

Di[M]O: Distilling Masked Diffusion Models into One-step Generator

Yuanzhi Zhu, Xi Wang, Stéphane Lathuilière, Vicky Kalogeiton

LIX, École Polytechnique, CNRS, IPP, Inria, Univ. Grenoble Alpes, CNRS, LJK



Content

- Preliminaries
- Methods
- Results

Text

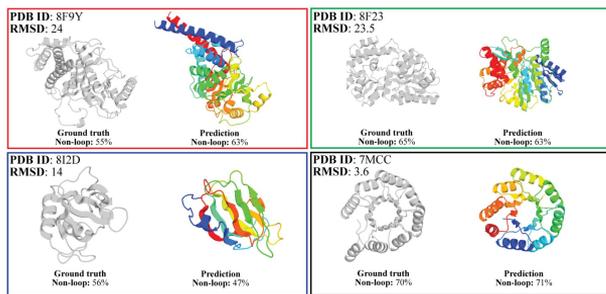
Multi-round Dialogue Case

User	Could you please share with me what the initial two lines of the renowned poem "The Road Not Taken"?
LLaDA	Certainly! The initial two lines of the renowned poem "The Road Not Taken" by Robert Frost are: "Two roads diverged in a yellow wood, And sorry I could not travel both." These lines set the stage for the poem, introducing the idea of a speaker facing a choice between two paths.
User	Please help me translate into Chinese.
LLaDA	Sure! The first two lines of "The Road Not Taken" by Robert Frost can be translated into Chinese as: "两条路分岔在黄色的树林中, 遗憾我不能同时走"
User	Please translate into German.
LLaDA	Sure! The first two lines of "The Road Not Taken" by Robert Frost can be translated into German as: "Zwei Wege trennten sich im gelben Wald, und leider konnte ich nicht beide tun."
User	Please also write a poem about life choices. I require 5 sentences, and the first word of each sentence starts with C as the first letter.
LLaDA	Certainly! Here's a poem about life choices, with each sentence starting with the letter "C": Choices cascade through life's journey, Creating connections with each decision. Careful consideration leads to better paths, Courageous choices shape our destiny, Choosing wisely guides us through the years.

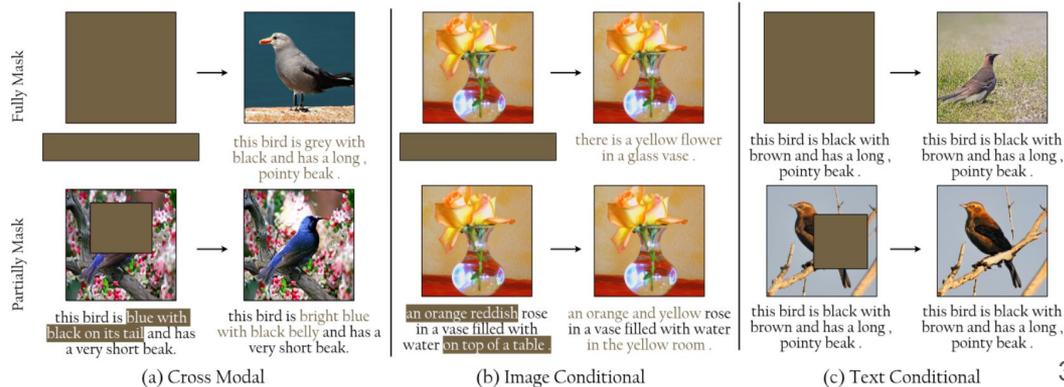
Image



Protein



Multi-modality



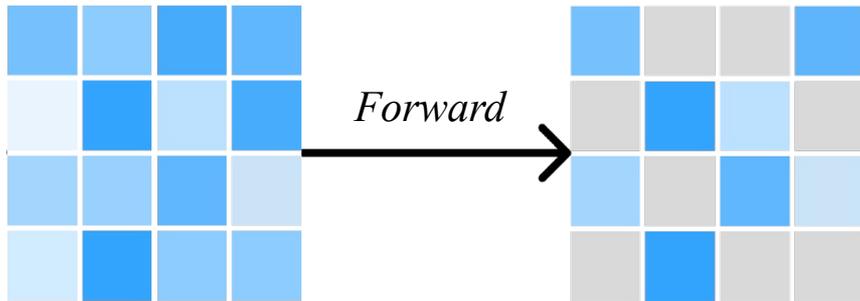
(a) Cross Modal

(b) Image Conditional

(c) Text Conditional

Mask Diffusion Models

Forward Process: mask *independently* each token at position i

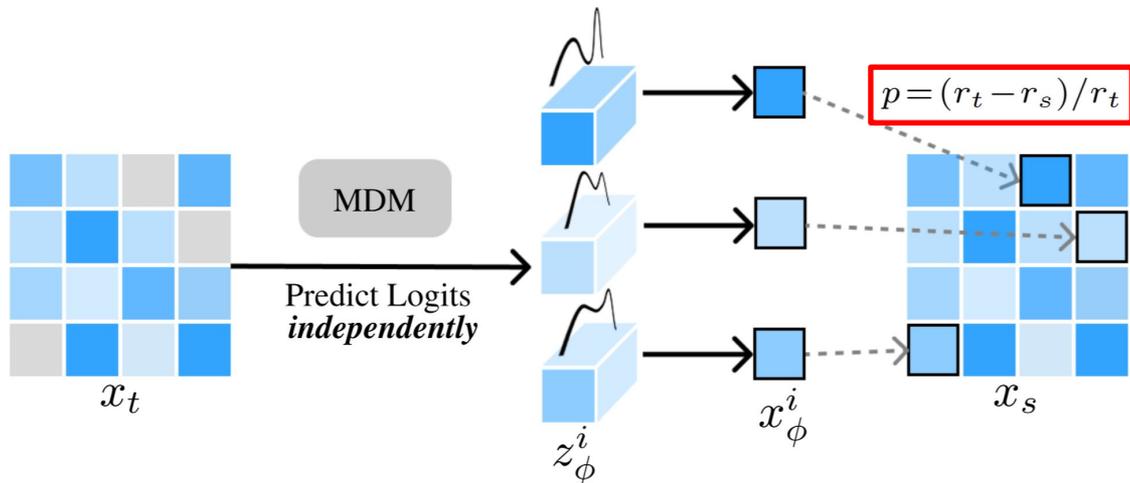


$$q_{t|0}(x_t|x_0) = \prod_{i=0}^{L-1} \text{Cat}(x_t^i; (1 - r_t)\delta(x_0^i) + r_t\delta([\mathbf{M}])))$$

