# A TICKET FROM MARGINALS TO JOINTS: COUPLED-NOISE DISTILLATION FOR ONE-STEP BLOCK GENERATION IN DIFFUSION LANGUAGE MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Can a diffusion language model generate a coherent token block in one forward pass? Masked models already predict every position at once, but each prediction is the marginal distribution given the visible context, so the tokens can be mutually inconsistent and later steps revise those already committed. We introduce CONDOR (Coupled-Noise Distillation for One-Step Readout), trained from scratch to map different noise samples to different coherent blocks. Initially, random noise is not naturally paired with a target. Winner-take-all supervision lets different samples specialize, and self-distillation trains the one-pass output to match the refined coherent block. TinyStories experiments show diverse, coherent continuations over successive blocks, one forward pass each. Qualitative MNIST experiments show that the same approach can extend to multimodal generation, such as text-to-image and unconditional text-and-image generation.

## 1 INTRODUCTION

Standard autoregressive (AR) decoding generates one token per forward pass (Vaswani et al., 2017; Radford et al., 2019). Conditioning each token on the preceding tokens supports coherent generation, but makes decoding sequential: a continuation of $L$ tokens requires $L$ forward passes. Subsequent work aims to preserve generation quality while improving decoding efficiency. Diffusion language models (dLLMs) extend the continuation one block after another, and improve the generation efficiency by decoding each new block in fewer than $L$ refinement steps (Fig. 1). Within a block, two designs are used. Continuous-latent dLLMs denoise a continuous representation from Gaussian noise and decode it into tokens at the end. The continuous latent to be recovered must be obtained before or within the training procedure: some models take it from a frozen pretrained model (Hu et al., 2026), while others train a text VAE alongside the diffusion model (Guo et al., 2026) or learn the embeddings jointly (Li et al., 2022). Masked dLLMs, including the LLaDA series (Nie et al., 2025; 2026; Bie et al., 2026), do not require this additional continuous latent. They operate directly on discrete tokens: each step fills several tokens in parallel until no mask remains in the current block. Parallel masked predictions add a further difficulty: each is based on the shared visible context, and tokens filled in the same step cannot see each other, so the predictions can be mutually inconsistent (Gu et al., 2018; Kang et al., 2026). Later steps have to remove that inconsistency by revising the visible tokens (Bie et al., 2026; Wang et al., 2025; Huang et al., 2026). Both designs still produce one block over several steps. These limits raise our question: can a dLLM generate a coherent block in one forward pass? Especially, can it do so without a continuous latent obtained before or within training, and without a later revision of the visible tokens?

A masked dLLM already predicts every position of a block in one forward pass, yet the result need not be a coherent block. Under token-wise cross-entropy, the optimal prediction at each masked position is its marginal given the visible tokens (Ou et al., 2025), so sampling all positions at once draws from a product of marginals rather than from the joint distribution of the block. The obstacle is not the number of steps but the lack of a per-sample variable that selects one joint mode. We give a masked dLLM such a variable by adding a Gaussian noise field $\varepsilon$ to its mask embeddings. One noise sample (one *ticket*) should select one coherent block, and different tickets should select different blocks (Fig. 1d). Because noise is drawn independently of the data, the model is never told which block a ticket should map to, so this coupling must be learned. We train the dLLM from scratch to learn this



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.

coupling. CONDOR (**Co**upled-**N**oise **D**istillation for **O**ne-Step **R**eadout) combines winner-take-all (WTA) supervision (Guzmán-Rivera et al., 2012; Lee et al., 2016; Rupprecht et al., 2017; Makansi et al., 2019) with self-distillation (Fig. 2). For a prefix and the one block about to be generated, we draw $k$ noise samples. These samples then enter two branches. In the first, a ratio of positions in that block is masked and the remaining positions keep the ground truth. Each masked position adds its noise to the mask embedding before the model predicts. Only the prediction closest to the ground truth in cross-entropy is kept as the loss; the others are discarded. This symmetry breaking lets different noises learn different continuations. In the second, the whole block is masked and the $k$ different noises are applied. For each noise ticket, we first iterate over several steps, filling only part of the mask each time, and obtain a relatively coherent sequence label. The one-step prediction is trained to match it, so one forward pass can produce that coherent block. WTA assigns each noise its own continuation, and self-distillation carries that continuation into the single pass.

Our contributions are as follows.

- We introduce a from-scratch model, CONDOR. By combining WTA supervision with self-distillation, it becomes the first diffusion language model that has the capability to generate one complete coherent token block per forward pass.
- We achieve coherent block-wise language generation. Each block is committed in one forward pass, and successive blocks remain a readable continuation.
- We achieve the same one-pass commitment in a multimodal setting, both for text-to-image and for unconditional generation of text and image, showing that the method generalizes beyond text and has potential for broader multimodal generation.

All experiments are conducted at small scale; our claims concern the mechanism rather than large-scale performance.




Figure 1: Committing a $B$=8-token block. **(a)** AR, one token per pass. **(b)** Continuous-latent dLM, $n$ denoising steps. **(c)** Masked dLLM, commit then edit. **(d)** Ours, one noise ticket per pass.

## 2 RELATED WORK

**Efficiency and joint coherence trade-off in autoregressive decoding.** Autoregressive decoding spends one forward pass on each token, so the number of passes grows with the length of the continuation. Autoregressive models try to reduce that number of passes in several ways. Speculative decoding lets a lightweight model draft a few tokens; the large model checks them in one forward pass and keeps only the prefix it would itself have continued (Leviathan et al., 2023). Medusa predicts several tokens from one prefix in a single pass. The guesses cannot see each other, so the draft can conflict; verification then keeps the longest prefix consistent with the model's own

predictions (Cai et al., 2024). Jacobi decoding and Consistency LLMs fill a span of positions at once. The next round rewrites every position from the previous round's left context, until each token is the greedy choice given that context (Santilli et al., 2023; Kou et al., 2024). These methods share a generate-then-self-verify procedure, and they still require multiple steps.

