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

Supplementary Material

The supplementary material is organized as follows:

- Appendix A: Broader impacts of this work.
- Appendix B: Relevant derivations of the approximated divergence gradient.
- Appendix C: Discussion of additional related works.
- Appendix D: Detailed experiment setup.
- Appendix E: Additional experiments and corresponding findings.
- Appendix F: Failure cases where the generation quality does not match that of the teacher model.
- Appendix G: Visualization and example of the mode-seeking/covering behaviors of the generalized Jeffrey divergence.
- Appendix H: Additional visual results of one-step generations from our distilled models.
- Appendix I: List of all prompts used in this paper for image generation.

A. Broader Impacts

Our work focuses on distilling the multi-step generation process of MDMs into one step, significantly reducing inference time and computational costs, therefore lowering the carbon footprint during the inference. This advancement has the potential to make high-quality generative models more accessible, facilitating applications in creative industries, content generation, and real-time systems. However, as with many generative modeling techniques, our method inherits biases from the teacher models. This could potentially lead to ethical concerns, including the generation of misleading or harmful content. Additionally, by enabling faster and more efficient content generation, our approach could lower the barrier to misuse, such as the creation of deepfakes or other deceptive media.

B. Relevant Derivations to the Token-level Divergence

B.1. Loss Gradient

Given Eq. (2) and Eq. (4), by assuming D is differentiable with respect to $p_\theta(x_0^i|\tilde{x}_t)$, we can calculate Eq. (6) using chain rule as:

$$
\begin{aligned}
\nabla_\theta \mathcal{L}_{\text{Di[M]O}} &= \nabla_\theta \mathbb{E}_{x_{\text{init}},t}\left[w(t)\left(\mathbb{E}_{q_{t|0}}\left[D\big((p_\phi||p_\theta)(\tilde{x}_t)\big)\right]\right)\right] \\
&= \mathbb{E}_{x_{\text{init}},t}\left[w(t)\left(\mathbb{E}_{q_{t|0}}\left[\nabla_\theta D\big((p_\phi||p_\theta)(\tilde{x}_t)\big)\right]\right)\right] \\
&= \mathbb{E}_{x_{\text{init}},t}\left[w(t)\left(\mathbb{E}_{q_{t|0}}\left[\frac{1}{L_M}\sum_{\substack{i=1\\ \tilde{x}_t^i=[\text{M}]}}^{L}\nabla_\theta D\big(p_\phi(x_0^i|\tilde{x}_t)\|p_\theta(x_0^i|\tilde{x}_t)\big)\right]\right)\right] \\
&= \mathbb{E}_{x_{\text{init}},t}\left[w(t)\left(\mathbb{E}_{q_{t|0}}\left[\frac{1}{L_M}\sum_{\substack{i=1\\ \tilde{x}_t^i=[\text{M}]}}^{L}\nabla_{p_\theta(x_0^i|\tilde{x}_t)} D\big(p_\phi(x_0^i|\tilde{x}_t)\|p_\theta(x_0^i|\tilde{x}_t)\big)\frac{\mathrm{d}p_\theta(x_0^i|\tilde{x}_t)}{\mathrm{d}z_\theta^i}\frac{\mathrm{d}z_\theta^i}{\mathrm{d}\theta}\right]\right)\right].
\end{aligned}
\tag{9}
$$

By defining $\nabla_{z_\theta} D\big((p_\phi||p_\theta)(\tilde{x}_t)\big)$ as a vector with the i-th element $[\nabla_{z_\theta} D\big((p_\phi||p_\theta)(\tilde{x}_t)\big)]_i = \frac{1}{L_M}\nabla_{p_\theta(x_0^i|\tilde{x}_t)} D\big(p_\phi(x_0^i|\tilde{x}_t)\|p_\theta(x_0^i|\tilde{x}_t)\big)\frac{\mathrm{d}p_\theta(x_0^i|\tilde{x}_t)}{\mathrm{d}z_\theta^i}$, and $\frac{\mathrm{d}z_\theta}{\mathrm{d}\theta}$ as a vector with the i-th element $\frac{\mathrm{d}z_\theta^i}{\mathrm{d}\theta}$, we have:

$$
\nabla_\theta \mathcal{L}_{\text{Di[M]O}} = \mathbb{E}_{x_{\text{init}},t}\left[w(t)\left(\mathbb{E}_{q_{t|0}}\left[\nabla_{z_\theta} D(p_\phi||p_\theta)(\tilde{x}_t)\ \frac{\mathrm{d}z_\theta(\tilde{x}_t)}{\mathrm{d}\theta}\right]\right)\right].
\tag{10}
$$

By default, we apply stop-gradient to the $\tilde{x}_t$, since the sample operation from z_θ to $x_\theta(x_{\text{init}})$ is non-differentiable.

B.2. Explicit Form of Divergence

We start the derivation with the FKL and RKL in the generalized Jeffrey divergence. The forward and reverse KL between the teacher ϕ and the one-step generator θ at each output location i are:

$$\begin{aligned} D_{FKL_i}(\tilde{x}_t) &= \sum_{k=1}^{V} p_\phi(x_0^i = k|\tilde{x}_t) \log\left(\frac{p_\phi(x_0^i = k|\tilde{x}_t)}{p_\theta(x_0^i = k|\tilde{x}_t)}\right), \\ D_{RKL_i}(\tilde{x}_t) &= \sum_{k=1}^{V} p_\theta(x_0^i = k|\tilde{x}_t) \log\left(\frac{p_\theta(x_0^i = k|\tilde{x}_t)}{p_\phi(x_0^i = k|\tilde{x}_t)}\right). \end{aligned} \tag{11}$$

The derivation of these KLs with respect to the student parameter θ can be written as:

$$\begin{aligned} \nabla_\theta D_{FKL_i} &= \sum_{j=1}^{V} \frac{\partial D_{FKL_i}}{\partial z_{\theta_j}^i} \frac{\partial z_{\theta_j}^i}{\partial \theta}, \\ \nabla_\theta D_{RKL_i} &= \sum_{j=1}^{V} \frac{\partial D_{RKL_i}}{\partial z_{\theta_j}^i} \frac{\partial z_{\theta_j}^i}{\partial \theta}. \end{aligned} \tag{12}$$

Given that the probability corresponding to the logits z as $p_\theta(x_0^i = k|\tilde{x}_t) = \frac{\exp(z_{\theta_k}^i)}{\sum_{n=1}^{V}\exp(z_{\theta_n}^i)}$, we can precalculate the following quantity for $j \neq k$:

$$\begin{aligned} \frac{\partial}{\partial z_{\theta_j}^i} p_\theta(x_0^i = j|\tilde{x}_t) &= \frac{\exp(z_{\theta_j}^i)}{\sum_{n=1}^{V}\exp(z_{\theta_n}^i)} - p_\theta(x_0^i = j|\tilde{x}_t)^2 = p_\theta(x_0^i = j|\tilde{x}_t)(1 - p_\theta(x_0^i = j|\tilde{x}_t)), \\ \frac{\partial}{\partial z_{\theta_j}^i} p_\theta(x_0^i = k|\tilde{x}_t) &= -\frac{\exp(z_{\theta_k}^i)\exp(z_{\theta_j}^i)}{(\sum_{n=1}^{V}\exp(z_{\theta_n}^i))^2} = -p_\theta(x_0^i = k|\tilde{x}_t) p_\theta(x_0^i = j|\tilde{x}_t). \end{aligned} \tag{13}$$

These are the gradients of the softmax function that we will use below to help the derivation.

B.2.1. Gradient of FKL

For each possible token i, we have:

$$\begin{aligned} \frac{\partial D_{FKL_i}}{\partial z_{\theta_j}^i} =& \frac{\partial}{\partial z_{\theta_j}^i} \sum_{k=1}^{V} p_\phi(x_0^i = k|\tilde{x}_t) \log\left(\frac{p_\phi(x_0^i = k|\tilde{x}_t)}{p_\theta(x_0^i = k|\tilde{x}_t)}\right) \\ =& \frac{\partial}{\partial z_{\theta_j}^i} \sum_{k=1}^{V} -p_\phi(x_0^i = k|\tilde{x}_t) \log p_\theta(x_0^i = k|\tilde{x}_t) \ \text{//From KL to cross entropy} \\ =& \frac{\partial}{\partial z_{\theta_j}^i} \left(\sum_{k=1,k\neq j}^{V} -p_\phi(x_0^i = k|\tilde{x}_t) \log p_\theta(x_0^i = k|\tilde{x}_t)\right) - \frac{\partial}{\partial z_{\theta_j}^i} p_\phi(x_0^i = j|\tilde{x}_t) \log p_\theta(x_0^i = j|\tilde{x}_t) \\ =& \left(\sum_{k=1,k\neq j}^{V} -\frac{p_\phi(x_0^i = k|\tilde{x}_t)}{p_\theta(x_0^i = k|\tilde{x}_t)} \frac{\partial}{\partial z_{\theta_j}^i} p_\theta(x_0^i = k|\tilde{x}_t)\right) - \frac{p_\phi(x_0^i = j|\tilde{x}_t)}{p_\theta(x_0^i = j|\tilde{x}_t)} \frac{\partial}{\partial z_{\theta_j}^i} p_\theta(x_0^i = j|\tilde{x}_t) \ \text{//derivative of } \log(\cdot) \\ =& \sum_{k=1,k\neq j}^{V} p_\phi(x_0^i = k|\tilde{x}_t) p_\theta(x_0^i = j|\tilde{x}_t) - p_\phi(x_0^i = j|\tilde{x}_t)(1 - p_\theta(x_0^i = j|\tilde{x}_t)) \ \text{//substitute } Eq.\ (13) \\ =& \sum_{k=1}^{V} p_\phi(x_0^i = k|\tilde{x}_t) p_\theta(x_0^i = j|\tilde{x}_t) - p_\phi(x_0^i = j|\tilde{x}_t) \\ =& p_\theta(x_0^i = j|\tilde{x}_t) - p_\phi(x_0^i = j|\tilde{x}_t). \ \text{//probability sums up to 1} \end{aligned} \tag{14}$$

