

A. Results overview

In this section, we provide an overview table for the speed-up we achieved for pixel-space and latent-space diffusion models (see Tab. 3). We also provide extra samples from the text-guided image generation model as well as comparison with DDIM [38], DPM [17] and DPM++ [18] solvers in Fig. 13 and Fig. 14. We provide more experimental details on pixel-space distillation in Appendix B and latent-space distillation in Appendix C.

B. Pixel-space distillation

B.1. Teacher model

The model architecture we use is a U-Net model similar to the ones used in [6]. The model is parameterized to predict \mathbf{v} as discussed in [33]. We use the same training setting as [6].

B.2. Stage-one distillation

The model architecture we use is a U-Net model similar to the ones used in [6]. We use the same number of channels and attention as used in [6] for both ImageNet 64x64 and CIFAR-10. As mentioned in Section 3, we also make the model take w as input. Specifically, we apply Fourier embedding to w before combining with the model backbone. The way we incorporate w is the same as how time-step is incorporated to the model as used in [10, 33]. We parameterize the model to predict \mathbf{v} as discussed in [33]. We train the distilled model using Algorithm 1. We train the model using SNR loss [10, 33]. For ImageNet 64x64, we use learning rate $3e-4$, with EMA decay 0.9999; for CIFAR-10, we use learning rate $1e-3$, with EMA decay 0.9999. We initialize the student model with parameters from the teacher model except for the parameters related to w -embedding.

Algorithm 1 Stage-one distillation

Require: Trained classifier-free guidance teacher model $[\hat{\mathbf{x}}_{c,\theta}, \hat{\mathbf{x}}_{\theta}]$

Require: Data set \mathcal{D}

Require: Loss weight function $\omega(\cdot)$

while not converged **do**

$\mathbf{x} \sim \mathcal{D}$ ▷ Sample data

$t \sim U[0, 1]$ ▷ Sample time

$w \sim U[w_{\min}, w_{\max}]$ ▷ Sample guidance

$\epsilon \sim N(0, I)$ ▷ Sample noise

$\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$ ▷ Add noise to data

$\lambda_t = \log[\alpha_t^2 / \sigma_t^2]$ ▷ log-SNR

$\hat{\mathbf{x}}_{\theta}^w(\mathbf{z}_t) = (1 + w)\hat{\mathbf{x}}_{c,\theta}(\mathbf{z}_t) - w\hat{\mathbf{x}}_{\theta}(\mathbf{z}_t)$ ▷ Compute target

$L_{\eta_1} = \omega(\lambda_t) \|\hat{\mathbf{x}}_{\theta}^w(\mathbf{z}_t) - \hat{\mathbf{x}}_{\eta_1}(\mathbf{z}_t, w)\|_2^2$ ▷ Loss

$\eta_1 \leftarrow \eta_1 - \gamma \nabla_{\eta_1} L_{\eta_1}$ ▷ Optimization

end while

B.3. Stage-two distillation for deterministic sampler

We use the same model architectures as the ones used in Stage-one (see Appendix B.2). We train the distilled model using Algorithm 2. We first use the student model from Stage-one as the teacher model. We start from 1024 DDIM sampling steps and progressively distill the student model from Stage-one to a one step model. We train the student model for 50,000 parameter updates, except for sampling step equals to one or two where we train the model for 100,000 parameter updates, before the number of sampling step is halved and the student model becomes the new teacher model. At each sampling step, we initialize the student model with the parameters from the teacher model. We train the model using SNR truncation loss [10, 33]. For each step, we linearly anneal the learning rate from $1e-4$ to 0 during each parameter update. We do not use EMA decay for training. Our training setting follows the setting in [33] closely.

Algorithm 2 Stage-two distillation for deterministic sampler

Require: Trained teacher model $\hat{\mathbf{x}}_{\eta}(\mathbf{z}_t, w)$

Require: Data set \mathcal{D}

Require: Loss weight function $\omega(\cdot)$

Require: Student sampling steps N

for K iterations **do**

$\eta_2 \leftarrow \eta$ ▷ Init student from teacher

while not converged **do**

$\mathbf{x} \sim \mathcal{D}$

$t = i/N, i \sim \text{Cat}[1, 2, \dots, N]$

$w \sim U[w_{\min}, w_{\max}]$ ▷ Sample guidance

$\epsilon \sim N(0, I)$

$\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$

 # 2 steps of DDIM with teacher

$t' = t - 0.5/N, t'' = t - 1/N$

$\mathbf{z}_{t'}^w = \alpha_{t'} \hat{\mathbf{x}}_{\eta}(\mathbf{z}_t, w) + \frac{\sigma_{t'}}{\sigma_t} (\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_{\eta}(\mathbf{z}_t, w))$

$\mathbf{z}_{t''}^w = \alpha_{t''} \hat{\mathbf{x}}_{\eta}(\mathbf{z}_{t'}^w, w) + \frac{\sigma_{t''}}{\sigma_{t'}} (\mathbf{z}_{t'}^w - \alpha_{t'} \hat{\mathbf{x}}_{\eta}(\mathbf{z}_{t'}^w, w))$

$\tilde{\mathbf{x}}^w = \frac{\mathbf{z}_{t''}^w - (\sigma_{t''}/\sigma_t) \mathbf{z}_t}{\alpha_{t''} - (\sigma_{t''}/\sigma_t) \alpha_t}$ ▷ Teacher $\hat{\mathbf{x}}$ target

$\lambda_t = \log[\alpha_t^2 / \sigma_t^2]$

$L_{\eta_2} = \omega(\lambda_t) \|\tilde{\mathbf{x}}^w - \hat{\mathbf{x}}_{\eta_2}(\mathbf{z}_t, w)\|_2^2$

$\eta_2 \leftarrow \eta_2 - \gamma \nabla_{\eta_2} L_{\eta_2}$

end while

$\eta \leftarrow \eta_2$ ▷ Student becomes next teacher

$N \leftarrow N/2$ ▷ Halve number of sampling steps

end for

B.4. Stage-two distillation for stochastic sampling

We train the distilled model using Algorithm 3. We use the same model architecture and training setting as Stage-two distillation described in Appendix B.3 for both ImageNet 64x64 and CIFAR-10: The main difference here is that our distillation target corresponds to taking a sampling step that

Space	Task	Dataset	Metric	Student diffusion step	Comparable teacher diffusion step	Speed-up
Pixel-space	class-conditional generation	CIFAR-10	FID	4	1024 DDIM×2	×512
	class-conditional generation	CIFAR-10	IS	4	1024 DDIM×2	×512
	class-conditional generation	ImageNet 64×64	FID	8	1024 DDIM×2	×256
	class-conditional generation	ImageNet 64×64	IS	8	1024 DDIM×2	×256
Latent-space	class-conditional generation	ImageNet 256×256	FID	2	16 DDIM ×2	×16
	class-conditional generation	ImageNet 256×256	Recall	2	16 DDIM ×2	×16
	text-guided generation	LAION-5B 512× 512	FID	2	16 DDIM / 8 DPM++ ×2	×16 / ×8
	text-guided generation	LAION-5B 512× 512	CLIP	4	8 DDIM / 4 DPM++ ×2	×8 / ×4

Table 3. Speed-up overview for pixel-space diffusion and latent-space diffusion. We note that the original model (without distillation) requires evaluating both the unconditional and the conditional diffusion model at each denoising step. Our model, on the other hand, only requires evaluating one diffusion model at each denoising step. This is because in our stage-one distillation, we distill the output of the unconditional and conditional models into the output of one model. Thus our method further decreases either the peak memory or sampling time by a half compared to the original model.

is twice as large as for the deterministic sampler. We provide visualization for samples with varying guidance strengths w in Fig. 15.

Algorithm 3 Stage-two distillation for stochastic sampler

Require: Trained teacher model $\hat{\mathbf{x}}_\eta(\mathbf{z}_t, w)$
Require: Data set \mathcal{D}
Require: Loss weight function $\omega(\cdot)$
Require: Student sampling steps N
for K iterations **do**
 $\eta_2 \leftarrow \eta$ ▷ Init student from teacher
 while not converged **do**
 $\mathbf{x} \sim \mathcal{D}$
 $t = i/N, i \sim \text{Cat}[1, 2, \dots, N]$
 $w \sim U[w_{\min}, w_{\max}]$ ▷ Sample guidance
 $\epsilon \sim N(0, I)$
 $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$
 if $t > 1/N$ **then**
 # 2 steps of DDIM with teacher
 $t' = t - 1/N, t'' = t - 2/N$
 $\mathbf{z}_{t'}^w = \alpha_{t'} \hat{\mathbf{x}}_\eta(\mathbf{z}_t, w) + \frac{\sigma_{t'}}{\sigma_t} (\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_\eta(\mathbf{z}_t, w))$
 $\mathbf{z}_{t''}^w = \alpha_{t''} \hat{\mathbf{x}}_\eta(\mathbf{z}_{t'}^w, w) + \frac{\sigma_{t''}}{\sigma_{t'}} (\mathbf{z}_{t'}^w - \alpha_{t'} \hat{\mathbf{x}}_\eta(\mathbf{z}_{t'}^w, w))$
 $\tilde{\mathbf{x}}^w = \frac{\mathbf{z}_{t''}^w - (\sigma_{t''}/\sigma_t) \mathbf{z}_t}{\alpha_{t''} - (\sigma_{t''}/\sigma_t) \alpha_t}$ ▷ Teacher $\hat{\mathbf{x}}$ target
 else ▷ Edge case
 # 1 step of DDIM with teacher
 $t' = t - 1/N$
 $\mathbf{z}_{t'}^w = \alpha_{t'} \hat{\mathbf{x}}_\eta(\mathbf{z}_t, w) + \frac{\sigma_{t'}}{\sigma_t} (\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_\eta(\mathbf{z}_t, w))$
 $\tilde{\mathbf{x}}^w = \frac{\mathbf{z}_{t'}^w - (\sigma_{t'}/\sigma_t) \mathbf{z}_t}{\alpha_{t'} - (\sigma_{t'}/\sigma_t) \alpha_t}$ ▷ Teacher $\hat{\mathbf{x}}$ target
 end if
 $\lambda_t = \log[\alpha_t^2 / \sigma_t^2]$
 $L_{\eta_2} = \omega(\lambda_t) \|\tilde{\mathbf{x}}^w - \hat{\mathbf{x}}_{\eta_2}(\mathbf{z}_t, w)\|_2^2$
 $\eta_2 \leftarrow \eta_2 - \gamma \nabla_{\eta_2} L_{\eta_2}$
 end while
 $\eta \leftarrow \eta_2$ ▷ Student becomes next teacher
 $N \leftarrow N/2$ ▷ Halve number of sampling steps
 end for

B.5. Baseline samples

We provide extra samples for the DDIM baseline in Fig. 16 and Fig. 17.