**More efficient token generation with dLLMs.** Diffusion language models reduce the number of passes in another way. They extend the continuation one block after another. A block of $B$ tokens is generated in $t$ iterative steps, with $t < B$. The two families differ in how that block is generated. Continuous-latent models denoise exactly as a standard diffusion model does, from Gaussian noise to a continuous representation (Li et al., 2022; Gong et al., 2023; Lovelace et al., 2023; Hu et al., 2026; Guo et al., 2026; Yu et al., 2026; Zhou et al., 2026). That continuous representation is built before or jointly with diffusion training, and it must be decodable back into tokens. It is taken from a frozen pretrained model (Hu et al., 2026), learned by a text VAE (Guo et al., 2026), or trained jointly as token embeddings (Li et al., 2022). Because this denoising follows traditional diffusion, progressive distillation, consistency models, rectified flow, distribution matching, and MeanFlow can be expected to reduce the number of denoising steps here as well, even to one step (Salimans & Ho, 2022; Song et al., 2023; Liu et al., 2023; Yin et al., 2024; Geng et al., 2025). To our knowledge, continuous-latent language models have not yet adopted this further reduction. Masked models write discrete tokens and fill several positions in a step (Austin et al., 2021; Lou et al., 2024; Sahoo et al., 2024; Shi et al., 2024; Ou et al., 2025; Nie et al., 2025; Ye et al., 2025; Arriola et al., 2025; Bie et al., 2026; Chang et al., 2022). Parallel fills can be mutually inconsistent (Kang et al., 2026), so later work chooses which positions to commit (Wu et al., 2026), revises them, or turns them back into masks on subsequent passes (Ghazvininejad et al., 2019; Bie et al., 2026; Wang et al., 2025; Zhai et al., 2026; Huang et al., 2026; Kim et al., 2026; He et al., 2025; Schiff et al., 2026). Some methods introduce distillation from an already trained multi-step teacher to reduce the number of sampling steps (Deschenaux & Gulcehre, 2025; Sahoo et al., 2025; Hayakawa et al., 2025). The sampling schedule becomes shorter, yet still requires multiple passes, except for DiMO, which has validated one-pass generation on images (Zhu et al., 2025). CONDOR is trained from scratch, without a pretrained teacher, and is capable of committing each block in one forward pass for both text and images.

**A necessary condition for one-step coherence.** One step can commit several tokens in parallel. Those tokens form a coherent block only when the input and the output form a one-to-one or many-to-one pair: one input must select one coherent combination. Methods differ in how they form this pair. (i) VAE-like methods encode the target $x$ into a high-dimensional latent $z$ and decode $z$ back into $x$, so $z$ and $x$ form the pair inside each training example (Kingma & Welling, 2014). An earlier discrete-latent decoders (Kaiser et al., 2018), FlowSeq (Ma et al., 2019), LaNMT (Shu et al., 2020), VADD (Xie et al., 2026), and Cola DLM (Guo et al., 2026) are trained in this way. The pair fails when the decoder ignores the latent, as a conditional generator can ignore an input noise code, or when the posterior collapses onto the prior (Isola et al., 2017; Zhu et al., 2017; Bowman et al., 2016). (ii) Flow models form the pair from the initial noise. Rectified flow (Liu et al., 2023) and MeanFlow (Geng et al., 2025) carry one noise sample to one result in a single forward pass. ELF (Hu et al., 2026) is a dLLM trained with flow, so it has the same one-pass potential, but it does not take that step. Consistency distillation (Song et al., 2023) and DMD (Yin et al., 2024) distill a diffusion model so that one noise sample yields one result in a single forward pass. DiMO (Zhu et al., 2025) distills a masked diffusion model, taking a random token initialization as the noise, so that one initialization yields one result in a single forward pass. (iii) Under one condition, IWAE (Burda et al., 2016) uses different inputs and obtains different predictions of the same label. Every prediction remains in the loss. Instead of token-level cross-entropy, it multiplies the token probabilities into the sentence probability. This product represents the joint distribution of the continuation, rather than the per-position marginals. The weight on each input is a choice: the inputs can be averaged uniformly, or the weights can be tilted by the loss. (iv) Winner-take-all (WTA) losses (Guzmán-Rivera et al., 2012; Lee et al., 2016; Rupprecht et al., 2017; Makansi et al., 2019) also draw several inputs, but supervise only the lowest-loss one, so different inputs specialize to different outputs, as in Social GAN (Gupta et al., 2018) and LoRA-MCL (Letzelter et al., 2026). CONDOR uses self-distillation as well as WTA. Each noise is self-distilled into a different label, so self-distillation also strengthens learning toward different labels.

# 3 METHOD

We first define $f_\theta$, a block-wise dLLM based on a Transformer. A prefix $c$ and a continuation block of $B$ tokens are the input of $f_\theta$. Attention inside a block is bidirectional. Across blocks, a token attends only to its own block and to earlier blocks (Arriola et al., 2025). Each block to be predicted has ground truth $x$, and $x_i$ is its token at position $i$. $z$ is the current input of that block, formed from $x$ and a mask position set $A$:

$$z_i = \begin{cases} \texttt{[M]} & i \in A, \\ x_i & i \notin A. \end{cases} \tag{1}$$

The masked fraction may be the whole block. Each visible token keeps its own embedding. Each mask embedding becomes a noised version,

$$\tilde{e}_i = e_{\texttt{[M]}} + \sigma \cdot \mathrm{RMS}(e_{\texttt{[M]}}) \cdot \varepsilon_i, \qquad i \in A, \tag{2}$$

where $\varepsilon \in \mathbb{R}^{B\times d}$ with $d$ the embedding dimension, $e_{\texttt{[M]}}$ is the mask embedding, and $\sigma$ scales the noise by $\mathrm{RMS}(e_{\texttt{[M]}})$. Each row is drawn $\varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, I_d)$, or one vector $u \sim \mathcal{N}(0, I_d)$ is copied onto every $i \in A$. $\bar{x}_i$ is the vector of logits that $f_\theta$ predicts at position $i$:

$$\bar{x}_i = f_\theta(c, z, \varepsilon)_i, \qquad i \in A. \tag{3}$$

CONDOR has two training tasks (Fig. 2). (i) Under a partial mask, $k$ noises each produce a prediction. Every prediction is scored against the ground truth, but a gradient is sent only along the lowest-loss noise. The rest are dropped. (ii) Under a full mask, those $k$ noises are self-distilled to $k$ labels, one label per noise, and the one-pass prediction for a noise is trained to match its label.

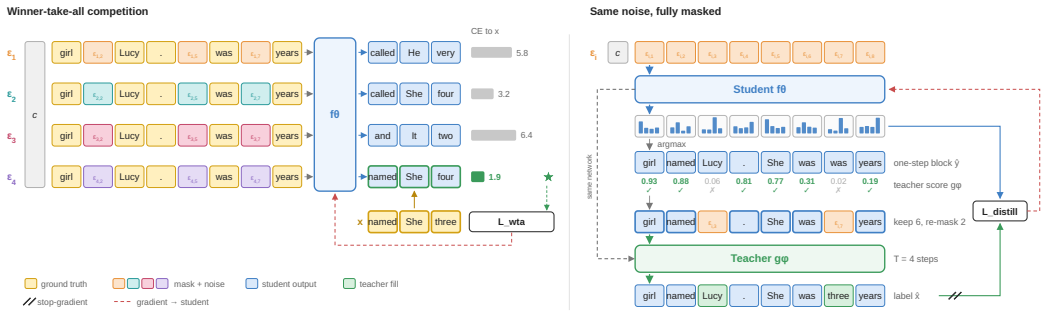



Figure 2: Training CONDOR. Left: the winner-take-all task. Right: self-distillation.