Mask Diffusion Models

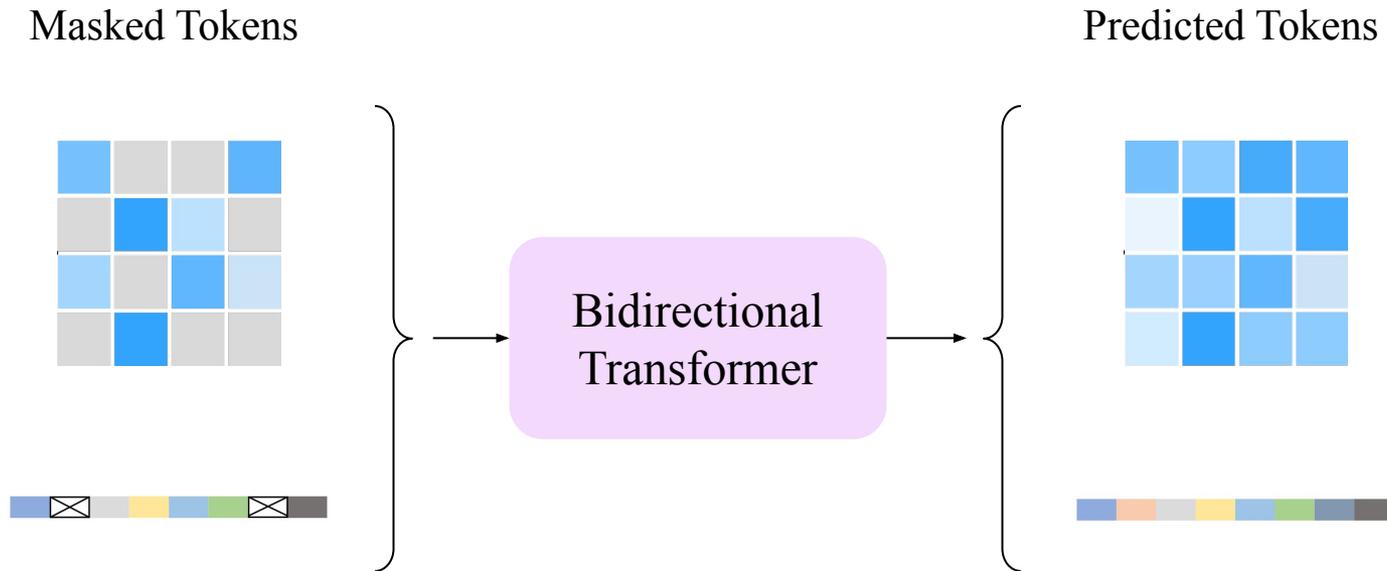
Reverse Process: fill masks

1. In parallel predict each token independently
2. Replace masks using strategies (e.g. **random**)



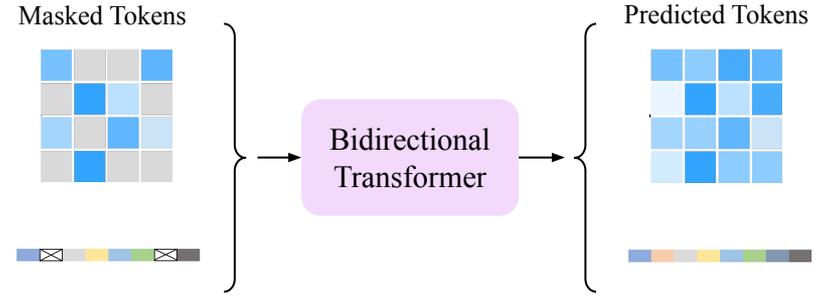
Predicted Probability: $p_\phi(x_0^i | x_t) := \text{softmax}(z_\phi^i(x_t) / \tau)$

Mask Diffusion Models

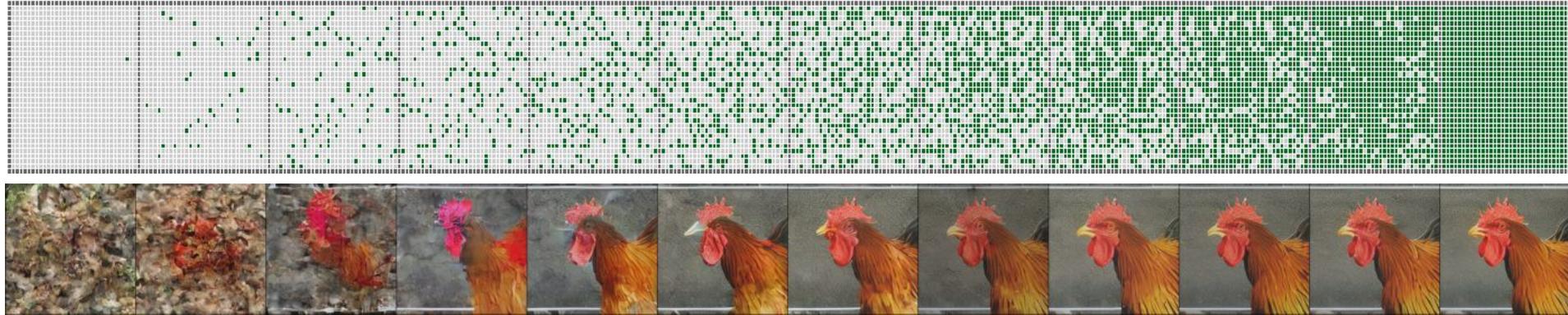


$$\mathcal{L}_{\text{MDM}} = \mathbb{E}_{x_0, t} \left[\left(\mathbb{E}_{q_{t|0}} \left[-\log p_{0|t}(x_0 | x_t, \phi) \right] \right) \right]$$

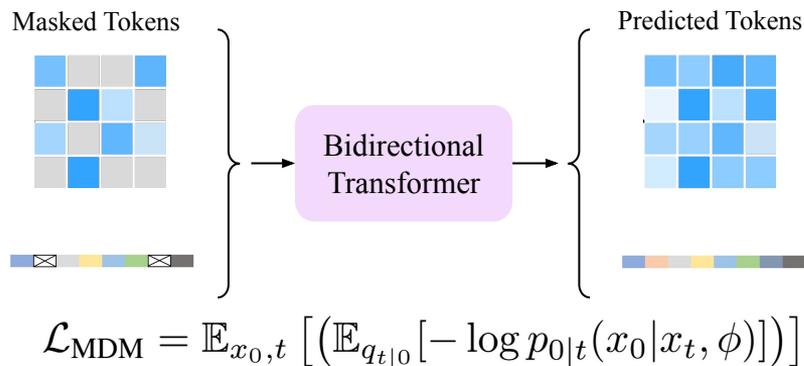
Mask Diffusion Models



Sampling Process



Promise of MDM



Theorem: Minimizing the *expected* cross-entropy loss is equivalent to maximizing the variational lower bound of the log-likelihood, i.e.:

$$-\mathbb{E}_{p_{\text{data}}(x_0)} [\log p_{\theta}(x_0)] \leq \mathcal{L}_{\text{MDM}}(\theta)$$

MDM loss (maximize likelihood) + Transformer (less inductive bias) + Data Scalability ?

Limitations of MDM

- Incorrect joint distribution for multi-token prediction



- Need sufficient steps for better performance



Ideas in Inference-time Scaling can Benefit Generative Pre-training Algorithms

Jiaming Song, Linqi Zhou

Luma AI

Theoretical Benefit and Limitation of Diffusion Language Model

Guhao Feng^{*1} Yihan Geng^{*1} Jian Guan² Wei Wu² Liwei Wang¹ Di He¹

Multi-token prediction (MTP) is of great interest to the language modeling community because of its potential to achieve faster inference [GIR⁺24], which allows efficient inference-time scaling. However, the current multi-token prediction models often predict the softmax values of multiple tokens in parallel, which is a naive conditional independence assumption (*i.e.*, naïve Bayes). We argue that this inference design greatly limits the capacity of the model distribution and more efforts should be spent resolving this fundamental issue.

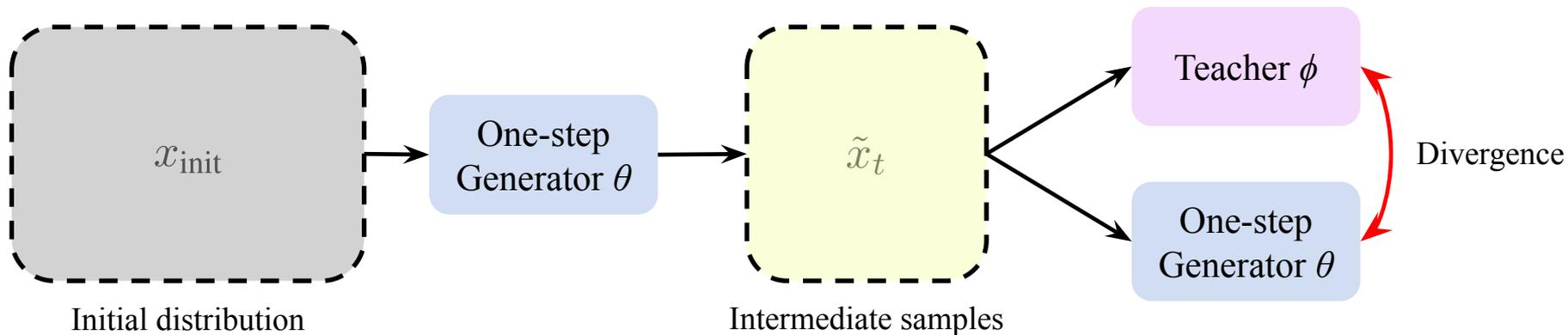
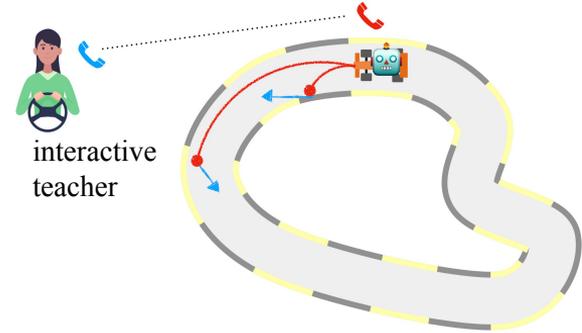
such as a reasoning chain—we show that the required sampling steps must scale linearly with sequence length to obtain “correct” sequences, thereby eliminating MDM’s efficiency advantage over autoregressive models. Our analysis estab-

