HIERARCHICAL TEXT GENERATION WITH NEXT-CONCEPT MODELING (NCM)

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion language models (DLMs) are appealing for their parallel sampling, which can enable faster generation than autoregressive models; this is increasingly important as language model inference has become a dominant computational workload. In practice, however, DLM quality degrades quickly as the number of sampling steps shrinks, every step processes the full sequence, and learning to generate in every possible order makes training expensive. We introduce **Next Concept Modeling (NCM)**, a two-stage, coarse-to-fine framework in which a multiscale residual VQ-VAE encodes text into a few levels of discrete codes and a transformer generates them one level at a time, predicting all codes within a level in parallel. At the $\sim$100M-parameter scale, **NCM** achieves a better generative perplexity on LM1B than the strongest DLM baseline while using $11\times$ fewer steps and running $21\times$ faster. On OpenWebText, it approaches the generative perplexity of the strongest diffusion baseline while running $14\times$ faster, and on XSum it matches the strongest diffusion baseline in ROUGE-L while sampling $31\times$ faster.

1 INTRODUCTION

Diffusion language models (DLMs) are the leading non-autoregressive language modeling paradigm (Austin et al., 2021; Lou et al., 2024; Sahoo et al., 2024). Their main appeal is their ability to produce many tokens at a time through iterative denoising, which can make generation substantially faster than in autoregressive (AR) models that generate one token at a time. This is becoming increasingly important, as language model inference is one of the dominant computational workloads (Kwon et al., 2023; Pope et al., 2022). DLMs are already moving from research to practice: open models such as LLaDA (Nie et al., 2025b) scale to billions of parameters, commercial systems such as Mercury (Labs et al., 2025) and Gemini Diffusion (Team et al., 2026) serve diffusion models in production, and diffusion drafters accelerate AR models through speculative decoding (Chen et al., 2026).

The promise of DLMs is parallel sampling: generating a sequence in far fewer sequential steps than it has tokens. In practice, however, current DLMs are expensive both to sample from and to train: **(i) Quality still requires many sampling steps.** In both unconditional and conditional generation tasks, DLM quality degrades sharply as the number of steps decreases (Figure 4). **(ii) Each DLM step is expensive.** Every step processes the full-length sequence, and because attention is bidirectional, the key–value cache that makes AR decoding cheap does not directly apply (Wu et al., 2026; Ma et al., 2026). **(iii) Training is inefficient.** Because masked DLMs are trained to fill in randomly masked positions, they must learn to generate the data in every possible order, not just left to right (Kim et al., 2025). As a result, they require substantially more training compute than AR models to reach a given quality (Nie et al., 2025a).

Prior work reduces the number of sampling steps mainly through few-step samplers and distillation, either by distilling a many-step teacher into a few-step student (Sahoo et al., 2025a; Deschenaux & Gulcehre, 2025) or by learning a direct map between noise levels (Lee et al., 2026). These approaches come at the cost of additional training stages on top of an already expensive model. Few-step sampling also comes at a price: distilled models tend to be less diverse than the models they are distilled from. On standard benchmarks such as OpenWebText (Gokaslan et al., 2019) and LM1B (Chelba et al., 2013), they reach a lower generative perplexity at the expense of sample

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.

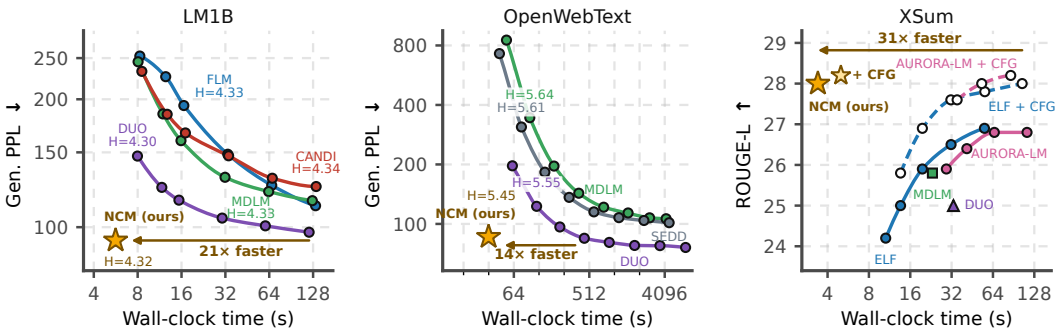


Figure 1: **Quality–speed trade-off.** Generative perplexity (Gen. PPL, lower is better) on LM1B (left) and OpenWebText (OWT, middle), and ROUGE-L on Extreme Summarization (XSum, right, higher is better), against the wall-clock time to generate 1,024 samples at batch size 16 on one NVIDIA H100 SXM 80GB GPU; H is the mean per-sample unigram entropy. On LM1B, **NCM** outperforms the strongest DLM baseline (DUO at 128 NFE) while running 21× faster. On OWT, it approaches the Gen. PPL of the strongest DLM baseline while running 14× faster. On XSum, it outperforms most baselines and is on par with the strongest one while generating 31× faster.

diversity, as measured by mean per-sample unigram entropy (Hu et al., 2026, Fig. 7). Moreover, these approaches leave the per-step cost unchanged.

In this work, we introduce **Next Concept Modeling (NCM)**, a two-stage, coarse-to-fine language modeling framework. **NCM** first learns a coarse-to-fine representation of the text with a multiscale residual VQ-VAE (van den Oord et al., 2018), which encodes a sequence into K levels of vector quantizer codes: coarse levels use only a few codes, and each finer level uses more. An **NCM** transformer is then trained to generate these levels one at a time, predicting all codes within a level in parallel, similar to next-scale prediction for images (Tian et al., 2024; Kumbong et al., 2025). **NCM**'s design answers each of the challenges above by construction. **(i)** The number of sampling steps is the same as the number of levels K, which is independent of sequence length, and **NCM** reaches high quality with good sample diversity at small K, without distillation (Figure 4). **(ii)** Each step processes only the current level, and the coarse levels are short, so for the same number of sampling steps **NCM** processes far fewer positions than DLMs such as MDLM (Sahoo et al., 2024), which process the full sequence at every step. This makes each step, and therefore generation, cheaper. **(iii)** **NCM** learns a single coarse-to-fine order rather than every possible order, which lets it train with far fewer passes over the data than masked DLMs, while still benefiting from generating global structure before local detail.

