Kouzelis et al. (2025), we define a joint diffusion process over k modalities, operating on vocabularies $\{\mathcal{V}^j\}_{j=1}^k$. In practice, we set $k = 2$ to distinguish plain tokens ($j = 1$) from *high level* semantic units ($j = 2$), or *clusters*. The diffusion process then noises continuous representations $x = \{x^j\}_{j=1}^k$, with $x^j \in E^{L\times d_j}$, of the data $y = \{y^j\}_{j=1}^k$, with $y^j \in (\mathcal{V}^j)^L$ and $\mathcal{V}^1 = \mathcal{V}$. The modalities are concatenated along the channel axis, so that

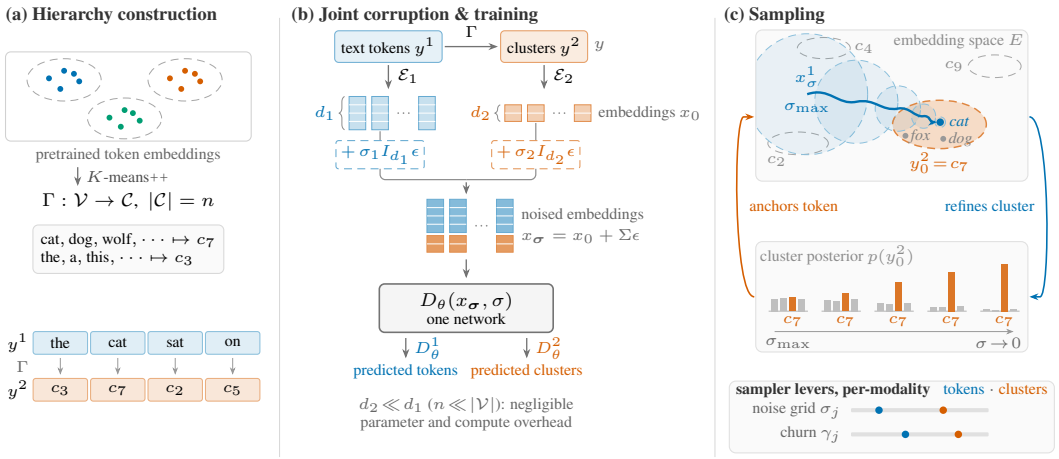


Figure 2: **Hierarchical continuous diffusion** (k=2)**. (a)** A surjective map Γ, obtained by balanced K-means++ on pretrained embeddings, sends every token to a cluster, yielding a second symbol sequence of the same length L. **(b)** The cluster index is encoded with the host model's own token encoder. The two codes are concatenated at each position and corrupted at their own noise levels σ^1, σ^2. One network denoises both modalities. **(c)** Clusters act as a semantic anchor for tokens. This interplay can be optimized via per-modality scheduler/sampler parameter tuning.

$x \in E^{L\times d}$ with $d = \sum_{j=1}^{k} d_j$. This integrates directly into an existing continuous DLM architecture (see Section 3) by concatenating the modalities' embeddings at each position. It does not increase sequence length. Each modality can be corrupted at its own noise level. The forward process of equation 1 becomes

$$x_{\boldsymbol{\sigma}} = x_0 + \Sigma\,\epsilon, \qquad \epsilon \sim \mathcal{N}\big(\epsilon; 0,\, I_{Ld}\big), \qquad \Sigma = \mathrm{blockdiag}\left(\sigma_1 I_{d_1}, \ldots, \sigma_k I_{d_k}\right) \otimes I_L, \tag{4}$$

where $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_k)$ collects the per-modality noise levels. The denoiser outputs one clean-data estimate Per-modality, $D_\theta(x_{\boldsymbol{\sigma}}, \boldsymbol{\sigma}) = (D_\theta^1, \ldots, D_\theta^k)$ with $D_\theta^j \in E^{L\times d_j}$. At training time, we minimize

$$\mathcal{L}_{\text{MSE-H}} = \mathbb{E}_{x_0,\,\epsilon,\,\boldsymbol{\sigma}}\left[\frac{1}{L}\sum_{i=1}^{L}\sum_{j=1}^{k} \lambda_j\, w(\sigma_j)\, \big\| D_\theta^j(x_{\boldsymbol{\sigma}}, \boldsymbol{\sigma})_i - x_{0,i}^j \big\|^2\right], \tag{5}$$

with λ_j per-modality loss weights. Note that expectation over time/noise is taken at the global level (see Appendix C). At sampling time, all hierarchical levels are diffused jointly. At the last step, all modalities except the first (token-level) are dropped, and it is decoded with its decoder $\mathcal{D}_1$. Discarding the auxiliary modalities is principled whenever they are deterministic functions of the tokens, which is the case for the constructions detailed in Section 4.2 (see Proposition 1 in appendix C).

4.2 Constructing hierarchical modalities for text

Our high-level modality provides an anchor for the diffusion of tokens. We instantiate it as *token clusters*, following the semantic hierarchy of Zhou et al. (2025). This allows us to keep our multimodal data representation intrinsically discrete. Each cluster can be attributed to a *semantic* group of tokens, which can be easier to infer at high noise levels than the exact token identity. In practice, a surjective map $\Gamma : \mathcal{V} \to \mathcal{C}$ assigns each token to one of $n = |\mathcal{C}|$ clusters, and the high-level sequence is obtained position-wise, $y^{2,l} = \Gamma(y^{1,l})$, so that both modalities share the sequence length L. We compute Γ by K-means++ on pretrained token embeddings taken from LangFlow (Chen et al., 2026) (for LM1B, OWT) and $\mathbb{S}$-FLM (Deschenaux & Gulcehre, 2026) (for TinyGSM). The cluster index must itself be given a continuous encoding $\mathcal{E}_2$. The most direct way of constructing $\mathcal{E}_2$ is by using the same mechanism as for $\mathcal{E}_1$, making the integration to an existing DLM seamless. For CoBit (Batzolis et al., 2026), this entails encoding clusters via bit sequences. This choice aims at simplicity. More complex multimodal token representations are left to future work. More generally, additional modalities ($k > 2$) can be obtained by clustering at multiple granularities, yielding a deeper hierarchy. We refer to Appendix A for further details, and to Wu et al. (2026) for a $k > 2$ modalities example.

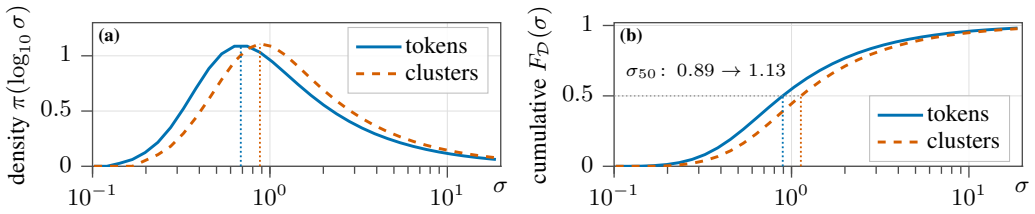


Figure 3: Token vs. cluster information-uniform noise schedules for H-CoBit: (a) density over log-noise levels; (b) cumulative distributions, with the median shifting from $\sigma_{50} = 0.89$ to 1.13.

4.3 Per-Modality Scheduling And Sampling

A key opportunity of joint diffusion is to adapt the scheduler and sampler *per-modality*, so as to enhance the interplay between modalities throughout the denoising process, and to adapt the noise schedule and sampler to each modality' underlying data structure.

Per-modality scheduler. In multimodal joint image diffusion, Baade et al. (2026) for instance order the modalities: the high level modality is kept cleaner than the pixels via per-modality schedulers, so that resolved semantics guide pixel denoising. In image diffusion, this requires setting an explicit ordering of the modalities' schedules. Our framework offers a simpler alternative: training a *per-modality* information-uniform noise schedule. For each modality j and sampling step i, $\sigma_{i,j}$ is set so that it yields a grid on which each step reveals approximately equal conditional information about the underlying data (see Section 3.3 and Appendix C.2 for details on how to train $\sigma_{i,j}$). Figure 3 shows such trained schedules for 2 modalities. Cluster density is right-shifted: cluster identity is a coarser variable, so information about it survives larger noise so that its entropy is released at higher σ. Figure 3 shows that each modality has a different underlying structure. This motivates adapting the scheduler and sampler *per-modality*. In our experiments, we set our scheduler per-modality by default, and ablate the possibility of using a single shared scheduler (see Table 3).

Per-modality sampler (PMS). One direct way to adapt the sampler Per-modality is to leverage stochastic sampling (see Section 3.3). We enable two key mechanisms: (i) adapt the churn rate Per-modality, so that $\hat{\sigma}_{i,j} = (1+\gamma_i^j)\sigma_{i,j}$, $\hat{x}_i^j = x_i^j + s_{\text{noise}}\sqrt{\hat{\sigma}_{i,j}^2 - \sigma_{i,j}^2}\epsilon$ for $\epsilon \sim \mathcal{N}(0, I)$. (ii) churn clusters on a predefined window $\mathcal{W}$. We set it to the first ($\mathcal{W}_{\text{early}}$) or last ($\mathcal{W}_{\text{late}}$) half of sampling steps. Combined, these mechanisms yield clear improvements in H-CoBit's performance We provide a comprehensive evaluation of stochastic sampling Per-modality on H-CoBit in Appendix E.2. By construction, any sampler mechanism or parameter can be tuned per-modality to improve the interplay between modalities. We leave further tuning to future work.