## 3.1 WINNER-TAKE-ALL (WTA) ON A PARTIAL MASK

On a fixed $(c, z)$ we draw $k > 1$ noises $\varepsilon^{(j)} \in \mathbb{R}^{B\times d}$, $j = 1, \ldots, k$. For noise $j$, the predicted logits are $\bar{x}_i^{(j)} = f_\theta(c, z, \varepsilon^{(j)})_i$. $\mathrm{CE}_j$ is the cross-entropy of these logits against the ground-truth tokens on the masked positions. $j_{\text{win}}$ is the index of the noise with the lowest $\mathrm{CE}_j$, and $\mathcal{L}_{\text{wta}}$ is that cross-entropy divided by $|A|$:

$$\mathrm{CE}_j = \sum_{i\in A} \mathrm{CE}\big(\bar{x}_i^{(j)}, x_i\big), \qquad j_{\text{win}} = \arg\min_j \mathrm{CE}_j, \qquad \mathcal{L}_{\text{wta}} = \frac{1}{|A|}\,\mathrm{CE}_{j_{\text{win}}}. \tag{4}$$

Only the winner sends a gradient. The other noises, the *losers*, are left free to stand for other continuations. Besides WTA, we ablate two other losses on these same noises, plain cross-entropy and IWAE:

$$\mathcal{L}_{\text{plain}} = \frac{1}{k|A|}\sum_{j=1}^{k} \mathrm{CE}_j, \qquad \mathcal{L}_{\text{iwae}} = -\frac{1}{|A|}\log\left(\frac{1}{k}\sum_{j=1}^{k} e^{-\mathrm{CE}_j}\right). \tag{5}$$

### 3.2 SELF-DISTILLATION ON A FULL MASK

Self-distillation uses two networks, student $f_\theta$ and teacher $g_\phi$. For a noise $\varepsilon$, the student and the teacher jointly form the distillation target $\hat{x}$. The parameters of $g_\phi$ can be the same as the current student, as in Fig. 2, or an exponential moving average of the student.

The block is fully masked, so $A = \{1, \dots, B\}$ and every position of the current input $z$ is a mask. One forward of the student returns logits. These logits are greedily discretized into the one-pass tokens. The probability of token $\bar{y}_i$ on this forward is

$$\bar{x}_i = f_\theta(c, z, \varepsilon)_i,\ \bar{y}_i = \arg\max_v f_\theta(c, z, \varepsilon)_{i,v},\ p_i = \mathrm{softmax}(\bar{x}_i)_{\bar{y}_i},\ \ i = 1, \dots, B. \tag{6}$$

The one-pass tokens $\bar{y}$ are not necessarily a good continuation, so a kept ratio $\rho$ of them stay, $\mathrm{round}(\rho B)$ positions, and the rest are masked again and refilled. The kept positions may be chosen from the student's $p_i$, the teacher's $p_i$, or at random. When the student is the judge, $p_i$ in Eq. (6) is sorted, the $\mathrm{round}(\rho B)$ positions of the block with the highest $p_i$ are kept, and the rest are remasked. When the teacher is the judge, position $i$ is rescored by the teacher on an input that keeps the other tokens of $\bar{y}$ and replaces position $i$ by the original noised mask $\tilde{e}_i$. This input is written $\bar{y}_{\neg i}$. $g_\phi$ returns logits, and $p_i$ is the softmax probability of the proposed token:

$$p_i \ = \ \mathrm{softmax}\big(g_\phi(c, \bar{y}_{\neg i}, \varepsilon)_i\big)_{\bar{y}_i}. \tag{7}$$

Then, by this $p_i$, the $\mathrm{round}(\rho B)$ highest positions are kept and the rest are remasked. The teacher is then used to fill the remaining masks. In each step, the teacher commits the $(1-\rho)B/T$ most confident argmax tokens until no mask remains, forming the filled block $\hat{x}$.

For each noise $j$, which is the same noise as in the WTA task, $\bar{x}^{(j)}$ is trained toward $\hat{x}^{(j)}$. The training loss is

$$\mathcal{L}_{\mathrm{distill}} = \frac{1}{kB} \sum_{j=1}^{k} \sum_{i=1}^{B} \mathrm{CE}\big(\bar{x}_i^{(j)}, \hat{x}_i^{(j)}\big), \quad \mathcal{L} = \mathcal{L}_{\mathrm{wta}} + w(t)\, \mathcal{L}_{\mathrm{distill}}, \tag{8}$$

with $w(t)$ increased early in training and then kept at 1.

### 3.3 SEVERAL BLOCKS, AND IMAGES

For block-wise generation inference, the model generates one block in a single forward pass, appends it to the prefix, and samples fresh noise for the next block. Each block is conditioned on the given prefix and all previously generated blocks. Repeating this procedure produces longer sequences while retaining one forward pass per block.

The same framework also supports multimodal generation, including text-to-image generation and unconditional generation of text and Images. In text-to-image generation, the text serves as the prefix for the image blocks. In unconditional generation, the model first generates text without a supplied text prefix, then generates the image conditioned on that text. Images are divided into patches, with each patch represented as a token. The model predicts continuous pixel values within each patch, and mean squared error replaces cross-entropy in both the winner-take-all and self-distillation objectives for image prediction. For image self-distillation, continuous patch predictions do not provide confidence scores for ranking patches. We therefore randomly select which patches of the one-pass prediction to remask, and randomly select which remaining masked positions to fill at each teacher step.

## 4 EXPERIMENTS

### 4.1 EXPERIMENTAL SETUP

We evaluate CONDOR on four tasks spanning text and image generation, using a single forward pass per block throughout. *Single-block text continuation* generates an eight-token continuation conditioned on an eight-token prefix. *Multi-block text continuation* extends a text prompt with eight consecutive blocks of eight tokens each. *Word-to-image* generates a digit image conditioned on its English class name (*zero–nine*). *Unconditional* generates a digit word followed by its corresponding image.

**Data.** Both text tasks use TinyStoriesV2-GPT4 (Eldan & Li, 2023). For *Single-block text continuation*, each window is the first 16 GPT-2 tokens starting at a sentence boundary. For *Multi-block text continuation*, training samples begin at document boundaries. We tokenize the stories with the LLaDA tokenizer, group the tokens into blocks of eight, and pack them into sequences of up to $1024$ tokens. Both image tasks use MNIST, with each $28 \times 28$ image partitioned into sixteen $7 \times 7$ patches.

**Models and training.** Table 1 summarizes core experimental settings. All CONDOR models are trained from scratch with AdamW at base learning rate $3 \times 10^{-4}$; defaults are $k = 4$ and $\sigma = 0.5$. Additional implementation details are given in Appendix A.1.