We train **NCM** models at the ∼100M-parameter scale and compare them with discrete-diffusion, flow-based and hybrid baselines, following standard evaluation practice (Sahoo et al., 2024; 2025a). For unconditional generation on LM1B (Chelba et al., 2013) and OpenWebText (Gokaslan et al., 2019), we report generative perplexity (Gen. PPL), the perplexity of the samples under a pretrained GPT-2-large (Radford et al., 2019), and the mean per-sample unigram entropy as a measure of diversity. For conditional generation, we report BLEU (Papineni et al., 2002; Post, 2018) on WMT14 De–En translation (Bojar et al., 2014) and ROUGE (Lin, 2004) on XSum summarization (Narayan et al., 2018):

- **High quality in few steps (i).** On LM1B, **NCM** achieves a better Gen. PPL in just 12 steps compared to the strongest diffusion baseline, DUO, at 128 steps (93.05 vs. 97.30), and does so at the unigram entropy of the ground-truth text (4.31; Figure 4). On OpenWebText, it approaches the Gen. PPL of the strongest diffusion baseline, DUO, while using only 16 steps, again at the entropy of the ground-truth text (5.45). On XSum, **NCM** reaches 28.0 ROUGE-L in 8 steps, a score that the strongest diffusion baseline, AURORA-LM (Liang et al., 2026), reaches only with 8× more forward passes (and classifier-free guidance).
- **Efficient inference and better quality at matched compute (ii).** Fewer, cheaper steps translate into wall-clock speed (Figure 1): **NCM** is 21× faster than the strongest baseline on LM1B, 14× faster at similar quality on OpenWebText, and 31× faster than the strongest

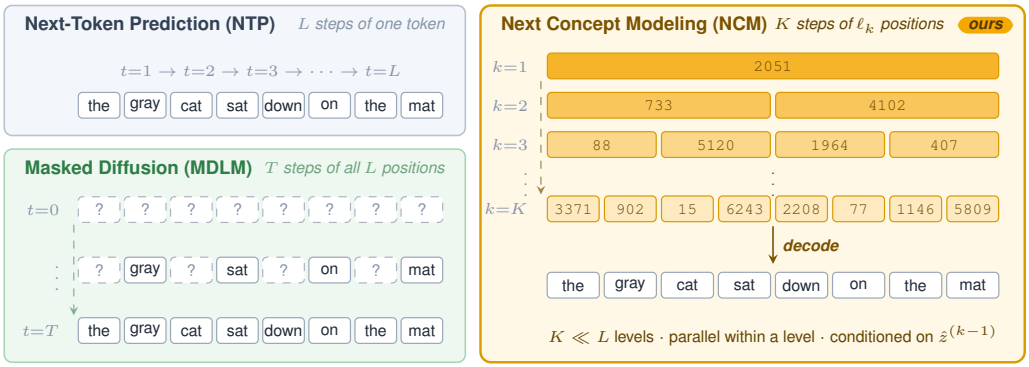


Figure 2: **Three language modeling paradigms.** Autoregressive next-token prediction generates one token per forward pass, while masked diffusion iteratively denoises multiple positions in parallel. **NCM** instead generates a hierarchy of discrete latent codes (shown as code indices) from coarse to fine, the last level holding one code per token, predicting codes in parallel within each level and conditioning level k on the reconstruction $\hat{z}^{(k-1)}$ of the earlier levels. Each step of **NCM** processes only the ℓ_k positions of its level, while each diffusion step processes all L. Since the number of levels K is much smaller than the sequence length, **NCM** requires far fewer forward passes.

diffusion baseline on XSum. Spending these savings on test-time scaling, **NCM** outperforms the baselines at matched compute (30.8 vs. 28.2 ROUGE-L on XSum; Figure 5).

- **Efficient training (iii).** **NCM** reaches these results with 12B training tokens on LM1B, $5.5\times$ fewer than the 66B used by the diffusion baselines.

2 Related Work

Discrete diffusion language models. Diffusion has emerged as the leading non-autoregressive paradigm for text. D3PM (Austin et al., 2021) introduced structured discrete diffusion via absorbing- or uniform-state Markov chains; SEDD (Lou et al., 2024) replaced the variational ELBO with a score-entropy objective on data ratios; and masked diffusion variants such as MDLM (Sahoo et al., 2024) simplified the training objective, with later models like LLaDA (Nie et al., 2025b) and GIDD+ (von Rütte et al., 2025) reaching competitive perplexity at scale. Block-diffusion variants such as BD3-LM (Arriola et al., 2025a) interpolate between autoregressive and discrete-diffusion language modeling. While these models sample many tokens in parallel, they typically require many refinement steps to achieve competitive quality (Feng et al., 2025). **NCM** is different on two axes: its forward process is *deterministic* and *hierarchical*—a coarsening operator over learned concept codes rather than masking—and its reverse chain has length $K \ll L$, so high-quality samples emerge in a handful of steps rather than hundreds.

Multiscale and next-scale generative models. **NCM**'s tokenizer extends the multiscale residual VQ paradigm developed in vision and audio. VQ-VAE (van den Oord et al., 2018) introduced learned discrete bottlenecks; VQ-VAE-2 (Razavi et al., 2019) and VQGAN (Esser et al., 2021) added multi-stage codebooks; and MaskGIT (Chang et al., 2022) demonstrated parallel decoding from VQ tokens. VAR (Tian et al., 2024) and HMAR (Kumbong et al., 2025) subsequently formalized *next-scale prediction* for images, while in audio, SoundStream (Zeghidour et al., 2021) and EnCodec (Défossez et al., 2023) popularized residual VQ across many levels. **NCM** carries this paradigm into language modeling, where coarse-level errors are far less forgiving than in pixel space; we address this with per-level residual refiners and a length-balanced VQ-VAE objective that maintains codebook utilization at coarse levels.

