

Masked Diffusion Language Models: Continuous to Discrete

Zhe Wang

Overview

- Background
- Foundations for Masked Diffusion Model
- Science Challenges
 - Generation order matters: Topk and Topk-Margin
 - Generate and then fix: Remasking strategy

Background - MDMs

- Masked diffusion models (MDMs) are recently emerged as a powerful class of generative models for language and image generation tasks.

Mercury: Ultra-Fast Language Models Based on Diffusion

Inception Labs

Samar Khanna*, Siddhant Kharbanda*, Shufan Li*, Harshit Varma*, Eric Wang*
Sawyer Birnbaum^, Ziyang Luo^, Yanis Miraoui^, Akash Palrecha^
Stefano Ermon[‡], Aditya Grover[‡], Volodymyr Kuleshov[‡]
*^{^‡} equal core, cross-function, senior contributors listed alphabetically.
hello@inceptionlabs.ai

 ByteDance | Seed

 AIR   GenSi

Seed Diffusion: A Large-Scale Diffusion Language Model with High-Speed Inference

¹ByteDance Seed ²Institute for AI Industry Research (AIR), Tsinghua University
³SIA-Lab of Tsinghua AIR and ByteDance Seed

 PRINCETON
UNIVERSITY

 ByteDance | Seed

MMaDA: Multimodal Large Diffusion Language Models

Ling Yang^{1,4†}, Ye Tian^{2*}, Bowen Li², Xinchen Zhang³,
Ke Shen¹, Yunhai Tong², Mengdi Wang¹
¹Princeton University ²Peking University ³Tsinghua University ⁴ByteDance Seed

Khanna et al. (2025). Mercury: Ultra-Fast Language Models Based on Diffusion. ArXiv, abs/2506.17298.

Song et al. (2025). Seed Diffusion: A Large-Scale Diffusion Language Model with High-Speed Inference.

Yang et al. MMaDA: Multimodal Large Diffusion Language Models. ArXiv, abs/2505.15809.

Background - The Advantage of MDM: Efficiency

- Fast generation speed

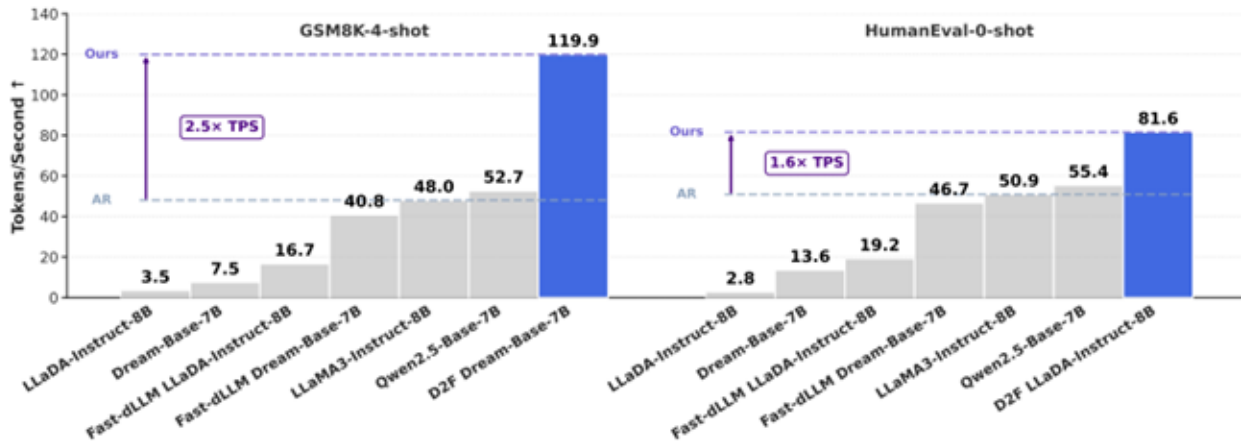


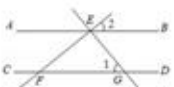
Figure 1: **D2F dLLMs surpass AR LLMs in inference speed for up to 2.5×.** Comparison of inference throughput among D2F dLLMs, vanilla dLLMs like Dream-Base-7B (Ye et al., 2025) and LLaDA-Instruct-8B (Nie et al., 2025), previous SOTA acceleration method Fast-dLLM (Wu et al., 2025), and similarly-sized AR baselines (Yang et al., 2024a; Grattafiori et al., 2024). The max generation length is set to 512.

The Advantage of MDM: One model for ...

Text -> text

| | | |
|--|---|---|
| <p>Task 1: Textual Reasoning</p> <p>Question:</p> <p>Given that</p> $\tan \beta = \frac{1}{2}$ <p>where β is an angle, determine the exact value of the following expression:</p> $\sin^2 \beta - 3 \sin \beta \cos \beta + 4 \cos^2 \beta$ | <p>Answers from Other Models</p> <p style="text-align: center;">☹️</p> <p>Show-o, Emu3 and Janus Pro 7B do not support this task.</p> | <p>Answer from MMA DA</p> <p><think> To find the value of $(\sin^2 \beta - 3 \sin \beta \cos \beta + 4 \cos^2 \beta)$ given that $\tan \beta = 1/2$, we can follow these steps:</p> <ol style="list-style-type: none"> Express $\sin \beta$ and $\cos \beta$: $\sin \beta = \frac{\tan \beta / \sqrt{1 + \tan^2 \beta}}{\sqrt{1 + \tan^2 \beta}} = \frac{(1/2) / \sqrt{1 + (1/2)^2}}{\sqrt{1 + (1/2)^2}} = \frac{1/\sqrt{5}}{2/\sqrt{5}}$ $\cos \beta = \frac{1/\sqrt{1 + \tan^2 \beta}}{2/\sqrt{5}} = \frac{2/\sqrt{5}}{2/\sqrt{5}}$ Calculate each term: $\sin^2 \beta = \left(\frac{1}{\sqrt{5}}\right)^2 = 1/5$ $3 \sin \beta \cos \beta = 3 \cdot \left(\frac{1}{\sqrt{5}}\right) \cdot \left(\frac{2}{\sqrt{5}}\right) = 3 \cdot (2/5) = 6/5$ $4 \cos^2 \beta = 4 \cdot \left(\frac{2}{\sqrt{5}}\right)^2 = 4 \cdot (4/5) = 16/5$ Substitute and simplify: $\sin^2 \beta - 3 \sin \beta \cos \beta + 4 \cos^2 \beta = 1/5 - 6/5 + 16/5 = 11/5$ <p>Final Answer: 11/5 </think> The exact value is 11/5. ✓</p> |
|--|---|---|