Table 1: Core experimental settings for CONDOR. The image column applies to both image tasks.

| | Single-block text continuation | Multi-block text continuation | Image generation |
|---|---|---|---|
| Architecture | Bidirectional Transformer | Block-causal Transformer | Bidirectional Transformer |
| Parameters | 17.6M | 0.5B | 4.77M |
| Layers / hidden width / FFN width | $6/256/1024$ | $20/1024/2048$ | $6/256/1024$ |
| Attention heads (query / key-value) | $8/8$ | $8/2$ | $8/8$ |
| Total updates | 150,000 | 16,000 | 50,000 |
| Global batch size | 512 | 32 | 512 |
| Teacher | EMA | Current student | Current student |
| Full-mask winner treatment | Winner excluded | Winner GT | Winner GT |
| Keep ratio $\rho$ / fill steps $T$ | $0.5/4$ | $0.75/2$ | $0.75/4$ |
| Retention | Confidence | Confidence | Random |
| Filling order | Confidence | Confidence | Random |
| Noise across positions | Independent | Shared within each block | Independent |

The single-block basic setting is a comparison baseline, not the best configuration. In the row *Full-mask winner treatment*, *winner* is the noise already selected by partial-mask WTA. The partial-mask and full-mask tasks use the same set of noises, so that winner is handled inside the full-mask branch: *Winner excluded* drops it from the full-mask loss, *Winner GT* trains it there against ground truth, and *Winner distilled* trains it there against the self-distillation result. The remaining noises use self-distillation results.

**Evaluation.** For fair comparisons, directly compared methods use matched evaluation examples and scoring protocols within their respective task settings. Each example provides an initial text prefix; each generated block contains $8$ tokens. *Validity* is the fraction of blocks judged grammatically and contextually acceptable by GPT-6-Astra; natural truncation is allowed, earlier errors do not automatically invalidate the current block. For $m$ generated blocks per example at a given block position, *Distinct* and $D_m^{\text{valid}}$ are the mean numbers of distinct strings and distinct valid strings per example, respectively, including examples with no valid block; $d_{\text{all}} = \text{Distinct}/m$. *Uniqueness* is the ratio of distinct valid strings to valid blocks per example, averaged over examples with at least one valid block. *gen-PPL* is GPT-2-large perplexity of generated tokens given the conditioning prefix. $\text{MI}_{16}$ measures whether a new noise sample changes the predictive distribution at the $8$ token positions of the same prefix. At each position it is the entropy of the softmax averaged over $16$ noise draws minus the mean entropy of those $16$ softmaxes, and $\text{MI}_{16}$ sums these eight values, in nats. The judging rubric appears in Appendix B.

For *Single-block text continuation*, we use the first $256$ validation examples, each with an $8$-token prefix, and generate $16$ alternative $8$-token blocks per example. Validity, $D_{16}^{\text{valid}}$, and Uniqueness use the first $32$ examples ($512$ judged blocks); Distinct, gen-PPL, and $\text{MI}_{16}$ use all $256$ examples. For *Multi-block text continuation*, we use $4$ evaluation examples and run generation $4$ times per example, each run covering $8$ successive $8$-token blocks. *Free continuation* extends the initial prefix with earlier generated blocks; *GT prefix* uses the ground-truth prefix at each block position. Both settings use fresh noise for each block. We report Validity, $d_{\text{all}}$, and Uniqueness at each block position over $4 \times 4 = 16$ generated blocks, grouped by example.

To test how inference noise affects the continuation, we hold the trained model and the text prefixes fixed and change only the noise (Table 3). $r_0$ is the root-mean-square noise norm during training. The radius sweep keeps each noise direction and sets the norm to a multiple of $r_0$. Automatic scores use $128 \times 64$ continuations, $64$ draws for every one of $128$ prefixes. $D_{64}$ is the mean number of distinct

strings among the 64 draws of one prefix. Judged Validity and $D_{16}^{\text{valid}}$ use $32 \times 16$ continuations, 16 draws for every one of the first 32 prefixes. The angle sweep keeps the norm at $r_0$ and sets the angle to a reference noise on the same prefix. The reference output is the block generated from that reference noise. Within each reference group, pairwise token disagreement is the mean fraction of token positions that differ between output pairs, and reference change is the fraction of blocks that differ from the reference output in any token. Conf. is the mean model probability of the generated tokens.

### 4.2 SINGLE-BLOCK TEXT CONTINUATION

Table 2: Single-block text continuation results. * assigns the same noise vector to every masked position in the block, in both training and inference. Other rows draw an independent noise vector at each masked position. The 30k-update CONDOR row is the wall-clock alignment with the no-self-distillation runs.

| Method / variant | NFE | Validity (%) ↑ | $D_{16}^{\text{valid}}$ ↑ | Uniqueness ↑ | Distinct ↑ | gen-PPL ↓ | $\text{MI}_{16}$ |
|---|---|---|---|---|---|---|---|
| *No self-distillation* | | | | | | | |
| Plain CE | 1 | 9.0 | 0.25 | 0.46 | 2.6 | 285.0 | 0.02 |
| Plain CE* | 1 | 6.4 | 0.12 | 0.14 | 2.2 | 316.7 | 0.01 |
| IWAE | 1 | 4.5 | 0.66 | 0.95 | 13.7 | 249.6 | 2.17 |
| IWAE* | 1 | 6.2 | 0.72 | 0.84 | 12.7 | 243.5 | 2.15 |
| WTA, $k = 4$ | 1 | 9.2 | 1.19 | 0.90 | 14.3 | 242.1 | 2.38 |
| WTA, $k = 4$* | 1 | 5.3 | 0.72 | 0.92 | 13.9 | 242.4 | 2.37 |
| WTA, $k = 16$ | 1 | 18.4 | 2.72 | 0.96 | 15.5 | 138.4 | 6.05 |
| WTA, $k = 4$, 8 passes | 8 | 97.3 | 14.91 | 0.96 | 15.1 | 20.7 | — |
| DiMO | 1 | 0.0 | 0.00 | — | 16.0 | 2702.5 | 13.45 |
| CONDOR | 1 | 81.6 | 11.31 | 0.87 | 13.6 | 24.5 | 7.66 |
| CONDOR* | 1 | 82.2 | 9.69 | 0.75 | 11.9 | 24.1 | 6.95 |
| CONDOR, 30k updates | 1 | 60.4 | 7.38 | 0.80 | 12.3 | 36.2 | 5.62 |
| Keep ratio $\rho = 0.25$ | 1 | 76.2 | 10.97 | 0.91 | 15.0 | 34.9 | 9.60 |
| Keep ratio $\rho = 0.75$ | 1 | 88.3 | 11.50 | 0.82 | 12.9 | 20.4 | 6.94 |
| Keep ratio $\rho = 0.75$* | 1 | 76.4 | 8.88 | 0.75 | 11.7 | 27.8 | 8.13 |
| $k = 8$, global batch 256 | 1 | 76.6 | 11.81 | 0.96 | 15.4 | 29.0 | 9.80 |
| $k = 16$, global batch 128 | 1 | 74.0 | 11.72 | 0.99 | 15.9 | 41.4 | 11.41 |
| Independent target noise $\varepsilon'$ | 1 | 90.0 | 1.38 | 0.10 | 1.7 | 15.6 | 0.37 |
| Retention: student confidence | 1 | 81.8 | 11.84 | 0.90 | 14.5 | 28.0 | 8.34 |
| Winner distilled | 1 | 83.6 | 11.94 | 0.90 | 13.9 | 22.4 | 8.11 |
| Winner distilled* | 1 | 81.1 | 10.06 | 0.78 | 12.2 | 24.9 | 8.08 |
| Teacher: frozen WTA | 1 | 90.8 | 11.31 | 0.78 | 11.6 | 21.9 | 4.81 |
| Teacher: current student | 1 | 82.2 | 11.16 | 0.85 | 14.3 | 24.9 | 8.74 |
| Teacher: current student* | 1 | 85.2 | 9.91 | 0.74 | 11.9 | 25.3 | 7.27 |
| Teacher: current student, $\rho = 0.75$, Winner GT | 1 | 79.3 | 9.78 | 0.79 | 12.6 | 22.7 | 4.69 |
| Teacher: current student, $\rho = 0.75$, Winner GT* | 1 | 80.3 | 8.22 | 0.67 | 11.0 | 37.3 | 4.94 |
| Teacher: current student, $\rho = 0.75$, Winner excluded* | 1 | 84.4 | 9.56 | 0.72 | 10.6 | 34.4 | 5.65 |
| Teacher: current student, $\rho = 0.75$, Winner distilled | 1 | 79.1 | 8.97 | 0.74 | 11.5 | 26.9 | 4.17 |
| Teacher: current student, $\rho = 0.75$, Winner distilled* | 1 | 93.9 | 11.62 | 0.77 | 12.3 | 26.3 | 6.02 |