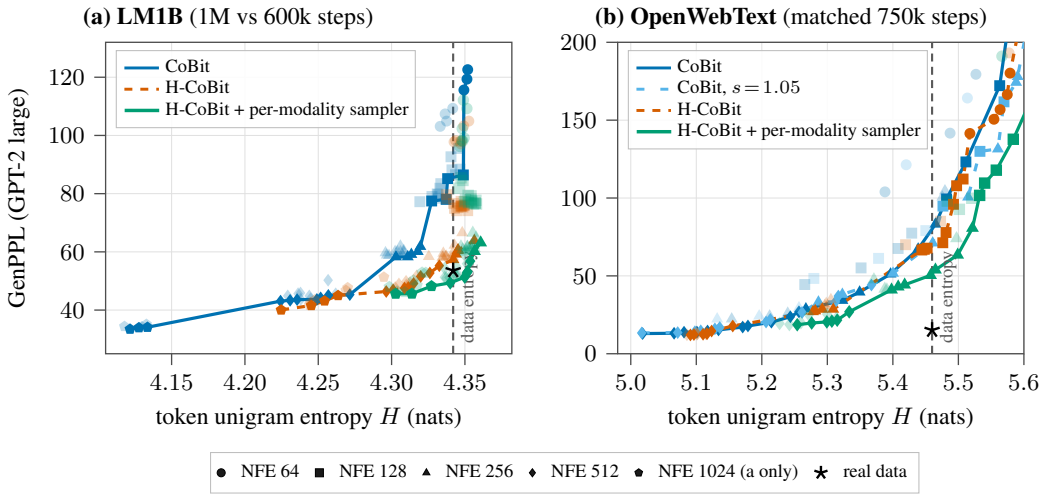
5 Experiments

We evaluate our framework on CoBit (Batzolis et al., 2026), a state-of-the-art (SOTA) continuous DLM in terms of downstream task accuracy and GenPPL frontier. We first present our experimental setup (Section 5.1), then our main training and evaluation results for H-CoBit on standard text generation and downstream tasks (Section 5.2), and finally our ablations (Section 5.3).

5.1 Experimental Setup

Training. Following Chen et al. (2026); Sahoo et al. (2024; 2025); Deschenaux & Gulcehre (2026); Batzolis et al. (2026); Hu et al. (2026), we train on LM1B (Chelba et al., 2013) using 128-token sequences and the BERT-base-uncased tokenizer ($V = 30522$), on OWT (Gokaslan & Cohen, 2019) using 1024-token sequences with the GPT-2 tokenizer ($V = 50257$), and on TinyGSM (Liu et al., 2023) using 512-token sequences with the SmolLM-135M tokenizer ($V = 49153$). Our denoiser is a 12-layer, 12-head Diffusion Transformer with hidden size 768, resulting in 134.6M and 135M parameters for LM1B and OWT respectively (parameter overhead of approximately +1% compared to the CoBit baseline). All other training parameters are set equal to Batzolis et al. (2026).

Figure 4: **Generation quality against diversity.** Gen-PPL versus token unigram entropy for (a) LM1B, 1M vs 600k steps, (b) OpenWebText, matched 750k steps. Each model is swept over churn γ and NFE budget; the solid curve is the Pareto frontier. The dashed rule is the data's unigram entropy and the star is real data. H-CoBit (n=16) uses cluster loss weight $\lambda_2 = \frac{4}{15}$ in (a) and $\lambda_2 = 0.1$ in (b). Per-modality sampler resamples the same checkpoint with cluster churn confined to the late half of the schedule, at $\gamma_2 = 0.4$ in (a) and $\gamma_2 = 0.3$ with global $s_{\text{noise}} = 1.05$ in (b). s_{noise} is model-agnostic, so (b) also plots CoBit at $s_{\text{noise}} = 1.05$ for fair comparison



Evaluation. We evaluate by measuring GPT-2 Large generative perplexity (GenPPL) - entropy (H) frontiers, following Sahoo et al. (2024); Chen et al. (2026); Deschenaux & Gulcehre (2026); Hu et al. (2026); Batzolis et al. (2026). We also compute MAUVE and GSM8K (Cobbe et al., 2021) accuracy for our model trained on TinyGSM. MAUVE and GenPPL are evaluated on 1024 generated sequences. All results are reported across a full sweep over Number of Function Evaluations (NFE) and sampling parameters, especially churn which is essential for CoBit. Per-modality loss weighting coefficients λ_j are tuned for each model/dataset. For the CoBit baseline, we keep the original paper's inference hyperparameters: global stochastic sampling noise is $s_{\text{noise}} = 1.03$ on LM1B/OWT, $s_{\text{noise}} = 1.003$ on GSM8K. We ablate $s_{\text{noise}} = 1.05$, for fair comparison with our per-modality samplers on H-CoBit.

5.2 Main Results

Figure 4 displays improvements of our model H-CoBit in the GenPPL-entropy frontiers on LM1B and OWT, shown by right shifts of the frontier. Table 1 shows GenPPL levels at given entropies. On LM1B, H-CoBit matches dataset entropy at GenPPL 57.3, more than 20 points better than CoBit, at only 60% of the training steps. On OWT at matched 750k steps, H-CoBit reaches GenPPl 67.8 at dataset entropy beating CoBit by 15.4 points. Adding per-modality churn further improves H-CoBit's overperformance: on OWT, combined with $s_{noise} = 1.05$, H-CoBit reaches Gen PPL 50.4, better than CoBit-S at same s_{noise} (71.1), and than any DLM baseline at dataset entropy. H-CoBit also improves MAUVE at dataset entropy, reaching 0.903 against 0.861 on LM1B, and 0.580 against 0.530 on OWT against CoBit (see Table 11, Appendix E). We note H-CoBit incurs only a negligible computational overhead (+4.2%) while matching CoBit's performance at much lower budget (Figure 1).

Table 2 displays H-CoBit's accuracy on GSM8K. On these 1319 mathematical reasoning problems, H-CoBit beats all other continuous DLMs, and narrows the gap to discrete DLMs.

5.3 Ablations

In this section, we discuss ablations of our framework. Due to limited compute, when retraining is required to conduct an ablation, we constrain to LM1B at 100k training steps.

Number of clusters and cluster loss weight. Figure 5(b,c) studies the optimal number of clusters n and cluster loss weight λ_2. Both panels (c) and (b) show that $n = 16$ is optimal: it is the only n to

Table 1: **Generation quality, diversity at one operating point.** Every row is one generation scored by GenPPL and entropy H. All reported continuous DLMs have ≈130M parameters (not ELF). H-CoBit is our hierarchical continuous DLM framework on CoBit-S. H-CoBit uses cluster-loss weights λ_2=$4/15$ / 0.1 for (LM1B / OWT). We report genPPL at matched dataset entropy when it is reached by the model. Per-Modality Sampler (PMS) samples with tuned churn per-modality: LM1B uses late window, $\gamma_2 = 0.4$ and OWT uses late window, $\gamma_2 = 0.3$ + global $s_{noise} = 1.05$ (compared to Cobit-S, $s_{noise} = 1.05$). Bold is best at matched H. Baselines are taken from the original paper, Franca & Tong (2026) or Pynadath et al. (2026b).

	LM1B (L=128)				**OpenWebText** (L=1024)			
Model	Steps	NFE	$H\uparrow$	GenPPL↓	Steps	NFE	$H\uparrow$	GenPPL↓
Real text (test split)	–	–	4.34	53.1	–	–	5.46	14.9
Autoregressive								
AR baseline (Franca & Tong, 2026)					–	–	5.61	40.2
Discrete diffusion								
SEDD (Lou et al., 2024)					–	–	5.65	125.7
Duo (Sahoo et al., 2025)	1M	–	4.28	98.6	1M	128	5.46	63.3
MDLM (Sahoo et al., 2024)	1M	1024	4.32	109.21	1M	128	5.46	66.2
Hybrid discrete–continuous								
CANDI (Pynadath et al., 2026a)	–	–	4.32	119.9	1M	128	5.46	71.0
CADD (Zheng et al., 2025)	–	–	–	–	1M	128	5.48	55.3
Continuous diffusion / flow								
LangFlow (Chen et al., 2026)	1M	128	4.31	92.2	1M	128	5.43	60.1
ELF-M (342M) (Hu et al., 2026)					95k	64	5.44	62.5
FLM (Lee et al., 2026)	1M	1024	4.28	119.1	1M	1024	5.40	96.8
FMLM (Lee et al., 2026)	100k	32	4.19	78.8	100k	32	5.22	61.0
S-FLM (Deschenaux & Gulcehre, 2026)					1M	256	5.46	108.2
CoBit (Batzolis et al., 2026) vs. H-CoBit (ours)								
CoBit-S	1M	128	4.34	85.2	750k	256	5.46	83.2
H-CoBit	600k	256	4.34	57.3	750k	128	5.45	67.8
CoBit-S, s_{noise}=1.05	1M	256	4.34	73.6	750k	256	5.46	71.1
H-CoBit + PMS	600k	512	4.34	**49.4**	750k	256	5.46	**50.4**

Table 2: **Mathematical reasoning accuracy.** GSM8K accuracy at matched training budget. CoBit/H-Cobit are reported at their best churn. H-CoBit ($\lambda_2 = 0.3$) beats CoBit by +2pts at half the NFE, +4pts at matched NFE. Baselines are from Deschenaux & Gulcehre (2026); Batzolis et al. (2026)).

Model	**Family**	**NFE**	**Acc. (%)**
Autoregressive (reference ceiling)			
AR (greedy)	autoregressive LM	–	63.3
AR (sampling)	autoregressive LM	–	53.9
Discrete DLMs			
MDLM (low-T)	masked discrete diffusion	1024	∼33
Duo (low-T)	discrete diffusion	1024	∼36
Continuous DLMs			
FLM	continuous flow	1024	0.3
S-FLM (+top-k=1)	hyperspherical flow	1024	18.0
CoBit-S (no cluster bits)	bitstream diffusion	512	23.4
CoBit-S (no cluster bits)	bitstream diffusion	1024	25.4
H-CoBit (n=16) **(ours)**	bitstream diffusion	512	**27.4**

beat the baseline in all NFEs and values of λ_2. Here, the multimodal process is beneficial at a very low number of clusters compared to vocabulary size ($V = 30522$ for LM1B). We hypothesize that this is closely tied to the clustering. At $n = 16$, the optimal cluster loss weight is $\lambda_2^{\text{opt}} = 5 \cdot 10^{-2}$ (we use $\lambda_2 = \frac{4}{15}$, implying LM1B performance could be further improved). This is likely highly dependent on n and the data set, thus requiring careful tuning for each new task.