Content

- Preliminaries
- Methods
- Results

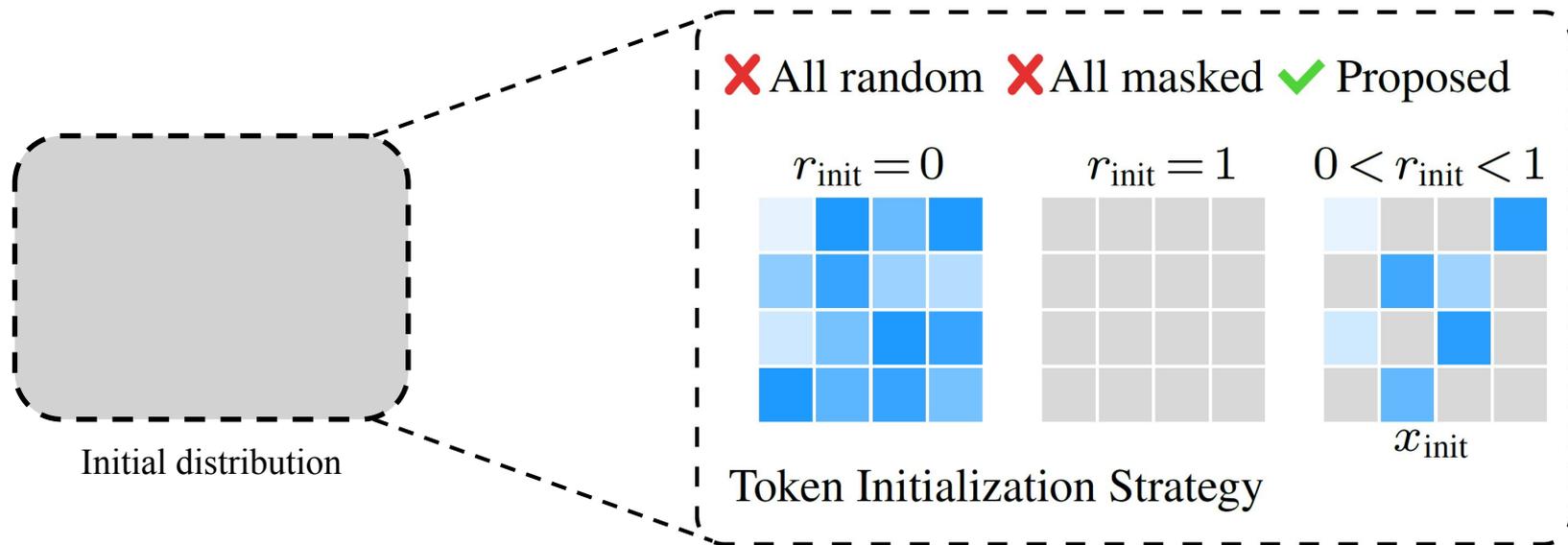
Main Idea: Distillation in an *On-Policy* fashion

Match teacher and generator for all possible intermediate samples



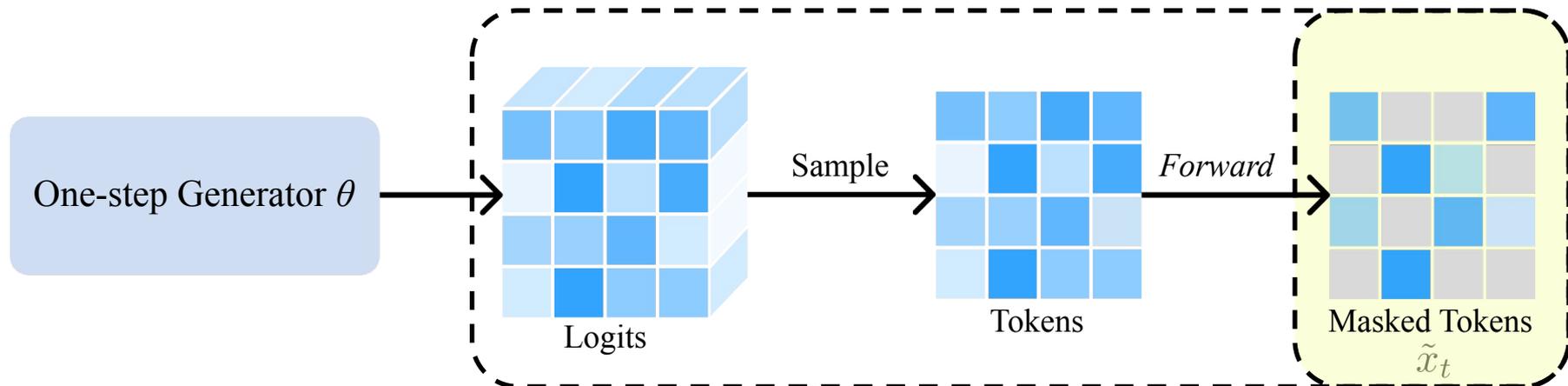
$$\mathcal{L}_{\text{Di[M]O}}(\theta) := \mathbb{E}_{x_{\text{init}}, t} \left[w(t) \left(\mathbb{E}_{q_t | o} \left[D(p_\phi || p_\theta)(\tilde{x}_t) \right] \right) \right]$$

Initial Distribution



Intermediate Samples

Via *forward mask diffusion process*



Consider loss on each intermediate state \tilde{x}_t

$$\mathcal{L}_{\text{Di}[M]\text{O}}(\theta) := \mathbb{E}_{x_{\text{init}}, t} \left[w(t) \left(\mathbb{E}_{q_t | o} \left[D(p_\phi || p_\theta)(\tilde{x}_t) \right] \right) \right]$$

Further decompose into *Token-level Divergence*

$$D(p_\phi || p_\theta)(\tilde{x}_t) := \frac{1}{L_M} \sum_{\substack{i=1 \\ \tilde{x}_t^i = [M]}}^L D(p_\phi(x_0^i | \tilde{x}_t) || p_\theta(x_0^i | \tilde{x}_t))$$

Gradient of Divergence:

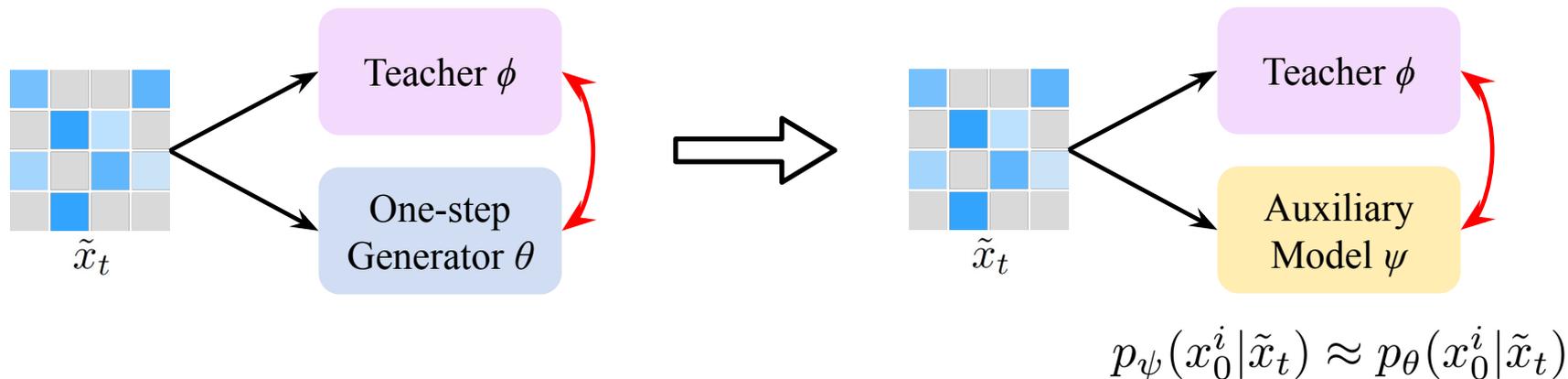
$$\nabla_{\theta} D((p_{\phi} || p_{\theta})(\tilde{x}_t)) = \nabla_{z_{\theta}} D(p_{\phi} || p_{\theta})(\tilde{x}_t) \frac{dz_{\theta}(\tilde{x}_t)}{d\theta}$$

Intractable terms

$$p_{\theta}(x_0^i | \tilde{x}_t) = \text{softmax}(z_{\theta}(\tilde{x}_t))$$

Approximation: $p_{\theta}(x_0^i | \tilde{x}_t)$

Introduce an *auxiliary model* to approximate the student output on intermediate samples