Table 2 compares single-block text continuation methods. Following DiMO (Zhu et al., 2025), the text run uses the Plain CE model as the frozen teacher and distills it into a one-step generator. This adaptation produces no valid outputs in the judged evaluation (Appendix A.2). The results support three observations. (1) Without self-distillation, Plain CE, IWAE, and WTA all underperform CONDOR in one-pass Validity and valid-output diversity. The 30k-update CONDOR row aligns training time with those 150k-update baselines: self-distillation slows each update, so the schedule stops at 30k updates (2.9 GPU-hours, within the 5.3 GPU-hours of WTA at $k = 4$ and the 8.3 GPU-hours at $k = 16$; Appendix A). On this budget, one-pass Validity is $60.4\%$ and $D_{16}^{\text{valid}}$ is 7.38, above Plain CE, IWAE, and one-pass WTA. The single-block case study qualitatively supports this finding for WTA versus CONDOR (Table 5, top). (2) No output collapse is observed when self-distillation targets are generated by the current student, an EMA of the student, or a frozen WTA model. (3) Ablations of the keep ratio, token-retention rule, competition size, and full-mask winner

treatment identify the best tested one-pass Validity of $93.9\%$: current student as teacher, $\rho = 0.75$, noise shared across tokens, and *Winner distilled*.

Holding the trained model and the text prefixes fixed, we change only the inference noise (Table 3). Under that probe, a larger angle from the reference noise increases the fraction of outputs that differ from the reference output, while gen-PPL stays stable (Table 3). A larger norm produces more distinct outputs, but an excessive norm reduces Validity. Both the direction and the magnitude of the noise affect the continuation.

Table 3: Inference-noise norm and direction for the default seed-0 student.

| *Fixed radius* | | | | | | *Fixed angle, norm $r_0$* | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Automatic: $128 \times 64$ | | | Judged: $32 \times 16$ | | Within each reference cone | | | | | |
| Radius | $D_{64}$ | Conf. (%) | gen-PPL ↓ | Validity (%) | $D_{16}^{\text{valid}}$ | Angle | $D_{64}$ | Pairwise token disagreement (%) | Reference change (%) | Conf. (%) | gen-PPL ↓ |
| Original Gaussian | 43.55 | 75.9 | 25.23 | 83.2 | 11.34 | | | | | | |
| $0 \times r_0$ | 1.00 | 71.2 | 21.67 | 71.9 | 0.72 | $0°$ | 1.00 | 0.0 | 0.0 | 75.2 | 24.82 |
| $0.25 \times r_0$ | 27.48 | 73.6 | 22.01 | 83.2 | 8.91 | $5°$ | 4.11 | 8.9 | 27.3 | 75.3 | 24.76 |
| $0.5 \times r_0$ | 36.95 | 75.4 | 23.46 | 84.2 | 10.59 | $15°$ | 11.48 | 23.4 | 54.2 | 75.6 | 24.36 |
| $0.75 \times r_0$ | 41.10 | 75.9 | 24.50 | 84.6 | 11.25 | $30°$ | 23.03 | 39.7 | 75.2 | 75.7 | 24.46 |
| $1 \times r_0$ | 43.55 | 75.9 | 25.22 | 82.8 | 11.28 | $45°$ | 32.23 | 50.8 | 86.0 | 75.9 | 24.74 |
| $1.25 \times r_0$ | 46.55 | 75.3 | 26.16 | 84.0 | 11.75 | $60°$ | 38.75 | 58.0 | 92.0 | 76.0 | 24.59 |
| $1.5 \times r_0$ | 49.53 | 74.1 | 27.84 | 79.5 | 11.28 | $90°$ | 43.80 | 62.8 | 97.2 | 76.1 | 24.79 |
| $2 \times r_0$ | 57.52 | 66.7 | 42.80 | 61.7 | 9.53 | $120°$ | 38.72 | 57.3 | 99.0 | 76.0 | 25.03 |
| $3 \times r_0$ | 64.00 | 41.5 | 1142.27 | 0.8 | 0.12 | $150°$ | 22.88 | 39.7 | 99.6 | 76.0 | 24.99 |
| | | | | | | $180°$ | 1.00 | 0.0 | 99.8 | 75.8 | 25.44 |

### 4.3 BLOCK-WISE TEXT GENERATION

We evaluate whether the $0.5$B CONDOR model can extend generation across $8$ successive $8$-token blocks (Table 4). Uniqueness increases across blocks in both Free continuation and GT prefix. Free-continuation Validity declines over the eight blocks, while GT-prefix Validity is largely retained.

Table 4: Block-wise generation with the $0.5$B student at step $16{,}000$. All values are percentages, based on $4$ evaluation examples with $4$ generation runs per example. Validity scores the current block, not the entire continuation.

| | Free continuation | | | GT prefix | | |
|---|---|---|---|---|---|---|
| Block | Validity | Uniqueness | $d_{\text{all}}$ | Validity | Uniqueness | $d_{\text{all}}$ |
| 1 | 100.00 | 25.00 | 25.00 | 100.00 | 25.00 | 25.00 |
| 2 | 87.50 | 50.00 | 50.00 | 100.00 | 43.75 | 43.75 |
| 3 | 87.50 | 56.25 | 50.00 | 93.75 | 54.17 | 56.25 |
| 4 | 87.50 | 81.25 | 81.25 | 100.00 | 50.00 | 50.00 |
| 5 | 75.00 | 91.67 | 93.75 | 68.75 | 50.00 | 62.50 |
| 6 | 87.50 | 91.67 | 93.75 | 93.75 | 45.83 | 50.00 |
| 7 | 81.25 | 91.67 | 93.75 | 62.50 | 52.78 | 50.00 |
| 8 | 68.75 | 87.50 | 93.75 | 93.75 | 56.25 | 56.25 |

The stories in Table 5 (bottom) show that Free continuation with the same $0.5$B checkpoint can extend a prefix through successive blocks into a longer text. Each new block is conditioned on the previously accepted generated blocks; unsatisfactory blocks are regenerated with fresh noise before continuation proceeds.