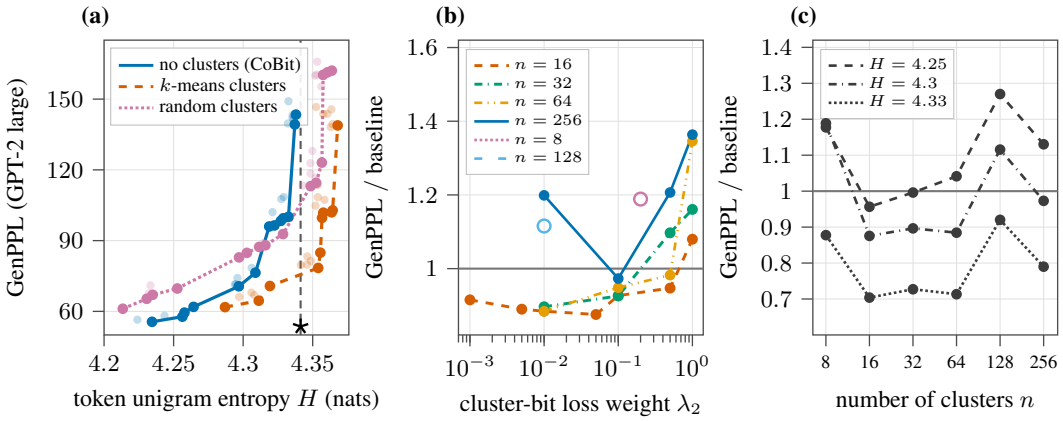


Figure 5: **H-CoBit cluster design: semantics, weight, count.** LM1B at 100k steps. **(a)** GenPPL/entropy frontier for n=16 clusters from k-means over LangFlow (Chen et al., 2026) embeddings, against a random cluster assignment control. Random clusters yield no improvement on the baseline. **(b)** GenPPL gain over the baseline at dataset entropy with respect to loss weight and n. **(c)** Cluster-count ablation with each n given the λ_2 that is best for it at three entropy levels. $n = 16$ **is best at every** H. For (b) and (c), the grey rule is what H-CoBit variants have to beat (lower is better).

Semantic vs. random cluster structure. In order to verify that the semantic structure of our *high level modality* is meaningful, we train and evaluate H-CoBit with *random* cluster assignments. This means that a token's cluster is random, not dictated by a clustering of pretrained token embeddings. Figure 5(a) compares the performance of the CoBit baseline to H-CoBit with random or with semantic clusters. It clearly shows that the random clusters model's frontier is intertwined with that of the baseline, while the semantic cluster model yields a right shift. Hence, cluster semantics matter. It is insufficient to provide *high level* categories; these categories should bear meaning.

Per-modality scheduling and sampling. For H-CoBit, we tune two per-modality sampling mechanisms (see Section 4.3). **(1)** The *per-modality* information uniform schedule. Table 3 compares this with a single shared schedule, showing the per-modality schedule is slightly better in MAUVE and GenPPL at dataset entropy, but worse at lower entropies. Neither schedule thus clearly dominates the other. **(2)** Second, the stochastic churn level is scheduled and scaled differently for the cluster modality. We adopt two churning strategies: churning the cluster modality (via γ_2) at a different level, and churning it with a phase. Churning it only late, at the low half of noise, achieves -7.9/-17.4 GenPPL gains on the shared churn control on LM1B/OWT (see Figure 4, Table 12).

Table 3: **Per-modality vs Single noise schedule.** LM1B n=16. MAUVE is the best achieved.

	Gen-PPL at token entropy H				
Arm	≈4.28	≈4.30	≈4.32	≈4.3413	**MAUVE**
dual (per-modality schedule)	46.4	46.9	**51.6**	**57.3**	**0.973**
single (shared schedule)	**44.6**	**45.1**	52.8	57.4	0.965

6 Conclusion

We have proposed a new multimodal joint diffusion framework for text with minimal parameter overhead. This is the first work to integrate hierarchical representations of tokens into continuous diffusion processes for text. It integrates innovations from the discrete hierarchical diffusion and the multimodal joint image diffusion literature. This multimodal setup allows consistent generation quality and downstream task accuracy gains when applied to CoBit, beating recent Diffusion Language Models in generative quality and on a mathematical reasoning task. Our simple instantiation's large empirical gains suggest a broader direction: generating text not from a single tokenizer, but jointly across several representations of a sequence, from sub-word units to sentence-level semantics.

AI USE STATEMENT

In this work, we used generative AI tools to create or edit software code, to edit figures and tables, to review grammar and formulation mistakes, and to polish text. We have not used generative AI tools for research directions, to formulate proofs, propose hypotheses, or interpret results, and the rest of the uses of AI required to disclose are not applicable for this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Generative models carry a substantial risk of misuse. Their application can lead to various negative societal impacts, most notably the spread of disinformation. Enhancements in generative performance, as achieved by our method, may further increase the realism of generated content, potentially making disinformation even more convincing.

REPRODUCIBILITY STATEMENT

We provide implementation details, dataset descriptions and preprocessing, hyperparameters and evaluation metrics for reproducing H-CoBit results in Section5 and Appendix Section D. All our implementations are based on open source code. Code and model checkpoints will be made publicly available upon acceptance.

REFERENCES

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, 2021.

Alan Baade, Eric Ryan Chan, Kyle Sargent, Changan Chen, Justin Johnson, Ehsan Adeli, and Li Fei-Fei. Latent forcing: Reordering the diffusion trajectory for pixel-space image generation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=LFUaHUCO1a.

Fan Bao, Shen Nie, Kaiwen Xue, Chongxuan Li, Shi Pu, Yaole Wang, Gang Yue, Yue Cao, Hang Su, and Jun Zhu. One transformer fits all distributions in multi-modal diffusion at scale. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Georgios Batzolis, Mark Girolami, and Luca Ambrogioni. Cobit: Language modeling with bitstream diffusion, 2026. URL https://arxiv.org/abs/2605.07013.

Victor Besnier, Mickael Chen, David Hurych, Eduardo Valle, and Matthieu Cord. Halton scheduler for masked generative image transformer. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=RDVrlWAb7K.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, T. Brants, Phillip Todd Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. In *Interspeech*, 2013. URL https://api.semanticscholar.org/CorpusID:14136307.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=3itjR9QxFw.

Yuxin Chen, Chumeng Liang, Hangke Sui, Ruihan Guo, Chaoran Cheng, Jiaxuan You, and Ge Liu. Langflow: Continuous diffusion rivals discrete in language modeling, 2026. URL https://arxiv.org/abs/2604.11748.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Justin Deschenaux and Caglar Gulcehre. Language modeling with hyperspherical flows, 2026. URL https://arxiv.org/abs/2605.11125.

Antonio Franca and Alexander Tong. Hacking generative perplexity: Why unconditional text evaluation needs distributional metrics. In *ICML 2026 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2026. URL https://openreview.net/forum?id=f1gwcTDwh9.

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

Keya Hu, Linlu Qiu, Yiyang Lu, Hanhong Zhao, Tianhong Li, Yoon Kim, Jacob Andreas, and Kaiming He. Elf: Embedded language flows, 2026. URL https://arxiv.org/abs/2605.10938.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022.

Naïl B. Khelifa, Richard E. Turner, and Ramji Venkataramanan. Diffusion models observe only gradients: A geometric perspective on score matching errors, 2026. URL https://arxiv.org/abs/2606.06179.

Theodoros Kouzelis, Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Boosting generative image modeling via joint image-feature synthesis. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=i4qAfV04rZ.

Chanhyuk Lee, Jaehoon Yoo, Manan Agarwal, Sheel Shah, Jerry Huang, Aditi Raghunathan, Seunghoon Hong, Nicholas M. Boffi, and Jinwoo Kim. Flow map language models: One-step language modeling via continuous denoising, 2026. URL https://arxiv.org/abs/2602.16813.

Bingbin Liu, Sebastien Bubeck, Ronen Eldan, Janardhan Kulkarni, Yuanzhi Li, Anh Nguyen, Rachel Ward, and Yi Zhang. TinyGSM: achieving 80% on GSM8k with one billion parameters. In *The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23*, 2023. URL https://openreview.net/forum?id=ROOVUBZp8v.

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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32819–32848. PMLR, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

Patrick Pynadath, Jiaxin Shi, and Ruqi Zhang. CANDI: Hybrid discrete-continuous diffusion models. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=03KQGUvw1e.

Patrick Pynadath, Jiaxin Shi, and Ruqi Zhang. Generative frontiers: Why evaluation matters for diffusion language models, 2026b. URL https://arxiv.org/abs/2604.02718.

Gabriel Raya, Bac Nguyen, Georgios Batzolis, Yuhta Takida, Dejan Stancevic, Naoki Murata, Chieh-Hsin Lai, Yuki Mitsufuji, and Luca Ambrogioni. Information-guided noise allocation for efficient diffusion training. 2026. URL https://arxiv.org/abs/2602.18647.

Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M Rush, Yair Schiff, Justin T Chiu, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=L4uaAR4ArM.

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.

Alexander Shabalin, Simon Elistratov, Viacheslav Meshchaninov, Ildus Sadrtdinov, and Dmitry Vetrov. Why gaussian diffusion models fail on discrete data and how to prevent it?, 2026. URL https://arxiv.org/abs/2604.02028.

Dejan Stancevic, Florian Handke, and Luca Ambrogioni. Entropic time schedulers for generative diffusion models. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=EfDIApcjgI. NeurIPS 2025.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Zihao Wu, Haoming Yang, Juncheng Dong, and Vahid Tarokh. Rethinking token prediction: Tree-structured diffusion language model, 2026. URL https://arxiv.org/abs/2604.03537.

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. In *ICLR*, 2025. URL https://arxiv.org/abs/2510.01329.

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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=kUzBGEuu7w.

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, 2026. URL https://arxiv.org/abs/2510.03206.