Hierarchical and concept-level text generation. Several recent lines explore text generation above the token level. MegaByte (Yu et al., 2023) predicts patches of bytes with a multiscale transformer; H-Net (Hwang et al., 2026) discovers two-level hierarchies end-to-end via dynamic chunking; Large Concept Models (Barrault et al., 2024) operate over sentence embeddings; and

concurrent work pursues latent-space concept reasoning (Liu et al., 2026; Qu et al., 2025) or hierarchical diffusion over semantic levels (HDLM) (Zhou et al., 2026). These approaches generally admit only one or two levels of abstraction, or rely on continuous concept spaces. **NCM** instead scales to an arbitrary number of *discrete* levels through a shared transformer trunk with lightweight per-level heads, embedding the hierarchy directly into the pretraining objective.

3 Next Concept Modeling

Here, we describe the key components of our method. **NCM** first generates a hierarchy of codes, then decodes this hierarchy into text. We train a multiscale VQ tokenizer to represent text at several levels and a transformer to predict these levels from coarse to fine. All codes within a level are predicted in parallel. The tokenizer is trained first and kept frozen while the transformer is trained second.

3.1 Formulation

Let x be a target sequence of L text tokens and let c denote its conditioning context. For downstream summarization and translation, c is the source text, whereas for unconditional generation, it is a fixed null prefix. The tokenizer maps x to discrete codes at K levels,

$$r_{1:K} = (r_1, \dots, r_K), \qquad r_k \in \{1, \dots, V_k\}^{\ell_k},$$

where $\ell_1 < \ell_2 < \cdots < \ell_K = L$. Each level is trained to encode the residual information not captured by earlier levels. Now, let $\hat{z}^{(k)}$ denote the reconstruction obtained by adding the contributions of the first k levels, with $\hat{z}^{(0)} = 0$. We call this the cumulative reconstruction. The generator predicts the levels in order:

$$p_\theta(r_{1:K} \mid c) = \prod_{k=1}^{K} p_\theta(r_k \mid \hat{z}^{(k-1)}, c).$$

Within each level, all positions are predicted in parallel:

$$p_\theta(r_k \mid \hat{z}^{(k-1)}, c) = \prod_{i=1}^{\ell_k} p_\theta(r_{k,i} \mid \hat{z}^{(k-1)}, c).$$

Once the last level is predicted, a decoder maps $\hat{z}^{(K)}$ to the token probabilities at all L positions.

3.2 Multiscale Residual Tokenization

To obtain the hierarchy, we learn a multiscale residual VQ-VAE (MS-RVQ-VAE). Its encoder maps the target text to L latent vectors, each of width D,

$$z = E_\phi(x) \in \mathbb{R}^{L \times D}.$$

Starting from $z^{(0)} = z$, we encode what remains after each reconstruction. At level k, we downsample this residual to ℓ_k positions and choose the nearest vector at each position from the level's codebook $\mathcal{C}_k = \{e_{k,1}, \dots, e_{k,V_k}\}$:

$$u_k = \mathrm{Down}_k(z^{(k-1)}), \qquad r_{k,i} = \arg\min_{j \in \{1,\dots,V_k\}} \|u_{k,i} - e_{k,j}\|_2^2.$$

We upsample the selected vectors to length L, apply a convolution refiner g_k, and subsequently add the result to the reconstruction:

$$\hat{z}^{(k)} = \hat{z}^{(k-1)} + g_k(\mathrm{Up}_k(e_k(r_k))), \qquad z^{(k)} = z - \hat{z}^{(k)}.$$

Here $e_k(r_k)$ denotes the vectors selected from the codebook. We use the convolutional refinement introduced in VAR (Tian et al., 2024), with a separate refiner and codebook at each level.

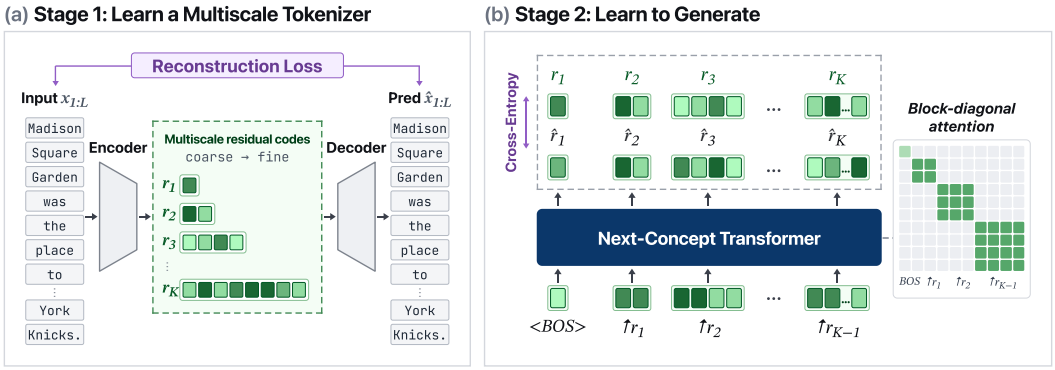


Figure 3: **NCM as a two-stage architecture.** In Stage 1, a multiscale tokenizer learns a hierarchical coarse-to-fine representation of text by encoding sequences into residual discrete scales. In Stage 2, a Next-Concept transformer sequentially generates these scales using block-diagonal attention. This process enables hierarchical text generation from coarse concepts to fine-grained details.

3.3 NCM TRANSFORMER

To predict level k, we resize $\hat{z}^{(k-1)}$ to ℓ_k positions and project it to the transformer's hidden width. The resulting positions form an input block where each position can attend to the context along with all other positions within the same block. Attention between blocks is masked, since earlier levels are already included in the cumulative reconstruction. During training, we can build all blocks from the tokenizer's codes and process them in one forward pass. During generation, we predict one level at a time and update the reconstruction after each prediction. Generation therefore requires one coarse-to-fine step per level.

Conditional generation. For the sequence-to-sequence XSum and WMT tasks, we encode the target summary or translation into multiscale codes. The source article or text to be translated enters as the prefix. Rather than introducing a separate source encoder, we reuse the frozen tokenizer embedding table to embed the prefix tokens and project them to the generator's hidden width. Within the generator, source positions attend to each other, and every target block can attend to the source. Source positions cannot attend to target blocks, and source padding is masked. At inference, only the source is given and the generator predicts the entire target hierarchy.

3.4 TRAINING

We train the multiscale tokenizer using reconstruction and residual quantization losses and update its codebooks with an exponential moving average (EMA). We then freeze the tokenizer and train the generator with weighted code cross-entropy.