B.2.2. Gradient of RKL

For each possible token i, we have:

$$
\begin{aligned}
\frac{\partial D_{RKL_i}}{\partial {z_\theta}^i_j} =& \frac{\partial}{\partial {z_\theta}^i_j} \sum_{k=1}^{V} p_\theta(x_0^i = k|\tilde{x}_t) \log\left(\frac{p_\theta(x_0^i = k|\tilde{x}_t)}{p_\phi(x_0^i = k|\tilde{x}_t)}\right) \\
=& \frac{\partial}{\partial {z_\theta}^i_j}\left(\sum_{k=1,k\neq j}^{V} p_\theta(x_0^i = k|\tilde{x}_t) \log\left(\frac{p_\theta(x_0^i = k|\tilde{x}_t)}{p_\phi(x_0^i = k|\tilde{x}_t)}\right)\right) + \frac{\partial}{\partial {z_\theta}^i_j}\left(p_\theta(x_0^i = j|\tilde{x}_t) \log\left(\frac{p_\theta(x_0^i = j|\tilde{x}_t)}{p_\phi(x_0^i = j|\tilde{x}_t)}\right)\right) \\
=& \sum_{k=1,k\neq j}^{V}\left(-p_\theta(x_0^i = k|\tilde{x}_t)p_\theta(x_0^i = j|\tilde{x}_t) \log\left(\frac{p_\theta(x_0^i = k|\tilde{x}_t)}{p_\phi(x_0^i = k|\tilde{x}_t)}\right) - p_\theta(x_0^i = k|\tilde{x}_t)p_\theta(x_0^i = j|\tilde{x}_t)\right) \\
& + p_\theta(x_0^i = j|\tilde{x}_t)(1 - p_\theta(x_0^i = j|\tilde{x}_t)) \log\left(\frac{p_\theta(x_0^i = j|\tilde{x}_t)}{p_\phi(x_0^i = j|\tilde{x}_t)}\right) + p_\theta(x_0^i = j|\tilde{x}_t)(1 - p_\theta(x_0^i = j|\tilde{x}_t)) \ \text{//substitute } Eq.\ (13) \\
=& \sum_{k=1}^{V}\left(-p_\theta(x_0^i = k|\tilde{x}_t)p_\theta(x_0^i = j|\tilde{x}_t) \log\left(\frac{p_\theta(x_0^i = k|\tilde{x}_t)}{p_\phi(x_0^i = k|\tilde{x}_t)}\right) - p_\theta(x_0^i = k|\tilde{x}_t)p_\theta(x_0^i = j|\tilde{x}_t)\right) \\
& + p_\theta(x_0^i = j|\tilde{x}_t) \log\left(\frac{p_\theta(x_0^i = j|\tilde{x}_t)}{p_\phi(x_0^i = j|\tilde{x}_t)}\right) + p_\theta(x_0^i = j|\tilde{x}_t) \\
=& \sum_{k=1}^{V}\left(-p_\theta(x_0^i = k|\tilde{x}_t)p_\theta(x_0^i = j|\tilde{x}_t) \log\left(\frac{p_\theta(x_0^i = k|\tilde{x}_t)}{p_\phi(x_0^i = k|\tilde{x}_t)}\right)\right) \cancel{- p_\theta(x_0^i = j|\tilde{x}_t)} \ \text{//probability sums up to 1} \\
& + p_\theta(x_0^i = j|\tilde{x}_t) \log\left(\frac{p_\theta(x_0^i = j|\tilde{x}_t)}{p_\phi(x_0^i = j|\tilde{x}_t)}\right) \cancel{+ p_\theta(x_0^i = j|\tilde{x}_t)} \\
=& p_\theta(x_0^i = j|\tilde{x}_t)\left(\sum_{k=1}^{V}\left(-p_\theta(x_0^i = k|\tilde{x}_t) \log\left(\frac{p_\theta(x_0^i = k|\tilde{x}_t)}{p_\phi(x_0^i = k|\tilde{x}_t)}\right)\right) + \log\left(\frac{p_\theta(x_0^i = j|\tilde{x}_t)}{p_\phi(x_0^i = j|\tilde{x}_t)}\right)\right) \\
=& p_\theta(x_0^i = j|\tilde{x}_t)\left(\log\left(\frac{p_\theta(x_0^i = j|\tilde{x}_t)}{p_\phi(x_0^i = j|\tilde{x}_t)}\right) - D_{RKL_i}\right). \ \text{//definition of RKL}
\end{aligned}
\tag{15}
$$

This is similar to the results derived for AR LLM in [111].

As a result, the approximated token-level gradients of FKL and RKL at each masked position i in Eq. (5) can be calculated as follows:

$$
\begin{aligned}
&\nabla_{z_\psi} D_{FKL_i}\left(p_\phi({x_0}^i|\tilde{x}_t)\|p_\psi({x_0}^i|\tilde{x}_t)\right) := \nabla_{z_\psi} D_{FKL_i}(\tilde{x}_t) = (p_\psi(x_0^i|\tilde{x}_t) - p_\phi(x_0^i|\tilde{x}_t)), \\
&\nabla_{z_\psi} D_{RKL_i}\left(p_\phi({x_0}^i|\tilde{x}_t)\|p_\psi({x_0}^i|\tilde{x}_t)\right) := \nabla_{z_\psi} D_{RKL_i}(\tilde{x}_t) = p_\psi(x_0^i|\tilde{x}_t)\left(\log\left(\frac{p_\psi(x_0^i|\tilde{x}_t)}{p_\phi(x_0^i|\tilde{x}_t)}\right) - D_{RKL_i}(\tilde{x}_t)\right).
\end{aligned}
\tag{16}
$$

B.2.3. Gradient of f-divergence

Our proposed token-level divergence can be seamlessly extended to general f-divergence [17] with the form [30, 82, 116]:

$$
D_{f_i} = \sum_{k=1}^{V} p_\theta(x_0^i = k|\tilde{x}_t) f\left(\frac{p_\phi(x_0^i = k|\tilde{x}_t)}{p_\theta(x_0^i = k|\tilde{x}_t)}\right). \tag{17}
$$

When the generator function f is differentiable, we can calculate its gradient as:

$$
\begin{aligned}
\frac{\partial D_{f_i}}{\partial {z_\theta}^i} =& \frac{\partial}{\partial {z_\theta}^i_j} \sum_{k=1}^{V} p_\phi(x_0^i = k|\tilde{x}_t) f\left(\frac{p_\phi(x_0^i = k|\tilde{x}_t)}{p_\theta(x_0^i = k|\tilde{x}_t)}\right) \\
=& \sum_{k=1}^{V}\left(\frac{\partial p_\theta(x_0^i = k|\tilde{x}_t)}{\partial {z_\theta}^i_j}\left[f\left(\frac{p_\phi(x_0^i = k|\tilde{x}_t)}{p_\theta(x_0^i = k|\tilde{x}_t)}\right) - \left(\frac{p_\phi(x_0^i = k|\tilde{x}_t)}{p_\theta(x_0^i = k|\tilde{x}_t)}\right) f'\left(\frac{p_\phi(x_0^i = k|\tilde{x}_t)}{p_\theta(x_0^i = k|\tilde{x}_t)}\right)\right]\right) \\
=& p_\theta(x_0^i = j|\tilde{x}_t) \sum_{k=1}^{V}\left(-p_\theta(x_0^i = k|\tilde{x}_t)\left[f\left(\frac{p_\phi(x_0^i = k|\tilde{x}_t)}{p_\theta(x_0^i = k|\tilde{x}_t)}\right) - \left(\frac{p_\phi(x_0^i = k|\tilde{x}_t)}{p_\theta(x_0^i = k|\tilde{x}_t)}\right) f'\left(\frac{p_\phi(x_0^i = k|\tilde{x}_t)}{p_\theta(x_0^i = k|\tilde{x}_t)}\right)\right]\right) \\
& + p_\theta(x_0^i = j|\tilde{x}_t)\left[f\left(\frac{p_\phi(x_0^i = j|\tilde{x}_t)}{p_\theta(x_0^i = j|\tilde{x}_t)}\right) - \left(\frac{p_\phi(x_0^i = j|\tilde{x}_t)}{p_\theta(x_0^i = j|\tilde{x}_t)}\right) f'\left(\frac{p_\phi(x_0^i = j|\tilde{x}_t)}{p_\theta(x_0^i = j|\tilde{x}_t)}\right)\right].
\end{aligned}
\tag{18}
$$

As shown in Tab. 4, the generalized Jeffrey divergence belongs to f-divergence, with generator function $f(u) = ((1-\beta)u - \beta)\log u$.