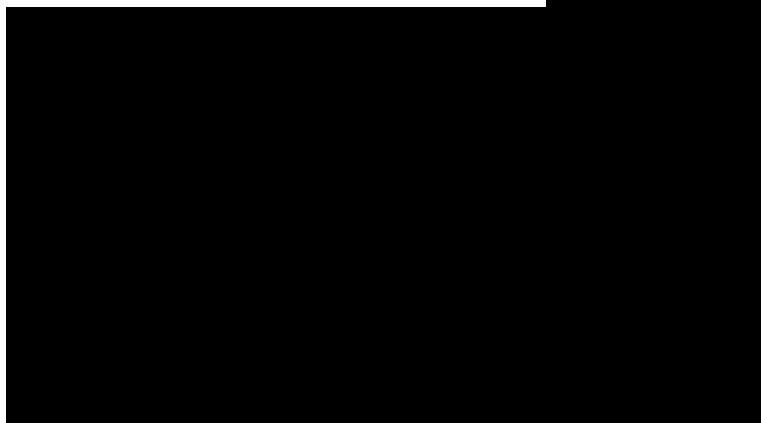
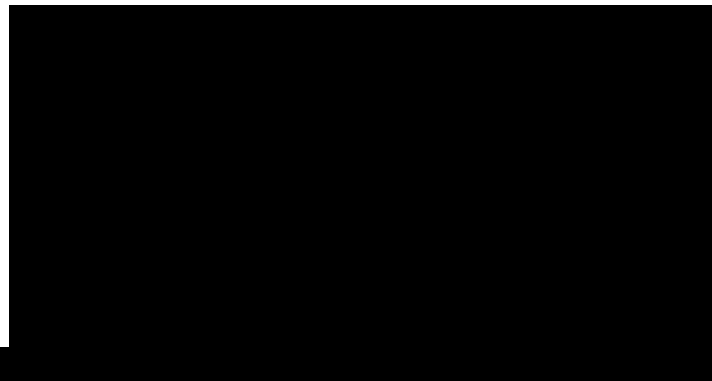
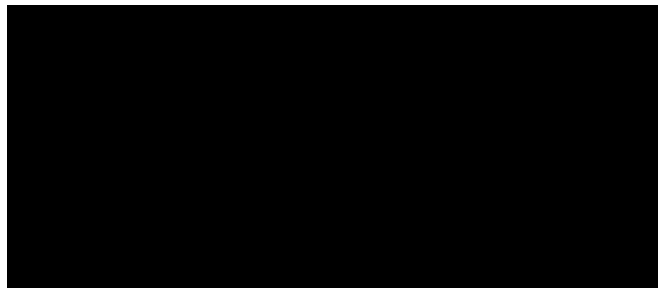
Text + image -> text

| | | |
|--|---|---|
| <p>Task 2: Multimodal Reasoning</p>  <p>Question:</p> <p>In the diagram provided, lines AB and CD are parallel. The line EF intersects AB at point E and CD at point F. From point E, a perpendicular EG is drawn to EF, meeting CD at point G. Given that angle 1 is 52 degrees, what is the measure of angle 2?</p> | <p>Answers from Other Models</p> <p>Show-o: Angle 2 measures 52 degrees. ✗</p> <p>Emu3: Angle 2 is equal to angle 1 due to the parallel lines AB and CD. Since angle 1 measures 52 degrees, angle 2 also measures 52 degrees. ✗</p> <p>Janus Pro 7B: Since line AB is parallel to line CD, corresponding angles are equal. Therefore, angle 1 and angle 2 are equal. Since angle 1 measures 52 degrees, angle 2 also measures 52 degrees. The answer is: 52 ✗</p> | <p>Answer from MMA DA</p> <p><think> To determine the measure of angle 2, we need to analyze the given information and use the properties of parallel lines and perpendicularity.</p> <ol style="list-style-type: none"> Identify the given information: <ul style="list-style-type: none"> Line AB is parallel to line CD. Points E and F are the intersections of line EF with lines AB and CD respectively. Line EG is perpendicular to line EF, with point E as the foot of the perpendicular. Line EG intersects line CD at point G. Angle 1 measures 52 degrees. Understand the relationship between the angles: <ul style="list-style-type: none"> Since EG is perpendicular to EF, angle FEG is 90 degrees. Therefore, angle 2 + angle 1 = 90 degrees. Determine the measure of angle 2: <ul style="list-style-type: none"> Angle 2 = 90 degrees - 52 degrees = 38 degrees. <p>So, the measure of angle 2 is 38. </think> The measure of angle 2 is 38 degrees. ✓</p> |
|--|---|---|

Text -> image

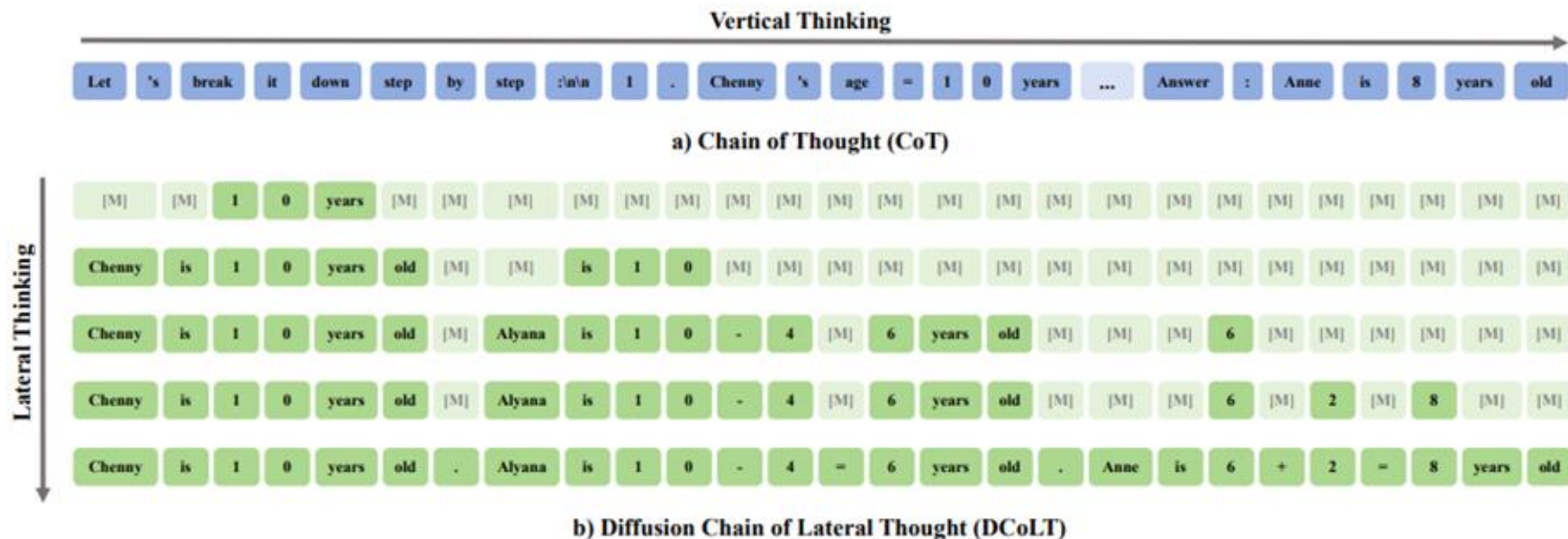
| | | |
|---|---|--|
| <p>Task 3: World Knowledge-Aware Text-to-Image Generation</p> <p>Prompt:</p> <p>The most popular kind of animal among the people of Istanbul.</p> | <p>Images from Other Models</p> <p>Show-o ✗ Emu3 ✗ Janus Pro 7B ✗</p>  | <p>CoT and Image from MMA DA</p> <p><think> The description states that the most popular animal among the people of Istanbul is the cat. This is because cats are a popular animal for companionship and people in the city. </think> A close-up of a cat with a light brown and white fur pattern. ✓</p>  |
|---|---|--|