B.6. Extra distillation results

We provide the FID and IS results for our method and the baselines on ImageNet 64x64 and CIFAR-10 in Fig. 22b, Fig. 22a and Tab. 4. We also visualize the FID and IS trade-off curves for both datasets in Fig. 18 and Fig. 19, where we select guidance strength $w = \{0, 0.3, 1, 2, 4\}$ for ImageNet 64x64 and $w = \{0, 0.1, 0.2, 0.3, 0.5, 0.7, 1, 2, 4\}$ for CIFAR-10.

B.7. Style transfer

We focus on ImageNet 64x64 for this experiment. As discussed in [41], one can perform style-transfer between domain A and B by encoding (performing reverse DDIM) an image using a diffusion model train on domain A and then decoding using DDIM with a diffusion model trained on domain B. We train the model using Algorithm 4. We use the same w -conditioned model architecture and training setting as discussed in Appendix B.3.



Figure 13. Text-guided image generation on LAION-5B (512×512). We compare our distilled model with the original model sampled with DDIM [38] and DPM++ [18]. We observe that our model, when using only two steps, is able to generate more realistic and higher quality images compared to the baselines using more steps. We note that both DDIM and DPM-Solver require evaluating both a conditional and an unconditional diffusion model at each denoising step, while we distill the two models into one model at our stage-one distillation and only require evaluating one model at each denoising step. Depending on the implementation, DDIM and DPM-Solver require either extra $\times 2$ peak memory or $\times 2$ sampling steps compared to our approach.

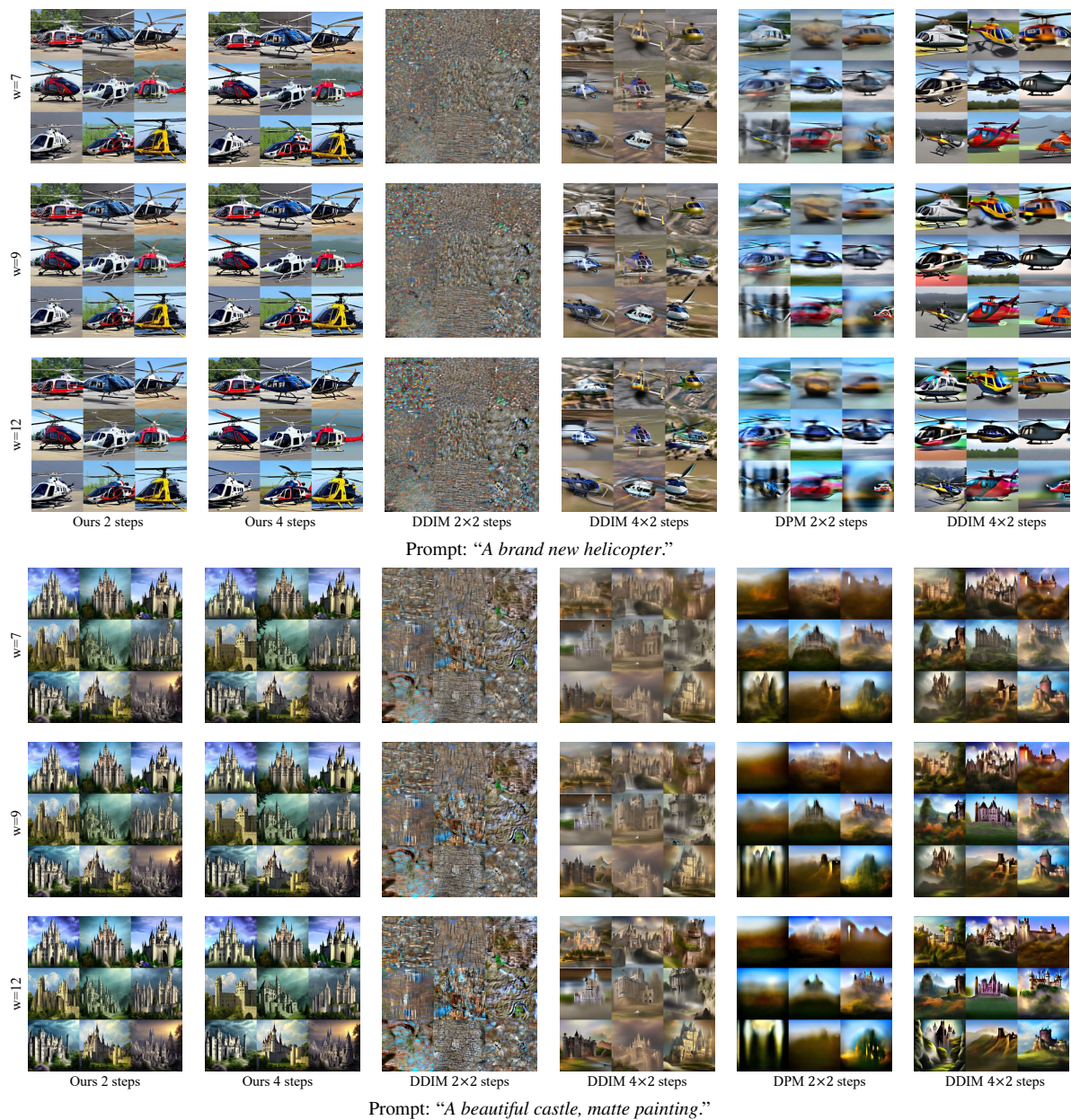


Figure 14. Text-guided image generation on LAION-5B (512×512). We compare our distilled model with the original model sampled with DDIM [38] and DPM++ [18]. We observe that our model, when using only two steps, is able to generate more realistic and higher quality images compared to the baselines using more steps. We note that both DDIM and DPM-Solver require evaluating both a conditional and an unconditional diffusion model at each denoising step, while we distill the two models into one model at our stage-one distillation and only require evaluating one model at each denoising step. Depending on the implementation, DDIM and DPM-Solver require either extra $\times 2$ peak memory or $\times 2$ sampling steps compared to our approach.



Figure 15. Class-conditional samples from our two-step (stochastic) approach on ImageNet 64x64. By varying the guidance weight w , our distilled model is able to trade-off between sample diversity and quality, while achieving visually pleasant results using as few as *one* sampling step.



Figure 16. ImageNet 64x64 class-conditional generation using DDIM (baseline) 8×2 sampling steps. We observe clear artifacts when $w = 0$.



Figure 17. ImageNet 64x64 class-conditional generation using DDIM (baseline) 16×2 sampling steps.

		ImageNet 64x64		CIFAR-10	
Guidance w	Model	FID (\downarrow)	IS (\uparrow)	FID (\downarrow)	IS (\uparrow)
$w = 0.0$	Ours 1-step (D/S)	22.74 / 26.91	25.51 / 23.55	8.34 / 10.65	8.63 / 8.42
	Ours 2-step (D/S)	9.75 / 10.67	36.69 / 37.12	4.48 / 4.81	9.23 / 9.30
	Ours 4-step (D/S)	4.14 / 3.91	46.64 / 48.92	3.18 / 3.28	9.50 / 9.60
	Ours 8-step (D/S)	2.79 / 2.44	50.72 / 55.03	2.86 / 3.11	9.68 / 9.74
	Ours 16-step (D/S)	2.44 / 2.10	52.53 / 57.81	2.78 / 3.12	9.67 / 9.76
	Single- w 1-step	19.61	24.00	6.64	8.88
	Single- w 4-step	4.79	38.77	3.14	9.47
	Single- w 8-step	3.39	42.13	2.86	9.67
	Single- w 16-step	2.97	43.63	2.75	9.65
	DDIM 16 \times 2-step [38]	7.68	37.60	10.11	8.81
	DDIM 32 \times 2-step [38]	5.03	40.93	6.67	9.17
	DDIM 64 \times 2-step [38]	3.74	43.16	4.64	9.32
	Target (DDIM 1024 \times 2-step)	2.92	44.81	2.73	9.66
	$w = 0.3$	Ours 1-step (D/S)	14.85 / 18.48	37.09 / 33.30	7.34 / 9.38
Ours 2-step (D/S)		5.052 / 5.81	54.44 / 54.37	4.23 / 4.74	9.45 / 9.45
Ours 4-step (D/S)		2.17 / 2.24	69.64 / 73.73	3.58 / 3.95	9.73 / 9.77
Ours 8-step (D/S)		2.05 / 2.31	76.01 / 83.00	3.54 / 3.96	9.87 / 9.90
Ours 16-step (D/S)		2.20 / 2.56	79.47 / 87.50	3.57 / 4.17	9.89 / 9.97
Single- w 1-step		11.70	36.95	5.98	9.13
Single- w 4-step		2.34	62.08	3.58	9.75
Single- w 8-step		2.32	68.76	3.57	9.85
Single- w 16-step		2.56	70.97	3.61	9.88
DDIM 16 \times 2-step		5.33	60.83	10.83	8.96
DDIM 32 \times 2-step		3.45	68.03	7.47	9.33
DDIM 64 \times 2-step		2.80	72.55	5.52	9.51
Target (DDIM 1024 \times 2-step)		2.36	74.83	3.65	9.83
$w = 1.0$		Ours 1-step (D/S)	7.54 / 8.92	75.19 / 67.80	8.62 / 10.27
	Ours 2-step (D/S)	5.77 / 5.83	109.97 / 108.38	6.88 / 7.52	9.64 / 9.55
	Ours 4-step (D/S)	7.95 / 8.51	128.98 / 135.36	7.39 / 7.64	9.86 / 9.87
	Ours 8-step (D/S)	9.33 / 10.56	136.47 / 147.39	7.81 / 7.85	9.9 / 10.05
	Ours 16-step (D/S)	9.99 / 11.63	139.11 / 153.17	7.97 / 8.34	10.00 / 10.05
	Single- w 1-step	6.64	74.41	8.18	9.32
	Single- w 4-step	8.23	118.52	7.66	9.88
	Single- w 8-step	9.69	125.20	8.09	9.89
	Single- w 16-step	10.34	127.70	8.30	9.95
	DDIM 16 \times 2-step	9.53	112.75	14.81	8.98
	DDIM 32 \times 2-step	9.26	126.22	11.44	9.36
	DDIM 64 \times 2-step	9.53	133.17	9.79	9.64
	Target (DDIM 1024 \times 2-step)	9.84	139.50	7.80	9.96
	$w = 2.0$	Ours 1-step (D/S)	10.71 / 10.55	118.55 / 108.37	13.23 / 14.33
Ours 2-step (D/S)		14.08 / 14.18	160.04 / 161.43	12.58 / 12.57	9.51 / 9.48
Ours 4-step (D/S)		17.61 / 18.23	178.29 / 184.45	13.83 / 13.24	9.70 / 9.77
Ours 8-step (D/S)		18.80 / 20.25	181.53 / 193.49	14.41 / 13.67	9.77 / 9.87
Ours 16-step (D/S)		19.25 / 21.11	183.17 / 197.71	14.80 / 14.28	9.79 / 9.84
Single- w 1-step		11.12	120.74	13.31	9.23
Single- w 4-step		18.14	172.74	14.04	9.70
Single- w 8-step		19.24	176.74	14.67	9.77
Single- w 16-step		19.81	177.69	15.04	9.79
DDIM 16 \times 2-step		15.92	157.67	20.25	8.97
DDIM 32 \times 2-step		16.85	175.72	17.27	9.29
DDIM 64 \times 2-step		17.53	182.11	15.66	9.48
Target (DDIM 1024-step)		17.97	190.56	13.60	9.81
$w = 4.0$		Ours 1-step (D/S)	18.72 / 17.85	157.46 / 148.97	23.20 / 23.79
	Ours 2-step (D/S)	23.74 / 24.34	196.05 / 200.11	23.41 / 22.75	9.16 / 9.11
	Ours 4-step (D/S)	26.45 / 27.33	207.45 / 216.56	25.11 / 23.62	9.23 / 9.33
	Ours 8-step (D/S)	26.62 / 27.84	203.47 / 219.89	25.94 / 23.98	9.26 / 9.55
	Ours 16-step (D/S)	26.53 / 27.69	204.13 / 218.70	26.01 / 24.40	9.33 / 9.50
	Single- w 1-step	19.857	170.69	23.17	8.93
	Single- w 4-step	27.75	219.64	24.45	9.32
	Single- w 8-step	27.67	218.08	24.83	9.38
	Single- w 16-step	27.40	216.52	25.11	9.37
	DDIM 16 \times 2-step	21.56	195.17	27.99	8.71
	DDIM 32 \times 2-step	23.03	213.23	25.07	9.07
	DDIM 64 \times 2-step	23.64	217.88	23.41	9.17
	Target (DDIM 1024 \times 2-step)	23.94	224.74	21.28	9.54