APPENDIX

A SEMANTIC CLUSTERING OF TOKEN EMBEDDINGS

This appendix details the construction of the surjective map $\Gamma : \mathcal{V} \to \mathcal{C}$ used in Section 4.2. Γ is a spherical K-means partition of a pretrained token-embedding matrix, computed offline and frozen before training, followed by a frequency-driven relabelling of the cluster *codes* (Algorithm 1).

A.1 REQUIREMENTS ON Γ

Determinism. Proposition 1 requires $y^2 = \Gamma(y^1)$ to be a deterministic function of the tokens, for our model of the joint distribution to be a valid model of the token distribution. Γ is therefore computed once, offline, and frozen: it is a property of the tokenizer and of the pretrained embeddings, not a learned component. It is independent of the underlying model's token representation. This is what allows discarding the cluster modality at the end of sampling.

Balance. The per-position entropy of the high-level modality (clusters) is $H(y^2)$, maximal and equal to $\log n$ for equiprobable clusters. Balanced clusters are more expressive, offering a better anchor to the diffusion process.

Semantics. Balance alone is not sufficient: Section 5.3 shows that a uniformly random cluster map, balanced by construction, recovers none of the gain. What the model exploits is that tokens sharing a cluster are close in embedding space, so resolving the cluster genuinely narrows the token posterior and provides a strong anchor to the denoising process.

A.2 INPUT EMBEDDINGS

We cluster the input embedding matrix $E \in \mathbb{R}^{|\mathcal{V}| \times D}$ of a pretrained LangFlow checkpoint (Chen et al., 2026), $D = 768$ (`langflow-lm1b` for LM1B, `langflow-owt` for OpenWebText): a flow LM using the same tokenizers. LangFlow normalizes each embedding row to the sphere of radius $\sqrt{D}$, so that all information is carried by directions. We accordingly unit-normalize the rows of E and cluster on the unit sphere.[1]

TinyGSM. On TinyGSM, following Deschenaux & Gulcehre (2026); Batzolis et al. (2026), we use the SmolLM-135M tokenizer plus an added `[PAD]` ($|\mathcal{V}| = 49{,}153$). No LangFlow checkpoint shares that tokenizer. We use the input embeddings of a S-FLM (Deschenaux & Gulcehre, 2026) checkpoint, which also constrains embeddings to a sphere.

H-CoBit on OpenWebText: span averaging. On OpenWebText, CoBit does not operate on GPT-2 tokens but on `gpt2id_bpe16` *code* tokens, a second-stage BPE tokenizer over GPT-2 ids, yielding $|\mathcal{V}| = 65{,}536$ – which no pretrained model embeds. We therefore take each code token's vector to be the mean of the LangFlow-OWT embeddings of the GPT-2 span it expands to. Four code tokens (`[PAD]`, `[UNK]`, `[EOSEQ]`, `'\n'`) expand to no span and receive no vector. The same happens for SmolLM-135M's tokenizer `[PAD]` token (for TinyGSM). They are attributed to a separate cluster, of id $n - 1$, and K-means with $n - 1$ centroids.

A.3 CLUSTERING

We run Lloyd's algorithm with K-means++ initialization (`sklearn.cluster.KMeans`) on the unit-normalized token embeddings. We keep the best of $n_{\text{init}} = 10$ restarts, each capped at $T = 300$ iterations and stopped early below a relative centroid shift of $\varepsilon = 10^{-4}$.

A.4 FREQUENCY-DRIVEN CODE REBALANCING FOR H-COBIT

K-means numbers its clusters arbitrarily and that number is written directly into the cluster bits, implying potentially imbalanced per-bit marginals $p_b = \mathbb{P}(\text{bit } b = 1)$ for H-CoBit. We therefore search for a permutation σ of the cluster codes maximizing the frequency-weighted $\sum_b H(p_b)$, by

[1] We read the raw embedding parameter, which LangFlow row-normalizes at inference time.

2-opt local search over code pairs. Frequencies are estimated over a prefix of the on-disk token cache (50M / 8M tokens for OWT / TinyGSM). For OWT $n = 16$, the realized per-bit entropy goes from 3.75 to 3.99 of 4.0 (35 swaps). We only apply code rebalancing on OWT and TinyGSM.

Algorithm 1 Construction of Γ (offline, once per (tokenizer, n) pair)

Require: pretrained embeddings E, clustered vocabulary $\mathcal{V}$, clusters n, corpus cache
Ensure: frozen map $\Gamma : \mathcal{V} \to \mathcal{C}$, $|\mathcal{C}| = n$
$e_v \leftarrow E_v$, or the mean of E over v's subword span for code-token vocabularies (H-CoBit on OWT)
$\mathcal{F} \leftarrow \{v : e_v \text{ exists}\}$; $\hat{e}_v \leftarrow e_v / \|e_v\|$ for $v \in \mathcal{F}$
if reserving a structural cluster **then** ▷ OWT, TinyGSM
    $\Gamma(v) \leftarrow n - 1$ for all $v \notin \mathcal{F}$; $k \leftarrow n - 1$
else
    $k \leftarrow n$ ▷ LM1B: $\mathcal{F} = \mathcal{V}$
end if
$\Gamma|_{\mathcal{F}} \leftarrow \text{KMEANS}\big(\{\hat{e}_v\}_{v \in \mathcal{F}},\, k;\, \text{K-means++},\big)$
$f_c \leftarrow$ corpus frequency of cluster c, over a prefix of the token cache
$\tau \leftarrow \arg\max_{\tau \in \mathfrak{S}_n} \sum_b H\big(p_b(\tau; f)\big)$ by 2-opt local search ▷ codes only
return $\tau \circ \Gamma$

A.5 Hyperparameters and measured quality

Table 4 lists the settings used for all runs. The clustering is cheap: a few minutes per map on CPU.

Table 4: **Clustering hyperparameters.** Γ is computed once per (tokenizer, n) pair and frozen before training.

Parameter	Value
Embeddings	pretrained LangFlow/S-FLM input embeddings, $D = 768$
Metric	Euclidean on unit-normalized rows ($\equiv$ cosine)
Clusters n	LM1B $\{8, \ldots, 256\}$, OWT / TinyGSM $\{16, 32, 64\}$; $n = 16$ default
Initialization	K-means++, $n_{\text{init}} = 10$ restarts, best inertia
Max iterations T / tolerance ε	300 / 10^{-4} (relative centroid shift)
Seed	42
Structural cluster	reserved on OWT, TinyGSM (one of the n ids); not LM1B

Table 5: **Measured clustering quality at $n = 16$.** n_{sem} excludes the reserved structural cluster. Silhouette / DBI / CH are computed on the fitted tokens only, silhouette on a 10,000-token subsample. H_{type} is the type-occupancy entropy normalized by $\log_2 n_{\text{sem}}$.

Corpus	$\lvert\mathcal{V}\rvert$	n_{sem}	size min / mean / max	Sil. ↑	DBI ↓	CH ↑	H_{type}
LM1B	30,522	16	848 / 1908 / 4210	0.050	3.50	506	0.969
OpenWebText	65,536	15	2279 / 4369 / 7671	0.057	3.41	1300	0.978

Silhouette values are small in absolute terms (0.003–0.075 across our maps), as expected for high-dimensional embedding clustering at these n. We read them comparatively between embedding sources at fixed n, not as evidence of well-separated clusters. Qualitatively the partitions are interpretable. On LM1B, $n = 16$ separates punctuation, subword continuations, function words and concrete nouns (*head, eyes, house, hand, room, water, door, river*). Its largest cluster is the block of BERT's reserved `[unused`k`]` ids, which never occur in the corpus. Table 6 gives the OWT case.

A.6 The random-cluster control

In Section 5.3 we present an ablation replacing Γ by a map Γ_{rand} drawn uniformly at random over $\mathcal{C}^{\mathcal{V}}$, independently of the embeddings, then frozen exactly like Γ. It is matched to our clustering in every non-semantic respect. The only thing it destroys is the alignment between cluster identity and embedding geometry.

Table 6: **OpenWebText clusters at $n = 16$.** The seven most-used semantic clusters, with their share of a 35.5M-token sample and their most frequent `gpt2id_bpe16` code tokens (which may span several GPT-2 tokens). Ids are the K-means labels, prior to the code rebalancing of Section A.4, which permutes them without changing membership.

Id	Usage	Most frequent members
11	16.5%	`the`, `of`, `in`, `a`, `for`, `of the`, `in the`, `with`, `by`, `on`
12	13.0%	`to`, `that`, `is`, `'s`, `are`, `as`, `it`, `were`, `has`, `have`
9	7.9%	`,`, `and`, `, the`, `, and`, `or`, `, a`, `, but`, `-`, `;`
0	6.6%	`.`, `:`, `. The`, `.\n\nThe`, `."`, `. He`, `?`
2	6.2%	`was`, `had`, `said`, `ed`, `has been`, `made`, `did`, `left`, `says`, `called`
4	4.7%	`two`, `2`, `1`, `3`, `4`, `three`, `5`, `$`, `10`
3	4.2%	`National`, `State`, `House`, `New`, `News`, `City`, `Google`, `New York`, `University`

B Extended Background

In this section, we provide further details about the general background set up for continuous diffusion language models in Section 3, and provide the link between our general setup and existing Continuous DLM's instantiations from the literature.

B.1 Auto-Regressive *vs* Diffusion Language Models

Auto-Regressive models conditionally factorize the joint data distribution left-to-right: $p_\theta(y) := \prod_{l=1}^{L} p_\theta(y^l | y^{<l})$. This allows computing the exact likelihood of p_θ and mimics human speech, at the cost of slow iterative sampling. Diffusion models instead generate all positions in parallel through iterative refinement, with the joint distribution defined by the probability-flow Ordinary Differential Equation (ODE) (equation 3), allowing for fast parallel sampling. Here, sampling costs are governed by the sampling steps rather than only the sequence length.