Multimodal Masked Diffusion Model



6

Generation Order



[1]: Jaeyeon et al. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. ICML 2025 outstanding paper.
 [2]: Zemin et al. Reinforcing the diffusion chain of lateral thought with diffusion language models. ArXiv, abs/2505.10446, 2025.

Current Performance of MDM

Table 1. Benchmark Results of Pre-trained LLMs. * indicates that LLaDA 8B Base, LLaMA2 7B Base, and LLaMA3 8B Base are evaluated under the same protocol, detailed in Appendix B.5. Results indicated by † and ‡ are sourced from Chu et al. (2024); Yang et al. (2024) and Bi et al. (2024) respectively. The numbers in parentheses represent the number of shots used for evaluation. "-" indicates unknown data.

| | LLaDA 8B* | LLaMA3 8B* | LLaMA2 7B* | Qwen2 7B† | Qwen2.5 7B† | Mistral 7B† | Deepseek 7B‡ |
|-----------------------|-----------------|-----------------|------------|-----------|-------------|-------------|--------------|
| Model | Diffusion | AR | AR | AR | AR | AR | AR |
| Training tokens | 2.3T | 15T | 2T | 7T | 18T | - | 2T |
| General Tasks | | | | | | | |
| MMLU | 65.9 (5) | 65.4 (5) | 45.9 (5) | 70.3 (5) | 74.2 (5) | 64.2 (5) | 48.2 (5) |
| BBH | 49.8 (3) | 57.6 (3) | 37.3 (3) | 62.3 (3) | 70.4 (3) | 56.1 (3) | 39.5 (3) |
| ARC-C | 47.9 (0) | 53.1 (0) | 46.3 (0) | 60.6 (25) | 63.7 (25) | 60.0 (25) | 48.1 (0) |
| Hellaswag | 72.5 (0) | 79.1 (0) | 76.0 (0) | 80.7 (10) | 80.2 (10) | 83.3 (10) | 75.4 (0) |
| TruthfulQA | 46.4 (0) | 44.0 (0) | 39.0 (0) | 54.2 (0) | 56.4 (0) | 42.2 (0) | - |
| WinoGrande | 74.8 (5) | 77.3 (5) | 72.5 (5) | 77.0 (5) | 75.9 (5) | 78.4 (5) | 70.5 (0) |
| PIQA | 74.4 (0) | 80.6 (0) | 79.1 (0) | - | - | - | 79.2 (0) |
| Mathematics & Science | | | | | | | |
| GSM8K | 70.7 (4) | 53.1 (4) | 14.3 (4) | 80.2 (4) | 85.4 (4) | 36.2 (4) | 17.4 (8) |
| Math | 27.3 (4) | 15.1 (4) | 3.2 (4) | 43.5 (4) | 49.8 (4) | 10.2 (4) | 6.0 (4) |
| GPQA | 26.1 (5) | 25.9 (5) | 25.7 (5) | 30.8 (5) | 36.4 (5) | 24.7 (5) | - |
| Code | | | | | | | |
| HumanEval | 33.5 (0) | 34.2 (0) | 12.8 (0) | 51.2 (0) | 57.9 (0) | 29.3 (0) | 26.2 (0) |
| HumanEval-FIM | 73.8 (2) | 73.3 (2) | 26.9 (2) | - | - | - | - |
| MBPP | 38.2 (4) | 47.4 (4) | 18.4 (4) | 64.2 (0) | 74.9 (0) | 51.1 (0) | 39.0 (3) |
| Chinese | | | | | | | |
| CMMLU | 69.9 (5) | 50.7 (5) | 32.5 (5) | 83.9 (5) | - | - | 47.2 (5) |
| C-Eval | 70.5 (5) | 51.7 (5) | 34.0 (5) | 83.2 (5) | - | - | 45.0 (5) |

LLaDA is an open source large scale MDM trained 2.3 trillion tokens with 0.13 million H800 GPU hours

SOTA Performance on Text-to-Image Generation

Table 3 Evaluation on Image Generation Benchmarks.

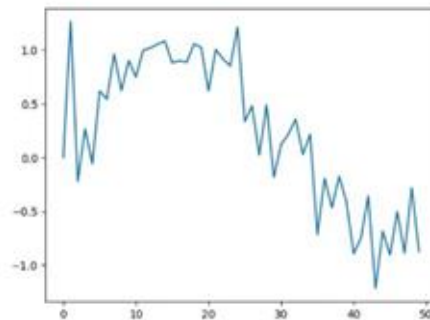
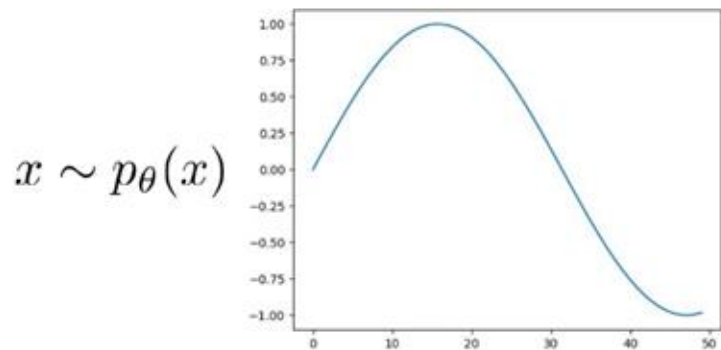
| Model | Wise (Cultural) ↑ | Image Reward ↑ | CLIP Score ↑ | GenEval ↑ | | | | | | |
|------------------------------------|----------------------|-------------------|-----------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | | | | Single Obj. | Two Obj. | Counting | Colors | Position | Color Attr. | Overall |
| Generation-Only | | | | | | | | | | |
| LlamaGen [58] | - | 0.79 | 13.43 | 0.71 | 0.34 | 0.21 | 0.58 | 0.07 | 0.04 | 0.32 |
| SDv1.5 [55] | 0.34 | 0.84 | 23.54 | 0.97 | 0.38 | 0.35 | 0.76 | 0.04 | 0.06 | 0.43 |
| SDv2.1 [55] | 0.30 | 0.95 | 27.41 | 0.98 | 0.51 | 0.44 | 0.85 | 0.07 | 0.17 | 0.50 |
| DALL-E 2 [56] | - | 0.83 | 25.20 | 0.94 | 0.66 | 0.49 | 0.77 | 0.10 | 0.19 | 0.52 |
| SDXL [57] | 0.43 | 1.13 | 32.12 | 0.98 | 0.74 | 0.39 | 0.85 | 0.15 | 0.23 | 0.55 |
| Unified Understanding & Generation | | | | | | | | | | |
| DreamLLM [16] | - | 0.76 | 18.33 | - | - | - | - | - | - | - |
| SEED-X [19] | - | 0.77 | 23.15 | 0.97 | 0.58 | 0.26 | 0.80 | 0.19 | 0.14 | 0.49 |
| Chameleon [8] | - | 0.83 | 20.32 | - | - | - | - | - | - | 0.39 |
| LWM [9] | - | 0.78 | 26.21 | 0.93 | 0.41 | 0.46 | 0.79 | 0.09 | 0.15 | 0.47 |
| Emu [11] | - | 0.81 | 22.29 | - | - | - | - | - | - | - |
| Show-o [20] | 0.28 | 0.92 | 28.94 | 0.95 | 0.52 | 0.49 | 0.82 | 0.11 | 0.28 | 0.53 |
| Janus [13] | 0.16 | 1.03 | 29.45 | 0.97 | 0.68 | 0.30 | 0.84 | 0.46 | 0.42 | 0.61 |
| Gemini-Nano-1 [12] | - | 0.89 | 24.58 | - | - | - | - | - | - | - |
| VAR-GPT [59] | - | 0.94 | 28.85 | 0.96 | 0.53 | 0.48 | 0.83 | 0.13 | 0.21 | 0.53 |
| MMaDA | 0.67 | 1.15 | 32.46 | 0.99 | 0.76 | 0.61 | 0.84 | 0.20 | 0.37 | 0.63 |