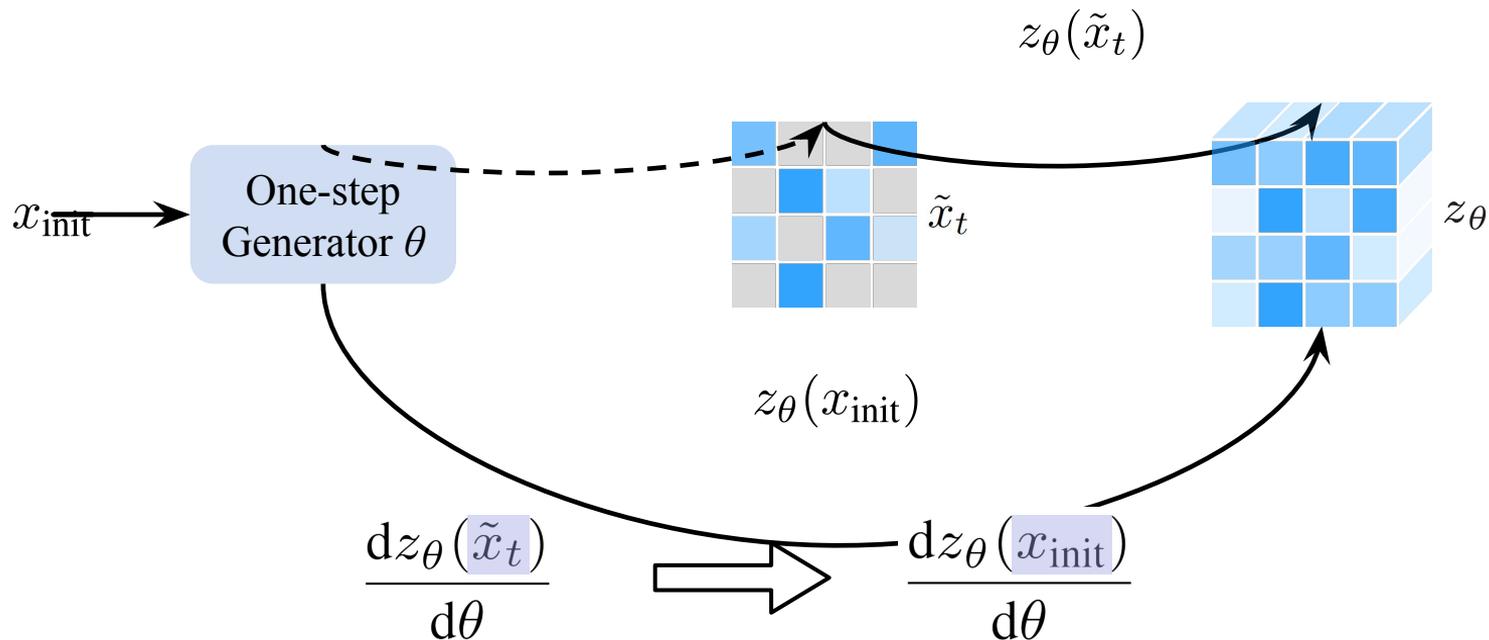


$$\nabla_{z_{\theta}} D(p_{\phi} || p_{\theta})(\tilde{x}_t) \quad \longrightarrow \quad \nabla_{z_{\psi}} D(p_{\phi} || p_{\psi})(\tilde{x}_t)$$

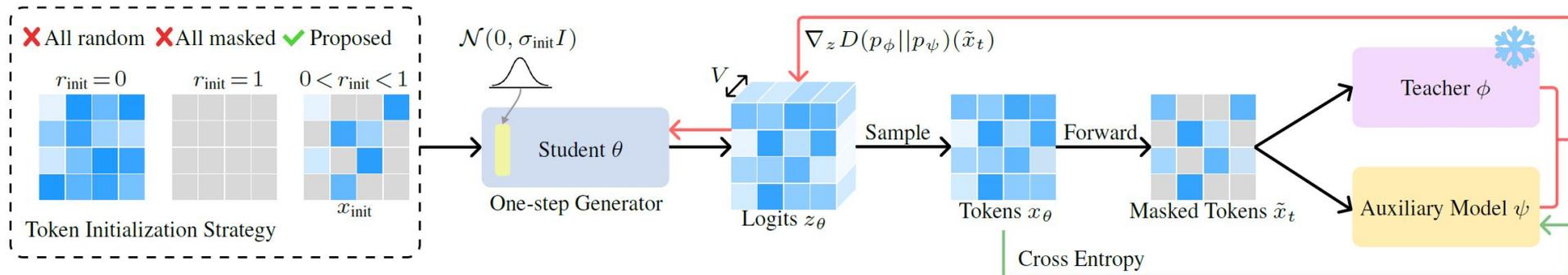
Approximation: $z_{\theta}(\tilde{x}_t)$

We approximate the output logits based on the *consistency assumption*

Also, we require this term as effective gradient for the generator

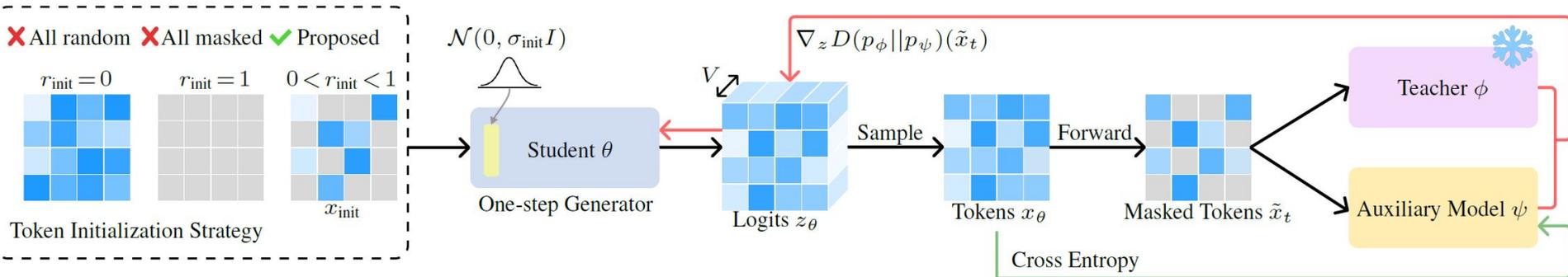


Overview of the Method

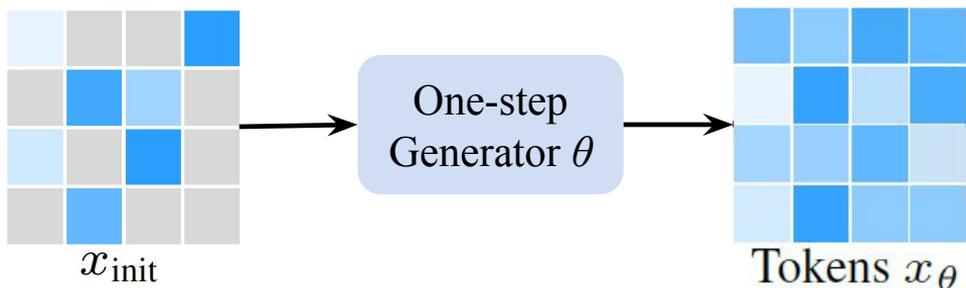


$$\nabla_\theta \mathcal{L}_{\text{Di}[M]O} \approx \mathbb{E}_{x_{\text{init}}, t} \left[w(t) \left(\mathbb{E}_{q_t | O} \left[\nabla_{z_\psi} D(p_\phi || p_\psi)(\tilde{x}_t) \quad \frac{dz_\theta(x_{\text{init}})}{d\theta} \right] \right) \right]$$

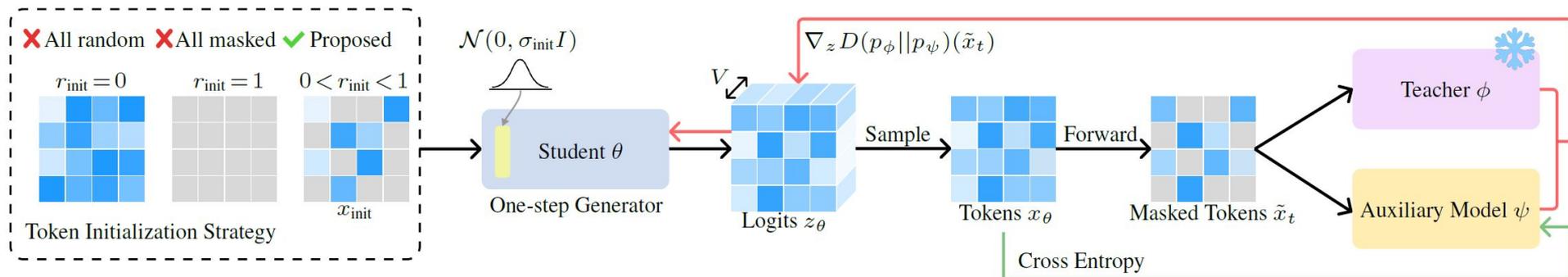
Overview of the Method



- From random initial states to samples from *correct joint distribution* 😊
- Greatly reduce the sampling steps \rightarrow One step generation 😊