Table 4. Summary of some typical f-divergences $D_f(p\|q)$ together with generator functions f, where $f : (0, \infty) \to \mathbb{R}$ is a convex function satisfying the condition $f(1) = 0$. This table is mainly adapted from [30].

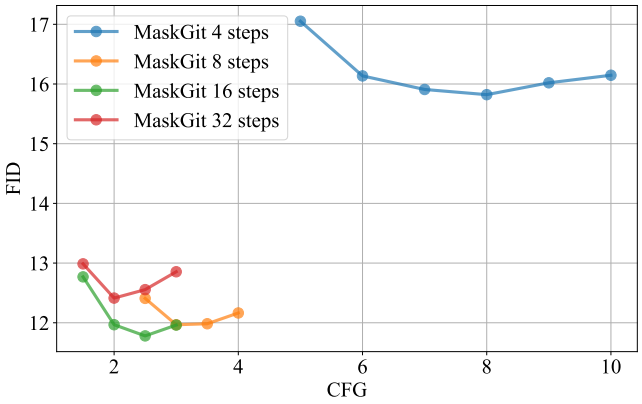
Name	$D_f(p\|q)$	Generator $f(u)$
Forward Kullback-Leibler	$\int p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x$	$u \log u$
Reverse Kullback-Leibler	$\int q(x) \log \frac{q(x)}{p(x)} \, \mathrm{d}x$	$-\log u$
α-divergence ($\alpha \notin \{0, 1\}$)	$\frac{1}{\alpha(\alpha-1)} \int \left(q(x) \left[\left(\frac{p(x)}{q(x)} \right)^{1-\alpha} - (1-\alpha) \left(\frac{p(x)}{q(x)} \right) - \alpha \right] \right) \mathrm{d}x$	$\frac{1}{\alpha(\alpha-1)} \left(u^{1-\alpha} - (1-\alpha)u - \alpha \right)$
Generalized Jeffrey	$\int [(1-\beta)p(x) - \beta q(x)] \log \left(\frac{p(x)}{q(x)} \right) \mathrm{d}x$	$((1-\beta)u - \beta) \log u$
Jensen-Shannon	$\frac{1}{2} \int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} \, \mathrm{d}x$	$-(u+1) \log \frac{1+u}{2} + u \log u$
Squared Hellinger	$\int \left(\sqrt{p(x)} - \sqrt{q(x)} \right)^2 \mathrm{d}x$	$(\sqrt{u} - 1)^2$

Table 5. Generative performance on class-conditional ImageNet-256. Results of methods in type AR are taken from the DD [53]. Percentage drop values (relative to the teacher) are shown in parentheses.

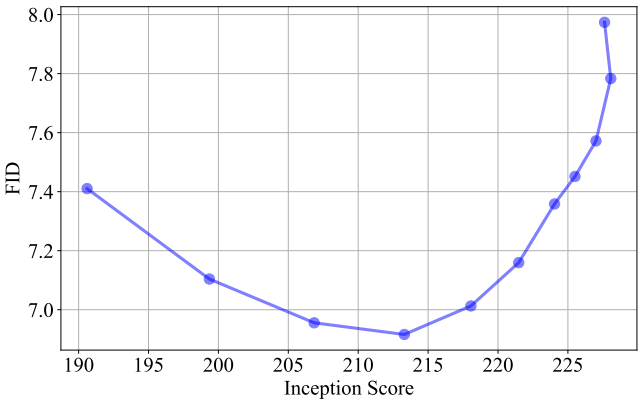
Type	Model	FID (↓)	IS (↑)	Precision (↑)	Recall (↑)	#Para	Step (↓)
AR	VAR-d16 [105]	4.19	230.2	0.84	0.48	310M	10
AR	VAR-d20 [105]	3.35	301.4	0.84	0.51	600M	10
AR	VAR-d24 [105]	2.51	312.2	0.82	0.53	1.03B	10
AR	VAR-d16-DD [53]	9.94 (137%)	193.6 (16%)	0.80 (5%)	0.37 (23%)	327M	**1**
AR	VAR-d16-DD [53]	7.82 (87%)	197.0 (14%)	0.80 (5%)	0.41 (15%)	327M	2
AR	VAR-d20-DD [53]	9.55 (185%)	197.2 (35%)	0.78 (7%)	0.38 (26%)	635M	**1**
AR	VAR-d20-DD [53]	7.33 (119%)	204.5 (32%)	0.82 (2%)	0.40 (22%)	635M	2
AR	VAR-d24-DD [53]	8.92 (255%)	202.8 (35%)	0.78 (5%)	0.39 (26%)	1.09B	**1**
AR	VAR-d24-DD [53]	6.95 (177%)	222.5 (29%)	0.83 (-1%)	0.43 (19%)	1.09B	2
AR	LlamaGen-B [103]	5.42	193.5	0.83	0.44	111M	256
AR	LlamaGen-L [103]	4.11	283.5	0.85	0.48	343M	256
AR	LlamaGen-B-DD [53]	15.50 (186%)	135.4 (30%)	0.76 (8%)	0.26 (41%)	98.3M	**1**
AR	LlamaGen-B-DD [53]	11.17 (106%)	154.8 (20%)	0.80 (4%)	0.31 (30%)	98.3M	2
AR	LlamaGen-L-DD [53]	11.35 (176%)	193.6 (32%)	0.81 (5%)	0.30 (38%)	326M	**1**
AR	LlamaGen-L-DD [53]	7.58 (84%)	237.5 (16%)	0.84 (1%)	0.37 (23%)	326M	2
MDM	MaskGit [11]	6.60	224.07	0.831	0.402	174M	16
MDM	Di[M]O	6.91 (5%)	214.05 (4%)	0.828 (0.4%)	0.377 (6.2%)	174M	**1**

C. More Discussion on Related Works

Recent research has explored the possibility of converting AR models into discrete diffusion versions [27]. Given this ongoing work, we plan to investigate the applicability of our approach to AR models. Furthermore, while existing limitations of MDM are well known, an efficient MDM could still serve as a draft model for AR speculative decoding [16]. Diffusion distillation has seen widespread adoption of the GAN objective, either directly for distillation [91, 109, 115] or to enhance performance [51, 118, 124]. However, these methods are not directly applicable to MDM due to its discrete nature. In a similar vein, recent works based on SiD [60, 61, 123–125] aim to minimize the Fisher divergence or a generalized score-based divergence for distillation. However, these approaches require backpropagating the gradient through the teacher model—akin to adversarial training—which is infeasible for MDM due to the non-differentiable sampling operation. Xu *et al*. [116] recently introduced a general framework for distilling continuous diffusion models using f-divergence. At the cost of training another additional discriminator, this method successfully extends the DMD framework from RKL to general f-divergence by utilizing the output of the discriminator to weight the loss gradient in DMD. However, their method relies on the assumption that the teacher model and the real data used for training the discriminator obey the same underline distribution (otherwise need to simulate the teacher model to get more accurate synthetic data). The training of additional discriminator in general increases the computational overhead. Moreover, while effective for continuous models, their method relies on an additional GAN loss to achieve better performance, making it unsuitable for MDMs. To show the performance of our method, we compare it with DD [53], which distills AR models into one-step or few-step generators. As demonstrated in Tab. 5, our method yields significantly smaller performance drops compared to the teacher models than [53]. To summarize, our method successfully address the multi-token prediction challenge pointed out in [31, 99], by transfer the stochasticity into the model input and always predict tokens from the correct joint distribution of multiple tokens.



(a) MaskGit teacher performance under difference inference steps with different CFG. Metric calculated with 5k generated samples.



(b) Performance of one-step generator distilled from MaskGit teacher under different temperatures. From left to right the temperatures are $[11.0, 10.0, 9.0, 8.0, 7.0, 6.0, 5.0, 4.5, 4.0, 3.0, 2.0]$. Metric calculated with 50k generated samples.

Figure 7. More results from the Maskgit experiments.

Table 6. Control experiments on the initialization choices to validate the hypothesis in Sec. 4.3.