SOTA Performance on Multimodal Understanding

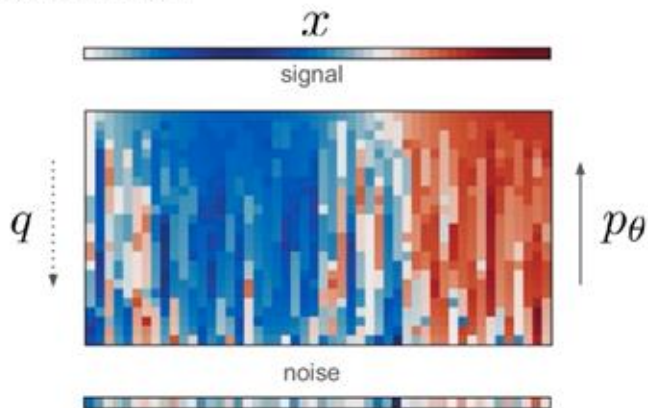
Table 2 Evaluation on Multimodal Understanding Benchmarks.

| Model | POPE↑ | MME↑ | Flickr30k↑ | VQAV2 _(test) ↑ | GQA↑ | MMMU ↑ | MMB ↑ | SEED ↑ |
|------------------------------------|-------------|---------------|-------------|---------------------------|-------------|-------------|-------------|-------------|
| Understanding-Only | | | | | | | | |
| LLaVA-v1.5 [32] | 85.9 | 1510.7 | - | 78.5 | 62.0 | 35.4 | 64.3 | 58.6 |
| InstructBLIP [33] | 78.9 | 1212.8 | - | - | 49.5 | - | - | - |
| Qwen-VL-Chat [34] | - | 1487.5 | - | 78.2 | 57.5 | - | 60.6 | 58.2 |
| mPLUG-Owl2 [35] | 85.8 | 1450.2 | - | 79.4 | 56.1 | - | - | - |
| LLaVA-Phi [36] | 85.0 | 1335.1 | - | 71.4 | - | - | 59.8 | - |
| Unified Understanding & Generation | | | | | | | | |
| DreamLLM [16] | - | - | - | 72.9 | - | - | - | - |
| SEED-X [19] | 84.2 | 1435.7 | 52.3 | - | 47.9 | 35.6 | - | - |
| Chameleon [8] | - | - | 74.7 | 66.0 | - | - | - | - |
| LWM [9] | 75.2 | 948.4 | - | 55.8 | 44.8 | - | - | - |
| Emu [11] | - | - | 77.4 | 57.2 | - | - | - | - |
| Show-o [20] | 80.0 | 1097.2 | 62.5 | 69.4 | 58.0 | 26.7 | - | - |
| Gemini-Nano-1 [12] | - | - | - | 62.7 | - | 26.3 | - | - |
| MMaDA | 86.1 | 1410.7 | 67.6 | 76.7 | 61.3 | 30.2 | 68.5 | 64.2 |

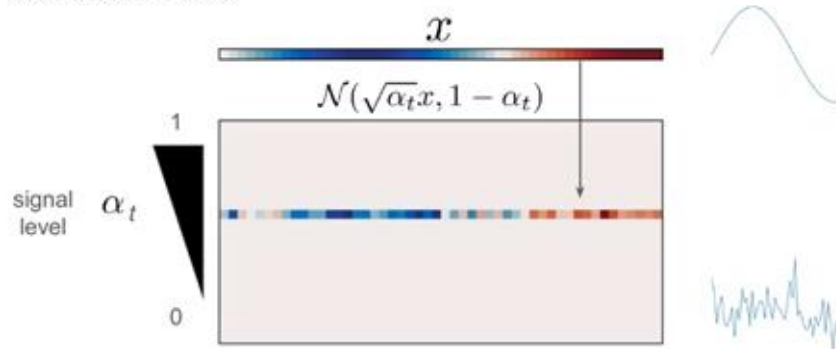
Masked Diffusion Models: How does it work