Table 4. Distillation results on ImageNet 64x64 and CIFAR-10 ($w = 0$ refers to non-guided models). For our method, D and S stand for deterministic and stochastic sampler respectively. We observe that training the model conditioned on an guidance interval $w \in [0, 4]$ performs comparably with training a model on a fixed w (see Single- w). Our approach significantly outperforms DDIM when using fewer steps, and is able to match the teacher performance using as few as 8 to 16 steps. We also note that DDIM and DDPM evaluates both an unconditional and a conditional diffusion model at each denoising step, giving rise to the $\times 2$ overhead either for peak memory or sampling steps.

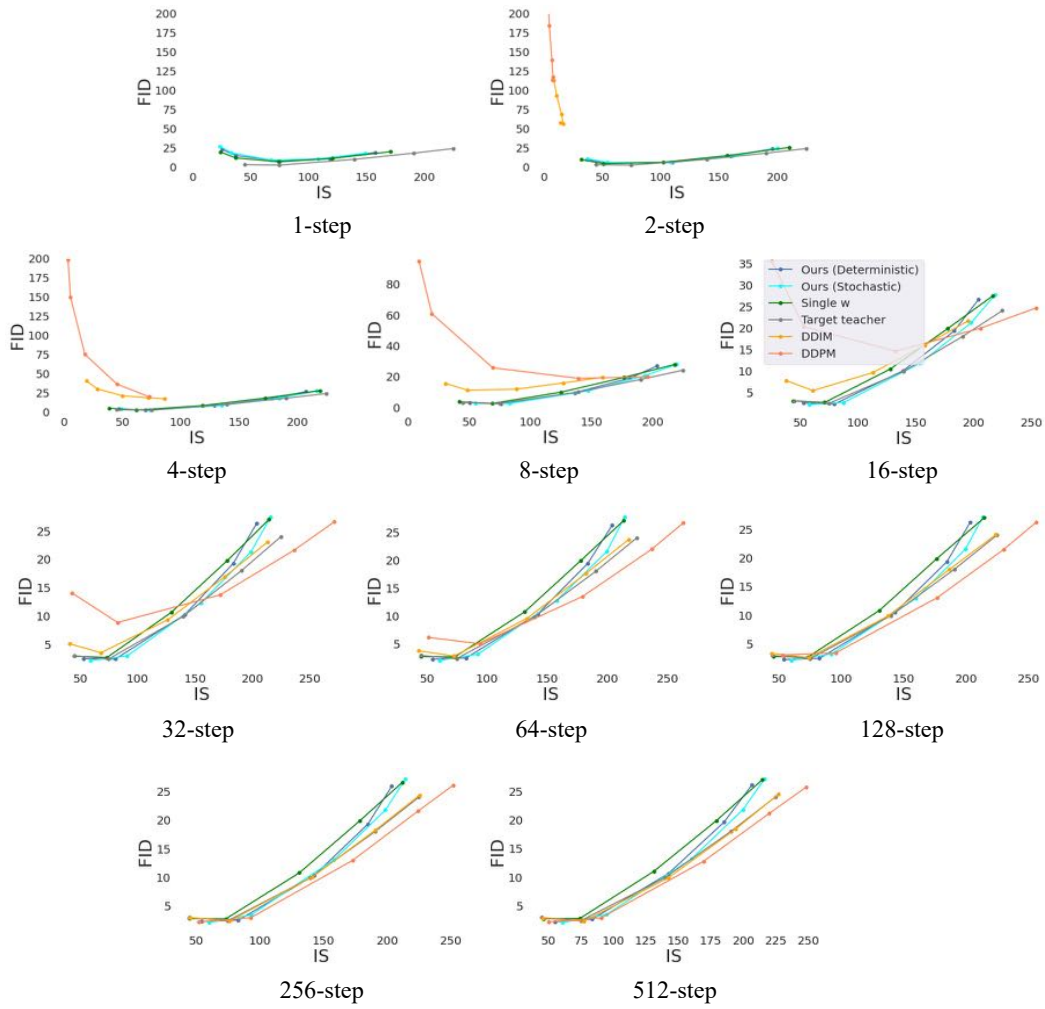


Figure 18. FID and IS score trade-off on ImageNet 64x64. We plot the results using guidance strength $w = \{0, 0.3, 1, 2, 4\}$. For the 1-step plot, the curves of DDIM and DDPM are too far to be visualized.

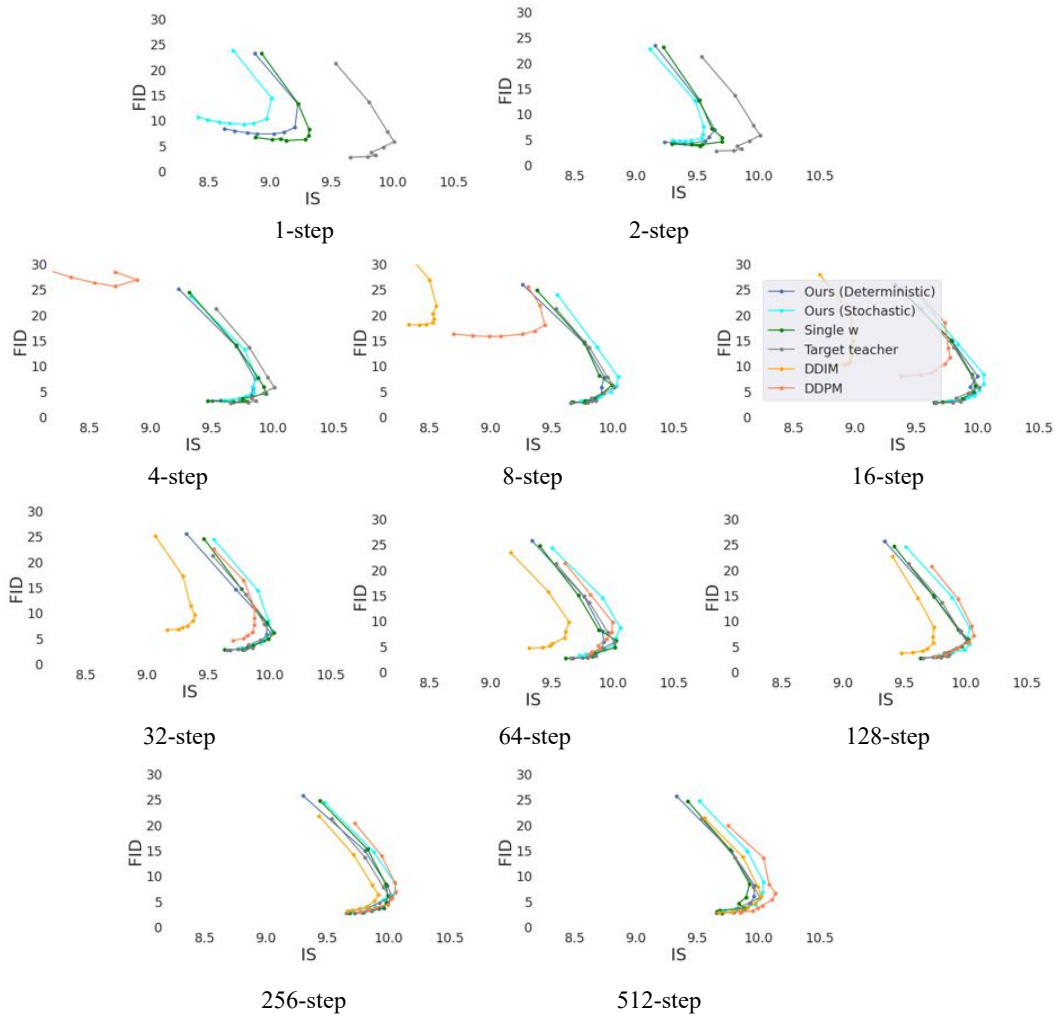


Figure 19. FID and IS score trade-off on CIFAR-10. We plot the results using guidance strength $w = \{0, 0.1, 0.2, 0.3, 0.5, 0.7, 1, 2, 4\}$. For the 1-step and 2-step plots, the curves of DDIM and DDPM are too far away to be visualized. For the 4-step plot, the curve of DDIM is too far away to be visualized.



Figure 20. Style transfer on ImageNet 64x64 for pixel-space models (orange to bell pepper). We use a distilled 16-step encoder and decoder. We fix the encoder guidance strength to be 0 and vary the decoder guidance strength from 0 to 4. As we increase w , we notice a trade-off between sample diversity and sharpness.



Figure 21. Style transfer on ImageNet 64x64 (orange to acorn squash). We use a distilled 16-step encoder and decoder. We fix the encoder guidance strength to be 0 and vary the decoder guidance strength from 0 to 4. As we increase the guidance strength w , we notice a trade-off between sample diversity and sharpness.