Let $\tilde{z}^{(k-1)}$ denote the possibly corrupted cumulative reconstruction used for conditioning.

$$\mathcal{L}_{\mathrm{NCM}} = -\sum_{k=1}^{K} w_k \sum_{i=1}^{\ell_k} \log p_\theta(r_{k,i} \mid \tilde{z}^{(k-1)}, c), \qquad w_k = \frac{\ell_k^{-\alpha}}{\sum_{j=1}^{K} \ell_j^{1-\alpha}}.$$

With $\alpha = 0$, all code positions have equal weight, and with $\alpha = 1$, each level has equal weight. For XSum and WMT, we pretrain the generator on unpaired text before training on source–target pairs. The target length and level schedule are unchanged between pretraining and training on source–target pairs.

Code corruption. During inference, errors in early levels can propagate and change inputs to later predictions. To mitigate this issue, we corrupt the conditioning codes during training and retain the original tokenizer codes as targets.

4 EXPERIMENTS

We train a separate tokenizer and generator for each of our four datasets. For unconditional generation, we use the One Billion Word (LM1B) (Chelba et al., 2013) and OpenWebText (OWT) (Gokaslan et al., 2019) datasets, sequence-packed to lengths $L = 128$ and $L = 1024$, respectively. LM1B uses `bert-base-uncased` WordPiece tokenization, while OWT uses GPT-2 BPE tokenization. For conditional generation, we use XSum summarization (Narayan et al., 2018) and WMT14 German–English translation (Bojar et al., 2014). XSum uses source and target lengths of 512 and 48 with GPT-2 BPE, while WMT14 uses lengths of 96 and 64 with the `Qwen3-0.6B` tokenizer (Yang et al., 2025). For both conditional tasks, we pretrain the generator on unpaired text before training on source–target pairs.

We follow the evaluation protocols established by prior diffusion language model literature wherever possible. We present results on tokenizer quality (§4.1), unconditional generation (§4.2), conditional generation (§4.3), scaling behavior (§4.4), and efficiency (§4.5). We also present generated samples and showcase how **NCM** enables conditioning of prompts in any order (§4.6). In all experiments, our models are comparable in size to or smaller than the baselines, and, except in the scaling experiments, they have ∼100M parameters.

4.1 TOKENIZER ANALYSIS

Because **NCM** generates entirely within the learned code hierarchy, the tokenizer must preserve the original text accurately. We therefore train a separate tokenizer for each dataset and measure exact token reconstruction on held-out data. As shown in Table 1, every tokenizer exceeds 98.5% reconstruction accuracy. We also measure codebook utilization by encoding $16,000$ examples and, at each level, computing the percentage of entries selected at least once. Averaged across levels, utilization ranges from 61.89% to 89.31% for our tokenizers.

Dataset	Acc. (%) ↑	Util. (%) ↑
OWT	99.97	83.10
LM1B	99.86	89.31
XSum	99.79	86.20
WMT14	98.79	61.89

Table 1: **Tokenizer Statistics.** On all datasets, reconstruction accuracy is 98.5%+. Utilization is the percentage of codebook entries hit.

4.2 UNCONDITIONAL GENERATION

We evaluate **NCM** by measuring the standard generative perplexity (Gen. PPL) metric of open-ended generations on LM1B and OWT. Here, we follow the standard established in the literature and use GPT-2-large as the reference. However, Gen. PPL alone is not an adequate reflection of text quality and can be "gamed" by producing degenerate text with lower entropy (Franca & Tong, 2026). Thus, we compare with diffusion baselines that generate samples with entropy greater than or comparable to the dataset entropy. In particular, our baselines consist of both discrete DLMs such as SEDD (Lou et al., 2024), MDLM (Sahoo et al., 2024), and Duo (Sahoo et al., 2025a), and continuous DLMs, including FLM (Lee et al., 2026), CANDI (Pynadath et al., 2025), Eso-LM (Sahoo et al., 2025b), and RDLM (Jo & Hwang, 2025).

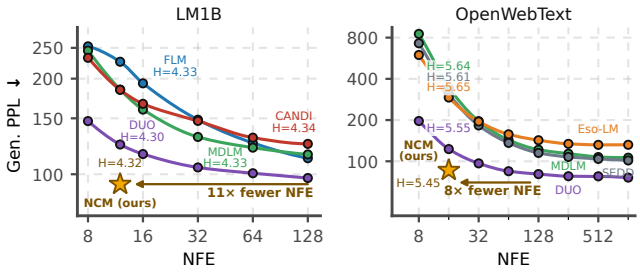


Figure 4: **Generation quality against NFE.** Gen. PPL of **NCM** and diffusion baselines on LM1B (left) and OpenWebText (right) at varying NFE. H is the unigram entropy of the generated samples. **NCM** achieves lower or equal Gen. PPL to the best baselines using 11× or 8× fewer sampling steps on LM1B and OWT, respectively.

NCM's generation quality on LM1B and OWT against diffusion baselines is displayed in Figure 4. Our **NCM** model achieves lower or equal Gen. PPL than diffusion baselines on LM1B and OWT using 11× and 8× fewer sampling steps, respectively. We also retain entropy greater than the dataset entropy. All baseline checkpoints for LM1B are from Lee et al. (2026), who trained them for the

same 1M iterations. OWT baseline checkpoints are from their respective papers. Moreover, we provide qualitative generation samples from **NCM** across Figure 7 and Appendix A.

4.3 Conditional Generation

We next test conditional sequence-to-sequence generation, evaluating machine translation on the WMT14 German–English (De–En) dataset (Bojar et al., 2014) and summarization on the XSum dataset (Narayan et al., 2018). We evaluate the translations on WMT14 using BLEU (Papineni et al., 2002; Post, 2018) and the summarizations on XSum using ROUGE-1, ROUGE-2, and ROUGE-L (Lin, 2004).

Conditional generation baselines consist of discrete DLMs, including MDLM, Duo, and E2D2 (Arriola et al., 2025b), as well as continuous DLMs such as SeqDiffuSeq (Yuan et al., 2023), ELF (Hu et al., 2026), and AURORA-LM (Liang et al., 2026). As shown in Table 2, **NCM** beats AURORA-LM without guidance on summarization using $16\times$ fewer sampling steps, at 8 versus 128.