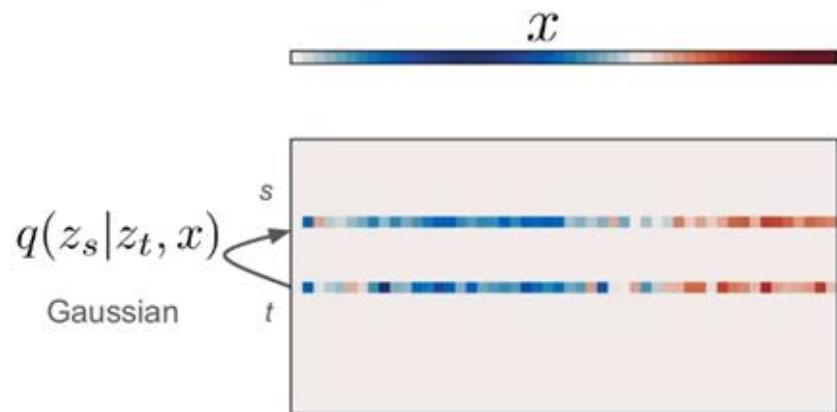
Continuous Diffusion



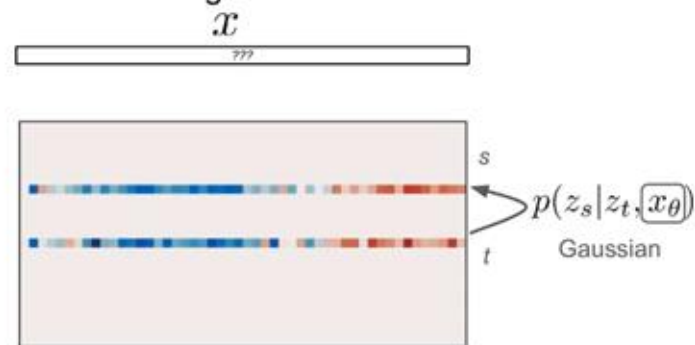
Gaussian Noise



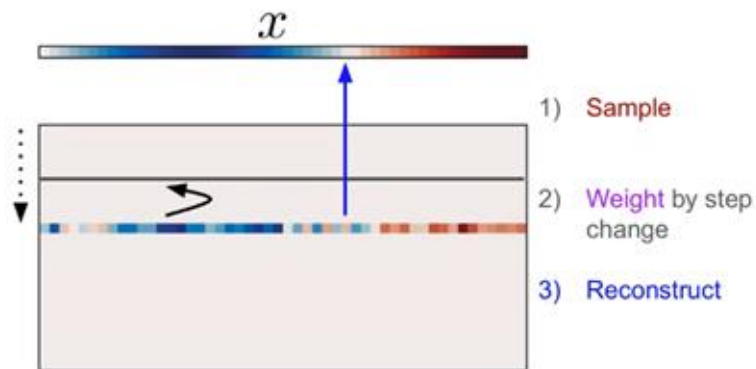
Gaussian Noise Implies Gaussian Posterior



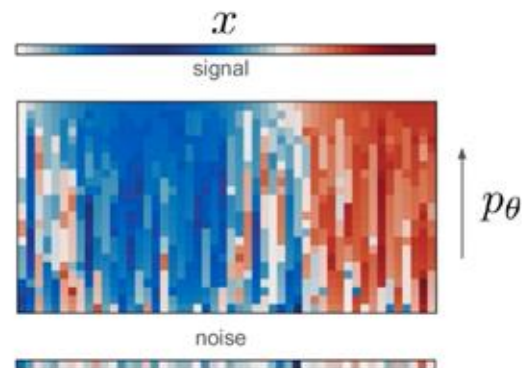
Reconstruction for Denoising



Learning to Denoise



Generation



Why it works

Think of it like a game:

1. I take a clean photo
2. I add random static/noise to it
3. I show you the noisy photo and tell you HOW MUCH noise I added
4. Your job: tell me exactly WHAT noise I added

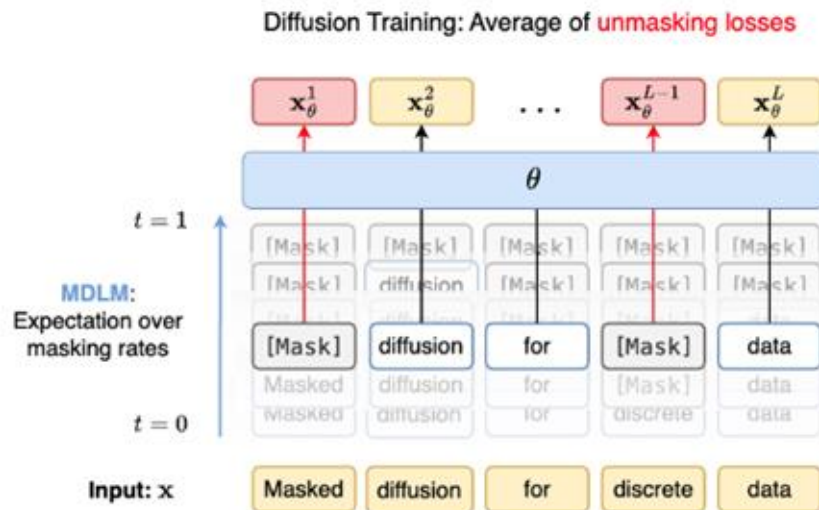
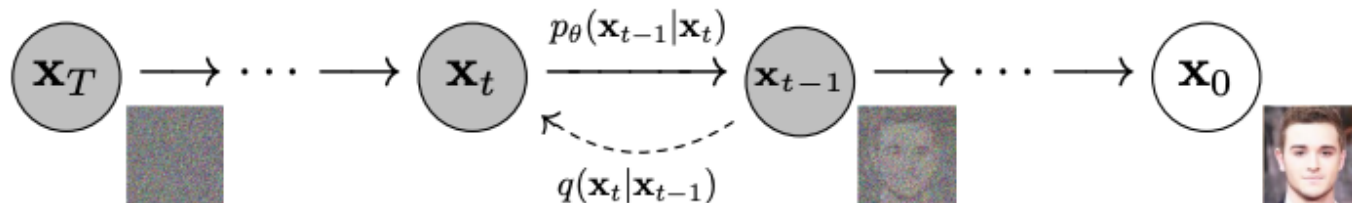
If you can do that perfectly, I can subtract the noise and recover the clean photo!

Comparing to Masked Diffusion (MDLM)

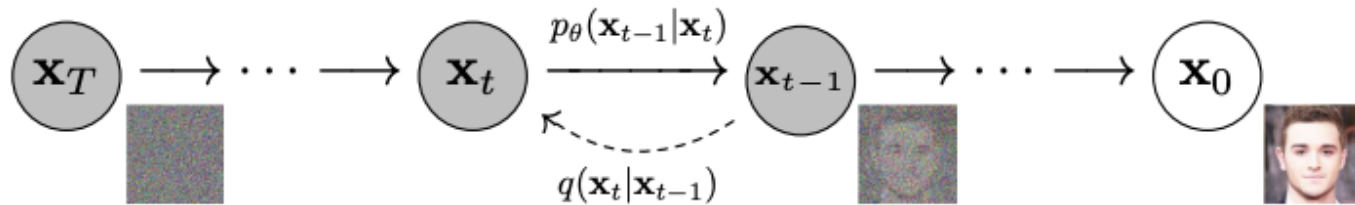
The core principle is identical — corrupt data gradually, learn to reverse the corruption — but the mathematical details differ because continuous noise and discrete masking require different tools.

| | Continuous | Discrete |
|----------------|---|--------------------------------------|
| Data | Real-valued | Categorical tokens |
| Corruption | Add Gaussian noise | Replace with [MASK] |
| Model predicts | The noise ε that was added | The original token x_0 |
| Loss | MSE: $\ \varepsilon - \varepsilon_\theta\ ^2$ | Cross-entropy: $-\log p_\theta(x_0)$ |
| Generation | Gradually remove noise | Gradually unmask tokens |