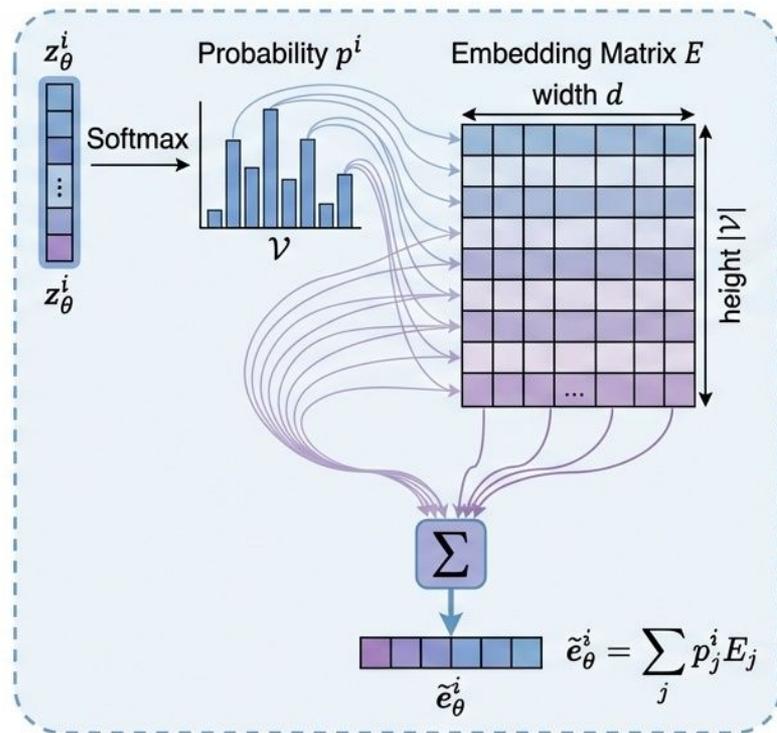
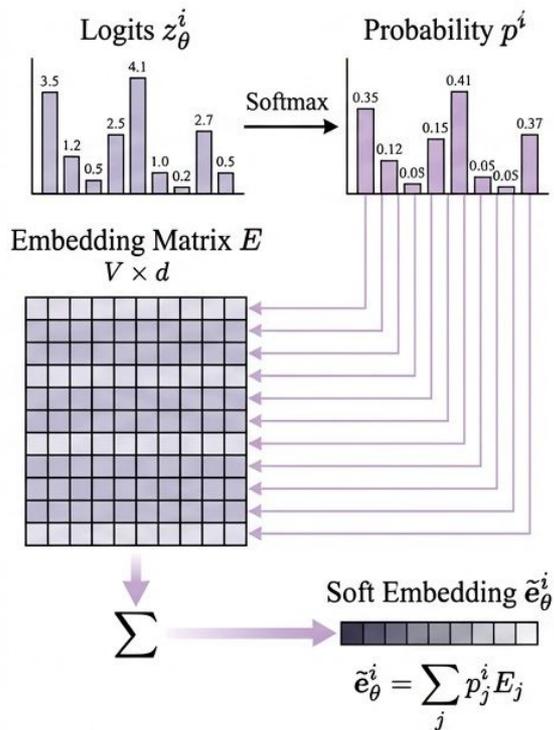
Limitation of the Di[M]O



All supervision comes from the teacher model, bottlenecked!!

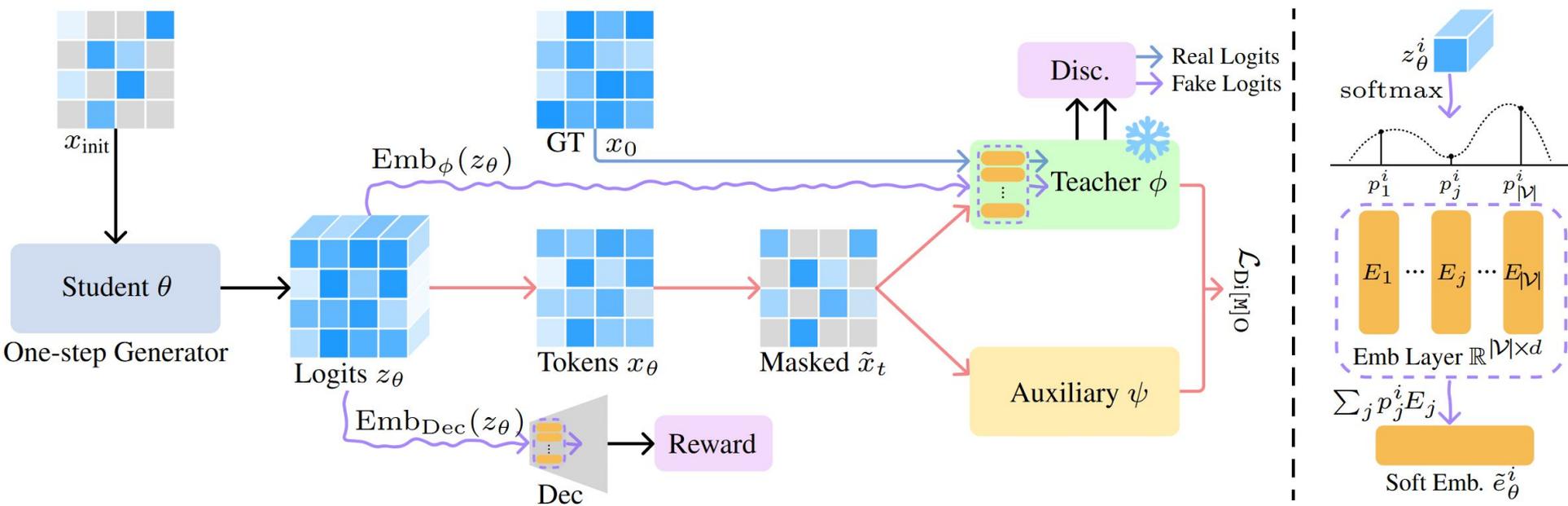
Augment with GAN and Reward via **Soft Embedding**

Two illustrations of **Soft Embedding**



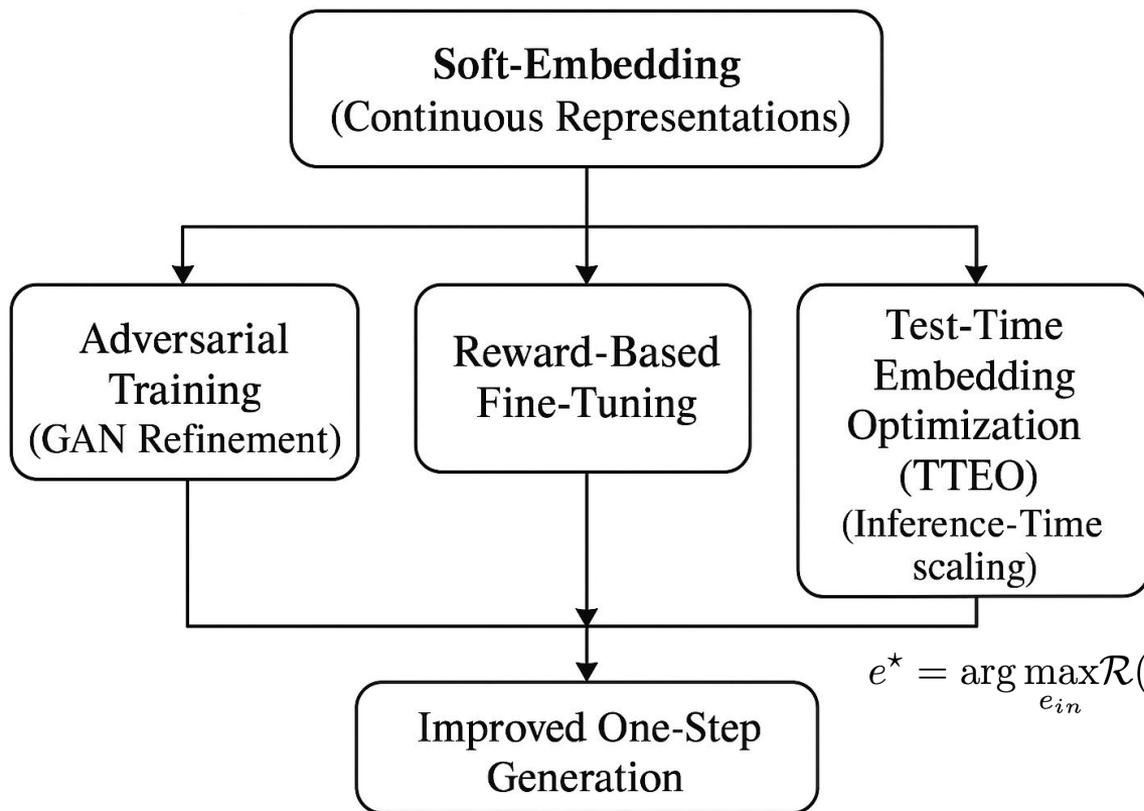
Augment with GAN and Reward via **Soft Embedding**

Soft-Di[M]O pipeline with **Soft Embedding**



$$\mathcal{L}_{gen}(\theta) = \mathcal{L}_{Di[M]O}(\theta) + w_{GAN} \mathcal{L}_{GAN}(\theta) + w_{reward} \mathcal{L}_{reward}(\theta)$$

Soft Embedding Enables Post-training and TTS



$$e^* = \arg \max_{e_{in}} \mathcal{R}(\text{Dec}(\text{Emb}_{\text{Dec}}(z_{\theta}(e_{in}))), c)$$