### 4.4 MULTIMODAL GENERATION

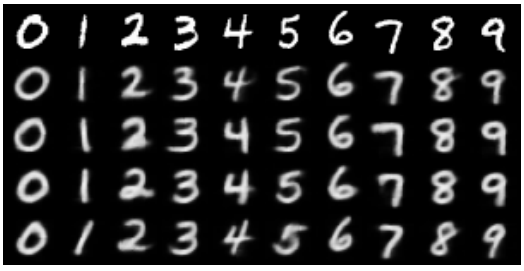
We demonstrate the same block-wise generation mechanism beyond text in two multimodal settings (Fig. 3). In *text-to-image generation*, an English digit word conditions an image block predicted in one forward pass. In *unconditional text-and-image generation*, the model first generates a digit word and then an image conditioned on that word, using one forward pass per block. These MNIST examples provide a qualitative demonstration of multimodal generation.

Table 5: Qualitative text-generation examples. Top: one-pass continuations of the same eight-token prefix; underlining marks errors. Bottom: selected stories from the 0.5B student at step 16,000. Each attempt uses one forward pass; these stories are not unfiltered rollouts.

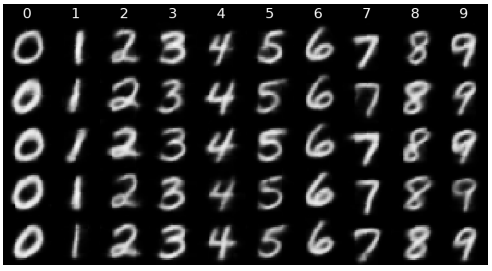
| WTA | CONDOR |
|---|---|
| **Prefix:** *At the market, Lily saw many fruits* | |
| *. She were to apples,,,* | *inside. She said, "Wow,* |
| *. She saw to apples,,,* | *on the ground. She said, "* |

**Multi-block text continuation.** *One day, a little girl named Sue went to the park. She saw a big tree and wanted to climb it. Sue started to climb the tree, but she was very small. Sue saw a big bird in the tree. The bird said, "Hi, Sue! Do you need help?" Sue said, "Yes, I want to climb the tree." The bird said, "I can help you, Sue." Sue and the bird climbed the tree. They were happy and said, "Thank you, bird!" The bird said, "You're welcome, Sue." Sue and the bird became good friends. They played in the park every day.*

**Multi-block text continuation.** *Sara and Tom are friends. They like to play with toys in Sara's room. Sara has many toys. She has dolls, cars, blocks and books. Tom likes to play with toys. One day, Sara says, "Tom, let's play together." Tom says, "OK, Sara, I will play with you." Sara and Tom play with the toys together. They have a lot of fun. They are happy.*



(a) Text-to-image generation (step 18,000). Columns are *zero–nine*; each sample row uses the same noise across words.




(b) Unconditional text-and-image generation (step 18,500). The digit word is generated before the image.

Figure 3: Qualitative multimodal generation on MNIST, with one forward pass per generated block.

## 5 CONCLUSION

We introduced CONDOR, a noise-conditioned masked diffusion model trained from scratch with winner-take-all supervision and same-noise self-distillation. The method learns to map different noise samples to different continuations, predicting a complete block in one forward pass. Small-scale experiments demonstrate single-block and block-wise text generation, as well as text-to-image and unconditional text-and-image generation.

## 6 LIMITATIONS

The experiments here study TinyStories continuation and a qualitative MNIST demonstration. Larger models, more complex and diverse datasets, reliable long-form generation, and broader generalization are natural directions for follow-up work.

### AI USE STATEMENT

We used generative AI tools to assist with manuscript polishing, code review and debugging, and the organization of experimental results. A language model also produced the text-validity labels in Section 4.1 and Appendix B. Those labels were locked before aggregation, and that review is not an independent human evaluation. We have reviewed the AI-assisted text, code, and judgments. The authors take responsibility for the final manuscript, code, and reported results.

### Ethics statement

This work trains masked models from scratch on TinyStories and MNIST. It does not involve human subjects or release a new dataset. Judged validity is local fragment acceptability under a permissive rubric, not factual correctness, fairness, or safety. The method does not remove biases that may come from the training text or from the language-model judge. Teacher refinement increases training compute. We report training GPU-hours in Appendix A and do not claim an inference-latency improvement.

### Reproducibility statement

Section 3 specifies the training objective, noise coupling, and one-step decoder. Section 4.1 and Table 1 specify the tasks, data processing, architectures, and training parameters. Appendix A provides training schedules, the DiMO adaptation, and training compute. Appendix B defines the judging rubric. Additional experiments and examples are in Appendix C. These sections provide the detail needed to reimplement the method and the reported evaluation setup.

## References

Marianne Arriola, Aaron Kerem Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *International Conference on Learning Representations*, pp. 50726–50753, 2025.

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*, volume 34, pp. 17981–17993. Curran Associates, Inc., 2021.

Tiwei Bie, Maosong Cao, Xiang Cao, Bingsen Chen, Fuyuan Chen, Kun Chen, Lun Du, Daozhuo Feng, Haibo Feng, Mingliang Gong, Zhuocheng Gong, Yanmei Gu, Jian Guan, Kaiyuan Guan, Hongliang He, Zenan Huang, Juyong Jiang, Zhonghui Jiang, Zhenzhong Lan, Chengxi Li, Jianguo Li, Zehuan Li, Huabin Liu, Lin Liu, Guoshan Lu, Yuan Lu, Yuxin Ma, Xingyu Mou, Zhenxuan Pan, Kaida Qiu, Yuji Ren, Jianfeng Tan, Yiding Tian, Zian Wang, Lanning Wei, Tao Wu, Yipeng Xing, Wentao Ye, Liangyu Zha, Tianze Zhang, Xiaolu Zhang, Junbo Zhao, Da Zheng, Hao Zhong, Wanli Zhong, Jun Zhou, Junlin Zhou, Liwang Zhu, Muzhi Zhu, and Yihong Zhuang. LLaDA2.1: Speeding up text diffusion via token editing. *arXiv preprint arXiv:2602.08676*, February 2026.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 10–21, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1002.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1509.00519.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5209–5235. PMLR, July 2024.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11315–11325, June 2022.