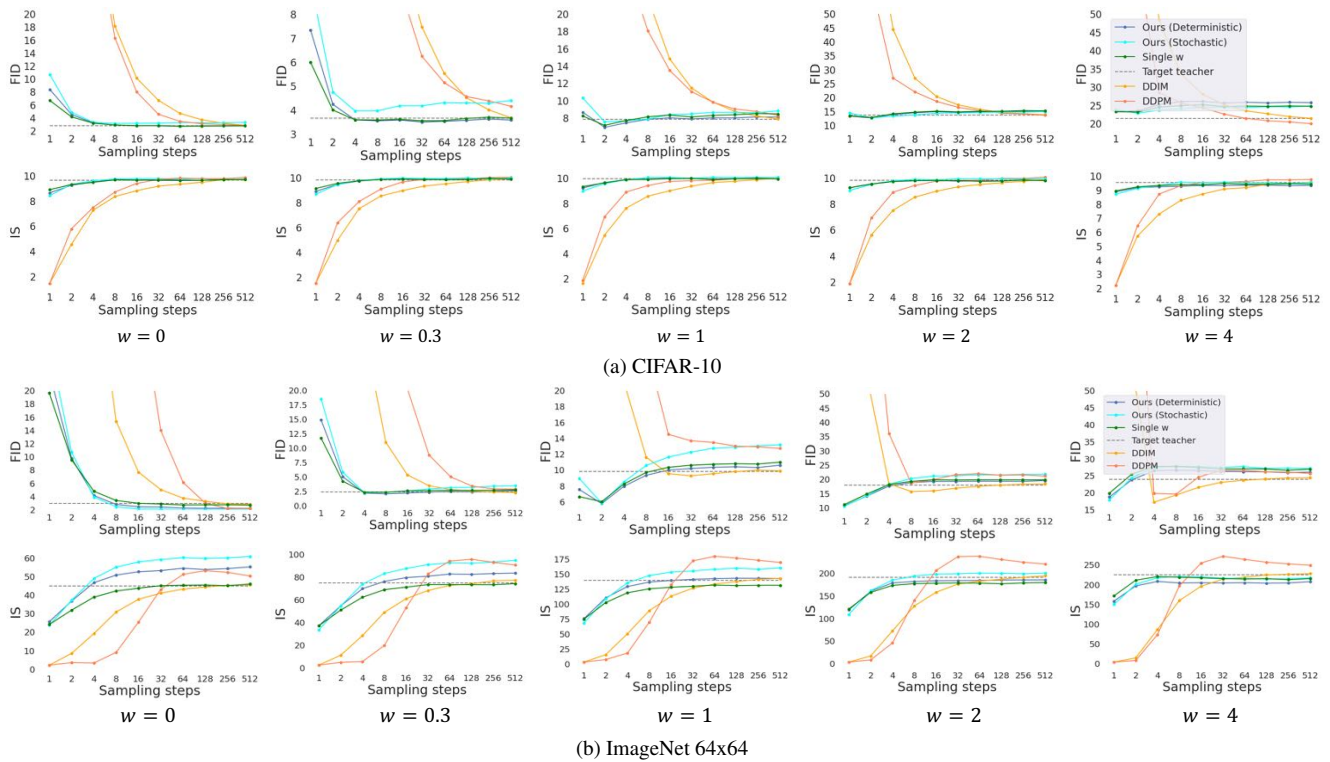


Figure 22. CIFAR-10 and ImageNet sample quality evaluated by FID and IS scores for pixel-space diffusion models. We follow the setting of [33] for our evaluation. We note that, the DDPM and DDIM baseline require evaluating both an unconditional and a conditional diffusion model at each denoising step for classifier-free guidance, giving rise to either an extra $\times 2$ overhead for peak memory or an extra $\times 2$ sampling steps than the “Sampling steps” value shown in the plot. Our distilled model significantly outperform the DDPM and DDIM baselines, and is able to match the performance of the teacher using as few as 4 to 16 steps. By varying w , a *single* distilled model is able to capture the trade-off between sample diversity and quality.

Algorithm 4 Encoder distillation

Require: Trained teacher model $\hat{\mathbf{x}}_\eta(\mathbf{z}_t, w)$ **Require:** Data set \mathcal{D} **Require:** Loss weight function $\omega(\cdot)$ **Require:** Student sampling steps N **for** K iterations **do** $\eta_2 \leftarrow \eta$ \triangleright Init student from teacher**while** not converged **do** $\mathbf{x} \sim \mathcal{D}$ $t = i/N, i \sim \text{Cat}[0, 1, \dots, N-1]$ $w \sim U[w_{\min}, w_{\max}]$ \triangleright Sample guidance $\epsilon \sim N(0, I)$ $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$

2 steps of reversed DDIM with

teacher

 $t' = t + 0.5/N, t'' = t + 1/N$ $\mathbf{z}_{t'}^w = \alpha_{t'} \hat{\mathbf{x}}_\eta(\mathbf{z}_t, w) + \frac{\sigma_{t'}}{\sigma_t} (\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_\eta(\mathbf{z}_t, w))$ $\mathbf{z}_{t''}^w = \alpha_{t''} \hat{\mathbf{x}}_\eta(\mathbf{z}_{t'}^w, w) + \frac{\sigma_{t''}}{\sigma_{t'}} (\mathbf{z}_{t'}^w - \alpha_{t'} \hat{\mathbf{x}}_\eta(\mathbf{z}_{t'}^w, w))$ $\tilde{\mathbf{x}}^w = \frac{\mathbf{z}_{t''}^w - (\sigma_{t''}/\sigma_t) \mathbf{z}_t}{\alpha_{t''} - (\sigma_{t''}/\sigma_t) \alpha_t}$ \triangleright Teacher $\hat{\mathbf{x}}$ target $\lambda_t = \log[\alpha_t^2/\sigma_t^2]$ $L_{\eta_2} = \omega(\lambda_t) \|\tilde{\mathbf{x}}^w - \hat{\mathbf{x}}_{\eta_2}(\mathbf{z}_t, w)\|_2^2$ $\eta_2 \leftarrow \eta_2 - \gamma \nabla_{\eta_2} L_{\eta_2}$ **end while** $\eta \leftarrow \eta_2$ \triangleright Student becomes next teacher $N \leftarrow N/2$ \triangleright Halve number of sampling steps**end for**

Algorithm 5 Two-student progressive distillation

Require: Trained classifier-free guidance teacher model $[\hat{\mathbf{x}}_{c,\theta}, \hat{\mathbf{x}}_\theta]$ **Require:** Data set \mathcal{D} **Require:** Loss weight function $\omega(\cdot)$ **Require:** Student sampling steps N **for** K iterations **do** $\eta \leftarrow \theta$ \triangleright Init student from teacher**while** not converged **do** $\mathbf{x} \sim \mathcal{D}$ $t = i/N, i \sim \text{Cat}[1, 2, \dots, N]$ $w \sim U[w_{\min}, w_{\max}]$ \triangleright Sample guidance $\epsilon \sim N(0, I)$ $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$ $\hat{\mathbf{x}}_\theta^w(\mathbf{z}_t) = (1+w) \hat{\mathbf{x}}_{c,\theta}(\mathbf{z}_t) - w \hat{\mathbf{x}}_\theta(\mathbf{z}_t)$ \triangleright

Compute target

2 steps of DDIM with teacher

 $t' = t - 0.5/N, t'' = t - 1/N$ $\mathbf{z}_{t'}^w = \alpha_{t'} \hat{\mathbf{x}}_\theta^w(\mathbf{z}_t) + \frac{\sigma_{t'}}{\sigma_t} (\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_\theta^w(\mathbf{z}_t))$ $\mathbf{z}_{c,t''}^w = \alpha_{t''} \hat{\mathbf{x}}_{c,\theta}(\mathbf{z}_{t'}^w) + \frac{\sigma_{t''}}{\sigma_{t'}} (\mathbf{z}_{t'}^w - \alpha_{t'} \hat{\mathbf{x}}_{c,\theta}(\mathbf{z}_{t'}^w))$ $\tilde{\mathbf{x}}_c^w = \frac{\mathbf{z}_{c,t''}^w - (\sigma_{t''}/\sigma_t) \mathbf{z}_t}{\alpha_{t''} - (\sigma_{t''}/\sigma_t) \alpha_t}$ \triangleright Conditional teacher $\hat{\mathbf{x}}$

target

 $\mathbf{z}_{t''}^w = \alpha_{t''} \hat{\mathbf{x}}_\theta(\mathbf{z}_{t'}^w) + \frac{\sigma_{t''}}{\sigma_{t'}} (\mathbf{z}_{t'}^w - \alpha_{t'} \hat{\mathbf{x}}_\theta(\mathbf{z}_{t'}^w))$ $\tilde{\mathbf{x}}^w = \frac{\mathbf{z}_{t''}^w - (\sigma_{t''}/\sigma_t) \mathbf{z}_t}{\alpha_{t''} - (\sigma_{t''}/\sigma_t) \alpha_t}$ \triangleright Unconditional teacher $\hat{\mathbf{x}}$

target

 $\lambda_t = \log[\alpha_t^2/\sigma_t^2]$ $L_\eta = \omega(\lambda_t) (\|\tilde{\mathbf{x}}_c^w - \hat{\mathbf{x}}_{c,\eta}(\mathbf{z}_t, w)\|_2^2 + \|\tilde{\mathbf{x}}^w - \hat{\mathbf{x}}_\eta(\mathbf{z}_t, w)\|_2^2)$ $\eta \leftarrow \eta - \gamma \nabla_\eta L_\eta$ **end while** $\theta \leftarrow \eta$ \triangleright Student becomes next teacher $N \leftarrow N/2$ \triangleright Halve number of sampling steps**end for**

Guidance w	Number of step	FID (\downarrow)	IS (\uparrow)
$w = 0.0$	1×2	212.20	3.66
	16×2	42.02	7.95
	64×2	35.37	8.47
	128×2	29.74	8.87
	256×2	20.14	9.50
$w = 0.3$	1×2	213.07	3.62
	16×2	48.74	7.70
	128×2	34.28	8.57
	256×2	24.54	9.21
$w = 1.0$	1×2	214.88	3.54
	16×2	64.92	7.21
	64×2	48.54	7.62
	128×2	42.56	8.00
	256×2	32.20	8.81
$w = 2.0$	1×2	217.37	3.48
	16×2	87.19	6.50
	64×2	57.15	7.22
	128×2	50.30	7.53
	256×2	39.76	8.26
$w = 4.0$	1×2	220.11	3.45
	16×2	115.57	6.16
	64×2	71.45	6.78
	128×2	61.75	7.02
	256×2	49.21	7.69

Table 5. Distillation results on CIFAR-10 using the naive approach mentioned in Appendix B.8. Note that the naive approach still requires evaluating both a conditional and an unconditional model at each denoising step, and thus requires $\times 2$ more steps or peak memory than our method. From the evaluated FID/IS scores, we observe that the naive distillation approach is not able to achieve strong performance.

B.8. Naive distillation approach

A natural approach to progressively distill [33] a classifier-free guided model is to use a distilled student model that follows the same structure as the teacher—that is with a jointly trained distilled conditional and unconditional diffusion component. Denote the pre-trained teacher model $[\hat{x}_{c,\theta}, \hat{x}_{\theta}]$ and the student model $[\hat{x}_{c,\eta}, \hat{x}_{\eta}]$, we provide the training algorithm in Algorithm 5. To sample from the trained model, we can use DDIM deterministic sampler [38] or the proposed stochastic sampler. We follow the training setting in Appendix B.3, use a w -conditioned model and train the model to condition on the guidance strength $[0, 4]$. We observe that the model distilled with Algorithm 5 is not able to generate reasonable samples when the number of sampling is small. We provide the generated samples on CIFAR-10 with DDIM sampler in Fig. 23, and the FID/IS scores in Tab. 5.