Initial Method	Step (↓)	FID-5k (↓)	IS-5k (↑)
use VAE encoded code of random noise	1	189.39	4.67
use $1 - r_{\text{init}}$ random image tokens, r_{init} fixed class token (e.g. 1025)	1	173.02	5.00
use $1 - r_{\text{init}}$ random image tokens, r_{init} fixed image token (e.g. 512)	1	188.3	4.65
use $1 - r_{\text{init}}$ random class tokens, r_{init} fixed mask token [M]	1	174.47	4.73
use $1 - r_{\text{init}}$ random image tokens, r_{init} fixed mask token [M] (ours strategy)	1	12.01	132.44

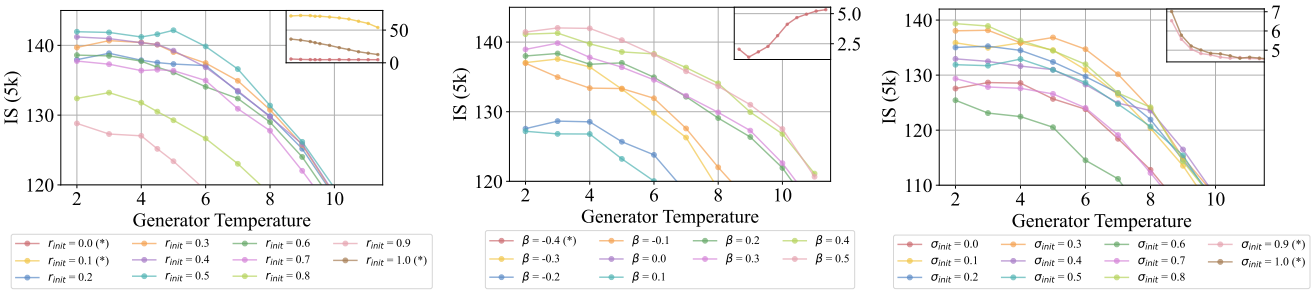
D. More Experiment Setup

For the sampling of teacher model, we use the heuristic parallel sampler by default, with temperature=1.3, schedule mode arccos, and choose token to remask randomly during sampling (as the greedy approach by keeping the most confident tokens leads to degraded generation). In Fig. 7a, we show the teacher's FID with with 5k generated images under-difference inference steps with different CFG. This result suggests the limitation of the parallel sampler for test time scaling, as pointed out in [81]. Given these results, we use 16 steps and CFG=2.5 for the teacher model by default in the paper. In order to add random perturbation to the token embeddings, we fix the embedding layer of all the models during distillation. During the experiment, we use the same mask schedule for getting the intermediate state $\tilde{x}_t$ from one-step model generation and training the auxiliary model. In addition, we choose the same schedule when training the teacher model. We use the arccos schedule for MaskGit distillation and the cosine schedule for Meissonic distillation, respectively. We adopt the loss weight from [119] with $w(t) = \frac{1}{p_\theta(x_0|\tilde{x}_t) - p_\phi(x_0|x_{\text{init}})}$. Our experiment suggests that this weight can prevent the gradient of the generator from exploding (while the one with 1000 times the gradient value can still generate a regular image). We use by default mixed precision training with *bf16* and a gradient clipping gradient normalization 1. We use a constant learning rate scheduler with a linear warmup of 100 steps. We use the *adam* optimizer with beta1 = 0.9 and beta2 =0.999 and no weight decay for all experiments. All temperatures for the three models are fixed to 1 during distillation. Exponential Moving Average (EMA) is applied with a rate of 0.9999 for all experiments. The codebook sizes for MaskGit and Meissonic are 1024 and 8192, and the sequence length of the latent codes for MaskGit and Meissonic are 1024 and 4096, respectively. For a sequence length of L, a codebook size of V, and the number of [M] tokens to be replaced $N \approx (1 - r_{\text{init}})L$, the possible initial token configure is $\binom{L}{N} V^N$. Given that L and V are usually large numbers, the possible initial token configure are sufficient to be the initial state. This explains why the noise perturbation yield marginal improvement, even though stabilize the training. The embedding layer in the one-step generator is fixed during distillation to improve the stability. By default, the FID, precision, recall, density and coverage are all calculated with features extracted from the InceptionV3 network.

E. More Experimental Results

In this section we show more experiment results.

1. **IS results from ablation.** We present the IS results of the ablation in Fig. 8. Unlike FID, we found that the best IS is achieved at a much lower temperature. This suggest that the temperature can control the trade-off between FID and IS. We show visually how the



(a) r_{init}: Initial Mask Ratio (b) β: Jeffrey Coefficient (c) σ_{init}: Perturbation Strength

Figure 8. IS results of the ablations corresponding to Fig. 5. * means the training is collapsed and falls outside the comparable range with other results, we therefore show them in the sub-figures at the right upper corner with the same range of the x-axis.

final output image changes with the generator temperature in Fig. 9.

2. **One-step generation metric score with different temperature settings.** In Fig. 7b, we show the FID and IS metric with varying temperatures of the generator. We observe that, as the temperature increases, the IS score decreases, while the FID score initially declines, reaching its minimum at a temperature of 7, before starting to rise again.
3. **Top-k visualization.** We apply the top-k trick when sampling each token from the predicted distributions $p_\phi(x_0^i|\tilde{x}_t)$ and $p_\theta(x_0^i|x_{\text{init}})$. As shown in Fig. 10, after distillation, our one-step generation with top-1 sampling and without top-k are nearly identical, unlike in the teacher model. This suggests that each initial code almost deterministically maps to a fixed set of image tokens, justifying the use of Gaussian perturbation for token embeddings in the generator.
4. **Analysis of output distribution.** To further investigate this, we examine the number of potential output tokens with probabilities greater than 0.001, as shown in Fig. 11a. Specifically, we analyze 256 randomly generated images (total 65536 tokens), where each token in every image has a total of 1024 possible output tokens (vocabulary size). We find that for nearly half of the tokens, the output logits from our distilled model collapse into a delta distribution, confirming the trends observed in Fig. 10 and Fig. 9. For comparison, Fig. 11b presents the corresponding distribution for the teacher model, where the potential output token probabilities are more evenly spread across different possible token values.
5. **AJustification for $z_\theta(\tilde{x}_t) \approx z_\theta(x_{\text{init}})$.** As stated in Sec. 4.2, an approximation of the $z_\theta(\tilde{x}_t)$ is required to provide meaningful gradients for optimizing the student θ. To justify our choice, we conduct an ablation study by comparing all three possible candidates to approximate $z_\theta(\cdot)$:
   (1) x_{init}: the input used in the main paper;
   (2) $\tilde{x}_t$: no approximation is used (while the student θ is trained to model $p_\theta(x_0^i|\tilde{x}_{\text{init}})$, here, we consider to directly use $z_\theta(\tilde{x}_t)$, which estimates $p_\theta(x_0^i|\tilde{x}_t)$);
   (3) $\tilde{x}_{r_{\text{init}}}$: constructed by applying the initial mask from x_{init} to the generated x_0, aiming to stay close to x_{init}.
   Fig. 12 compares the visual quality of one-step generations at various training iterations when using these different inputs for loss calculation. We observe that using x_{init} yields the best generation quality and fastest convergence. The alternative choices fail because they both lie outside the training initial distribution p_{init} of the student θ , making them unsuitable inputs. In particular, training with $\tilde{x}_t$ diverges due to its larger deviation from p_{init}.
6. **Information in the initial sequence.** Similar to several recent works on the influence of initial noise on the final generated images [13, 58, 78, 126]. In this work, we designed a token initialization strategy which injects randomness in the initial sequence x_0. We here investigate the influence in the initial code. For a distilled model trained with $r_{\text{init}} = 0.6$, we tested its performance using initial sequences composed of either 40% random image tokens or 40% image tokens derived from encoding real images with VQ-VAE. The results, shown in Fig. 13, present three image examples with three different random seeds applied to each for both the random image tokens case and the encoded image token case. We find that, unlike using random image tokens when the one-step generator can produce diverse images, using the real image tokens instead results in generations that closely resemble the original images. This suggests that, similar to continuous diffusion processes, the information contained in randomly initialized codes significantly influences the final generation. This property enables test-time scaling techniques for our model to improve performance [62, 107].
7. **Interpolation between initial token sequences.** In addition, we show the interpolation results between random initial sequence and encoded sequence in Fig. 14.
8. **Control experiments on token initialization strategy.** In our experiments, we conducted a controlled study on the token initialization strategy to verify our hypothesis in Sec. 4.3 that the initial sequence of the student should be similar to those used to train the teacher model (see Tab. 6). Similar to our main strategy, we replaced a fraction $1 - r_{\text{init}}$ of the [M] tokens with random *visual* tokens. However, in this case, we replaced the remaining [M] tokens with a fixed token, such as image token 512 or class token 1025. Additionally, we tested another variation where we replaced $1 - r_{\text{init}}$ of the [M] tokens with random *class* tokens while keeping the rest as [M] tokens. In both settings, the models either failed to generate meaningful images. This outcome reinforces the student model's preference for a