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

Ronen Eldan and Yuanzhi Li. TinyStories: How small can language models be and still speak coherent English? *arXiv preprint arXiv:2305.07759*, May 2023.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, volume 38, pp. 75460–75482. Curran Associates, Inc., 2025. doi: 10.52202/085713-2534.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-Predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6112–6121. Association for Computational Linguistics, November 2019. doi: 10.18653/v1/D19-1633.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. DiffuSeq: Sequence to sequence text generation with diffusion models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=jQj-_rLVXsj.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1l8BtlCb.

Hongcan Guo, Qinyu Zhao, Yian Zhao, Shen Nie, Rui Zhu, Qiushan Guo, Feng Wang, Tao Yang, Hengshuang Zhao, Guoqiang Wei, and Yan Zeng. Continuous latent diffusion language model. *arXiv preprint arXiv:2605.06548*, May 2026.

Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264. IEEE, June 2018. doi: 10.1109/CVPR.2018.00240.

Abner Guzmán-Rivera, Dhruv Batra, and Pushmeet Kohli. Multiple choice learning: Learning to produce multiple structured outputs. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

Satoshi Hayakawa, Yuhta Takida, Masaaki Imaizumi, Hiromi Wakaki, and Yuki Mitsufuji. Distillation of discrete diffusion through dimensional correlations. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 22259–22297. PMLR, July 2025.

Haoyu He, Katrin Renz, Yong Cao, and Andreas Geiger. MDPO: Overcoming the training-inference divide of masked diffusion language models. *arXiv preprint arXiv:2508.13148*, August 2025.

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*, May 2026.

Zemin Huang, Yuhang Wang, Zhiyang Chen, and Guo-Jun Qi. Don't settle too early: Self-reflective remasking for diffusion language models. In *International Conference on Learning Representations*, pp. 56998–57026, April 2026.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125–1134, July 2017.

Łukasz Kaiser, Aurko Roy, Ashish Vaswani, Niki Parmar, Samy Bengio, Jakob Uszkoreit, and Noam Shazeer. Fast decoding in sequence models using discrete latent variables. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2390–2399. PMLR, July 2018.

Wonjun Kang, Kevin Galim, Seunghyuk Oh, Minjae Lee, Yuchen Zeng, Shuibai Zhang, Coleman Hooper, Yuezhou Hu, Hyung Il Koo, Nam Ik Cho, and Kangwook Lee. ParallelBench: Understanding the trade-offs of parallel decoding in diffusion LLMs. In *International Conference on Learning Representations*, pp. 89900–89938, April 2026.

Jaeyeon Kim, Seunggeun Kim, Taekyun Lee, David Z. Pan, Hyeji Kim, Sham Kakade, and Sitan Chen. Fine-tuning masked diffusion for provable self-correction. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=2LRaNghfoM.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014. URL https://arxiv.org/abs/1312.6114.

Siqi Kou, Lanxiang Hu, Zhezhi He, Zhijie Deng, and Hao Zhang. CLLMs: Consistency large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 25426–25440. PMLR, July 2024.

Stefan Lee, Senthil Purushwalkam Shiva Prakash, Michael Cogswell, Viresh Ranjan, David Crandall, and Dhruv Batra. Stochastic multiple choice learning for training diverse deep ensembles. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Victor Letzelter, Hugo Malard, Mathieu Fontaine, Gaël Richard, Slim Essid, Andrei Bursuc, and Patrick Pérez. Multiple choice learning of low-rank adapters for language modeling. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=CCO35e4DCO.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286. PMLR, July 2023.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-LM improves controllable text generation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 4328–4343. Curran Associates, Inc., 2022. doi: 10.52202/068431-0313.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

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, July 2024.

Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Q. Weinberger. Latent diffusion for language generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 56998–57025. Curran Associates, Inc., 2023. doi: 10.52202/075280-2492.

Xuezhe Ma, Chunting Zhou, Xian Li, Graham Neubig, and Eduard Hovy. FlowSeq: Non-autoregressive conditional sequence generation with generative flow. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4282–4292. Association for Computational Linguistics, November 2019. doi: 10.18653/v1/D19-1437.

Osama Makansi, Eddy Ilg, Özgün Çiçek, and Thomas Brox. Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7137–7146. IEEE, June 2019. doi: 10.1109/CVPR.2019.00731.

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. In *Advances in Neural Information Processing Systems*, volume 38, pp. 50608–50646. Curran Associates, Inc., 2025. doi: 10.52202/085713-1689.

Shen Nie, Qiyang Min, Shaoxuan Xu, Zihao Huang, Yuxuan Song, Yong Shan, Yankai Lin, Wayne Xin Zhao, Chongxuan Li, and Ji-Rong Wen. Improved large language diffusion models. *arXiv preprint arXiv:2606.25331*, June 2026.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *International Conference on Learning Representations*, pp. 64972–65009, 2025.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, February 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Christian Rupprecht, Iro Laina, Robert DiPietro, Maximilian Baust, Federico Tombari, Nassir Navab, and Gregory D. Hager. Learning in an uncertain world: Representing ambiguity through multiple hypotheses. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3591–3600, 2017.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander M. Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 130136–130184. Curran Associates, Inc., 2024. doi: 10.52202/079017-4135.

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin T. Chiu, and Volodymyr Kuleshov. The diffusion duality. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 52584–52619. PMLR, July 2025.

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

Andrea Santilli, Silvio Severino, Emilian Postolache, Valentino Maiorca, Michele Mancusi, Riccardo Marin, and Emanuele Rodolà. Accelerating transformer inference for translation via parallel decoding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12336–12355. Association for Computational Linguistics, July 2023. doi: 10.18653/v1/2023.acl-long.689.

Yair Schiff, Omer Belhasin, Roy Uziel, Guanghan Wang, Marianne Arriola, Gilad Turok, Ran Zilberstein, Michael Elad, and Volodymyr Kuleshov. Learn from your mistakes: Self-correcting masked diffusion models. *arXiv preprint arXiv:2602.11590*, February 2026.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, volume 37, pp. 103131–103167. Curran Associates, Inc., 2024. doi: 10.52202/079017-3277.

Raphael Shu, Jason Lee, Hideki Nakayama, and Kyunghyun Cho. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8846–8853, April 2020. doi: 10.1609/aaai.v34i05.6413.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, July 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. In *Advances in Neural Information Processing Systems*, volume 38, pp. 147282–147339. Curran Associates, Inc., 2025. doi: 10.52202/085713-4927.

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*, pp. 57027–57051, April 2026.

Tianyu Xie, Shuchen Xue, Zijin Feng, Tianyang Hu, Jiacheng Sun, Zhenguo Li, and Cheng Zhang. Variational autoencoding discrete diffusion with enhanced dimensional correlations modeling. In *International Conference on Learning Representations*, pp. 39109–39132, April 2026.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7B: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, August 2025.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6613–6623, June 2024.

Longxuan Yu, Yunshu Wu, Yu Fu, Siheng Xiong, Rob Brekelmans, Hui Liu, Yue Dong, and Greg Ver Steeg. DSL-LLaDA: Scaling continuous denoising to 8B masked diffusion LMs. *arXiv preprint arXiv:2606.01024*, May 2026.