Figure 23. Samples using the distillation algorithm mentioned in Appendix B.8. The model is trained with guidance strength $w \in [0, 4]$ on CIFAR-10. The samples are generated with DDIM (deterministic) sampler at $w = 0$. We observe clear artifacts when the number of sampling step is small.

C. Latent-space distillation

C.1. Class-conditional generation

C.1.1 Training details

In this experiment, we consider class-conditional generation on ImageNet 256×256 . We first fine-tune the original ϵ -prediction model to a v -prediction model, and then start from the DDIM teacher model with 512 sampling steps, where we use the output as the target to train our distilled model. For

stage-one, we train the model for 2000 gradient updates with constant loss [10,33]. For stage-two, we train the model with 2000 gradient updates except when the sampling size equals to 1,2, or 4, where we train for 20000 gradient updates. We train the second stage model with SNR-truncation loss [10,33]. For both stages, we train with extra 500 learning rate warm-up steps, where we linearly increase the learning rate from zero to the target learning rate. We use a batch size of 2048 and uniformly sample the guidance strength $w \in [w_{min} = 0, w_{max} = 14]$ during training.

Additional results We provide quantitative results evaluated by precision and recall in Fig. 25. These results confirm a significant performance boost of our method in the small-step regime, especially for 1-4 sampling steps. Our distilled latent diffusion model for 2- and 4-step sampling nearly matches DDIM performance at 32 steps in terms of precision and significantly outperforms it in terms of recall for low numbers of steps. For more qualitative results, see Fig. 25, where we depict random samples for the 1- and 2-step model and contrast them to DDIM sampling.

C.2. Text-guided image generation

C.2.1 Training details

We consider the LAION-5B datasets with resolution 256×256 and 512×512 in this experiment.

LAION-5B 256×256 Similar to Appendix C.1, we first fine-tune the original ϵ -prediction model to a v -prediction model. We start from the DDIM teacher model with 512 sampling steps, and use the output as the target to train our distilled model. For stage-one, we train the model for 2000-5000 gradient updates with constant loss [10,33]. For stage-two, we train the model with 2000-5000 gradient updates except when the sampling size equals to 1,2, or 4, where we train for 10000-50000 gradient updates. We train the second stage model with SNR-truncation loss [10,33]. For both stages, we train with extra 100-1000 learning rate warm-up steps, where we linearly increase the learning rate from zero to the target learning rate. We use a batch size of 1024 and uniformly sample the guidance strength $w \in [w_{min} = 2, w_{max} = 14]$ during training.

Fig. 26 provides a convergence analysis of the different training setting described above. We observe that our method approaches DDIM sampling of the base model after a few thousand training iterations and outperforms it quickly in the 1- and 2-step regime. However, for maximum performance, longer training is required.

LAION-5B 512×512 Similarly, we first fine-tune the original ϵ -prediction model to a v -prediction model. We start from the DDIM teacher model with 512 sampling steps, and

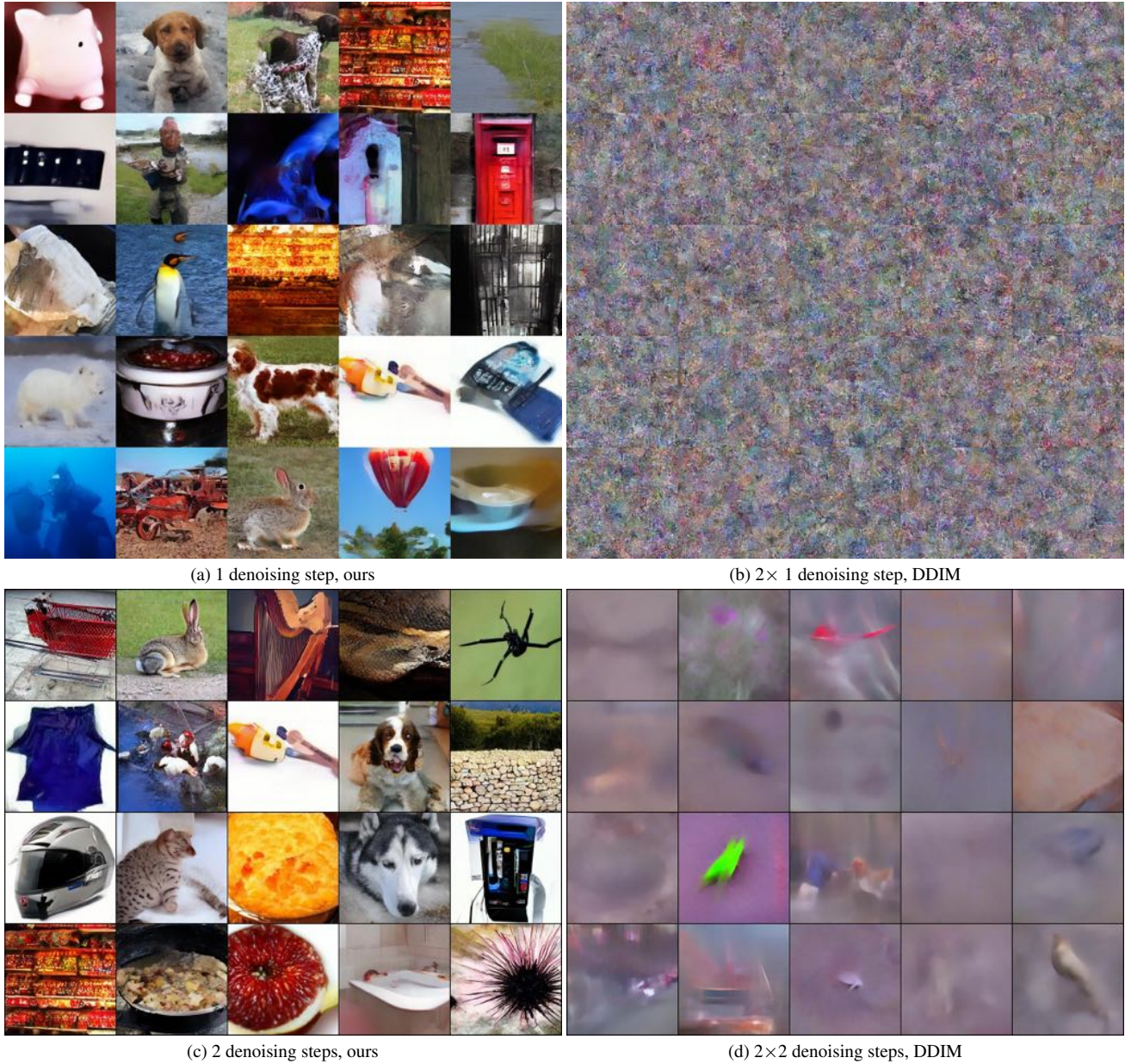


Figure 24. Random 256×256 class-conditional samples from our distilled model and from the DDIM teacher for 1 and 2 denoising steps for $w = 3.0$.

use the output as the target to train our distilled model. For stage-one, we train the model for 2000-5000 gradient updates with constant loss [10, 33]. For stage-two, we train the model with 2000-5000 gradient updates except when the sampling step equals to 1, 2, or 4, where we train for 10000-50000 gradient updates. We train the second-stage model with SNR-truncation loss [10, 33]. For both stages, we train with extra 1000 learning rate warm-up steps, where we linearly increase the learning rate from zero to the target learning rate. We use a batch size of 512 and uniformly

sample the guidance strength $w \in [w_{min} = 2, w_{max} = 14]$ during training.

Additional results Besides DDIM, we also compare our method here with DPM++-Solver [16, 18], a state-of-the-art sampler that requires no additional training and has achieved good results for ≥ 10 sampling steps for latent diffusion models. Unlike our distilled model, this method, similar to DDIM, must use classifier-free guidance to achieve good results. This doubles the number of U-Net evaluations com-

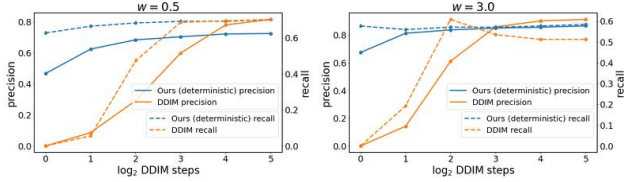


Figure 25. Precision and recall [13] for class-conditional image generation on ImageNet (256×256) with distilled latent diffusion. The results are evaluated on 5000 samples. Our distilled latent diffusion model for 2- and 4-step sampling nearly matches DDIM performance at 32×2 steps in terms of precision, and strictly outperforms it in terms of recall.

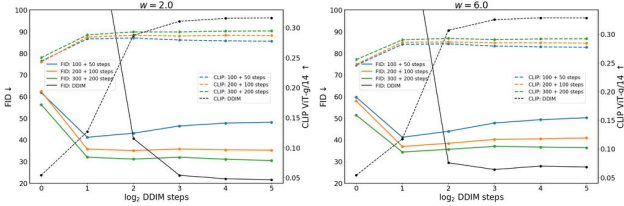


Figure 26. FID and Inception Score for text-guided image generation on LAION (256×256) with distilled latent diffusion. The results are evaluated on 5000 captions from COCO2017. We observe that our distillation method approaches DDIM sampling after only a few thousand training steps, see Appendix C.2.

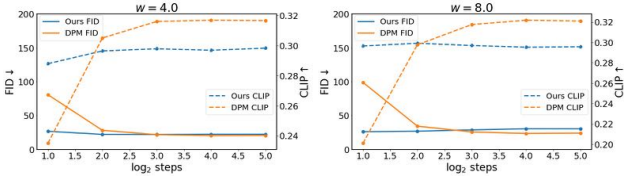


Figure 27. FID and CLIP ViT-g/14 score for text-to-image generation at 512×512 px using the distilled *Stable Diffusion* model. The results are evaluated on 5000 captions from the COCO2017 [14] validation set. Our distilled model outperforms the state-of-the-art accelerated sampler *DPM-Solver* (DPM++) [16, 18] in the 2- and 4- step regime. We believe the difference in CLIP scores for > 10 -step sampling can be closed by longer training. We stress that DPM-Solver, as DDIM, uses classifier-free guidance during sampling, which requires evaluating both an unconditional and a conditional diffusion model at each denoising step, giving rise to an extra $\times 2$ overhead compared to our method.

Setting	vs. DDIM (FID)	vs. DPM++ (FID)
2-step, $w = 2.0$	+89.8%	+69.4%
4-step, $w = 2.0$	+68.9%	+32.5%
2-step, $w = 8.0$	+89.5%	+73.7%
4-step, $w = 8.0$	+42.6%	+21.6%

Table 6. Relative performance of our distilled 512×512 LAION model compared to DDIM [38] and DPM++ [18] sampling of the base model. Note that DDIM and DPM-Solver use $2 \times$ more steps than the one listed under “Setting”, as they rely on classifier-free guidance instead of w -conditioning. This requires DDIM and DPM-Solver to evaluate both an unconditional and a conditional diffusion model at each denoising step, giving rise to the $\times 2$ overhead.