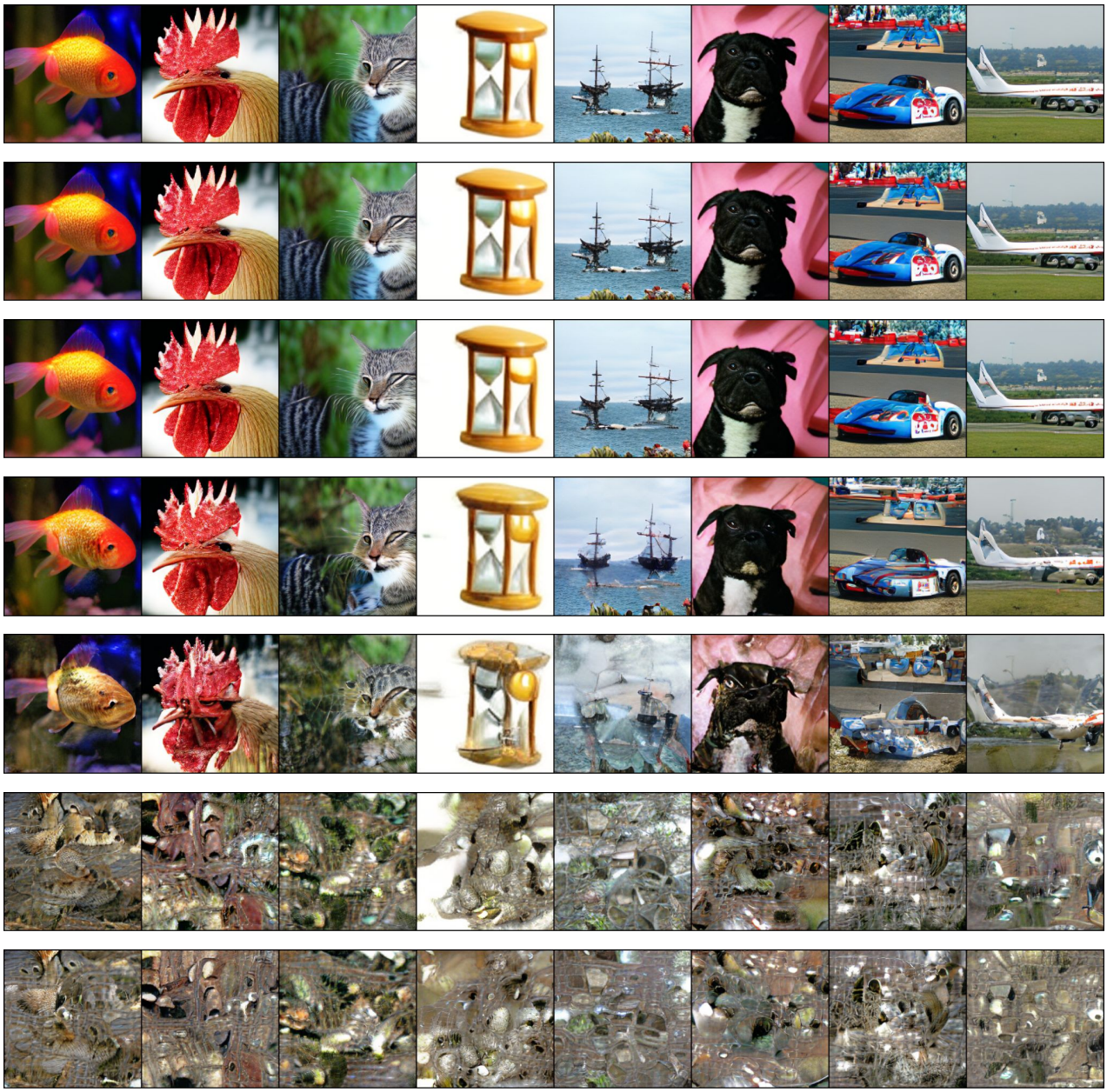


Figure 9. One-step generation at different temperature scale. From top to bottom the temperature is [1e-6,1e-3,1,10,20,50,100]

'familiar' input, a hybrid of visual and [M] tokens, during distillation. Furthermore, we experimented with sampling random sequences encoded from Gaussian noise of the same size as an image. As shown in Tab. 6, the training of these control experiments leads to divergence metric values.

9. **Additional metrics (FID and CLIP score) on Meissonic.** We evaluated the FID, Fréchet Dino-v2 [70] Distance (FDD) and CLIP Score [33, 45] on the MsCoCo 30k validation dataset [52], as shown in Table 8. We use the default CFG=9 as suggested in the official codebase. Two key observations emerge from the results: (1) as the number of generation steps decreases, the teacher model's FID deteriorates rapidly (e.g., 8-step FID is 96.75 compared to 64-step FID of 48.27); and (2) our one-step student achieves superior FID/FDD while maintaining a CLIP Score comparable to the teacher. This suggests that (i) our generator performs competitively with the teacher and (ii) FID/FDD may not always be a reliable fidelity metric, particularly for the Meissonic teacher model. We hypothesize that the biased style of the teacher model causes discrepancies compared to realistic images from the MS COCO dataset. Consequently, the distilled student model records better performance relative to the teacher.

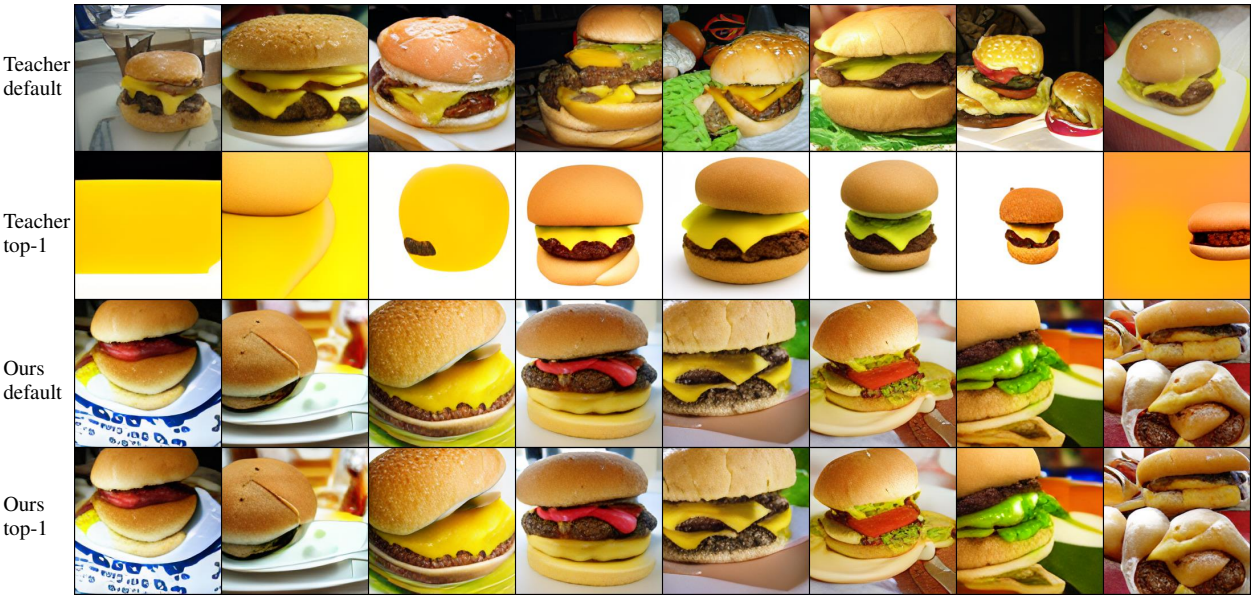
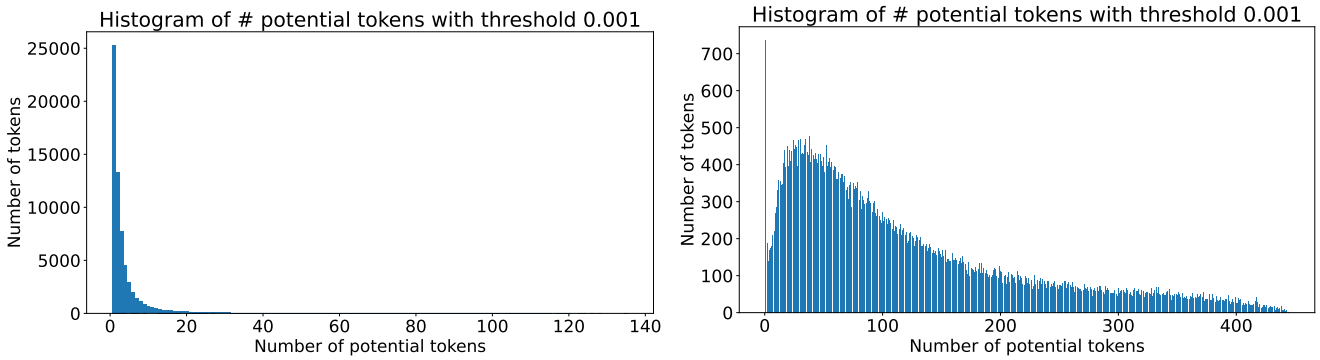


Figure 10. The top two rows display results from the teacher model using 16-step generation with default setup, with the first row generated without top-k filtering and the other with top-1 sampling. The bottom two rows showcase results from our distilled one-step generator. The class label is `933`.