Masked Diffusion Models: How does it work



Background on Masked Diffusion Model



Forward
process

$$q_{t|0}(x_t^i | x_0^i) = \text{Cat}(\alpha_t \mathbf{e}_{x_0^i} + (1 - \alpha_t) \mathbf{e}_0)$$

Reverse
process

$$q_{s|t}(x_s^i | x_t, x_0) = \begin{cases} \text{Cat}(\mathbf{e}_{x_t^i}) & x_t^i \neq m \\ \text{Cat}\left(\frac{1-\alpha_s}{1-\alpha_t} \mathbf{e}_m + \frac{\alpha_s - \alpha_t}{1-\alpha_t} \mathbf{e}_{x_0}\right) & x_t^i = m \end{cases}$$

Training objective

$$\mathcal{L}_\theta = \int_0^1 \frac{\alpha'_t}{1 - \alpha_t} \mathbb{E}_{\substack{x_0 \sim p_{\text{data}} \\ x_t \sim q_{t|0}(\cdot|x_0)}} [\delta_{x_t,0} \mathbf{e}_{x_0}^\top \log p_\theta(x_t, t)] dt$$

Forward

Think of a timeline from $t=0$ (clean) to $t=1$ (fully masked). At any time t , each token is independently masked with probability $(1 - \alpha_t)$, where α_t is a noise schedule that goes from $\alpha_0 \approx 1$ (almost no masking) to $\alpha_1 \approx 0$ (everything masked).

"The cat sat on mat"

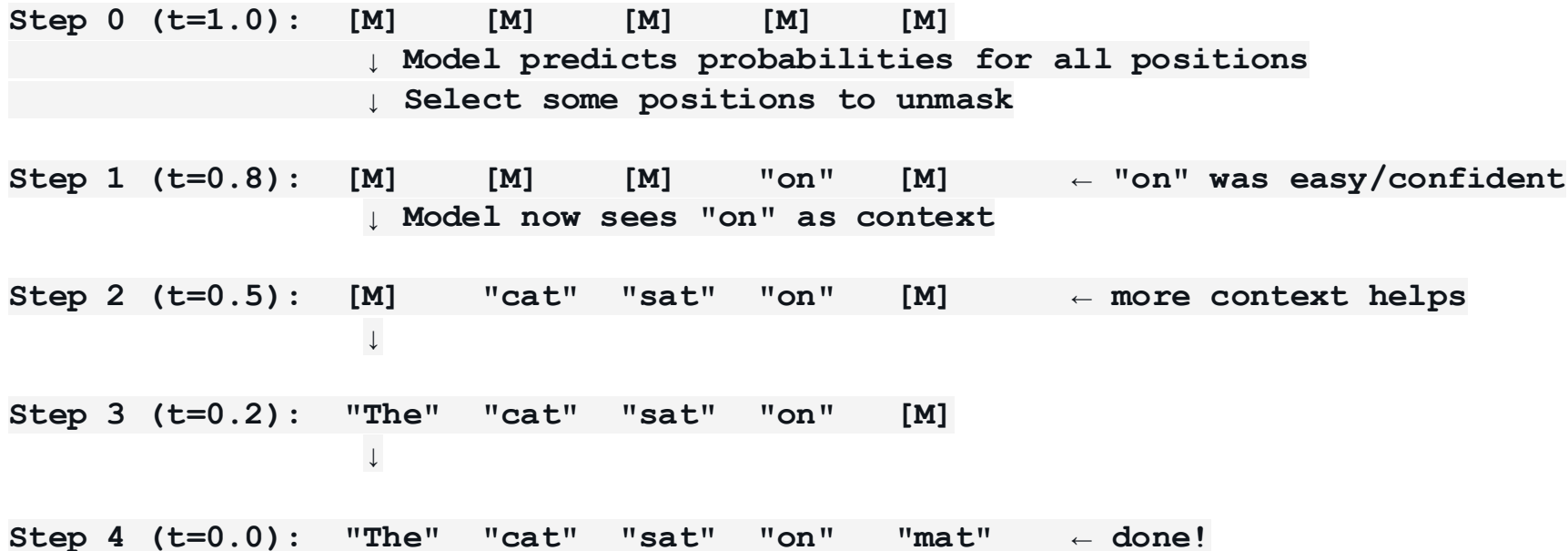
$t = 0.0$ (clean): "The" "cat" "sat" "on" "mat"

↓ ↓ ↓ ↓ ↓
(each token independently: keep with prob α_t , mask with prob $1-\alpha_t$)

| | | | | | | |
|---------------------------------------|-------|-------|-----|------|-------|---------------|
| $t = 0.2$ ($\alpha_t \approx 0.8$): | "The" | "cat" | [M] | "on" | "mat" | ← ~20% masked |
| $t = 0.5$ ($\alpha_t \approx 0.5$): | [M] | "cat" | [M] | "on" | [M] | ← ~50% masked |
| $t = 0.8$ ($\alpha_t \approx 0.2$): | [M] | [M] | [M] | "on" | [M] | ← ~80% masked |
| $t = 1.0$ ($\alpha_t \approx 0.0$): | [M] | [M] | [M] | [M] | [M] | ← 100% masked |

$$q(z_t^i | x^i) = \text{Cat}(\alpha_t \cdot e_{\{x^i\}} + (1 - \alpha_t) \cdot e_{\{[M]\}})$$

Reverse



If $z_t^i \neq [M]$ (already unmasked): $q(z_s^i | z_t, x^0) = \text{Cat}(e_{\{z_t^i\}})$ -> That is, just keep the token unchanged.

If $z_t^i = [M]$ (still masked): $q(z_s^i | z_t, x^0) = \text{Cat}(\left(\frac{1 - \alpha_s}{1 - \alpha_t}\right) \cdot e_{\{[M]\}} + \left(\frac{\alpha_s - \alpha_t}{1 - \alpha_t}\right) \cdot e_{\{x^{0i}\}})$

That is:

- Stay masked with probability $(1 - \alpha_s)/(1 - \alpha_t)$
- Get unmasked to the true token with probability $(\alpha_s - \alpha_t)/(1 - \alpha_t)$

Approximation of the reverse

Since we don't know x^0 during generation, we replace it with the model's prediction:

$$p_{\theta}(z_s^i | z_t) = q(z_s^i | z_t, x^0 \leftarrow x_{\theta}(z_t))$$

Where $x_{\theta}(z_t)$ is the denoising neural network that predicts the clean data from the noisy input.