Setting	vs. DDIM (CLIP)	vs. DPM (CLIP)
2-step, $w = 2.0$	+550%	+27.9%
4-step, $w = 2.0$	+19.2%	+0.1%
2-step, $w = 8.0$	+348%	+47.5%
4-step, $w = 8.0$	+8.6%	+0.6%

Table 7. Relative performance of our distilled 512×512 LAION model compared to DDIM [38] and DPM-Solver (DPM++) [16, 18] sampling of the base model. Note that DDIM and DPM use $2 \times$ more steps than the one listed under “Setting”, as they rely on classifier-free guidance instead of w -conditioning. This requires DDIM and DPM to evaluate both an unconditional and a conditional diffusion model at each denoising step, giving rise to the $\times 2$ overhead. We use CLIP ViT-g/14 for evaluation [7, 25].

pared to our w -conditional approach.

We provide a qualitative comparison of these sampling methods in Fig. 28, where we clearly see the benefits of our distillation approach for low numbers of sampling steps: our method produces sharper and more coherent results than the training-free samplers. This behavior is reflected by the quantitative FID and CLIP analysis in Fig. 27 and Tab. 6, Tab. 7. While the speed-up here is not quite as significant as in pixel-space, our method still achieves very good results with 2 or 4 sampling steps. Our approach further reduces the maximum memory or denoising step by a half compared to existing methods due to w -conditioning (since here we no longer need to evaluate both the unconditional model and conditional model for classifier-free guidance, we only need one distilled w -conditional model). We hope that our work will lead to progress in real-time applications of general high-resolution text-to-image systems.

We also provide human evaluation results by leveraging Amazon Mechanical Turk. We generate images using text prompts from [45]. We compare our distilled model sampled using 2 or 4 denoising steps with DDIM and DPM++ solver sampled using 2×2 or 4×2 denoising steps. For each setting, we generate 100 HITs each with 17 pair-wise comparisons between samples generated with our approach and the baseline. In each of the question, the user is shown the text prompt used to generate the image and asked to select the image that looks better to them. We provide a snapshot of our user interface in Fig. 29. We provide the results in Tab. 9. Although we observe noisy answers (for instance some user would prefer the right image to the left image in Fig. 29c), our distilled model still consistently outperforms the baselines in all the settings we considered in Tab. 9. To get higher-quality user feedback and reduce the noise in the answers, in the future work, we will perform a new human evaluation with a larger sample size and extra constraints to ensure the quality of the response. We will also build a framework to automatically ignore HITs with random selections.

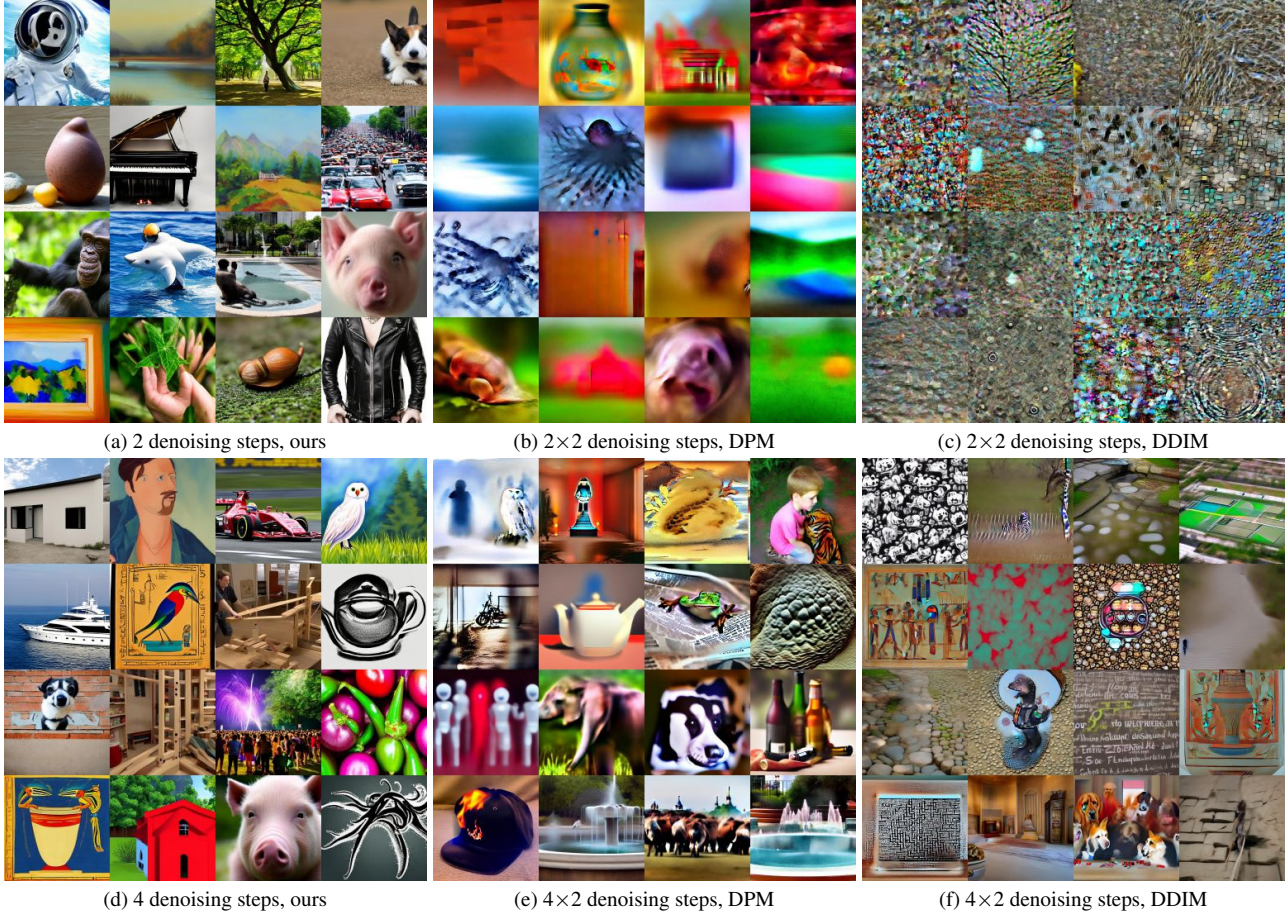


Figure 28. Random 512×512 text-guided samples from our distilled *Stable Diffusion* model compared to the DDIM teacher and DPM-solver for 2 and 4 denoising steps for $w = 11.5$.

C.3. Text-guided image-to-image translation

C.3.1 Training details

We use the model trained for text-guided image generation. The training details can be found in Appendix C.2.

C.3.2 Extra analysis

We provide more analysis on the trade-off between sample quality, controllability and efficiency in Fig. 30 and Fig. 31. Similar to [20], we also observe a trade-off between realism, controllability and faithfulness as we increase the initial perturbed noise level: the more noise we add, the more aligned the images are to the text prompt, but less faithful to the input image (see Fig. 30 and Fig. 31).

C.4. Image inpainting

C.4.1 Training details

Similar to our previous experiments, we fine-tune the ϵ -prediction model to a v -prediction model, using the large

Setting	Ours (FID ↓)	DDIM (FID ↓)
2-step, $w = 4.0$	29.50	109.35
4-step, $w = 4.0$	24.90	26.89
2-step, $w = 11.0$	31.43	105.71
4-step, $w = 11.0$	24.36	27.22

Table 8. Quantitative inpainting results as evaluated by FID. We evaluate on 2000 examples from COCO2017. Note that DDIM, which is evaluated with classifier-free guidance, uses two times more function evaluations than the one listed under “Setting”.


mask generation scheme suggested in LAMA [42] and train on LAION-5B at 512×512 resolution. We start from the DDIM teacher model with 512 sampling steps, and use the output as the target to train our distilled model. For stage-one, we train the model for 2000 gradient updates with constant loss [10, 33]. For stage-two, we train the model with 10000 gradient updates except when the sampling size equals to 1 or 2, where we train for 5000 gradient updates. We train the second stage model with SNR-truncation loss [10, 33]. For

Given the text prompt: "a glass of orange juice", how would you imagine this image to look like?
 Choose the image that looks **more reasonable** to you.
 Your selection should be based on how **realistic** and **less blurry** the image is, and whether it **follows the text prompt**.

About this HIT:

- Please only participate in this HIT if you have normal color vision.
- It should take about 1 minute.
- You will take part in an experiment involving visual perception. You'll see a text prompt and a series of pairs of images. In each pair, given the text prompt, the images are "fake" images generated using a computer program. Choose the image that looks **more reasonable** to you. Your selection should be based on how **realistic** and **less blurry** the image is, and whether the image **follows the text prompt**.

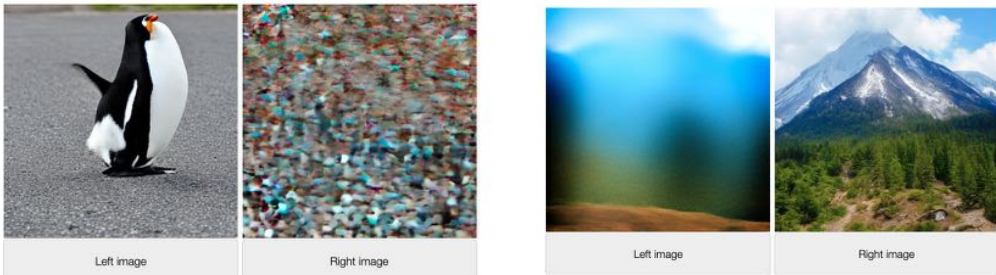
Start!



(a) Instructions for the human evaluators on Amazon Mechanical Turk. (b) Images generated by our 4-step distillation model (left) and images generated by the 4×2 -step baseline (right).

Given the text prompt: "a penguin standing on a sidewalk", how would you imagine this image to look like?
 Choose the image that looks **more reasonable** to you.
 Your selection should be based on how **realistic** and **less blurry** the image is, and whether it **follows the text prompt**.

Given the text prompt: "a mountain", how would you imagine this image to look like?
 Choose the image that looks **more reasonable** to you.
 Your selection should be based on how **realistic** and **less blurry** the image is, and whether it **follows the text prompt**.



(c) Images generated by our 2-step distillation model (left) and (d) Images generated by the 2×2 -step baseline (left) and images generated by the 2×2 -step baseline (right).

Figure 29. A snapshot of the human evaluation interface we used on Amazon Mechanical Turk.