Conditional generation examples from **NCM** on OWT are showcased in Figure 7 in addition to the conditional generations from WMT and XSum across Figure 7 and Appendix A.

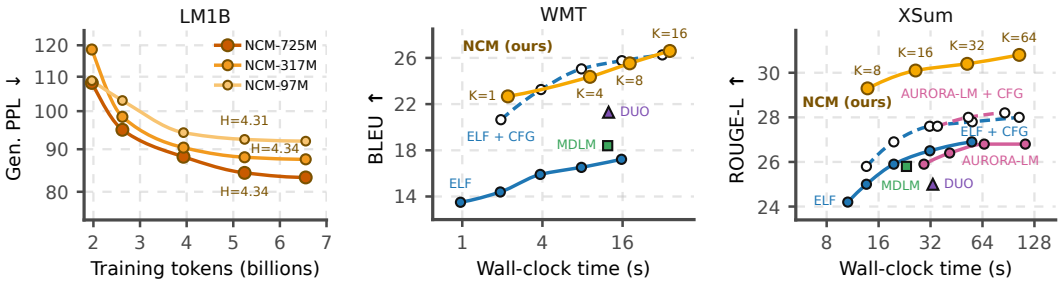
Type	Method	Size	De–En	XSum			NFE ↓
			BLEU ↑	ROUGE-1 ↑	ROUGE-2 ↑	ROUGE-L ↑	
Without guidance							
Diff.	MDLM†	99M	18.4	33.4	11.6	25.8	64
	Duo†	205M	21.3†	31.4	10.1	25.0	64
	E2D2†	99M	**24.8**	28.4	8.3	22.0	64
	SeqDiffuSeq†	–	21.3	19.3	1.7	14.1	64
Flow	ELF	140M	17.2	34.2	11.1	26.4	64
	AURORA-LM	130M	–	<u>34.7</u>	<u>12.2</u>	<u>26.6</u>	128
Ours	**NCM**	88M	<u>22.6</u>	**35.0**	**12.8**	**28.0**	**8**
With guidance or MBR decoding							
Flow	ELF†	140M	**26.4**	36.0	12.2	27.8	128
	AURORA-LM	130M	–	<u>36.3</u>	<u>13.4</u>	<u>28.6</u>	184
Ours	**NCM**	88M	23.9	35.3	13.0	28.2	**16**
	NCM + MBR	88M	<u>25.2</u>	**37.3**	**14.0**	**29.6**	128

Table 2: **Results on machine translation and summarization.** **NCM** attains the best score on all three XSum ROUGE metrics. NFE represents the number of forward passes per sample, and sizes are total generator parameters. Bold marks the best and <u>underline</u> the second-best score per column within its block (guidance on/off). Results marked † are from Hu et al. (2026), and all others are our own measurements.

4.4 Scaling

We study the scaling behavior of **NCM** on LM1B across three model sizes: **NCM**-97M, **NCM**-317M, and **NCM**-725M. These models share the same frozen tokenizer and have generator parameter counts of 97M, 317M, and 725M as their namesakes suggest. Figure 5a displays the Gen. PPL of these three **NCM** models against the number of tokens seen during training. All three fall steeply over the first 3–4B tokens and then flatten out, while unigram entropy stays between 4.31 and 4.36 throughout (dataset entropy is 4.31), so the gains do not come from less diverse samples.

The larger models reach lower Gen. PPL: at 6.6B tokens, 92.0 for the 97M model, 87.5 for the 317M model and 83.2 for the 725M model. Thus, we observe that scaling model size consistently improves the Gen. PPL versus training-token frontier. All three models need only a few billion tokens to begin converging, as opposed to the 66B on which the diffusion baselines are trained as shown in Figure 6.



(a) Scaling with training data (b) Test-time scaling on WMT (c) Test-time scaling on XSum

Figure 5: **Training and inference efficiency. (a)** The Gen. PPL of three **NCM** models (97M, 317M and 725M generator parameters) on LM1B against training tokens, evaluated at 30k, 40k, 60k, 80k and 100k training steps. The H next to each curve is the unigram entropy of the samples, averaged over these checkpoints (dataset entropy is 4.31). Scaling model size consistently improves the Gen. PPL versus training token frontier. **(b)** BLEU against wall-clock time on WMT14 De–En; ELF at 4–64 steps, with and without guidance (weight 2). Additionally, we report MDLM and DUO at 64 sampling steps, with BLEU scores from Table 2. Here, K denotes the number of MBR candidates for **NCM**. **(c)** ROUGE-L against wall-clock time to generate 1,024 summaries at batch size 16. K is the number of candidates in MBR decoding, which selects one summary from K samples. Greedy **NCM** is shown in Figure 1.

4.4.1 Faster Test-Time Scaling

Because a single **NCM** sample is cheap, **NCM** can spend additional inference compute on several samples rather than on additional denoising steps for a single one. We use minimum Bayes risk (MBR) decoding (Bertsch et al., 2023) to draw K candidates and return the one with the highest average similarity to the others. In this manner, the cost grows linearly in K, and the K candidates are generated as a single batch. Figures 5b and 5c compare **NCM** with MBR against the baselines in wall-clock time.

We observe that quality improves monotonically across the whole range. With $K = 8$ candidates (64 NFE), **NCM** already exceeds the best ROUGE-L of every baseline in Table 2, each of which spends 128–184 NFE on a single sample. On translation (WMT14), where single-sample **NCM** trails the strongest baselines, $K = 16$ candidates is sufficient to overtake ELF with classifier-free guidance. At a fixed budget of 64 NFE, spending compute on more candidates is also more effective than classifier-free guidance (29.10 ROUGE-L for MBR with $K = 8$ candidates versus 28.57 for 4 guided candidates with AURORA-LM).

4.5 Efficiency

4.5.1 Training Efficiency

A masked diffusion model must learn to generate text in many possible orders, which makes training expensive. However, **NCM** learns a single coarse-to-fine order. On LM1B, **NCM** is trained on 5.5× fewer tokens than the diffusion baselines: 12B tokens in total, including both tokenizer and generator training, compared with 66B for the baselines. Despite this smaller training budget, **NCM** achieves lower Gen. PPL than all diffusion baselines as shown in Figure 6. **NCM** is trained on 5.5× fewer tokens (12B vs. 66B) and evaluated at 12 sampling steps, while the diffusion baselines use 128. Figure 5a shows how **NCM**'s Gen. PPL evolves over training.