Overview of the Method

Algorithm 1 Di[M]O Distillation

Require: Pre-trained teacher model ϕ , condition dataset \mathcal{D}

- 1: $\theta \leftarrow \text{copyWeights}(\phi)$, $\psi \leftarrow \text{copyWeights}(\phi)$ // initialize
 - 2: **repeat**
 - 3: *// Generate tokens x_0*
 - 4: Sample $x_{\text{init}} \sim p_{\text{init}}$, $c \sim \mathcal{D}$ // with strategy in [Sec. 4.3](#)
 - 5: Get generator logits $z_\theta(x_{\text{init}}, c) \in \mathbb{R}^{B \times h \times w \times V}$
 - 6: $x_\theta \in \mathbb{R}^{B \times h \times w} \xleftarrow{\text{sample}} p_\theta(x_0 | x_{\text{init}}) = \text{softmax}(z_\theta(x_{\text{init}}, c))$
 - 7: *// Update generator θ*
 - 8: Sample $t \sim \mathcal{U}[0, 1]$, $\tilde{x}_t \sim q_{t|0}(\tilde{x}_t | x_\theta(x_{\text{init}}, c))$ // Forward
 - 9: Calculate $p_\phi(x_0 | \tilde{x}_t, c)$ and $p_\psi(x_0 | \tilde{x}_t, c)$
 - 10: Update θ with the loss gradient $\nabla_\theta \mathcal{L}_{\text{Di[M]O}}$ (Eq. (7))
 - 11: *// Update auxiliary model ψ*
 - 12: Sample $t' \sim \mathcal{U}[0, 1]$, $\tilde{x}_{t'} \sim q_{t'|0}(\tilde{x}_{t'} | x_\theta(x_{\text{init}}, c))$
 - 13: Update ψ with cross entropy loss (Eq. (3))
 - 14: **until** convergence
 - 15: **Return** one-step generator θ
-

Algorithm 1 Soft-Di[M]O Distillation

Require: Pre-trained teacher model ϕ , Pre-trained one-step generator θ_0 (optional), condition dataset \mathcal{D}_c , ground truth dataset \mathcal{D}_d , loss weights w_{GAN} and w_{reward}

- 1: $\theta \leftarrow \text{copyWeights}(\phi)$, $\theta \leftarrow \text{copyWeights}(\theta_0)$ (optional), $\psi \leftarrow \text{copyWeights}(\phi)$ // initialize
 - 2: **repeat**
 - 3: *### Generate logits z_θ and tokens x_θ*
 - 4: Sample $x_{\text{init}} \sim p_{\text{init}}$, $c \sim \mathcal{D}_c$ // with strategy in Di[M]O
 - 5: Get generator logits $z_\theta(x_{\text{init}}, c) \in \mathbb{R}^{B \times h \times w \times |V|}$
 - 6: $x_\theta \in \mathbb{R}^{B \times h \times w} \xleftarrow{\text{sample}} p_\theta(x_0 | x_{\text{init}}) = \text{softmax}(z_\theta(x_{\text{init}}, c))$
 - 7: *### Update generator θ*
 - 8: Sample $t \sim \mathcal{U}[0, 1]$, $\tilde{x}_t \sim q_{t|0}(\tilde{x}_t | x_\theta(x_{\text{init}}, c))$ // Forward
 - 9: Calculate $p_\phi(x_0 | \tilde{x}_t, c)$ and $p_\psi(x_0 | \tilde{x}_t, c)$
 - 10: *# calculate Di[M]O loss*
 - 11: $\mathcal{L}_{\text{Di[M]O}}(\theta) \leftarrow \mathbb{E}_{x_{\text{init}}, t, x = G_\theta(x_{\text{init}})} \left[w(t) \left(\mathbb{E}_{q_{t|0}} [\nabla_{z_\psi} D_{\text{div}}(p_\phi || p_\psi)(\tilde{x}_t)] \right) \right]$
 - 12: Calculate soft embedding $\tilde{e}_\theta = \text{Emb}(z_\theta) = E^\top p_\theta(x_0 | x_{\text{init}})$
 - 13: *# calculate GAN loss*
 - 14: $\mathcal{L}_{\text{GAN}}(\theta) \leftarrow \mathbb{E}_{x_{\text{init}} \sim p_{\text{init}}, r \sim p_{r_{\text{GAN}}}} [-\log(D_\eta(\text{Emb}(z_\theta)_r, r))]$
 - 15: *# calculate reward loss*
 - 16: $\mathcal{L}_{\text{reward}}(\theta) \leftarrow -\sum_i \lambda_i \mathcal{R}_i(\text{Dec}(\text{Emb}_{\text{Dec}}(z_\theta), c))$
 - 17: *# calculate total loss and update*
 - 18: Update θ using gradient of $\mathcal{L}_{\text{gen}}(\theta) = \mathcal{L}_{\text{Di[M]O}}(\theta) + w_{\text{GAN}} \mathcal{L}_{\text{GAN}}(\theta) + w_{\text{reward}} \mathcal{L}_{\text{reward}}(\theta)$
 - 19: *### Update auxiliary model ψ*
 - 20: Sample $t' \sim \mathcal{U}[0, 1]$, $\tilde{x}_{t'} \sim q_{t'|0}(\tilde{x}_{t'} | x_\theta(x_{\text{init}}, c))$
 - 21: Update ψ with cross entropy loss (Eq. (1))
 - 22: *### Update discriminator η*
 - 23: Sample $t'' \sim \mathcal{U}[0, 0.95]$, calculate $\tilde{e}_{t''} = \text{Emb}(z_\theta)_{t''}$
 - 24: Sample real data $(x_{\text{gt}}, c) \sim \mathcal{D}_d$, calculate $\text{Emb}(x_{\text{gt}})_{t''}$
 - 25: Update η with GAN objective (Eq. (4))
 - 26: **until** convergence
 - 27: **Return** one-step generator θ
-

Content

- Preliminaries
- Methods
- Results

Quantitative Results (ImageNet)

Table 1: Class-conditional ImageNet-256 results with MaskBit teacher.

	Method	Steps (\downarrow)	#Params	FID(\downarrow)	IS(\uparrow)	Pr(\uparrow)	Rec(\uparrow)
Diffusion & Flow	ADM (Dhariwal & Nichol, 2021)	250	554M	4.59	186.7	0.82	0.52
	U-ViT-H (Bao et al., 2023)	50	501M	2.29	263.9	0.82	0.57
	DiT-XL/2 (Peebles & Xie, 2023)	250	675M	2.27	278.2	0.83	0.57
	SiT-XL/2 (Ma et al., 2024)	250	675M	2.06	277.5	0.83	0.59
Masked & AR	LlamaGen-3B (Sun et al., 2024)	576	3.1B	2.18	263.3	0.81	0.58
	MAR (Li et al., 2024e)	100	400M	1.98	-	-	-
	MAGViT-v2 (Yu et al., 2023)	64	307M	1.78	319.4	-	-
	MaskBit (Weber et al., 2024)	64	305M	1.66	320.0	0.81	0.60
Few-Step from Scratch	iCT (Song et al., 2023)	1	675M	34.24	-	-	-
		2	675M	20.3	-	-	-
	MeanFlow (Geng et al., 2025)	1	676M	3.43	-	-	-
	+ train longer	2	676M	2.93	-	-	-
Discrete Distillation	LlamaGen-L-DD (Liu et al., 2024)	2	326M	7.58	237.5	0.84	0.37
	Di[M]O-MaskGit (Zhu et al., 2025b)	1	174M	6.91	214.1	0.83	0.38
	Di[M]O-MaskBit (Zhu et al., 2025b)	1	305M	2.89	310.1	0.87	0.49
	Soft-Di[M]O-MaskBit	1	305M	1.96	281.4	0.84	0.55
	+ train longer	1	305M	1.56	273.2	0.81	0.60

Table 2: Quantitative results on class-conditional ImageNet-256 with MaskGit teacher. \dagger We reproduced Halton Sampler on ImageNet-256 with MaskGit.