Final Objective

$$\text{Loss}(\theta) = E_{\{t, x_0, x_t\}} [(\alpha'_t / (1 - \alpha_t)) \cdot \sum_{i=1}^L \mathbb{1}[x_t^i = [M]] \cdot (-\log p_{\theta}(x_0^i | x_t))]$$

- $t \sim \text{Uniform}[0, 1]$ is a randomly sampled timestep
- $x_0 \sim p_{\text{data}}$ is a clean sample; $x_t \sim q(\cdot | x_0)$ is the masked version at time t
- $\alpha'_t = d\alpha_t/dt$ is the derivative of the noise schedule
- L is the sequence length

Results - Training on 1B tokens

Test perplexities (PPL; ↓) on LM1B. †. Best diffusion value is bolded.

| | | Parameters | PPL (↓) |
|-----------------------------------|----------------------------|------------|----------------|
| <i>Autoregressive</i> | Transformer-X Base | 0.46B | 23.5 |
| | <i>OmniNet_T</i> | 100M | 21.5 |
| <i>Diffusion</i> | BERT-Mouth | 110M | ≤142.89 |
| | D3PM (absorb) | 70M | ≤77.50 |
| | Diffusion-LM | 80M | ≤118.62 |
| | DiffusionBert | 110M | ≤63.78 |
| | SEDD | 110M | ≤32.79 |
| <i>Autoregressive (Retrained)</i> | Transformer (33B tokens) | 110M | 22.32 |
| | Transformer (327B tokens) | | 20.86 |
| <i>Diffusion (Ours)</i> | MDLM (33B tokens) | 110M | ≤27.04 |
| | MDLM (327B tokens) | | ≤ 23.00 |

Results - Scale up to 524B tokens

Zero-shot validation perplexities (↓) of models trained for 524B tokens on OpenWebText. All perplexities for diffusion models are upper bounds.

| | PTB | Wikitext | LM1B | Lambada | AG News | Pubmed | Arxiv |
|--------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| AR (Retrained) | 82.05 | 25.75 | 51.25 | 51.28 | 52.09 | 49.01 | 41.73 |
| SEDD (Retrained) | 100.09 | 34.28 | 68.20 | 49.86 | 62.09 | 44.53 | 38.48 |
| MDLM (Ours) | 95.26 | 32.83 | 67.01 | 47.52 | 61.15 | 41.89 | 37.37 |

What's the problem? - No going back

Reverse
process

$$q_{s|t} (x_s^i | x_t, x_0) = \begin{cases} \text{Cat} (e_{x_t^i}) & x_t^i \neq m \\ \text{Cat} \left(\frac{1-\alpha_s}{1-\alpha_t} e_m + \frac{\alpha_s-\alpha_t}{1-\alpha_t} e_{x_0} \right) & x_t^i = m \end{cases}$$

If $z_t^i \neq [M]$ (already unmasked): $q(z_s^i | z_t, x^0) = \text{Cat}(e_{\{z_t^i\}})$ -> That is, just keep the token unchanged.

What's the problem? - No going back

Example: Math problem

Question: "What is $12 + 15 = ?$ "

Target output: "1" "2" "+" "1" "5" "=" "2" "7"

Start: [M] [M] [M] [M] [M] [M] [M] [M]

Bad order — pick the answer digit first (position 7):

Step 1: Pick position 7 → This is the tens digit of the answer

Model guesses → "3"  locked!

[M] [M] [M] [M] [M] [M] ["3"] [M]

Step 2: ...

Step 3: Until all tokens are unmasked →

"1" "6" "+" "1" "5" "=" "3" "1"

X The question was $12+15$, not $16+15$!

Cannot fix "31" — it's locked in.

In theory - what is MDM learning

MDM Loss = Order-Agnostic Training

Proposition 2.1. Assume $\alpha_0 = 1$, $\alpha_1 = 0$ and de-noising network p_θ is time-embedding free. Then $\mathcal{L}_\theta \leq -\mathbb{E}_{x_0 \sim p_{\text{data}}} [\log p_\theta(x_0)]$ and

$$\mathcal{L}_\theta = - \sum_{M \subseteq [L], i \in M} \frac{1}{|M|} \frac{1}{\binom{L}{|M|}} \mathbb{E}_{x_0 \sim p_{\text{data}}} [\log p_\theta(x_0^i | x_0[M])], \quad (1)$$

where $|M|$ is the size of the set M and $p_\theta(x_i | x_0[M])$ indicates the conditional probability of the i -th coordinate from $p_\theta(x_t)$.

- MDM loss = weighted sum over all possible mask patterns M
- ARM: only L subproblems (left-to-right prefix masks)
- MDM: $\Theta(L \cdot 2^L)$ subproblems — some are inevitably hard

MDM training objective decomposes into a sum over all possible masking patterns. This is equivalent to training a model that must work for every possible permutation ordering. ARMs only handle one ordering.

Planning for the best - Adaptive Inference

- An **ideal** MDM (perfect on all subproblems) generates correctly regardless of unmasking order
- **In practice**, MDM errors are imbalanced → different orderings give different quality
- Solution: Adaptive MDM inference — instead of randomly selecting which tokens to unmask, use a strategy:

| | Vanilla MDM | Adaptive MDM |
|----------|---|--|
| Step (a) | S = random subset of masked positions | S = F(θ , x_t) — strategically chosen |
| Step (b) | Sample $x_{s^i} \sim p_{\theta}(x^i x_t)$ | Same |

Planning for the best - Adaptive Inference

- Top-K Probability (Zheng et al., 2023):
 - Certainty at position $i = \max_j p_{\theta}(x^i = j \mid x_{-t})$
 - Unmask the K positions with highest max probability
 - Problem: Can be fooled when two values have near-equal high probabilities
- Top-K Probability Margin
 - Certainty at position $i = |p_{\theta}(x^i = j_1 \mid x_{-t}) - p_{\theta}(x^i = j_2 \mid x_{-t})|$
 - where j_1, j_2 are the two most probable values
 - Unmask the K positions with largest margin
 - Better uncertainty estimate when multiple values compete

Results

Table 2. Comparison of accuracy for solving the Sudoku puzzle.

| Method | # Param | Accuracy |
|------------------------|---------|----------|
| ARM (w/o ordering) | 42M | 9.73% |
| ARM (with ordering) | | 87.18% |
| MDM (vanilla) | 6M | 6.88% |
| MDM (Top probability) | | 18.51% |
| MDM (Top prob. margin) | | 89.49% |

Table 5. Comparison of accuracy for solving the hard Sudokus.

| Method | #Param | Accuracy |
|------------------------|--------|----------|
| ARM (with ordering) | 42M | 32.57 % |
| MDM (random) | 6M | 3.62 % |
| MDM (Top probability) | | 9.44 % |
| MDM (Top prob. margin) | | 49.88 % |

Table 4. Performance of different inference strategies for LLaDa 8B model on coding and math tasks.

| Method | HumanEval-Single | HumanEval-Multi | HumanEval-Split | Math | MMLU | ROCStories |
|------------------|------------------|-----------------|-----------------|--------------|--------------|---------------|
| Vanilla | 31.8% | 16.5% | 14.2% | 28.5% | 33.2% | 21.23% |
| Top probability | 32.9% | 20.8% | 18.4% | 31.3% | 36.5% | 21.10% |
| Top prob. margin | 33.5% | 25.4% | 22.3% | 34.3% | 35.4% | 21.41% |

But...