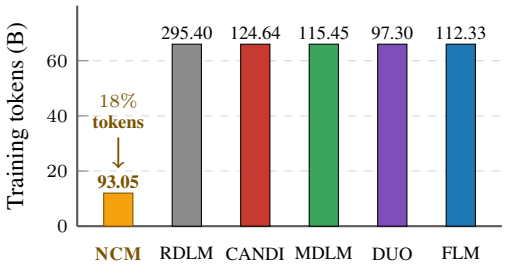


Figure 6: **Training tokens on LM1B. NCM** uses 5.5× fewer tokens and reaches a lower Gen. PPL (above each bar).

Unconditional Generation **Gen. PPL:** 55.6 | **Entropy:** 4.39

Generated: [CLS] at the far end of the series around 18 games. [CLS] "it could be good if he doesn't simply name the party that has won his past two on the other side and the corners, " he said. [CLS] "if you are in power then you can put your players down and work together and try find out the changes with which we need. [CLS] i work in the various schools or into children's school. [CLS] saunders, who was also hit after britain's most successful player in 30 years, had been for six appearances since starting 11 with an 11–14 major league record, a 28-year high [CLS]

Conditional Generation Prompt | Generated

Sample 1: An Egyptian court has ordered a probe into fresh clashes between police and protesters in Cairo. "There are strikes at the Suez Canal,". witnesses said. "It's going to be happening in the streets. It's the new regime." Those police, responding to the protests, are now getting the money out of the government, he said. . .

Sample 2: But Church explosion and billions of thought, frustration, morals, and religion had itself bound inextricably in the fabric of beauty rather than reason. The decay of society fell into the wealth of divine advancement at hand. The power of social engineering perused human godhood into a sense of direction.

Summarization **ROUGE-1:** 48.5 | **ROUGE-2:** 12.9 | **ROUGE-L:** 36.4

Context: The 33-year-old has featured only twice for the Foxes this term, having signed a new one-year deal with the Premier League newcomers in the summer. Former Blackpool forward Taylor-Fletcher scored three goals in 23 games for his parent club last season. He joined the 13th-placed Owls prior to Saturday's Championship game against Norwich City.

Reference: Leicester City striker Gary Taylor-Fletcher has joined Sheffield Wednesday on an initial month-long loan.

Generated: Sheffield Wednesday have signed Leicester striker James Taylor-Fletcher on loan until the end of the season.

Translation **BLEU:** 47.6

German: Die Bürger Pfullendorfs müssen sich auf höhere Stromkosten einstellen.

Reference: The citizens of Pfullendorf will have to adjust to high electricity costs.

Generated: The citizens of Pfullendorfs will have to adjust to higher electricity costs.

Figure 7: **Qualitative samples from NCM** The unconditional generation is from our model trained on LM1B. The conditional generation samples are with prompts (shown in blue) drawn from a held-out OWT validation set. Summarization samples are on XSum, and the German–English translation is from WMT-14. Additional samples are given in Appendix A.

4.5.2 Inference Efficiency

NCM requires one sequential generator pass per level: 12 passes on LM1B and 8 on XSum, compared with 128 sampling steps for the strongest diffusion baselines. Its passes are also cheaper because each one processes only the current level rather than repeatedly processing the full output sequence. In the end-to-end measurements of Figure 1, these two advantages make **NCM** 21× faster than DUO on LM1B and 31× faster than the strongest XSum baseline at better quality.

4.6 Qualitative Samples

Figure 7 shows **NCM** samples across unconditional and conditional settings. The conditional outputs remain anchored to their input contexts. Additional samples are provided in Appendix A.

5 Conclusion

We introduce **N**ext-**C**oncept **M**odeling (**NCM**), a two-stage framework that first learns a discrete coarse-to-fine representation of text and then trains a transformer to generate its levels in order. Predicting all codes within each level in parallel provides a short sequential generation path. On LM1B, **NCM** reaches a Gen. PPL of 93.05 in 12 forward evaluations, improving on DUO's 97.30 at 128, while generating 21× faster in our end-to-end benchmark. It achieves this result with 12B training tokens, 5.5× fewer than the diffusion baselines. On XSum, **NCM** sets a new state-of-the-art among the discrete and continuous diffusion models evaluated here, reaching ROUGE-1/2/L scores of 37.3/14.0/29.6 with MBR decoding. Scaling the generator from 97M to 725M parameters also consistently improves the quality–training-token frontier.

AI USE STATEMENT

The authors used generative AI tools to suggest wording improvements, spell-checking, and copy editing. We did not use generative AI tools to design the methodology and experiments, interpret results, or generate scientific claims. We reviewed all AI-assisted work and take responsibility for the final content of this work, including its text, claims, and artifacts.

ETHICS STATEMENT

We assert that our submission fully adheres to the ICLR Code of Ethics. **NCM** inherits the usual risks common to pretrained language models such as misinformation, low-quality automated content, impersonation, and harmful content generation, and our efficiency gains could amplify these by lowering generation cost. We do not believe **NCM** introduces qualitatively new misuse vectors beyond those of existing models at comparable scale. Our training corpora are publicly available research datasets used as released, but we recommend downstream users add domain-appropriate content filtering before any user-facing deployment.

REPRODUCIBILITY STATEMENT

Section 3 describes the model, training objective, and conditional-generation procedure, and Section 4 specifies the datasets, evaluation metrics, and baselines for each experiment. We report generator parameter counts, training-token budgets, and the number of forward evaluations alongside the corresponding results. We will release the training and inference code, the configuration files that fully specify the architecture and training hyperparameters for each dataset, tokenizer and generator checkpoints, and evaluation scripts upon acceptance of this work.

REFERENCES

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

Marianne Arriola, Yair Schiff, Hao Phung, Aaron Gokaslan, and Volodymyr Kuleshov. Encoder-decoder diffusion language models for efficient training and inference. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=5jneOToPou.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=h7-XixPCAL.

Loïc Barrault, Paul-Ambroise Duquenne, Maha Elbayad, Artyom Kozhevnikov, Belen Alastruey, Pierre Andrews, Mariano Coria, Guillaume Couairon, Marta R Costa-jussà, David Dale, et al. Large concept models: Language modeling in a sentence representation space. *arXiv preprint arXiv:2412.08821*, 2024.