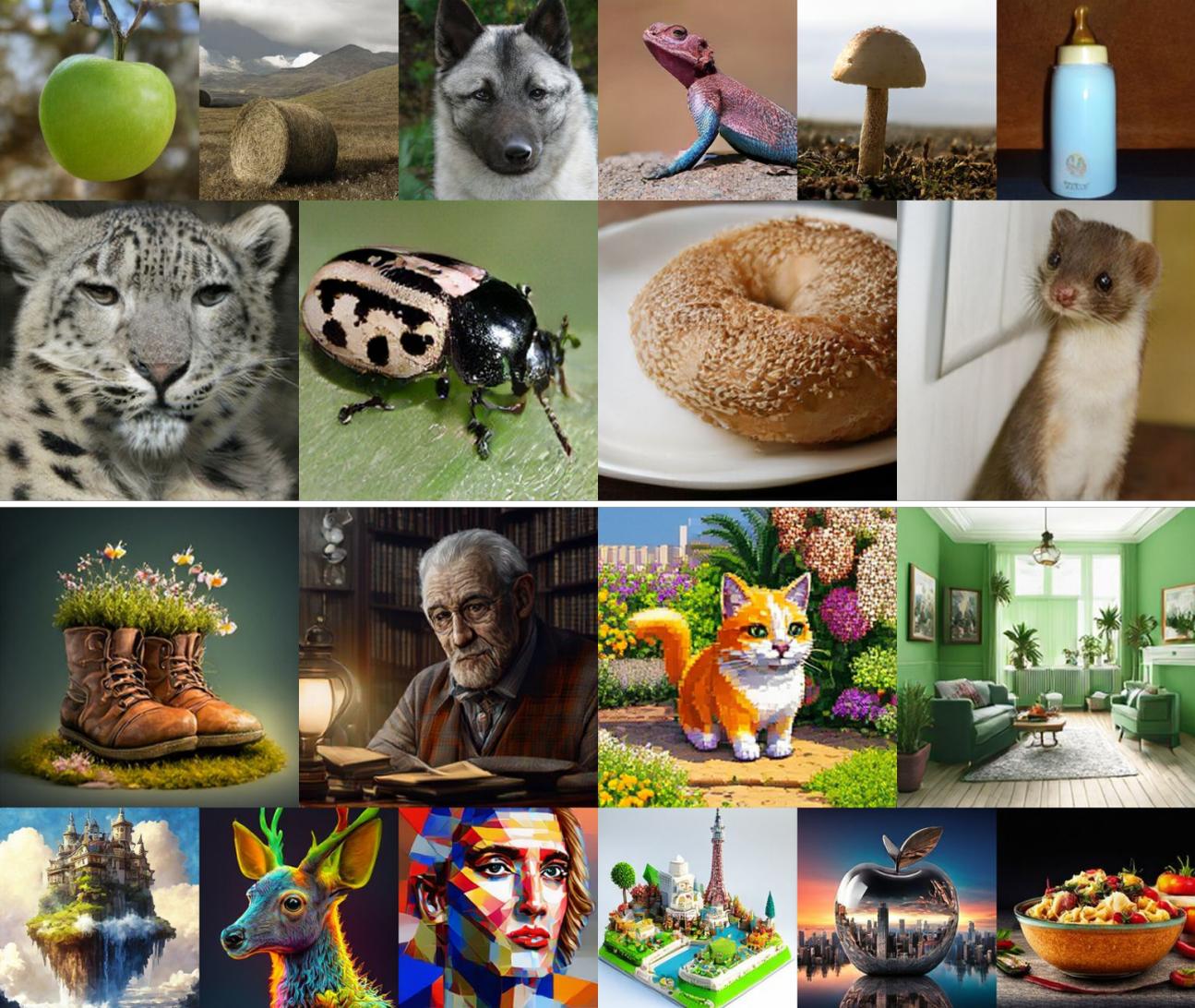
	Method	Steps (\downarrow)	FID (\downarrow)	IS (\uparrow)	Prec. (\uparrow)	Rec. (\uparrow)	Den. (\uparrow)	Cov. (\uparrow)
Teacher	MaskGit (Besnier & Chen, 2023)	16	6.60	224.1	0.83	0.40	1.25	0.98
	MaskGit (Besnier & Chen, 2023)	8	6.66	221.6	0.83	0.40	1.23	0.97
	MaskGit (Besnier & Chen, 2023)	4	10.73	192.3	0.75	0.31	1.01	0.92
	MaskGit (Besnier & Chen, 2023)	2	91.35	13.4	0.18	0.16	0.09	0.12
Sampler	θ -trapezoidal (Ren et al., 2025)	64	6.70	-	-	-	-	-
	θ -trapezoidal (Ren et al., 2025)	32	7.10	-	-	-	-	-
	Halton \dagger (Besnier et al., 2025)	32	9.62	296.2	0.90	0.28	1.47	0.98
	Halton \dagger (Besnier et al., 2025)	16	8.73	283.2	0.89	0.28	1.46	0.98
	Halton \dagger (Besnier et al., 2025)	8	7.73	248.8	0.86	0.31	1.37	0.97
Discrete Distillation	SDTT (Deschenaux & Gulcehre, 2024)	4	8.97	205.0	0.88	0.41	1.43	0.97
	SDTT (Deschenaux & Gulcehre, 2024)	1	90.40	14.0	0.31	0.13	0.21	0.34
	di4c (Hayakawa et al., 2024)	4	6.79	209.2	-	-	-	-
	di4c-d (Hayakawa et al., 2024)	4	6.57	213.6	-	-	-	-
	ReDi ¹ (Yoo et al., 2025)	4	7.58	228.0	0.87	0.46	1.33	0.98
	ReDi ² (Yoo et al., 2025)	4	7.86	240.0	0.87	0.44	1.31	0.97
	ReDi ³ -distill (Yoo et al., 2025)	1	11.68	182.0	0.83	0.44	1.25	0.96
	Di[M]O (Zhu et al., 2025b)	1	6.91	214.0	0.83	0.38	1.26	0.97
Soft-Di[M]O	1	6.40	214.8	0.83	0.39	1.27	0.97	

Quantitative Results (Text to Image)

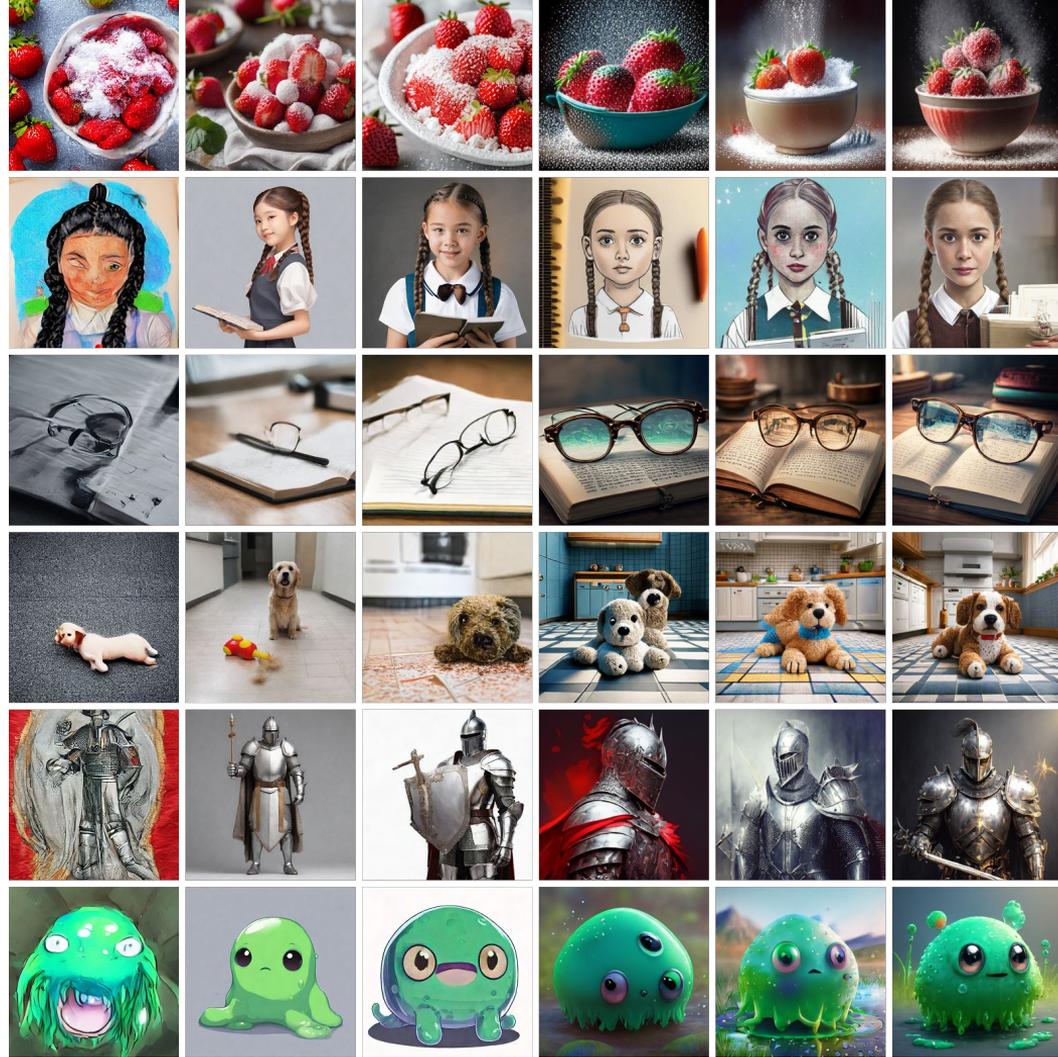
Table 3: Comparison of text-to-image generation methods across multiple metrics. Methods marked with † are evaluated by running their publicly available checkpoints; all other numbers are taken from the corresponding papers as reported. † denotes the higher is better and ‡ denotes the lower is better.