B.2 Link to score matching

As presented in Section 3, we consider a denoiser $D_\theta(x_\sigma, \sigma) \approx \mathbb{E}[x_0 | x_\sigma] \in E^{L \times d}$. By Tweedie's formula, it gives direct access to the score of the marginal density p_σ,

$$\nabla_x \log p_\sigma(x) = \frac{D_\theta(x, \sigma) - x}{\sigma^2}. \tag{6}$$

B.3 From general (α_t, σ_t) parametrizations to the variance-exploding convention

Section 3 adopts the variance-exploding (VE) convention of equation 1, also used by Batzolis et al. (2026). Flow matching and variance-preserving formulations instead use an affine Gaussian path

$$x_t = \alpha_t x_0 + \sigma_t \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I_{Ld}), \qquad \alpha_t > 0, \tag{7}$$

with α_t and σ_t being monotone along the path. Setting

$$\sigma \;=\; \frac{\sigma_t}{\alpha_t}, \qquad x_\sigma \;=\; \frac{x_t}{\alpha_t} \qquad \text{gives} \qquad x_\sigma = x_0 + \sigma\epsilon, \tag{8}$$

which is equation 1.

At fixed t the map $x_t \mapsto x_\sigma$ is a bijection, so conditioning on either variable gives the same posterior, $\mathbb{E}[x_0 \mid x_t] = \mathbb{E}[x_0 \mid x_\sigma] = D_\theta(x_\sigma, \sigma)$: any network trained to predict clean data under equation 7 is a denoiser, up to input rescaling.

Furthermore, σ is a monotone function of t. A monotone change of time variable leaves the continuous ODE trajectory unchanged. It affects only the density of training noise levels which is what the schedules of Section 3.3 control.

B.4 From our general setup to the continuous DLMs literature

Tables 7 and 8 instantiate, respectively, the representation and corruption process of Section 3, and its objective, schedule and sampler, for several existing continuous DLMs.

Table 7: **Representation and corruption process.** Instantiation of the encoder $\mathcal{E}$, of the forward process, and of the decoder $\mathcal{D}$ of Section 3. The VE-equivalent noise level follows from the rescaling equation 8.

Model	Representation $x_0 = \mathcal{E}(y)$	Corruption path	VE-equivalent σ	Decoder $\mathcal{D}$
Analog Bits	$m = \lceil \log_2 \lvert\mathcal{V}\rvert \rceil$ analog bits in $\{\pm 1\}$; $d = m$	VP (cosine), $x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon$	$\sqrt{(1-\bar{\alpha}_t)/\bar{\alpha}_t}$	threshold bits at 0
CoBit	analog bits in $\{0,1\}$, $m = 15$ (LM1B) or 16 (OWT codec); $d = m$, patched to length L	VE, native: $x_\sigma = x_0 + \sigma\epsilon$, $\sigma_{\text{data}} = 1/2$	σ (native)	threshold bit probabilities, invert codec
LangFlow	learned embeddings, ℓ_2-normalized and scaled by $\sqrt{D}$; $d = D = 768$	VP γ-path, $z_\gamma = \alpha_\gamma z + \sigma_\gamma \epsilon$, $\sigma_\gamma^2 = \text{sigmoid}(\gamma)$	$e^{\gamma/2}$, i.e. $\gamma = 2\log\sigma$	arg max of predicted token probabilities
FLM, FMLM	one-hot vectors; $d = \lvert\mathcal{V}\rvert$	linear interpolant $I_t = (1-t)\,\epsilon + t\, x_1$	$(1-t)/t$	arg max per position
ELF	frozen T5-small *contextual* embeddings (512), linear bottleneck to 128	rectified flow, $z_t = t\, x + (1-t)\,\epsilon$	$(1-t)/t$	network in *decode* mode at $t = 1$, then unembedding
S-FLM	learned unit-norm embeddings, $E = \mathbb{S}$; $d = 768$	geodesic, $z_t = \text{SLERP}(z_0, z_1, \alpha_t)$, $z_0 \sim \mathcal{U}(\mathbb{S}^{d-1})$	— (not additive Gaussian)	arg max of $p^\theta_{1\lvert t}$

Analog Bits (Chen et al., 2023). It is not a language model, but it is a key towards CoBit's framework (Batzolis et al., 2026) on which we rely on extensively in this work.

CoBit (Batzolis et al., 2026). The closest model to our setup. Its denoiser outputs bitwise clean probabilities. Bits being binary, output probabilities equal $\mathbb{E}[x|x_\sigma]$, so $D_\theta(x_\sigma, \sigma) = \mathbb{E}[x_0|x_\sigma]$ holds exactly and the score follows from equation 6. It further utilizes a matched residual parametrization of the bitwise probabilities.

LangFlow (Chen et al., 2026). LangFlow uses a Flow parametrization and represents tokens via learned embeddings. Under equation 8, its time variable γ is exactly $\gamma = 2\log\sigma$. The network predicts token probabilities and is trained with cross-entropy; the denoiser from Section 3 is recovered as the embedding expectation $\hat{z}_\theta = E^\top \hat{x}_\theta$. Sampling is restricted to the deterministic ODE to allow flow-map distillation.

FLM / FMLM (Lee et al., 2026). FLM represents tokens as one-hot vectors. Hence, the posterior mean is the categorical posterior, which allows a tokenwise softmax denoiser trained with cross-entropy. FLM relies on a decoding error rate $P_e(t)$ based time reparametrization. This is very close conceptually to our information-uniform schedule in Section 3, as we approximate conditional entropy via validation loss.

ELF (Hu et al., 2026). ELF is the only model that is not applying token-level supervision along the trajectory: discretization is confined to $t = 1$, where the same network, in decoding mode, is followed by an unembedding layer. This keeps ELF close to the image-domain instantiation, allowing direct use of existing techniques from that domain. ELF also keeps an EDM-style logit-normal noise density rather than fitting an information-uniform schedule. In general, ELF is the model that departs most from our framework, and from existing continuous DLMs.

S-FLM (Deschenaux & Gulcehre, 2026). S-FLM embeds tokens on the hypersphere. Noise is injected by rotation rather than addition, along the geodesic from the clean embedding to a uniform

Table 8: **Prediction target, objective, schedule and sampler of continuous DLMs.** All models estimate the same posterior, but differ in how it is parametrized, how noise levels are allocated, and how much stochasticity the sampler injects.

Model	Network output	Objective	Noise / time schedule	Sampler, stochasticity
Analog Bits (2023)	$\hat{x}_0$ (analog bits)	MSE, $w \equiv 1$	cosine (VP)	DDIM / ancestral; SC ($p = 0.5$); asymmetric time intervals
CoBit (2026)	bit posteriors in $(0,1)^{Lm}$ via matched-filter residual $\ell_\theta = r_\theta + \mathrm{clip}\big((x_\sigma - \frac{1}{2})/\sigma^2\big)$	weighted MSE, EDM $w(\sigma) = \frac{\sigma^2+\sigma^2_{\mathrm{data}}}{\sigma^2\sigma^2_{\mathrm{data}}}$	online entropy-rate, $\pi_\alpha(u) \propto g(\sigma)\,\hat{h}_{\log}(\sigma)^\alpha$ with $\alpha = \frac{1}{2}$ and $\hat{h}_{\log} \approx e(\sigma)/\sigma^2$	DDIM PF-ODE + full-band EDM churn $\gamma_i = S_{\mathrm{churn}}/N$, entropy-gated $\lambda_{\mathrm{ent}} \approx S_{\mathrm{churn}}\pi_\alpha$; SC ($p = 0.5$, carry mode)
LangFlow (2026)	token probabilities $\hat{x}_\theta \in \Delta^{\lvert\mathcal{V}\rvert-1}$; denoiser $\hat{z}_\theta = E^\top \hat{x}_\theta$	cross-entropy (Bregman flow matching) + scheduler loss	learnable Gumbel density over γ fitted to the information gain $H'_\gamma \approx \lvert\mathrm{d}\hat{\mathcal{L}}/\mathrm{d}t\rvert$	Euler on the γ-path, deterministic by design; SC ($p = 0.25$)
FLM, FMLM (2026)	posterior $p_{1\lvert t}$ through a tokenwise softmax ($= \mathbb{E}[x_1 \mid I_t]$)	cross-entropy; FMLM: KL semigroup loss on the two-time denoiser $\delta_{s,t}$	decoding-error reparametrization $\tau(t) = 1 - \frac{\lvert\mathcal{V}\rvert}{\lvert\mathcal{V}\rvert-1}P_e(t)$	Euler PF-ODE; FMLM performs one/few-step flow-map jumps; no SC; autoguidance
ELF (2026)	clean embeddings (x-prediction), shared-weight decoding head at $t = 1$	velocity MSE ($w \propto (1-t)^{-2}$, 80%) + token cross-entropy at $t = 1$ (20%)	logit-normal over t ($P_{\mathrm{mean}} = -1.5$, $P_{\mathrm{std}} = 0.8$), at training and inference	Euler ODE, or noise re-injection ("SDE", scale γ); SC ($p = 0.5$) + training-time CFG
S-FLM 2026	posterior $p^\theta_{1\lvert t}$; tangent average $\bar{u} = \sum_v p^\theta_{1\lvert t}(v)\,\log_{z_t}(\hat{e}_v)$	cross-entropy	truncation to the high-noise band $\alpha \le \alpha^\star(\delta)$, plus adaptive refit from $\lvert\mathrm{d}\hat{\mathcal{L}}/\mathrm{d}t\rvert$ (information gain).	geodesic Euler; exact, one-sample stochastic or top-k velocity; temperature T; no SC

sample on $\mathbb{S}^{d-1}$. The marginal velocity averages the tangent directions $\log_{z_t}(\hat{e}_v)$ under the posterior. Tokens are then decoded via a greedy approach.