Amanda Bertsch, Alex Xie, Graham Neubig, and Matthew R Gormley. It's mbr all the way down: Modern generation techniques through the lens of minimum bayes risk. In *Proceedings of the Big Picture Workshop*, pp. 108–122, 2023.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. Findings of the 2014 workshop on statistical machine translation. In Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, and Lucia Specia (eds.), *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pp. 12–58, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-3302. URL https://aclanthology.org/W14-3302/.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer, 2022. URL https://arxiv.org/abs/2202.04200.

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

Jian Chen, Yesheng Liang, and Zhijian Liu. Dflash: Block diffusion for flash speculative decoding. *arXiv preprint arXiv:2602.06036*, 2026.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=ivCd8z8zR2. Featured Certification, Reproducibility Certification.

Justin Deschenaux and Caglar Gulcehre. Beyond autoregression: Fast llms via self-distillation through time. In *International Conference on Learning Representations*, volume 2025, pp. 5007–5045, 2025.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis, 2021. URL https://arxiv.org/abs/2012.09841.

Guhao Feng, Yihan Geng, Jian Guan, Wei Wu, Liwei Wang, and Di He. Theoretical benefit and limitation of diffusion language model. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=fGBCRZQVse.

Antonio Franca and Alexander Tong. Hacking generative perplexity: Why unconditional text evaluation needs distributional metrics. *arXiv preprint arXiv:2606.08417*, 2026.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. 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. *arXiv preprint arXiv:2605.10938*, 2026.

Sukjun Hwang, Brandon Wang, and Albert Gu. Dynamic chunking for end-to-end hierarchical sequence modeling. In *International Conference on Learning Representations*, volume 2026, pp. 149273–149313, 2026.

Jaehyeong Jo and Sung Ju Hwang. Continuous diffusion model for language modeling. In *Advances in Neural Information Processing Systems*, 2025.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. *arXiv preprint arXiv:2502.06768*, 2025.

Hermann Kumbong, Xian Liu, Tsung-Yi Lin, Xihui Liu, Ziwei Liu, Daniel Y Fu, Ming-Yu Liu, Christopher Re, and David W Romero. Hmar: Efficient hierarchical masked autoregressive image generation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2506.04421.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention, 2023. URL https://arxiv.org/abs/2309.06180.

Inception Labs, Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, Stefano Ermon, Aditya Grover, and Volodymyr Kuleshov. Mercury: Ultra-fast language models based on diffusion, 2025. URL https://arxiv.org/abs/2506.17298.

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.

Jiajun Liang, Yucheng Liao, Yukang Cao, Jiazhe Wei, Ken Li, Wende Tan, Jiankun Zhang, ZY Cui, Jingkang Yang, Liucheng Guo, et al. Aurora-lm: Autoencoding unified representation for continuous-latent diffusion language modeling. *arXiv preprint arXiv:2608.02602*, 2026.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Yuliang Liu, Yunchong Song, Yixuan Wang, Kewen Ge, Alex Lamb, Qipeng Guo, Kai Chen, Bowen Zhou, and Zhouhan Lin. Next concept prediction in discrete latent space leads to stronger language models. *arXiv preprint arXiv:2602.08984*, 2026.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution, 2024. URL https://arxiv.org/abs/2310.16834.

Xinyin Ma, Runpeng Yu, Gongfan Fang, and Xinchao Wang. dkv-cache: The cache for diffusion language models. *Advances in Neural Information Processing Systems*, 38:149009–149033, 2026.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL https://aclanthology.org/D18-1206/.

Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. Scaling up masked diffusion models on text. In *International Conference on Learning Representations*, volume 2025, pp. 82974–82997, 2025a.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models, 2025b. URL https://arxiv.org/abs/2502.09992.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference, 2022. URL https://arxiv.org/abs/2211.05102.

Matt Post. A call for clarity in reporting BLEU scores. In Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor (eds.), *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6319. URL https://aclanthology.org/W18-6319/.

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

Xingwei Qu, Shaowen Wang, Zihao Huang, Kai Hua, Fan Yin, Rui-Jie Zhu, Jundong Zhou, Qiyang Min, Zihao Wang, Yizhi Li, et al. Dynamic large concept models: Latent reasoning in an adaptive semantic space. *arXiv preprint arXiv:2512.24617*, 2025.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Technical Report*, 2019.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2, 2019. URL https://arxiv.org/abs/1906.00446.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models, 2024. URL https://arxiv.org/abs/2406.07524.

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*, 2025a. URL https://openreview.net/forum?id=9P9Y8FOSOk.

Subham Sekhar Sahoo, Zhihan Yang, Yash Akhauri, Johnna Liu, Deepansha Singh, Zhoujun Cheng, Zhengzhong Liu, Eric Xing, John Thickstun, and Arash Vahdat. Esoteric language models: A family of any-order diffusion llms. *arXiv preprint arXiv:2506.01928*, 2025b.

DiffusionGemma Team, Adrien Ali Taïga, James Assiene, Daniele Calandriello, Rahma Chaabouni, João Gante, Tamara von Glehn, Nate Keating, Chris Knutsen, Martin Kukla, et al. Diffusiongemma technical report. *arXiv preprint arXiv:2608.00146*, 2026.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. URL https://arxiv.org/abs/1711.00937.

Dimitri von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=rvZv7sDPV9.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. In *International Conference on Learning Representations*, volume 2026, pp. 57027–57051, 2026.

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

Lili Yu, Dániel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. Megabyte: Predicting million-byte sequences with multiscale transformers. *Advances in Neural Information Processing Systems*, 36:78808–78823, 2023.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. Seqdiffuseq: Text diffusion with encoder-decoder transformers, 2023. URL https://arxiv.org/abs/2212.10325.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec, 2021. URL https://arxiv.org/abs/2107.03312.

Cai Zhou, Chenyu Wang, Dinghuai Zhang, Shangyuan Tong, Yifei Wang, Stephen Bates, and Tommi Jaakkola. Next semantic scale prediction via hierarchical diffusion language models. *Advances in Neural Information Processing Systems*, 38:41496–41531, 2026.