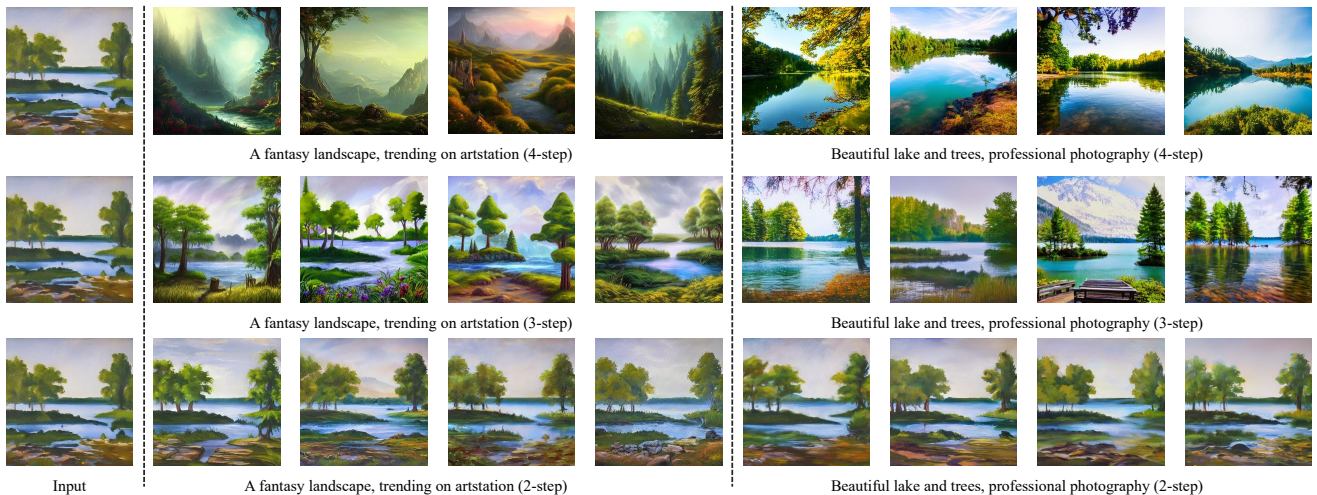


Figure 30. In this example, we study the trade-off between efficiency, realism, and controllability for guided image translation with SDEdit [20]. We use a 4-step distilled text-guided image generation model trained on LAION-5B (512×512). The training detail is discussed in Appendix C.2. Given an input image (guide), we consider perturbing the input image with different noise level, with 2 denoising step corresponding to perturb the image with around 50% noise, and 4 denoising step corresponding to perturb the image with around 100% noise according to the DDIM noise schedule. We observe that the more noise we perturb, the more aligned the images are with the text prompt, but the less faithful they are to the input image.

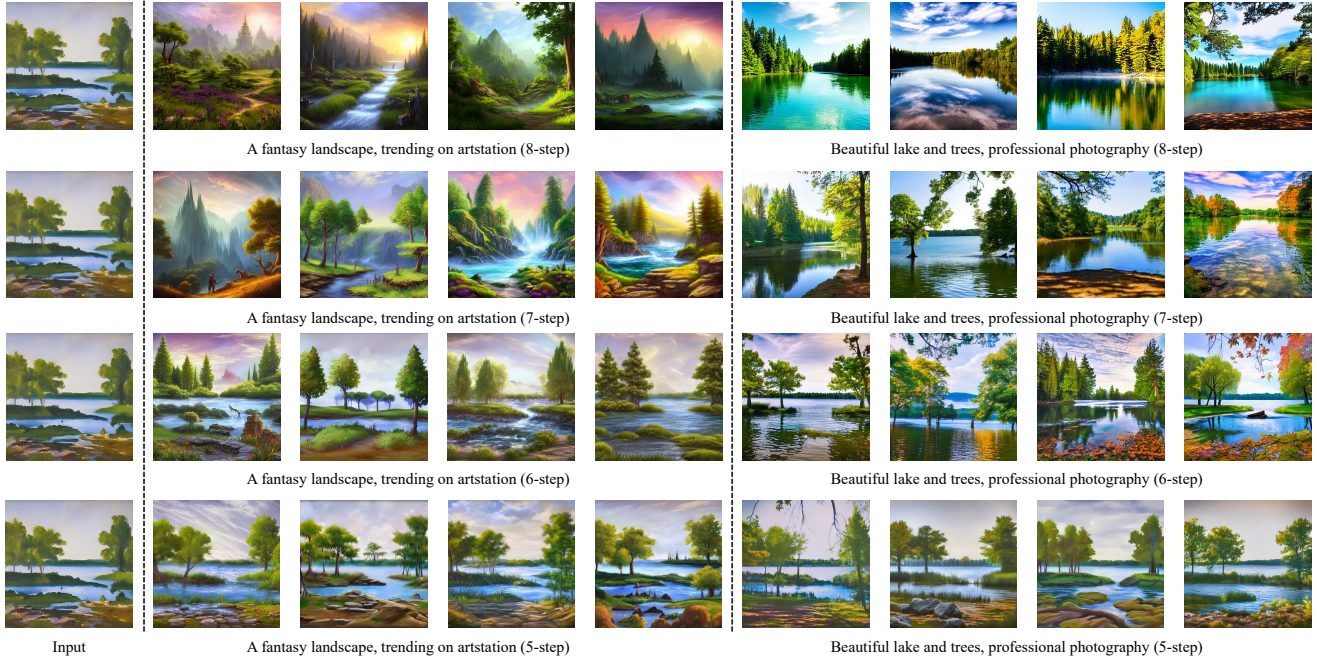


Figure 31. In this example, we study the trade-off between efficiency, realism, and controllability for guided image translation with SDEdit [20]. We use a 8-step distilled text-guided image generation model trained on LAION-5B (512×512). The training detail is discussed in Appendix C.2. Given an input image (guide), we consider perturbing the input image with different noise level, with 5 denoising step corresponding to perturb the image with around 60% noise, and 8 denoising step corresponding to perturb the image with around 100% noise according to the DDIM noise schedule. We observe that the more noise we perturb, the more aligned the images are with the text prompt, but the less faithful they are to the input image.



Figure 32. Random 512×512 inpainting samples from our distilled model and from the DDIM teacher for 2 denoising steps for $w = 11.0$.

both stages, we train with extra 1000 learning rate warm-up steps, where we linearly increase the learning rate from zero to the target learning rate. We use a batch size of 512 and uniformly sample the guidance strength $w \in [w_{min} = 2, w_{max} = 14]$ during training.

Additional evaluation results A quantitative comparison with DDIM sampling at low sampling numbers of sampling steps can be found in Tab. 8, additional samples are in Fig. 32.

Ours	Baseline	Our method is better (\uparrow)
Distillation 2-step	DDIM 2 \times 2-step	66.32%
Distillation 2-step	DPM++ 2 \times 2-step	68.97%
Distillation 2-step	DDIM 4 \times 2-step	57.44%
Distillation 2-step	DPM++ 4 \times 2-step	59.88%
Distillation 4-step	DDIM 4 \times 2-step	67.36%
Distillation 4-step	DPM++ 4 \times 2-step	64.71%

Table 9. Human evaluation on text-guided image generation. Here the model is trained on LAION-5B (512 \times 512). We leverage Amazon Mechanical Turk for human evaluation. We perform pairwise comparison between our method and the baselines. We compare our method using 2 or 4 denoising steps with DDIM [38] and DPM++ [18] samplers using 2 \times 2 or 4 \times 2 denoising steps. We use a guidance strength of 12.5 for all methods. For each setting, we distribute 100 HITs each with 17 pairwise comparison questions. We show MTurk workers the text prompt as well as the two generated images, and then ask them to select the one they think is better. We provide a snapshot of the interface in Fig. 29. In the table, we report the percentage that the MTurk workers think our method is better than the baseline. Although, we observe noise in the response (some user would prefer the right image to the left image in Fig. 29c), our method still consistently outperform the baselines in all settings. For the future work, we will incorporate schemes to ignore invalid HITs with random answers. We will also perform another human evaluation study with larger sample sizes and more constraints to ensure high-quality responses.

D. Extra samples for pixel-space distillation

In this section, we provide extra samples for the pixel-space distillation models. We generate samples using the deterministic sampler (see Algorithm 2) and the stochastic sampler (see Algorithm 3).



Figure 33. Ours (deterministic in pixel-space) on CIFAR-10. Distilled 256 sampling steps.



Figure 34. Ours (stochastic in pixel-space) on CIFAR-10. Distilled 256 sampling steps.

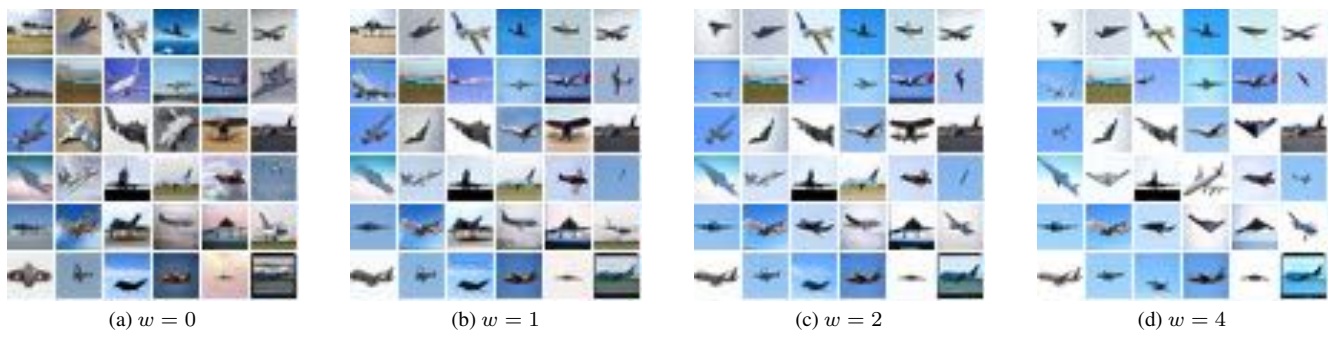


Figure 35. Ours (deterministic in pixel-space) on CIFAR-10. Distilled 256 sampling steps. Class-conditioned samples.

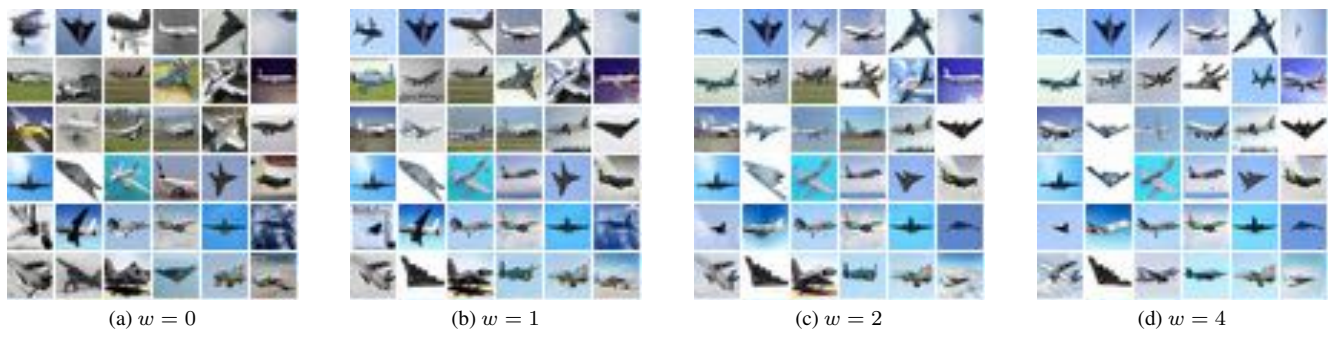


Figure 36. Ours (stochastic in pixel-space) on CIFAR-10. Distilled 256 sampling steps. Class-conditioned samples.