Limitations of the framework. Our general setting (Section 3 applies seamlessly to most existing continuous DLMs, with however 3 caveats which do not affect the construction of Section 4: *(i)* ELF's encoder is contextual, so that $x_0 = \mathcal{E}(y)$ is not obtained position-wise and $\mathcal{D} \circ \mathcal{E} = \mathrm{id}$ only up to decoder error. *(ii)* S-FLM's forward process is not a pure rescaling of equation 1, since it swaps additive for geodesic interpolation. The per-modality construction of Section 4.3 however still transfer unchanged. *(iii)* LangFlow, FLM and S-FLM train with cross-entropy rather than the regression loss of equation 2, but still estimate the same object.

B.5 Information-uniform noise schedules

The information-guided noise allocation approach, originally proposed by Stancevic et al. (2025) and Raya et al. (2026), is adopted by the recent continuous diffusion for text works in varying forms.

We adopt the following general approach following Chen et al. (2026); Batzolis et al. (2026). Let π denote the density of sampling steps over $u = \log\ \sigma$, and let $\hat{h}_{log}(u)$ be an estimator of the conditional entropy rate per unit of log noise,

$$h_{\log}(\sigma) = \frac{d}{d\log\sigma}\, H(y \mid x_\sigma)\,. \tag{9}$$

We aim to set π so that noise allocation matches information gain $\pi(u) \propto \hat{h}_{log}(u)$, and place the N solver points uniformly in the cumulative distribution function (CDF) of π, $\sigma_i = F^{-1}(1 - i/N)$

with F the CDF of π over u. This yields a grid on which each step reveals approximately equal conditional information about the tokens. In practice, $h_{log}(u)$ is approximated via validation loss after a loglinear schedule warmup phase. π is either constrained to a known parametric family such as Gumbel densities (Chen et al., 2026) or fit as an unnormalized density (Batzolis et al., 2026). We note that Lee et al. (2026) instead define $\hat{h}_{log}(u)$ as the decoding error of their diffused one hot token vectors, which results in a similar shaped reparametrization although its mechanism differs. Deschenaux & Gulcehre (2026) fit the schedule to the derivative of the loss profile and additionally truncate its support, while among the models we cover only Hu et al. (2026) keeps a fixed, image-style noise density.

B.6 Self conditioning

Self-conditioning (SC), introduced by Chen et al. (2023), feeds the denoiser its own previous clean-data estimate as an auxiliary input, so that consecutive sampling steps are no longer independent forward passes. Shabalin et al. (2026) argue that this is what prevents the trajectory from drifting away from the discrete modes of the data, which explains why SC is markedly more useful for text than for images.

At training time, this entails running a no-gradient forward pass of the model with probability p_{sc} whose output clean-data estimate x_{sc} is fed to the model in the subsequent steps. When no self conditioning is enabled, the model receives zeroed-out data estimates. In practice $p_{sc} = 0.5$ for Chen et al. (2023); Batzolis et al. (2026); Hu et al. (2026) and $p_{sc} = 0.25$ for Chen et al. (2026).

At sampling time, this entails one first step without SC, after which each previous step data estimates are fed back to the model in the next step, thus entailing no additional sampling steps.

B.7 Instantiating the hierarchical framework

In Section 4, we describe H-CDLM a general setting and apply it to CoBit for experiments in Section 5. H-CDLM can also be extended to other continuous diffusion formulations; we leave the empirical evaluation of these variants to future work.

Applying H-CDLM to a given model requires three decisions only (for $k = 2$): (i) the cluster encoder $\mathcal{E}_2$, which we take to be the model's own token encoder applied to the cluster index (Section 4.2); (ii) the loss weight λ_2 of equation 5; (iii) and the per-modality instantiation of the schedule and sampler of Section 4.3. Table 9 shows these for each model. No component of the base model changes: the cluster channel is concatenated to the state at every position, so architecture, objective family and sampler are inherited unchanged.

B.8 Evaluating Diffusion Language Models.

Evaluating Language Models is hard, because there is no single agreed-upon measure of generation quality. Generative Perplexity (GenPPL), that is the perplexity measured by an auxiliary model (typically GPT-2 Large) on the sequences generated by the model under evaluation, has been extensively used in the continuous diffusion literature as the main evaluation metric (Lee et al., 2026; Chen et al., 2026; Batzolis et al., 2026). This is preferred to Evidence Lower Bounds (ELBO) since they were shown as ill-adapted to continuous diffusion (Khelifa et al., 2026), unlike for discrete DLMs for which the ELBO is a key metric. GenPPL however requires careful reporting, since it can be artificially reduced by reducing sample entropy. Even at dataset entropy, the pertinence of GenPPL remains debatable (Franca & Tong, 2026). The standard most recent practice consists in reporting GenPPL-Entropy frontiers, which are typically increasing, with a focus on GenPPL at the dataset entropy. Franca & Tong (2026) also advocate for the use of distributional metrics, including MAUVE, although they come at a higher cost. Deschenaux & Gulcehre (2026) also introduce downstream task performance on *sudoku* and GSM8k, which has the advantage of allowing fully interpretable results.

Table 9: **Hierarchical instantiation of existing continuous DLMs.** How the $k = 2$ construction of Section 4 applies to each model, with $n = |\mathcal{C}|$ clusters and d_2 the cluster embedding dimension. Added state is reported for $n = 16$, the value we find optimal (Figure 5). The CoBit and FLM rows correspond to H-CoBit (Section 5) and H-FLM (Section **??**).

Model	Cluster encoder $\mathcal{E}_2$	Added state per position, $n = 16$	Per-modality schedule and remarks
CoBit	$\lceil \log_2 n \rceil$ analog bits appended to the token's bit patch	$+4$ bits on $m = 15/16$ (approx. $+1\%$ parameters, Section 5.1)	Own entropy-rate density π_α^2 fitted on the cluster loss; churn can be set per-modality. Encoding the cluster *inside* the token code instead of appending it underperforms (Section **??**).
LangFlow	learned cluster embedding table $E^{(2)} \in \mathbb{R}^{n \times d_2}$, concatenated per position	$+d_2$ channels on $D = 768$	A second learnable Gumbel scheduler over γ^2; the deterministic sampler leaves no churn to tune.
FLM, FMLM	one-hot over $\mathcal{C}$	$+16$ on $\lvert\mathcal{V}\rvert = 30522$	A second decoding-error curve $P_e^2(t)$, hence a second reparametrization τ_2.
ELF	extra embedding channels for the cluster, plus a second unembedding head at the decoding step	$+d_2$ channels	No information-uniform schedule to fit; the logit-normal density would have to be shifted per-modality by hand, as in image latent forcing (Baade et al., 2026).
S-FLM	second unit-norm cluster table on $\mathbb{S}^{d_2-1}$, with its own SLERP interpolation	$+d_2$ channels on $d = 768$	Own truncation threshold $\alpha_2^\star(\delta)$, which predicts the coarse-to-fine ordering.

C Our Method's Details

C.1 The joint distribution is a valid model of the data

Proposition 1 is what justified discarding the auxiliary modalities at the end of sampling. It is the text-domain counterpart of the argument underlying *latent forcing* in image diffusion (Baade et al., 2026), where pixels are recovered from jointly diffused DINOv2 features, and of the ordered-scheduler argument by Besnier et al. (2025). We restate it in our framework and give the short proof.

Proposition 1. *Let $y^1 \sim p_{\mathrm{data}}$ and let $y^j = \Gamma_j(y^1)$ for measurable maps Γ_j, $j = 2, \dots, k$. Write $y = (y^1, \dots, y^k)$ and let p denote its joint distribution. Then the first marginal of p is p_{data}. Consequently, any sampler whose output is distributed according to p yields an exact sampler for p_{data} by returning y^1, irrespective of the order in which the coordinates are resolved and of the per-modality noise schedules used to resolve them.*

Proof. By construction, $y = (y^1, \Gamma_2(y^1), \dots, \Gamma_k(y^1))$ with $y^1 \sim p_{\mathrm{data}}$, so the first coordinate of $y \sim p$ is distributed as p_{data}. Hence, if a sampler outputs $\hat{y} \sim p$, then $\hat{y}^1 \sim p_{\mathrm{data}}$. This depends only on the law of $\hat{y}$, not on the order or schedules used to produce it. □

C.2 Per-modality information uniform schedulers

A key opportunity of joint diffusion is to let the denoiser exploit different noise levels per-modality. In multimodal image diffusion, this can be achieved by *ordering* the modalities: the high level modality is kept cleaner than the pixels, so that resolved semantics guide pixel denoising (Baade et al., 2026; Kouzelis et al., 2025). In image diffusion, this requires setting an explicit ordering of the modalities' schedules, after careful sweeps over time-shift parameters.

Our framework offers a simpler alternative: training a *per-modality* information-uniform noise schedule (see Section 3.3). For each modality j, let π^j denote the density of sampling steps over $u^j = \log \sigma_j$, and let $\hat{h}^j_{log}(u^j)$ be an estimator of the conditional entropy rate per unit of log noise

$$h^j_{\log}(\sigma_j) = \frac{d}{d \log \sigma_j} H(y_j \mid x_{\boldsymbol{\sigma}}) \,. \tag{10}$$

Let $\pi^j(u^j) \propto \hat{h}^j_{\log}(u^j)$ with F_j its Cumulative Distribution Function (CDF), so that noise allocation matches information gain. For modality j, placing the N solver steps i uniformly in the CDF of π^j, $\sigma_{i,j} = F_j^{-1}(1 - i/N)$, this yields a grid on which each step reveals approximately equal conditional information about the underlying data. We refer to Appendix B for details on estimating $\hat{h}^j$. Set *per-modality*, this information-uniform noise schedule from Section 3 provides a natural candidate for the per-modality schedules. Figure 3 shows the two measured densities for $k = 2$. Cluster density is visibly right-shifted: cluster identity is a coarser variable, so information about it survives larger noise so that its entropy is released at higher σ. This leads to two sampling strategies:

Shared schedule: implicit ordering. If both modalities follow a single shared noise grid ($\sigma_{i,1} = \sigma_{i,2}$), the right shift implies that at any step along the sampling trajectory, a larger part of the cluster information than the token information is resolved. The coarse-to-fine ordering recovered by image diffusion through explicit time-shifts (Baade et al., 2026) hence emerges *implicitly*.