(a) Histogram of student probabilities

(b) Histogram of teacher probabilities

Figure 11. Histogram of number of potential output with probability greater than 0.001

10. **Additional HPSV2 results on Meissonic.** In Tab. 7, we present a comparison of our teacher model, Meissonic [4], across different classifier-free guidance settings. Our results indicate that Meissonic operates optimally at CFG$=9$ for the HPSV2 metric.
11. We experimented with the Two Time-scale Update Rule [118] to update the fake score multiple times per iteration. However, this did not improve model performance or further stabilize the training loss.
12. Inspired by [35], we added soft targets for training the auxiliary model using the following loss function:

$$\mathcal{L}_{\text{MDM}} = \mathbb{E}_{x_{\text{init}},t}\left[\gamma(t)\left(\mathbb{E}_{q_{t|0}}[(1-\alpha)(-\log p_{0|t}(x_\theta(x_{\text{init}})|\tilde{x}_t,\phi)) + \alpha D_{\text{KL}}(p_\psi(x_0|\tilde{x}_t)||p_\theta(x_0|x_{\text{init}}))]\right)\right] \tag{19}$$

where α is a hyperparameter controlling the interpolation between the hard target cross-entropy loss and the soft target KL loss.
13. The default CFG for MaskGit during distillation is 2. We also explored adaptive CFG, where the guidance scale varies with r_t, similar to the linear CFG used in MDM parallel sampling. However, this approach did not yield improvements.
14. Inspired by Proximal Policy Optimization (PPO) [94], we introduced an entropy bonus term $-p_\theta(x_0|x_{\text{init}})\log p_\theta(x_0|x_{\text{init}})$ to encourage generation diversity. However, this did not show practical benefits.
15. While $\beta \in [-0.3, 1]$ works well for the ImageNet teacher, we found that in the Meissonic experiment, the distillation process diverges when $\beta < -0.1$ or $\beta > 0.2$.
16. In Fig. 15, we illustrate the consistency assumption visually.

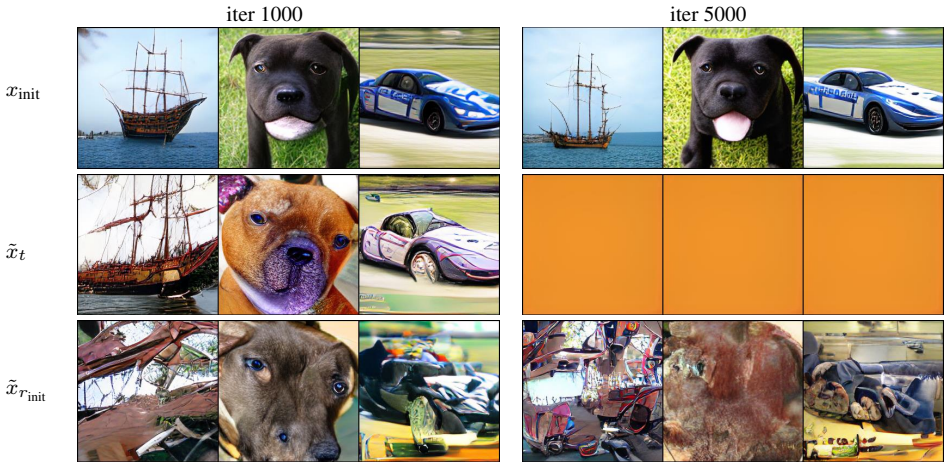


Figure 12. Ablation on different choice for the approximation of $z_\theta(\tilde{x}_t)$.

Figure 13. One-step image generation from encoded image tokens and random image tokens. The class labels are `417`, `279` and `497`, respectively.

F. Failure Cases

While the distilled one-step generator performs comparably to the teacher model in most cases, certain classes exhibit slight color shifts and mode collapse, as shown in Fig. 16. This discrepancy may explain the observed gap in evaluation metrics between the teacher and the one-step generator.

G. Mode-seeking vs. Mode-covering

In Fig. 17, we show with a Gaussian example the visualization of the mode-seeking vs. mode-covering behaviors of the generalized Jeffrey divergence with different β. When β is small, the divergence approaches the FKL, which is known for its mode-covering tendency, assigning high importance to matching all modes of the target distribution. As β increases, the behavior transitions toward a balanced form, and for large β, the divergence exhibits a mode-seeking tendency, akin to the RKL, which focuses on fitting high-density regions while ignoring low-probability modes.

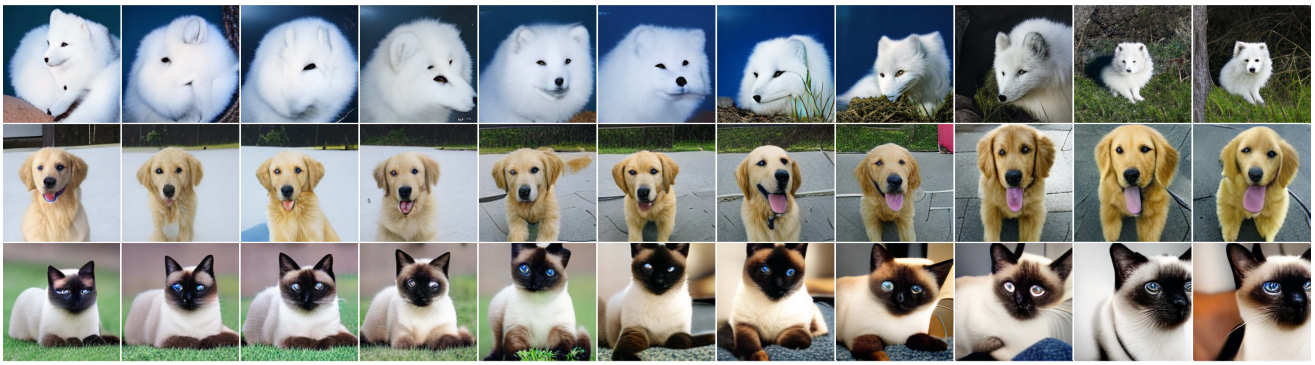


Figure 14. Interpolation between the encoded image token (left most column) and random image token (right most column)in the initial sequence. The class labels are `279`, `284` and `207`, respectively.

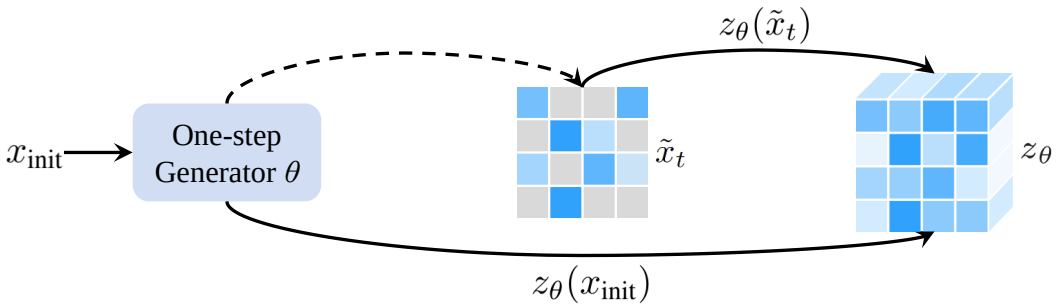


Figure 15. A visual representation of the consistency assumption. Ideally, the model's prediction based on the correct intermediate state, $\tilde{x}_t$, should be identical to its prediction derived from the initial sequence, x_{init}.

Table 7. Complete HPS v2.0 benchmark. Scores are collected from `https://github.com/tgxs002/HPSv2`. We highlight the **best**.

		HPS v2.0				
Model	**NFE**	**Animation**	**Concept-art**	**Painting**	**Photo**	**Averaged**
Latent Diffusion [83]	25	25.73	25.15	25.25	26.97	25.78
DALL·E 2 [79]	-	27.34	26.54	26.68	27.24	26.95
Stable Diffusion v1.4 [83]	50	27.26	26.61	26.66	27.27	26.95
Stable Diffusion v2.0 [83]	50	27.48	26.89	26.86	27.46	27.17
DeepFloyd-XL [19]	25	27.64	26.83	26.86	27.75	27.27
SDXL Base 1.0 [76]	50	28.88	27.88	27.92	28.31	28.25
SDXL Refiner 1.0 [76]	50	28.93	27.89	27.90	28.38	28.27
InstaFlow [55]	1	25.98	25.79	25.93	26.32	26.01
SD Turbo [90]	1	27.98	27.59	27.16	27.19	27.48
SwiftBrush v2 [18]	1	27.25	27.62	26.86	26.77	27.15
Meissonic (cfg=9) [4]	48	**29.57**	**28.58**	**28.72**	**28.45**	**28.83**
	32	29.18	28.32	28.28	27.96	28.44
	16	28.61	27.82	27.84	27.32	27.90
	8	25.62	26.49	26.67	27.07	26.46
	4	25.01	24.95	24.87	23.80	24.66
	2	23.06	23.28	23.22	22.38	22.98
Meissonic (cfg=4) [4]	48	28.52	27.44	27.54	27.17	27.67
	32	28.59	27.54	27.60	27.22	27.74
	16	28.49	27.52	27.65	27.20	27.71
	8	27.99	27.24	27.31	26.54	27.27
	4	26.33	26.03	26.01	24.79	25.79
	2	23.61	23.87	23.72	22.50	23.43
Di[M]O	1	28.64	27.91	27.99	27.92	28.11

H. More Qualitative Results

In Fig. 18, we present randomly sampled ImageNet images generated in a single step by our distilled models with MaskGit teacher. Fig. 19 compares our one-step generator with the Meissonic teacher model using different sampling steps. Notably, our one-step generation achieves superior visual quality compared to the teacher model's 16-step generation. Finally, in Fig. 20, we provide additional text-to-

Table 8. Comparison of FID, FDD and CLIP-Score for Meissonic [4] across varying generation steps and our one-step generator. The results are evaluated on MSCOOC-val 30k dataset. The teacher CFG is set to 9.