Kevin Zhai, Sabbir Mollah, Zhenyi Wang, and Mubarak Shah. CORE: Context-robust remasking for diffusion language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=bmKHxLWkz9.

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. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=80nuPd5Gvb.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Yuanzhi Zhu, Xi Wang, Stéphane Lathuilière, and Vicky Kalogeiton. DiMO: Distilling masked diffusion models into one-step generator. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18606–18618, October 2025.

# SUPPLEMENTARY MATERIAL

Appendix A provides architecture, training schedules, the DiMO adaptation, and training compute. Appendix B defines the judging rubric. Appendix C reports additional experiments and examples.

## A ADDITIONAL EXPERIMENTAL DETAILS

### A.1 ARCHITECTURE AND TRAINING DETAILS

Table S1 complements the core settings in Table 1. The single-block model uses pre-norm layers, learned positional embeddings, and tied input/output token embeddings. Its EMA teacher has decay $0.9999$, is initialized from the student, and is updated after each optimizer step.

Table S1: Optimizer details for CONDOR. The image column applies to both image tasks.

| | Single-block text | Multi-block text | Images |
|---|---|---|---|
| AdamW $(\beta_1, \beta_2)$ | $(0.9, 0.999)$ | $(0.9, 0.95)$ | $(0.9, 0.999)$ |
| Weight decay | 0.01 | 0.1 | 0.01 |
| Warm-up updates $W$ | 2000 | 800 | 2500 |
| Final learning rate | 0 | $3 \times 10^{-5}$ | 0 |

**Learning-rate and distillation schedules.** Let $S$ denote total updates and $\eta_0 = 3 \times 10^{-4}$. Single-block training uses $\eta_t = \eta_0 \min(1, t/W)[1 + \cos(\pi t/S)]/2$ at update $t$, with distillation weight $w(t) = 0.1 + 0.9 \min(t/80{,}000, 1)$. The 30k-update runs set $S = 30{,}000$ and shorten the distillation ramp to 16,000 updates, so wall-clock time stays with the 150k-update runs that do not use self-distillation. Multi-block and image runs use linear warm-up over $W$ updates followed by cosine decay to the final rate in Table S1. Their distillation weights are $w(t) = \min(1, \max(0, (t - 3200)/3200))$ and $w(t) = 0.1 + 0.9 \min(t/20{,}000, 1)$, respectively. The multi-block full-mask loss is normalized by $k$; for images, both the full-image ground-truth term and the refilled-pixel distillation term are normalized by $k$.

**Training compute.** All single-block runs use one A100. Default CONDOR takes 14.1 GPU-hours, the $k = 4$ WTA baseline 5.3 (8.3 at $k = 16$), and each 30k-update run 2.9. DiMO takes about 2 GPU-hours per configuration.

### A.2 BASELINE TRAINING

**Plain CE, IWAE, and WTA.** These runs use the same base learning rate and weight decay as the single-block basic setting, with uniform mask sampling unless specified otherwise. Their learning rate uses 2000 linear warm-up updates followed by cosine decay to zero.

**DiMO.** DiMO originally distills a pretrained teacher on discrete images. The text adaptation follows that protocol. The 150k-update Plain CE checkpoint is the frozen teacher. Student, teacher, and fake model are initialized from this checkpoint. The four adaptations cross fully masked or hybrid inputs with learning rate $10^{-5}$ (embeddings frozen) or $10^{-4}$ (all parameters trainable), each for 50k additional updates at batch size 512. Table 2 reports the hybrid-input adaptation at learning rate $10^{-5}$. Each update samples a student continuation and remasks a random subset. The teacher–fake forward KL at masked positions updates the student. A fresh student sample trains the fake model with masked cross-entropy. Fully masked inputs use $\sigma = 0.5$ noise; hybrid inputs use $60\%$ masks and $40\%$ uniformly sampled tokens, with embeddings perturbed as $\sqrt{1 - 0.1^2}\, e + 0.1\, \mathrm{RMS}(E)\epsilon$. All four adaptations use AdamW with zero weight decay and the single-block learning-rate formula above with $S = 50{,}000$ and $W = 500$. Sampling temperature is 1; each student update is paired with one fake-model update using label smoothing 0.1.

### A.3 Ablation variants

The independent-target-noise run builds the distillation target under a fresh $\varepsilon'$ and trains the student under the original $\varepsilon$.

## B Evaluation protocols and metrics

### B.1 Frozen-rubric text-validity re-review

**Criterion.** A continuation is judged together with its visible prefix. Natural truncation is allowed. Internal grammar errors, merged words, token loops, control tokens, and punctuation degeneration fail. A close-quote can pass when the visible text already reads as the end of speech. An extra close-quote or a nesting the visible text already breaks still fails. Boundary items are concrete ambiguities, such as a name switch. The reported *mixed* policy uses each item's recorded decision. The judge is a language model that had seen earlier rubric discussion, so this is not an independent human review. Labels were locked before aggregation.

## C Additional experiments and examples

### C.1 Noise geometry

Tables 3 and S2 probe the default seed-0 student at 150k updates, with no retraining. A radius-$\alpha r_0$ input uses a unit direction $u$ in the $D = 2048$ completion field, $\delta = \sigma \mathrm{RMS}(e_{\texttt{[M]}})\,\alpha\sqrt{D}\,u$, so $\|\delta\|_2 = \alpha r_0$. On this checkpoint $\mathrm{RMS}(e_{\texttt{[M]}}) = 0.720720$ and $r_0 = 16.308026$. An angle-$\theta$ input has the same norm and angle $\theta$ to a reference direction, with the tangent uniform on the orthogonal complement. The $0^\circ$ and $180^\circ$ cases are a single direction.

Table S2: Unedited continuations of *They would swim, splash, and pretend* at norm $r_0$. Panel A follows one tangent; panel B uses other directions at $30^\circ$.

| Angle | Tangent | Generated continuation | Conf. (%) | $\log P$ |
|---|---|---|---|---|
| *A. One paired rotation trajectory* | | | | |
| $0^\circ$ | — | `to play.  The fish said, "` | 78.4 | $-2.163$ |
| $30^\circ$ | $v_0$ | `to play.  Fin said, "Yes` | 89.2 | $-1.063$ |
| $90^\circ$ | $v_0$ | `to swim.\nOne day, the` | 91.0 | $-0.961$ |
| $180^\circ$ | — | `to be friends.\nOne day,` | 86.5 | $-1.259$ |
| *B. Multiple directions with the same angle to the same reference* | | | | |
| $30^\circ$ | $v_1$ | `together.  The fish said, "Okay` | 86.1 | $-1.472$ |
| $30^\circ$ | $v_2$ | `to play.  The fish said, "` | 92.6 | $-0.624$ |
| $30^\circ$ | $v_4$ | `to be a fish.  One day,` | 61.2 | $-4.808$ |
| $30^\circ$ | $v_6$ | `to play..  fish said, "` | 74.8 | $-2.545$ |