Coupled per-modality schedule: synchronizing modalities. Alternatively, each modality can follow its *own* information-uniform grid, coupled through a single progress variable, so that every modality traverses its own noise axis information-uniformly. By construction, this places both modalities at the same *information* quantile. The implicit ordering above is thus deliberately removed.

These two grids represent two competing hypotheses. Either *clusters should lead*, or *information should be released uniformly within each modality*. We compare these empirically in Section 5.3. By default, we adopt the coupled per-modality schedule.

C.3 Training-time noise sampling.

The distribution of $\boldsymbol{\sigma}$ in equation 5 admits two natural choices, mirroring the multi- versus single-schedule distinction of Baade et al. (2026): sampling the σ_j *independently*, which trains the model on all relative noise configurations and permits arbitrary inference trajectories; or sampling them *coupled* through a shared quantile, $v \sim \mathcal{U}[0, 1]$ and $\sigma_j = F_j^{-1}(v)$, which concentrates capacity on the trajectory actually used at inference. We use the coupled variant by default.

For a budget of N sampling steps, this yields a noise grid

$$\boldsymbol{\sigma}_i = \big(\sigma_{i,1}, \ldots, \sigma_{i,k}\big), \qquad \sigma_{i,j} = F_j^{-1}\big(1 - \tfrac{i}{N}\big), \qquad i = 0, \ldots, N. \tag{11}$$

C.4 Variations and links

Beyond token clusters. The framework accommodates arbitrary hierarchies: deeper cluster trees ($k > 2$), or sequence-level semantic variables (e.g., sentence-level summaries), the latter requiring only a modality with its own length and alignment. We evaluate the per-token, $k = 2$ instantiation and leave these to future work. Wu et al. (2026) especially consider discrete diffusion along hierarchical trees. They however find tree depth 2, that is $k = 2$, to yield the best results.

Model Dependent Parameter tuning. Hierarchical Continuous Diffusion effectively designs parallel diffusion processes that are denoised jointly. This framework is applicable to any continuous DLM. As such, each specific model's sampling mechanisms and parameters can be tuned *per-modality*, as we show for the entropy gated schedule and the stochastic churn in Section 4.3. This model specific tuning is expected to enhance the interplay between modalities. We explore varying per-modality stochastic samplers for the special case of H-CoBit, with substantial sample quality improvements (see Tables 1, 12).

Links to discrete diffusion. Our method can be seen as a continuous diffusion analog to HDLM (Zhou et al., 2025), which unmasks *high level* clusters before tokens. We draw from their clustering method, and also present a framework for k arbitrary modalities evaluated in the special case $k = 2$. In our method however, all modalities are still diffused *jointly*, despite modality-specific schedulers. HDLM is based on masked diffusion, and hence fully predicts a token's cluster before predicting the actual token.

Table 10: **Main-run settings.** All other hyperparameters follow Batzolis et al. (2026).

	LM1B	OpenWebText	TinyGSM
Tokenizer	BERT-base-uncased	`gpt2id_bpe16`	SmolLM-135M
Vocabulary $\lvert\mathcal{V}\rvert$	30,522	65,536	49,153
Sequence length L	128	1024	512
Bits/token (CoBit / H-CoBit)	$15 \to 19$	$16 \to 20$	$16 \to 20$
Steps (CoBit / H-CoBit)	1M / 600k	750k / 750k	250k / 250k
LR schedule	cosine (1M)	cosine (1M)	constant
(β_1, β_2)	$(0.9, 0.99)$	$(0.9, 0.99)$	$(0.9, 0.999)$
Weight decay	0.01	0.01	0
Clusters n / weight λ_2	16 / 0.23	16 / 0.1	16 / 0.3
Cluster embeddings	LangFlow-LM1B	LangFlow-OWT (span mean)	S-FLM

D Experimental Details

We take CoBit's training and sampling recipe unchanged (Batzolis et al., 2026), and only add the cluster modality of Section 4. CoBit and H-CoBit share every hyperparameter except the cluster bits and λ_2. Table 10 summarizes the per-dataset settings.

Architecture. A 12-layer, 12-head Diffusion Transformer with hidden size 768, SwiGLU feed-forward of width 3072, RoPE, adaLN noise conditioning, dropout 0.1, and CoBit's matched-filter residual output head (Batzolis et al., 2026). Each token's bit code is one patch, and H-CoBit appends the $\lceil \log_2 n \rceil = 4$ cluster bits to it. The denoiser is conditioned on σ^1 only, which keeps the backbone's conditioning identical to the baseline.

Optimization. AdamW, learning rate $3 \cdot 10^{-4}$, 2.5k warmup steps, gradient clipping at 1, EMA 0.9999, global batch size 512, bf16. Training noise is log-normal ($P_{\text{mean}} = -1.2$, P_{std}=1.2) with EDM weighting and σ_{data}=$1/2$ (Karras et al., 2022), and self-conditioning with p_{sc}=0.5 (Chen et al., 2023). On LM1B and OWT the learning rate follows a cosine decay over a 1M-step horizon. The 600k (LM1B H-CoBit) and 750k (OWT, both arms) checkpoints are therefore taken before the decay completes. On TinyGSM we follow CoBit's TinyGSM recipe: a constant learning rate, β_2=0.999 and no weight decay. We train on 4–8 A100-64GB GPUs. The batch size is global and there is no gradient accumulation.

Information-uniform schedule. We use CoBit's online entropy-rate estimator. A 40k-step log-linear warmup is followed by refits every 2k steps from a 128-bin loss histogram, each blended in over 10k steps. H-CoBit fits one schedule per modality.

Sampling. DDIM probability-flow ordinary differential equation with EDM churn self-conditioning (Batzolis et al., 2026), stopping at $\sigma_{\min}$=0.08. We sweep the churn γ jointly with NFE $\in \{64, \ldots, 1024\}$. The per-modality sampler churns clusters in the late window $q \in [0, 0.5]$ only, with γ_2 as given in Table 1.

Evaluation. GenPPL is GPT-2 Large perplexity over 1024 unconditional samples, which are decoded to text and retokenized. Entropy is the token unigram entropy per sample, averaged over samples, and the real-text references are 4.34 (LM1B) and 5.46 (OWT). MAUVE uses GPT-2 Large features on n=1024 samples against held-out text, and we report the mean over 6 k-means seeds (Franca & Tong, 2026). For GSM8K, following Deschenaux & Gulcehre (2026), we train conditionally on the question, keeping the prompt noise-free and applying the loss to the answer only. We sample one answer for each of the 1319 problems and score the exact match of the final numeric answer.

E Additional Ablations

E.1 MAUVE evaluation

We evaluate MAUVE at matched entropy levels and report results in table 11. At dataset entropy and above, and at matched or lower NFE, H-CoBit beats the baseline by approximately 0.04 and 0.05 on LM1B and OWT respectively. Below dataset entropy, CoBit surpasses our model.

Table 11: **MAUVE at fixed entropy levels, best budget per level.** Entropy-matched MAUVE (H_1 = $+0.02$ nats off it, H_{data} = the dataset's own, H_2 = -0.05 nats off it); cell is the best MAUVE any swept NFE reaches at that level, winning NFE in parentheses. Target entropies (nats): LM1B 4.36/4.34/4.29; OWT 5.48/5.46/5.41.

			MAUVE ↑ at fixed entropy (best NFE)		
Model	**steps**	λ_2	H_1	H_{data}	H_2
LM1B (final checkpoints)					
target H			*4.36 (H_1)*	*4.34 (H_{data})*	*4.29 (H_2)*
CoBit	1M	–	0.870 (64)	0.861 (128)	**0.697 (256)**
H-CoBit	600k	$\frac{4}{15}$	**0.900 (64)**	**0.903 (128)**	0.648 (512)
OpenWebText (final checkpoints)					
target H			*5.48 (H_1)*	*5.46 (H_{data})*	*5.41 (H_2)*
CoBit	750k	–	0.329 (256)	0.530 (128)	**0.613 (128)**
H-CoBit	750k	0.1	**0.435 (128)**	**0.580 (128)**	0.563 (128)

E.2 Per-modality Sampling Evaluation

Table 12 provides a comprehensive evaluation of per-modality sampling for H-CoBit. As detailed in Section 5.3, we test adjusting global noise s_{noise}, the churn window $\mathcal{W}$ (early or late noise half), and the churn level γ *per-modality*. Per-modality sampling adjustments allow multiple degrees of freedom. On our main benchmark, GenPPL at dataset entropy, they allow substantial gains compared to the baseline.

E.3 Training and sampling speed comparison

Table 13 provides training and sampling speeds of CoBit-S and H-CoBit. As expected due to the larger channel width of H-CoBit, we observe a small sampling and training time overhead. Figure 1 and Table 1 however shows that we match CoBit-S' quality at lower NFE, which more than compensates this speed gap.

F Generation Examples

In this section, we show examples of unconditional text generation of our best performing models on LM1B and OWT. We report text sequences corresponding to the hyperparameters that yield the best GenPPL at dataset entropy.

G Limitations and Extensions

G.1 Limitations

We study the simplest instantiation of H-CDLM, which uses a single auxiliary modality obtained from a fixed clustering of pretrained token embeddings. First, we consider only one level of granularity. Richer hierarchies spanning multiple levels, from fine-grained clusters to broad semantic categories, are a natural extension that our framework supports but that we leave unexplored. Second, our cluster assignments are fixed before training. Learning them jointly with the model could yield groupings better suited to denoising, at the cost of additional training complexity. Finally, our experiments are limited to relatively small models, reflecting the computational resources available in an academic

Table 12: **Every multimodal-churn arm on both datasets.** ΔGen-PPL: the mean % change against the control at matched entropy over the interval both arms' envelopes cover at that budget (negative is better; – where they share none). Bold is best within each dataset. Magnitude alone is a null, it must be combiend with late churning to yield improvements. Composing the global s_{noise} that helps OWT onto this already-winning recipe does not transfer to LM1B.