Steps (↓)	64	32	16	8	1 *(ours)*
FID (↓)	48.27	50.13	63.29	96.75	38.45
FDD (↓)	620.9	625.6	709.6	980.8	548.6
CLIP-Score (↑)	0.321	0.318	0.307	0.280	0.322

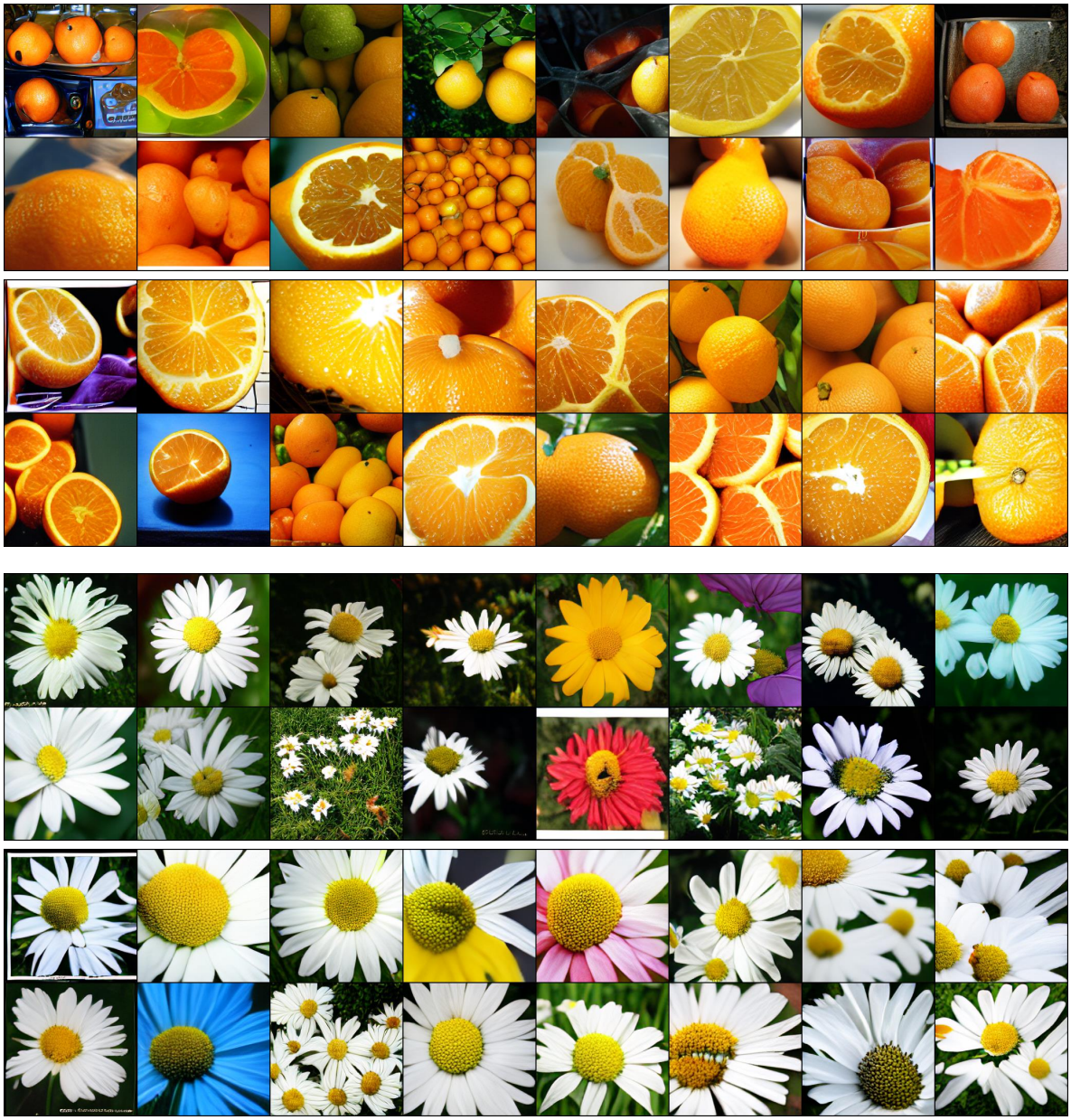


Figure 16. Distribution shift in the student. For both classes, the top two rows are results from the teacher with 16 steps and the lower two rows are the results from our one-step generator. The class labels are `950` and `985`, respectively.

image one-step generation results from our distilled model with the Meissonic teacher.

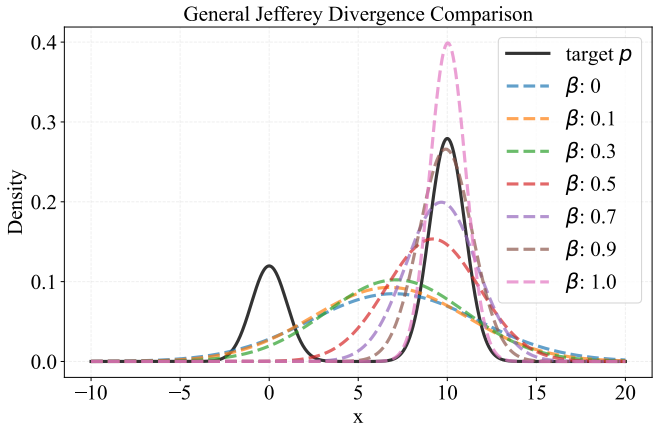


Figure 17. Toy example to visualize the mode-seeking VS mode-covering behavior of different β values in generalized Jeffrey divergence. We grid search the mean and std to minimize the generalized Jeffrey divergence.

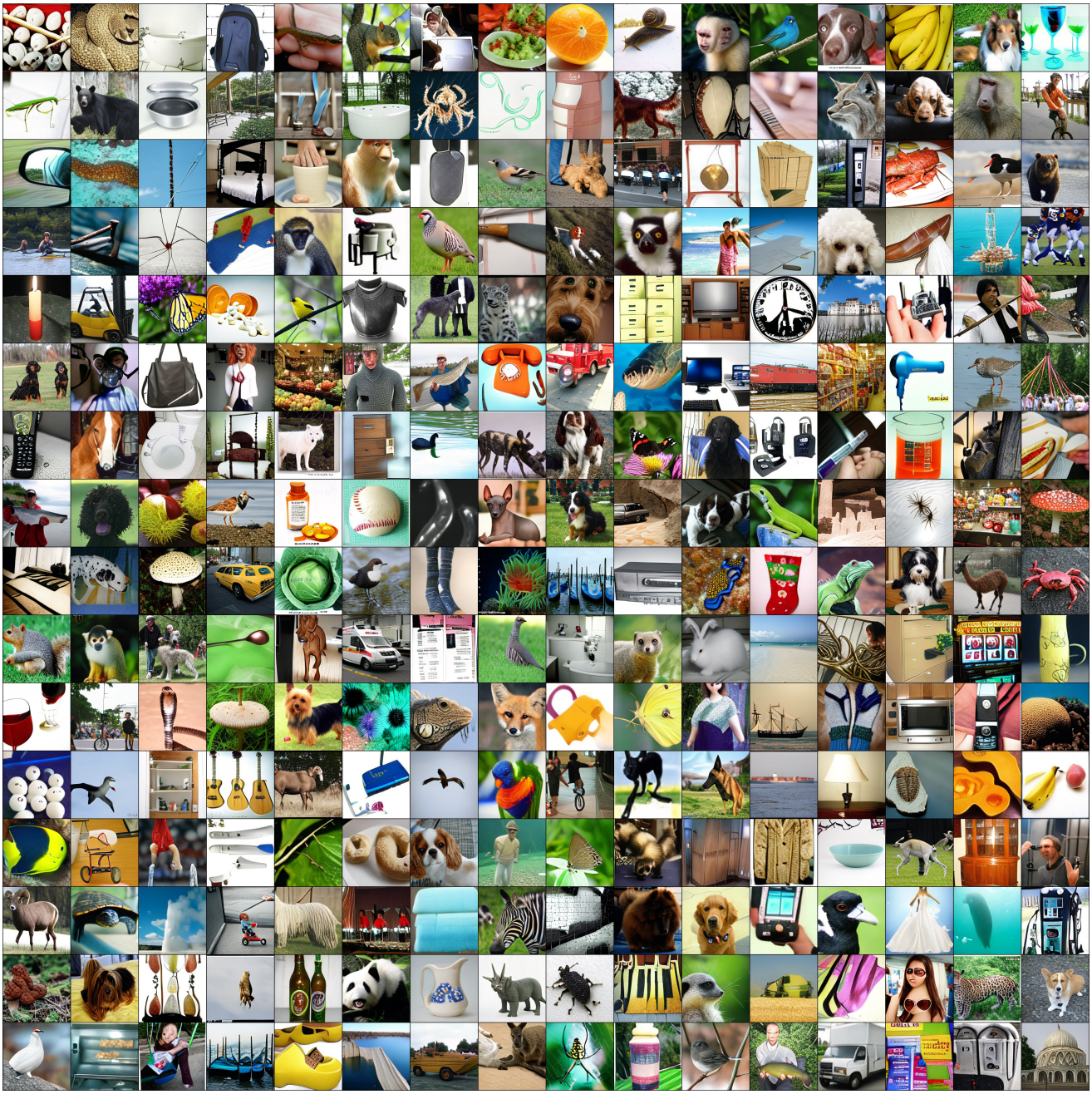


Figure 18. One-step samples from our class-conditional model on ImageNet.