APPENDIX CONTENTS

A Additional Qualitative Samples **15**

A.1 WMT14 De–En 15

A.2 XSum 15

A.3 LM1B 16

A Additional Qualitative Samples

Figure 7 presents short samples from WMT-14 German–English, XSum, and LM1B. In this section, we give three additional samples for each of these corpora **NCM** is evaluated on, with the relevant metrics reported at the top of every sample.

A.1 WMT14 De–En

Translations of held-out WMT14 German sentences, shown against the reference translation. By sentence-level BLEU on the 3,003-sentence test set, the three samples sit at the 78th percentile, the 97th, and the median, so the typical case is represented alongside the stronger ones.

Sample 1 **BLEU:** 30.6

German: Volkswirtschaftler sorgen sich auch wegen der hohen Verbraucherverschuldung in Thailand, das diese Woche knapp aus einer technischen Rezession herausgekommen ist.

Reference: Economists are also worried about high levels of consumer debt in Thailand, which this week narrowly emerged from technical recession.

Generated: Economists also worry about high consumer debt in Thailand, which has just come out of a technical recession this week.

Sample 2 **BLEU:** 55.7

German: Außerdem erlebte er auf seiner Bahnfahrt von Titisee-Neustadt nach Freiburg die bereits unternommenen Anstrengungen für den Ausbau des öffentlichen Personennahverkehrs.

Reference: In addition to this, on his train journey from Titisee-Neustadt to Freiburg, he experienced first-hand the efforts already undertaken to develop public transport.

Generated: Furthermore, on his train journey from Titisee-Neustadt to Freiburg, he experienced the efforts already made to develop local public transport.

Sample 3 **BLEU:** 18.6

German: Samsung, Huawei und HTC stellen Handys her, die mit Googles Betriebssystem Android arbeiten, das in scharfem Wettbewerb zu den Mobilprodukten von Apple und Microsoft steht.

Reference: Samsung, Huawei and HTC all manufacture phones that operate on Google's Android operating system, which competes fiercely with Apple and Microsoft mobile products.

Generated: Samsung, BMW and HTC make mobile phones that work with Android's operating [Edit] system, which is firmly competing with Apple [Edit] and Microsoft products.

A.2 XSum

Summaries conditioned on a held-out XSum article, shown against the reference summary. The three samples are drawn from the top, the 98th percentile, and the median of the 11,233-example test set by ROUGE-1, rather than hand-picked, so that the typical case is represented alongside the best one.

Sample 1 **ROUGE-1:** 73.3 | **ROUGE-2:** 57.1 | **ROUGE-L:** 73.3

Context: The incident happened in Gerard Street, Ashton-in-Makerfield in Wigan at about 14:45 BST on Saturday. The woman, who has not been named but was in her 70s, was critically injured and died at the scene. Gerard Street was closed for a time while investigations were carried out. Police have appealed for any witnesses to contact them.

Reference: A woman has died after being hit by a lorry in Greater Manchester, police have said.

Generated: A woman has died after she was hit by a car in Greater Manchester.

Sample 2 **ROUGE-1:** 66.7 | **ROUGE-2:** 43.2 | **ROUGE-L:** 66.7

Context: Berahino, 21, was travelling at 110mph when he was stopped by Cheshire Police... He was found to have 101 mg of alcohol per 100 ml of blood — the drink-drive limit is 80 mg. North Cheshire Magistrates in Runcorn fined Berahino £3,400 and offered him a drink-driver referral course. If he completed the course, it would reduce the length of his ban to nine months.

Reference: West Brom striker Saido Berahino has been banned from driving for 12 months after pleading guilty to drink-driving.

Generated: West Brom Albion striker Saido Berahino has been banned for nine months after being caught over the drink-drive limit.

Sample 3 **ROUGE-1:** 35.0 | **ROUGE-2:** 21.1 | **ROUGE-L:** 35.0

Context: ... the heir to the British throne, who is turning 67... Their 12-day tour will take in 12 different towns and cities across Australia and New Zealand. Charles and Camilla arrive in the New Zealand capital Wellington on 4 November. They will then travel around Australia between 10 and 15 November... His son Prince William and his wife Kate proved a hit with Australians when they visited last year.

Reference: Prince Charles and his wife Camilla are to visit Australia and New Zealand in November, his office says.

Generated: The Charles and Duchess of Cambridge will arrive in Australia for their two-day tour trip ahead of New Zealand in November.

A.3 LM1B

We provide additional unconditional generations at 128 tokens from our **NCM** models trained on LM1B. The Gen. PPL and entropy values reported are the per-sample values. The special token `[CLS]` is the sentence separator of the `bert-base-uncased` vocabulary.

Sample 1 **Gen. PPL:** 39.4 | H**:** 4.29

[CLS] swine flu cases have been confirmed across the united states. [CLS] there are actually more people that see it—since you don't worry about other things. [CLS] the labor department says. [CLS] the two proposals are approved by the end of the week. [CLS] he soon said that he always felt him about health care which was not a practical issue and insisted on comprehensive reforms to improve a child's health. [CLS] the euro also added 1.33 per cent to $1.46829 against the dollar and gained 0.8 percent tob72,54 against the yen and slipped 2.4 percent barrel to [CLS]

Sample 2 **Gen. PPL:** 66.1 | H**:** 4.39

[CLS] to be welcome—the united nations security resolution that was imposed by russia earlier this month, the united states, european union and the baltic states to take sanctions against iran, whose nuclear program also is its west and nuclear programme. [CLS] they will cover the amount of state subsidies on power and equipment. [CLS] any power cuts rates will be likely on those who are still in talks over a deal late thursday, according to a government report released late tuesday. [CLS] conservative advocates have done their effort to helping people who have fallen before given to come into life while others has in particular turn them off work, he said. [CLS] more than two [CLS]

Sample 3 **Gen. PPL:** 72.0 | H**:** 4.35

[CLS] that individuals should be given an perspective on where the things are right now, and when we know things are doing well you know that the federal government should still take measures to get us back over the next three months. [CLS] "it is certainly saying, by 50 it isn't going to be put this week in charge, but this has been a ridiculous prospect.. 'i thought would say we have been very good before election," he added. [CLS] gs said the profit dropped 8 percent to £96.7 million, and revenues grew 9 percent from $10bn to £2.7m. [CLS] " [CLS]