	Gen-PPL		Δ vs. control (%) at NFE			
Arm	**@H_{data} (NFE)**	**H reach**	**64**	**128**	**256**	**512**
LM1B: n=16**, 600k steps,** λ_2=4/15**,** $H_{\text{data}} = 4.34$						
control (shared churn)	59.2 (256)	4.356	0	0	0	0
Cluster-churn magnitude γ_2						
γ_2=0 (frozen)	57.8 (512)	4.356	-1.5	+8.2	+4.7	**-2.0**
γ_2=0.1	59.7 (256)	4.357	-3.5	+0.9	-0.8	-1.1
γ_2=0.2	58.9 (256)	4.350	-2.9	**-2.1**	-0.1	+2.0
Cluster-churn window (phase)						
early, $q\in[0.5, 1]$	62.3 (256)	4.357	+5.2	+3.3	+4.4	+7.1
late, $q\in[0, 0.5]$	51.6 (512)	4.365	-4.2	+3.7	-1.1	–
late, γ_2=0.3	51.5 (512)	4.364	**-4.4**	-0.1	-2.5	–
late, γ_2=0.4	**49.4 (512)**	4.361	-0.8	–	**-4.8**	–
Compositions						
late γ_2=0.3 + s_{noise}=1.05	56.3 (1024)	4.407	–	–	–	–
late γ_2=0.4 + s_{noise}=1.05	57.9 (1024)	**4.408**	–	–	–	–
OpenWebText: n=16**, 750k steps,** λ_2=0.1**,** $H_{\text{data}} = 5.46$						
control (shared churn)	67.8 (128)	5.645	0	0	0	0
Cluster-churn magnitude γ_2						
γ_2=0 (frozen)	72.6 (128)	5.674	+8.6	-0.3	–	–
Cluster-churn window (phase)						
late, γ_2=0.3	69.2 (128)	5.649	+3.6	-4.1	–	–
Global noise scale s_{noise} (both modalities)						
s_{noise}=1.02	71.9 (128)	5.659	-5.5	-3.3	+6.2	+1.9
s_{noise}=1.05	80.0 (128)	5.709	-10.7	-1.1	+14.5	+6.8
s_{noise}=1.10	–	**5.755**	**-13.6**	+6.3	–	–
Compositions						
late γ_2=0.3 + s_{noise}=1.05	**50.4 (256)**	5.711	-8.1	**-15.4**	–	–

Table 13: **Inference and training costs.** Both arms built and timed in one process on one A100-SXM-64GB; training is forward, backward and optimizer step with no dataloading, inference is deterministic sampling with self-conditioning at NFE 128. Per-call time is the inference unit because it is what multiplies by NFE to give generation time. Each dataset is read at the largest batch it was profiled at, where the GPU is saturated.

		Training (ms/step)			Inference (μs/call)		
Dataset	**bits/tok**	**CoBit**	**H-CoBit**	**Δ**	**CoBit**	**H-CoBit**	**Δ**
LM1B	15→19	123.5	125.8	+1.9%	292	301	+2.9%
TinyGSM / GSM8K	16→20	231.5	237.3	+2.5%	1174	1213	+3.3%
OpenWebText	16→20	244.3	250.0	+2.3%	2401	2503	+4.2%
k=64	16→22	244.3	252.2	+3.2%	2401	2520	+5.0%

setting. As for diffusion language models more broadly, whether these gains persist at larger scales remains to be established.

G.2 Extensions

Cluster guidance. The high-level channel provides a natural handle for guidance: as in multimodal image diffusion (Baade et al., 2026), classifier-free guidance can be applied on the cluster modality to strengthen semantic coherence. We leave this to future work.

Sample 1 genPPL 49.3

. [SEP] toll free : (212) 623 - 3228. [SEP] howard hit. 200 with 83 homers and 105 rbis all season. [SEP] caracas, venezuela (cnn) - - venezuela ' s president says venezuelan president hugo chavez and his venezuelan counterpart are ready to work closely with venezuelan president hugo chavez to bolster the island ' s most populous nation. [SEP] how far has it been going? 1 the big question : how does the pharmaceutical industry affect our health? [SEP] gov. tim kaine is expected to deliver his acceptance speech to the national academy of sciences as he prepares to recreate a global scam that could affect millions of

Sample 2 genPPL 49.2

in the closing stages. [SEP] " legitimate insomnia could have a negative effect on developing this efficient derivatives, " he said. [SEP] notwithstanding - goers at dumfries and galloway borough council were arrested following the incident and are being questioned by shropshire police. [SEP] it was last updated at 22. 49 gmt on sunday 17 may 2009. oxford city council. london borough of hackney, hackney. £37, 779 - £32, brewers pa inclusive. london borough of hackney. east london. £33, zaragoza - £27, 258 per annum. london borough of hackney. hackney. £34, 456 - £3

Sample 3 genPPL 49.2

called on freddie mac and fannie mae financier freddie mac to regain the government ' s 50 percent stake in fannie mae. [SEP] three men, all from bangor, have been arrested in connection with alleged sexual sex and are due to appear before preston magistrates ' court. [SEP] tracy mcgrady, the scrum - half chief executive, made a full recovery after saturday ' s clash with fulham. [SEP] for many exporters, beijing has been hailed as a scapegoat for exports. [SEP] johnson was unveiling by mr clegg ' s resignation today after the party decided not to address parliament. [SEP] kirchner

[SEP] wal - mart

Figure 6: Unconditional samples from H-CoBit (n=16 cluster bits) with per-modality churn on LM1B (600k steps; late churn window, $\gamma_c = 0.4$), the generation the "H-CoBit + PMS" row of the main table (Table 1) reports at the corpus entropy: NFE 512, γ=0.3, H=4.34 (data 4.34), genPPL 49.4 under GPT-2-large (data 53.1). The 3 samples shown are those of the 1024 generated at this setting whose own per-sample genPPL is closest to that batch-level number. Each sample is 128 model tokens, shown in full.

Sample 1 genPPL 50.4

said. “This is consistent with ways to bring energy generation centres.[26 go, adding homes attached to coal, you prototype any electric car otherwise needed.”

Both Hyderabad and T Haiata are both concerned about the development the people of northern Delhi to continue implementing skyscintention on October 29, CUK member Art decahri NakAli said.

Riana also told Karachi’s news agency that with her mother on Friday she hadjump thoroughly to the “reaster” of tyre.

The Energy and Environment Agency toured an inquiry into the Deadly tornado hits information and accused the company of tyre continuing to tape Hyderabad and T Haiata.

And theNG failed to respond to details on the matter.

Firstpost is now on WhatsApp. For the latest analysis, commentary and news updates, sign up for our WhatsApp services. Just go to Firstpost.com/Whatsapp and hit the Subscribe button<|endoftext|>A top attorney who spent the past two decades in office known for his voter job at driveway Hill, is accused of haircinding and wounding pro-commercial Country.

New Hampshire Attorney General Canterbury Hunt and City Council ex-Chief Jeff Sessions announced Monday.

Senate senators Dianne Feinstein, Majority [. . .]

Sample 2 genPPL 50.4

divided terrorist pledged Europe – al Qaeda.

About a month later Abdel Sheikh bin al-Yaminhtar was arrested, and avoided that on that occasion.

Mr Masawi’s official has not yet been arrested, but did not remain in custody L., five Anamaufie state was quoted as saying by condition of anonymity. He was carrying Horse and ammunition+ identifiable air bombs, police added.

"I’m going to say that we are angry and fortunate afraid of," Mr Masawi said: "We might blow to the embarrassment. We must return back to us and take care of us, our basket," he said.

CNN News correspondent Alan Hitsje contributed to this report.<|endoftext|>Breaking News Emails Get breaking news alerts and special reports. The news and stories that matter, delivered weekday mornings.

Aug. 12, 2011, 4:05 PM GMT / Updated Oct. 12, 2011, 12:05 PM GMT By Vindariad

House Minority Leader Thilo Pelosi (D-Calif.) said Friday afternoon that Pelosi should also be at odds because Senate Majority Leader Mitch McConnell was “almost confident” to ascertain the interests of American climate talks, in a decision that goes ahead seriously.

Rep. Richard Jontz (D-Calif.) said Pelosi were “highly concerned” and that some [. . .]

Sample 3 genPPL 50.3

’s been very aware of what he’s going to do.

ADVERTISEMENT

"Among all the ex-Millbor officials of Congress — one of whom you’re president — he has on matters," House Minority Leader Nancy Pelosi said in a telephone interview Monday. "There’s at work president about this style of what’s doing and condemning him to be president. The president-in-chief’s not always able to do what he’s going to do, and they basically do our best. But the president-in-chief’s not always going to seem to be human."<|endoftext|>Former Democratic presidential nominee Hillary Clinton Hillary Diane Rodham ClintonREAD: Cohen testimony alleges Trump knew Stone talked with WikiLeaks about DNC emails County GOP in Minnesota shares comparing Sanders to Hitler Holder: ’Time to make the Electoral College a vestige of the past’ MORE (D-N.H.), about Donald Trump’s “functional legislation.”

Sen. Paul Ryan Donald John TrumpHouse committee believes it has evidence Trump requested putting ally in charge of Cohen probe: report Vietnamese airline takes steps steps to open flights on sidelines of Trump-Kim summit Manafort’s attorneys say he should get less than 10 years in prison MORE (Clairff) Raney fork (Robert) [. . .]

Figure 7: Unconditional samples from H-CoBit (n=16 cluster bits) with per-modality churn on OpenWebText (750k steps; late churn window + global $s_{\text{noise}} = 1.05$), the generation the “H-CoBit + PMS” row of the main table (Table 1) reports at the corpus entropy: NFE 256, γ=0.25, H=5.46 (data 5.46), genPPL 50.4 under GPT-2-large (data 14.9). The 3 samples shown are those of the 1024 generated at this setting whose own per-sample genPPL is closest to that batch-level number. Each sample is 1024 model tokens; the first 1200 characters are shown and [. . .] marks the cut.