Figure 19. Comparison with the teacher: Meissonic [4] on different steps, we see clearly that the teacher model's results drop very quickly (e.g., around 4 steps).

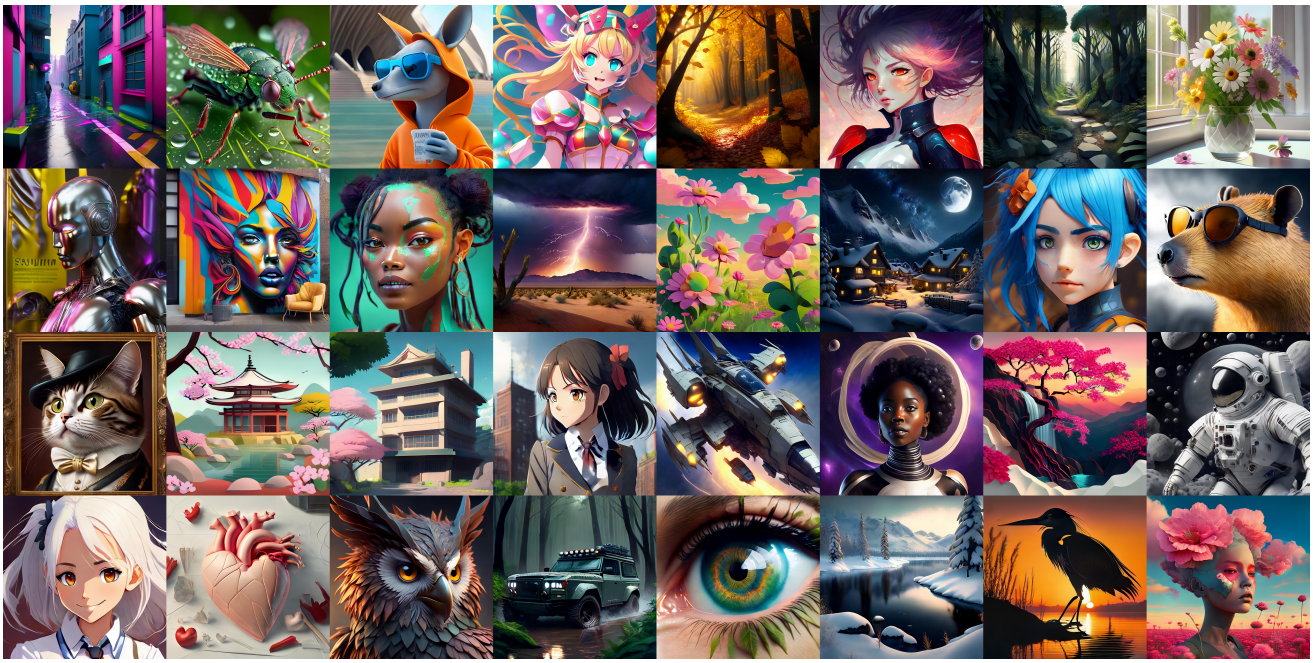


Figure 20. Qualitative results of our one-step generator distilled from Meissonic [4].

I. Misc.

Prompts Below is a collection of creative prompts we used to generate images in Figs. 1, 6, 19 and 20:

- A plushy tired owl sits on a pile of antique books in a humorous illustration.
- A photograph of a woman from Steven Universe with gigantic pink ringlets and a white dress.
- A white bichon frise puppy dog riding a black motorcycle in Hollywood at sundown with palm trees in the background.
- A photorealistic image of a giant floating glass sphere in a rocky landscape surrounded by a gentle mist.
- A cosmonaut otter poses for a portrait painted in intricate detail by Rembrandt.
- A beaver in formal attire stands next to a stack of books in a library.
- An egirl with pink hair and extensive makeup.
- Portrait of a monkey wearing a spacesuit and an astronaut helmet.
- Closeup of a seinen manga film still showing the interior of a shinkansen train with a leather seat and a window view, with a hyperrealistic film still from a Nepali movie projecting in the background.
- Swedish lake at night with heavy snowfall depicted in hyper-realistic and detailed art.
- Pencil sketch of Danny DeVito by Milt Kahl.
- A realistic anime painting of a cosmic woman wearing clothes made of universes with glowing red eyes.
- Photo of Ty Lee from Avatar.
- A green field with flowers and pink and yellow clouds under a bright sun at sunset, illustrated by Peter Chan in a colorful Day of the Tentacle style on Artstation.
- A raccoon in formal attire, carrying a bag and cane, depicted in a Rembrandt-style oil painting.
- A girl in school uniform standing in the city.
- Serene, anime-style landscape with vibrant flowers and trees, picturesque clouds, and no signs of human activity.
- A kangaroo wearing an orange hoodie and blue sunglasses holding a sign in front of the Sydney Opera House.
- A pikachu in a forest illustration.
- An oil painting close-up portrait of a young black woman wearing a crown of wildflowers, surrounded by hazy golden light.
- A capybara wearing sunglasses.
- A frog wearing an anime-inspired onesie.
- A blue-haired girl with soft features stares directly at the camera in an extreme close-up Instagram picture.
- Digital art of Prince of Roses.
- A landscape featuring a Kyoto Animation-style building.
- A path winding through a forest depicted in digital art.
- A close-up portrait of a beautiful girl with an autumn leaves headdress and melting wax.
- A neon-soaked cyberpunk alleyway with rain-drenched streets and futuristic holograms, gritty yet vibrant, hyper-realistic, ultra-detailed, cinematic scene.
- A serene mountain landscape at sunrise, mist rolling over rugged peaks, ultra-detailed, photorealistic, soft lighting, high-resolution, digital art.
- A hyper-detailed closeup of a dew-covered insect on a vibrant leaf, extreme macro photography style, ultra-realistic, high-resolution, intricate textures.
- An anime-style magical girl in a dynamic pose, vibrant colors, ultra-detailed costume and background, energetic, high-resolution, cinematic lighting.
- An enchanted autumn forest with falling leaves and warm, glowing light, ultra-detailed, photorealistic, rich textures, digital art, serene mood.
- An elegant Renaissance portrait of a noble figure, detailed textures, soft natural lighting, ultra-detailed, classical, high-resolution, oil painting style.
- A cybernetic humanoid robot portrait with metallic textures and neon accents, ultra-detailed, photorealistic, cinematic, futuristic digital art.
- A vibrant street art mural on an urban wall, ultra-detailed, energetic, bold colors, high-resolution, digital painting, modern art style.
- A dark fantasy warrior in intricately detailed armor standing in a stormy battlefield, ultra-detailed, hyper-realistic, cinematic, dynamic action scene.
- An intense lightning storm over a vast desert landscape, ultra-detailed, dramatic, high-resolution, cinematic, digital art, atmospheric.
- A detailed nature macro shot of a vibrant flower with dewdrops, ultra-detailed, photorealistic, high-resolution, digital painting, delicate textures.
- An ultra-realistic snowy mountain village under a starry sky, ultra-detailed, atmospheric, cinematic, high-resolution, digital winter wonderland.
- A humorous portrait of a cat dressed as a Victorian aristocrat, in vintage photorealism.
- A photorealistic shot of a bouquet of wildflowers in a clear glass vase on a sunlit windowsill.
- A mecha jet fighter engages in an air battle with an explosion as a backdrop, set against a dark, starry sky in a highly-detailed art piece by Stephan Martiniere.

- A young black woman stands in front of a ringed planet in space.
- Digital art of a cherry tree overlooking a valley with a waterfall at sunset.
- An astronaut in white futuristic cybernetic armor running on the surface of the moon, featured in an artwork illustration on Artstation.
- The image is a headshot of a happy girl with white hair in a school uniform, illustrated by Ilya Kuvshinov.
- A minimalistic heart drawing created using Adobe Illustrator.
- The image is a digital art headshot of an owlfolk character with high detail and dramatic lighting.
- Close up of an eye with the Earth inside the pupil, inspired by Wes Anderson's art.
- Landrover drives through a rain-soaked forest in a highly-detailed digital artwork by Greg Rutkowski and Artgerm.
- A snowy lake in Sweden captured in a vibrant, cinematic style with intense detail and raytracing technology showcased on Artstation.
- A heron silhouetted against a beautiful sunrise, created by Greg Rutkowski.
- A surreal portrait of a woman with a giant carnation face in a flower field at sunset with colorful clouds and a large sky, created by artist Simon Stålenhag.