If model-logits alone are sufficient?

Example

Rule: The sequence must alternate between A and B

Valid sequences: A B A B A B
 B A B A B A

Scenario 1: Logits are misleading

Current state: [M] [M] [M] [M] [M] [M]

Model logits at every position:

Position 1: $P(A) = 50\%$, $P(B) = 50\%$

Position 2: $P(A) = 50\%$, $P(B) = 50\%$

...

Every position looks equally uncertain. The logits say "I have no idea." Top-K or Top-K Margin would say: all positions are equally confident — just pick randomly.

But actually:

- If you pick any position and commit to any value, the rest is fully determined!

The logits say "I'm uncertain everywhere" but the right strategy is "just pick anywhere and go."

Why?

The model predicts each position independently:

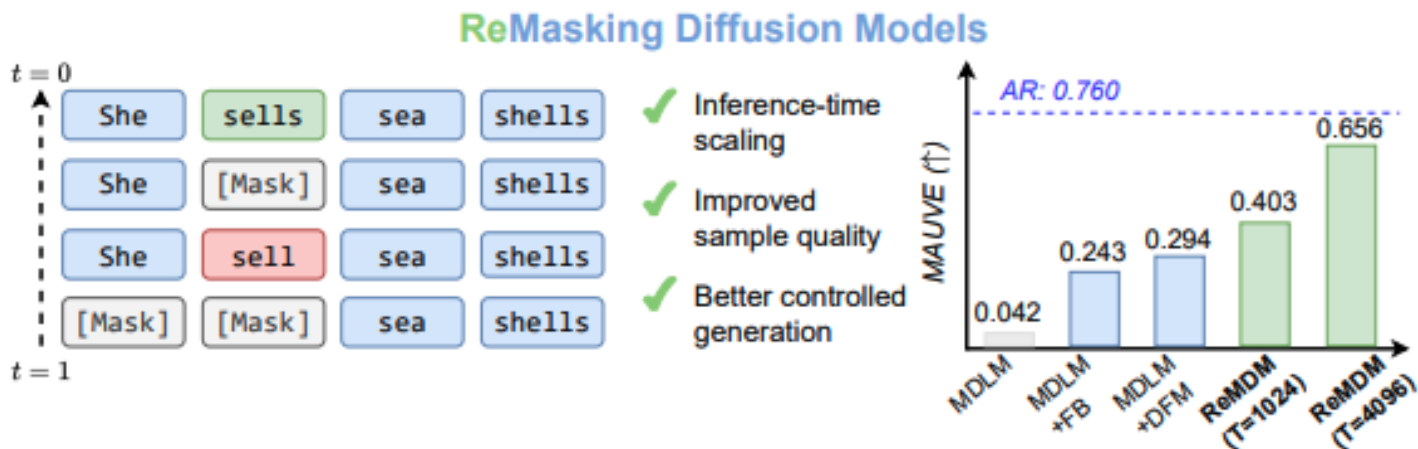
$p_{\theta}(x^i \mid x_{-t})$ ← predicts position i given current visible tokens

It does NOT reason about:

- How positions constrain each other
- What happens after committing to a value
- The global consistency of the full sequence

What else can we do?

We capture the error token and then remask it



What else can we do?

Pretrained MDMs can already detect their own mistakes

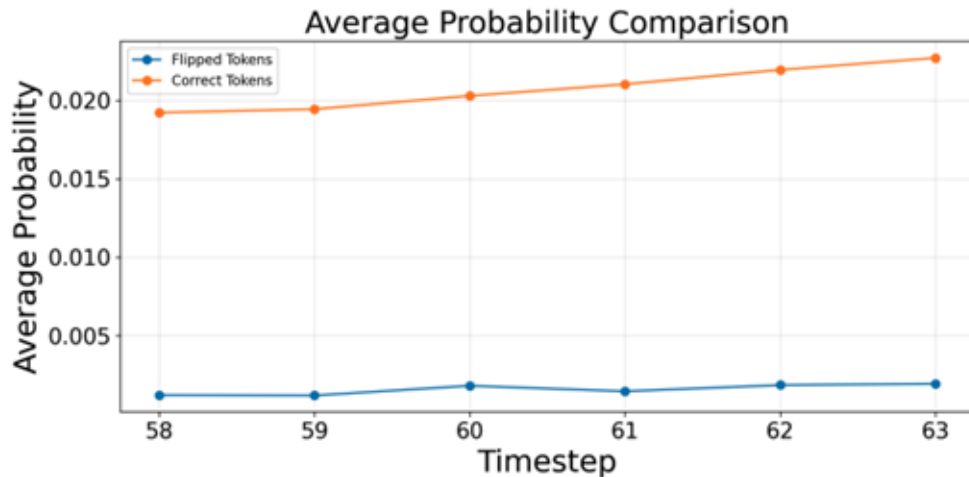


Figure 1: Average predicted probability of flipped tokens and correct tokens over 2000 samples. The x-axis denotes the time steps for generation (64 steps in total for text-to-image generation), while the y-axis denotes the average probability over all flipped positions and the correct position.



Figure 3: Comparison of generated images of Lumina-DiMOO, ReMDM, and our methods. Our method achieves better quality with less sampling steps.

Table 5: Evaluation of the proposed method on the GenEval benchmark based on MMaDA-8B-MixCoT [38].

| Method | Single | Two | Count | Color | Pos. | Attr. | Overall \uparrow |
|--------------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|--------------------|
| MMaDA-8B-MixCoT ¹ (Top K) | 0.93 | 0.47 | 0.31 | 0.81 | 0.16 | 0.26 | 0.49 |
| MMaDA-8B-MixCoT (Top K margin) | 0.88 | 0.49 | 0.29 | 0.79 | 0.15 | 0.28 | 0.48 |
| MMaDA-8B-MixCoT (ReMDM) | 0.91 | 0.59 | 0.38 | 0.78 | 0.20 | 0.19 | 0.51 |
| MMaDA-8B-MixCoT (Ours) | 0.91 | 0.67 | 0.39 | 0.84 | 0.15 | 0.18 | 0.52 |

We reevaluate the pre-trained MMaDA-8B-MixCoT on GenEval benchmarks for fair comparison. The generation step is 64.

Table 3: Evaluation of multimodal understanding on several benchmarks.

| Model | MMB \uparrow | SEED \uparrow | MMMU \uparrow |
|---------------------------|----------------|-----------------|-----------------|
| Lumina-DiMOO ^a | 58.7 | 71.4 | 41.4 |
| Lumina-DiMOO (ReMDM) | 57.8 | 74.3 | 43.4 |
| Lumina-DiMOO (Ours) | 60.7 | 74.2 | 44.0 |

^a We reevaluate the pre-trained Lumina-DiMOO on several multimodal understanding benchmarks for fair comparison. The generation step is 64 and the generated sequence length is 256.

THANK YOU