	Methods	Steps (‡)	#Params	FID ‡	CLIP †	GenEval †						HPS V2.1 †					
						Single	Two	Counting	Colors	Position	Color Attr.	Overall	Anim.	Concept	Painting	Photo	Averaged
Pre-train Models	LDM (Rombach et al., 2022)	50	1.4B	12.64	-	0.92	0.29	0.23	0.70	0.02	0.05	0.37	20.63	19.65	19.79	21.26	20.34
	DALLE 2 (Ramesh et al., 2022)	-	4.2B	10.39	-	0.94	0.66	0.49	0.77	0.10	0.19	0.52	26.38	24.51	24.93	25.55	25.34
	SDXL (Podell et al., 2023)	50	2.6B	6.63	0.290	0.98	0.74	0.39	0.85	0.15	0.23	0.55	32.84	31.36	30.86	27.48	30.63
	Meisssonic (Kim et al., 2025)	32	1.0B	50.13	0.318	0.92	0.53	0.33	0.80	0.08	0.13	0.46	31.67	27.27	29.67	29.93	29.63
	MaskGen-L (Kim et al., 2025)	16	0.6B	22.64	0.312	0.97	0.55	0.38	0.80	0.08	0.14	0.48	29.09	27.54	27.32	25.87	27.60
Continuous Distillation	InstaFlow (Liu et al., 2023b)	1	0.9B	13.10	-	0.88	0.21	0.20	0.66	0.03	0.03	0.33	21.25	21.12	21.41	20.92	21.18
	SiD-LSG† (Zhou et al., 2024a)	1	0.9B	8.15	0.304	0.93	0.37	0.21	0.57	0.03	0.03	0.36	23.08	21.37	22.55	21.53	22.13
	RG-LCM (HPS)† (Li et al., 2024b)	2	0.9B	24.04	-	0.97	0.54	0.35	0.82	0.07	0.14	0.48	30.85	33.66	33.35	33.66	32.88
	TDM† (Luo et al., 2025c)	4	0.9B	20.44	-	0.99	0.57	0.49	0.78	0.09	0.09	0.50	32.91	31.73	32.18	29.95	31.37
	SDXL-LCM† (Luo et al., 2023a)	1	2.6B	72.50	0.286	0.75	0.11	0.14	0.59	0.01	0.03	0.27	18.48	19.57	17.88	18.95	18.72
	SDXL-LCM† (Luo et al., 2023a)	4	2.6B	17.83	0.327	0.99	0.57	0.39	0.86	0.09	0.18	0.51	27.17	29.02	25.52	27.26	27.24
	SDXL-Turbo† (Sauer et al., 2024b)	1	2.6B	19.40	0.342	0.99	0.65	0.52	0.87	0.12	0.19	0.55	28.67	30.91	28.45	26.62	28.66
	SDXL-DMD2† (Yin et al., 2024a)	1	2.6B	14.49	0.343	0.99	0.68	0.48	0.90	0.08	0.19	0.55	28.29	30.36	27.19	27.92	28.44
Discrete Distillation	Di[M]O-Meisssonic (Zhu et al., 2025b)	1	1.0B	38.45	0.322	0.91	0.53	0.22	0.75	0.07	0.11	0.43	27.29	28.34	28.25	30.47	28.59
	Di[M]O-MaskGen-L (Zhu et al., 2025b)	1	0.6B	24.15	0.299	0.93	0.39	0.35	0.74	0.07	0.08	0.42	27.30	28.83	27.06	25.38	27.14
	Soft-Di[M]O-Meisssonic	1	1.0B	28.33	0.319	0.98	0.75	0.39	0.83	0.10	0.14	0.53	30.45	32.80	32.54	33.63	32.35
	Soft-Di[M]O-Maskgen-L	1	0.6B	23.43	0.321	0.98	0.59	0.41	0.81	0.09	0.18	0.51	29.51	30.62	29.34	28.06	29.38
	+ TTS	1	0.6B	-	-	0.99	0.78	0.68	0.83	0.14	0.33	0.63	30.01	32.33	32.20	33.25	31.95

Visual Results



Visual Results



InstaFlow s-1

LCM-SDXL s-4

SDXL-DMD2 s-4

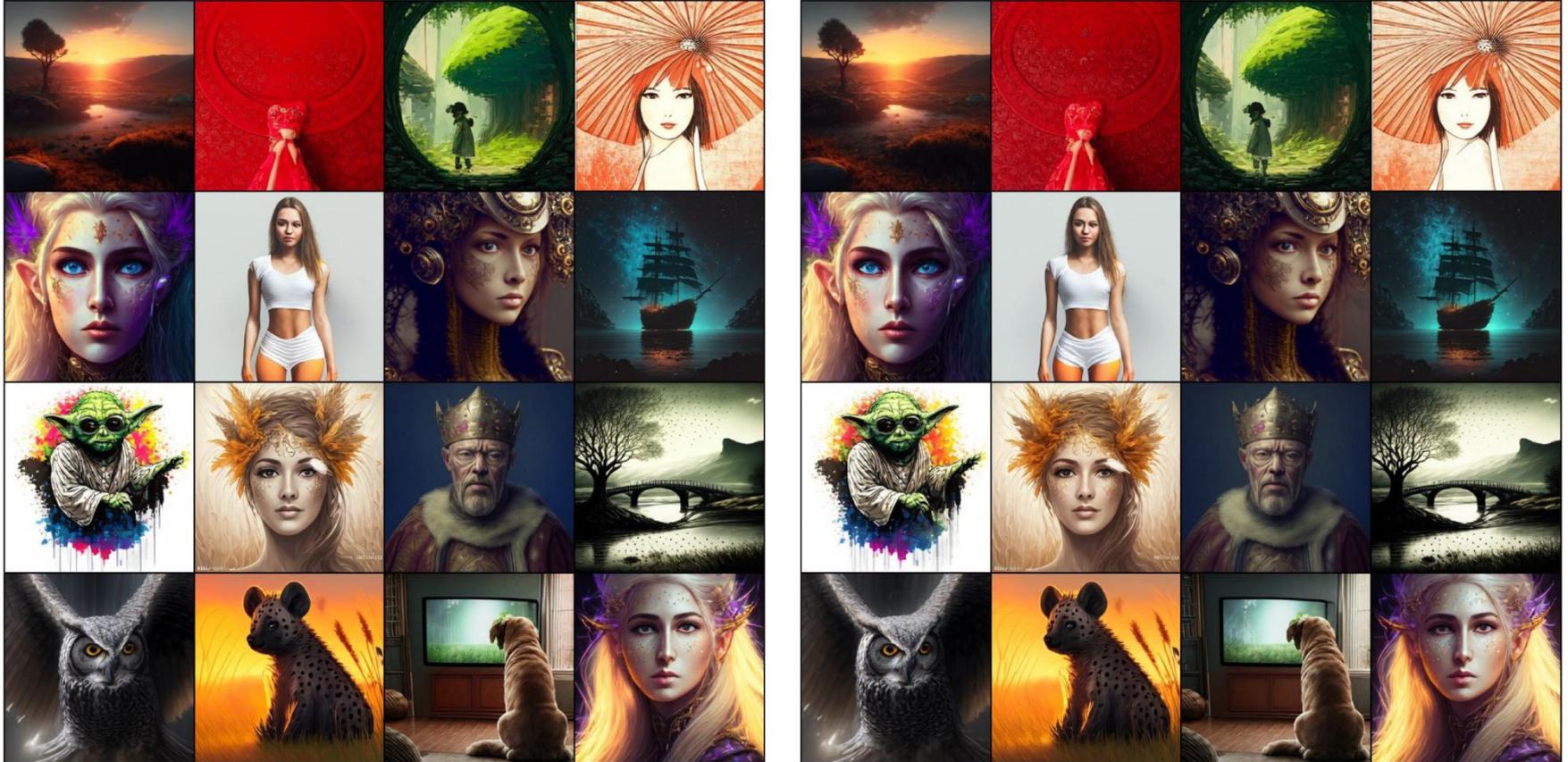
Di[M]O s-1

Soft-Di[M]O s-1

Soft-Di[M]O +TTS

Thank You!

Comparison of decoded images produced from sampled discrete tokens (left) and from soft embeddings (right)



H.8 TEST-TIME EMBEDDING OPTIMIZATION (TTEO) WITH MULTI-START STRATEGY

To reduce computational cost while maintaining performance, we employ TTEO combined with a multi-start Best-of-N (BoN) strategy, where multiple TTEO runs are executed in parallel for each prompt. In Fig. 14, we present the GenEval scores as a function of both the number of BoN seeds N and the number of TTEO optimization iterations. Our results demonstrate that while the standard BoN baseline provides performance improvements, incorporating TTEO yields additional gains, thanks to soft embeddings. For all experiments reported in the main paper, we use $N = 4$ seeds and 4 TTEO iterations, which requires approximately the same inference time as pure BoN with $N = 64$.