Figure 37. Ours (deterministic in pixel-space) on CIFAR-10. Distilled 4 sampling steps.



Figure 38. Ours (stochastic in pixel-space) on CIFAR-10. Distilled 4 sampling steps.



Figure 39. Ours (deterministic in pixel-space) on CIFAR-10. Distilled 4 sampling steps. Class-conditioned samples.



Figure 40. Ours (stochastic in pixel-space) on CIFAR-10. Distilled 4 sampling steps. Class-conditioned samples.



Figure 41. Ours (deterministic in pixel-space) on CIFAR-10. Distilled 2 sampling steps.



Figure 42. Ours (stochastic in pixel-space) on CIFAR-10. Distilled 2 sampling steps.

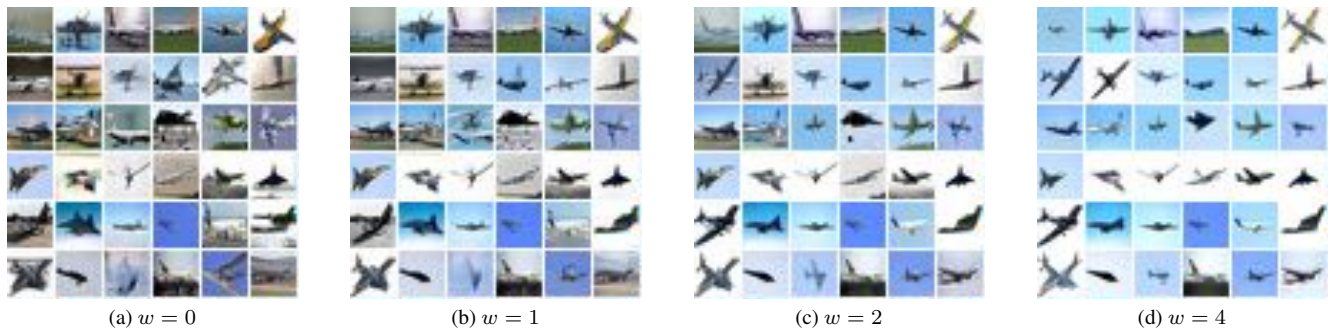


Figure 43. Ours (deterministic in pixel-space) on CIFAR-10. Distilled 2 sampling steps. Class-conditioned samples.

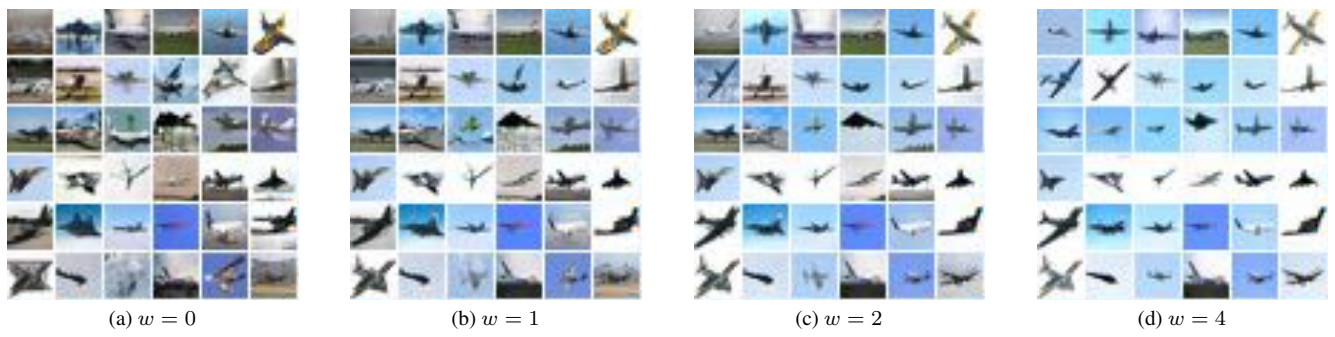


Figure 44. Ours (stochastic in pixel-space) on CIFAR-10. Distilled 2 sampling steps. Class-conditioned samples.



Figure 45. Ours (deterministic in pixel-space) on CIFAR-10. Distilled 1 sampling step.



Figure 46. Ours (stochastic in pixel-space) on CIFAR-10. Distilled 1 sampling step.

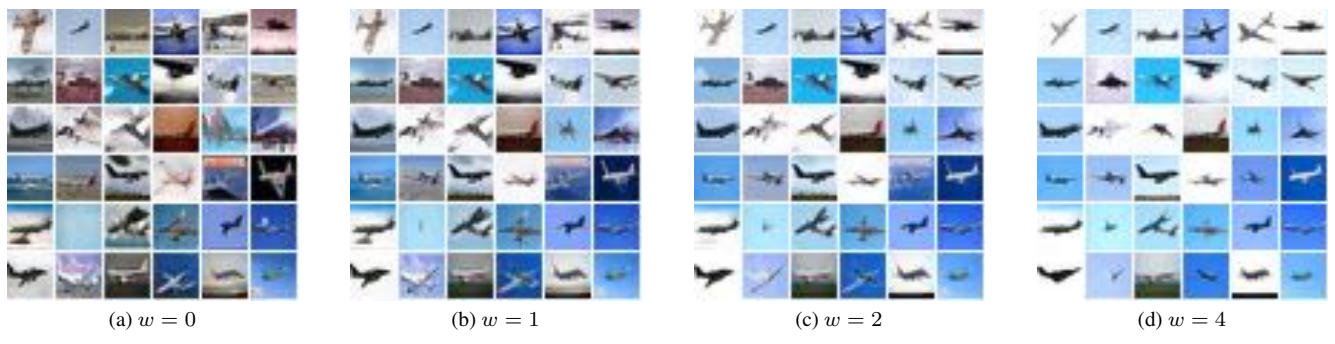


Figure 47. Ours (deterministic in pixel-space) on CIFAR-10. Distilled 1 sampling step. Class-conditioned samples.

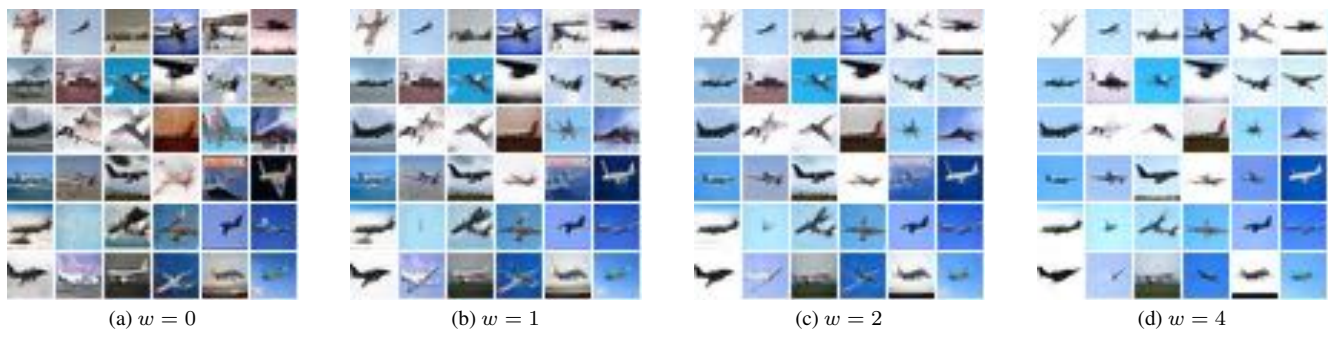


Figure 48. Ours (stochastic in pixel-space) on CIFAR-10. Distilled 1 sampling step. Class-conditioned samples.

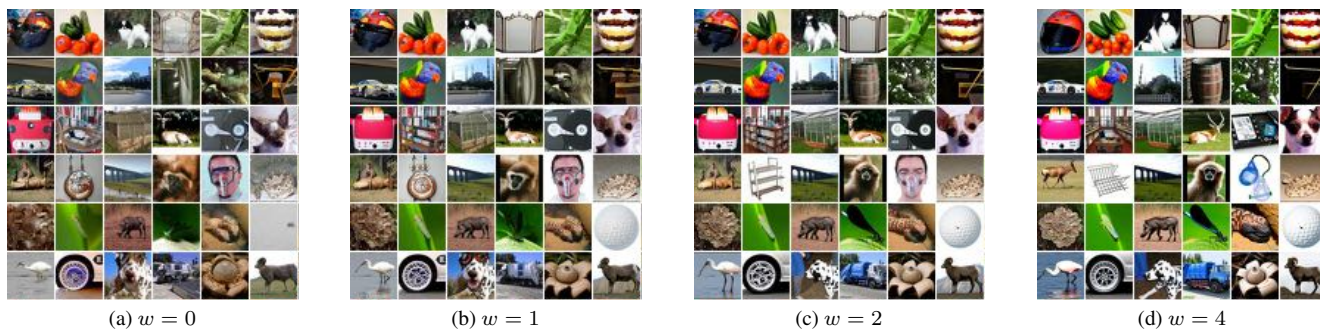


Figure 49. Ours (deterministic in pixel-space) on ImageNet 64x64. Distilled 256 sampling steps.

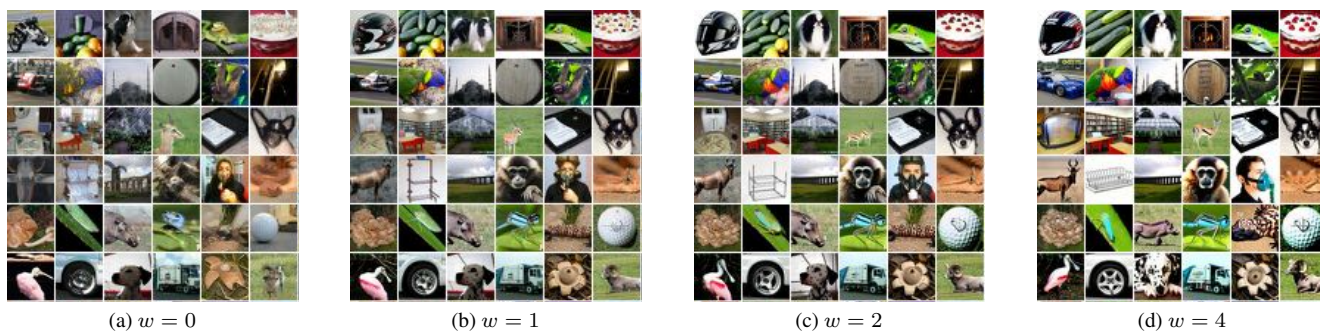


Figure 50. Ours (stochastic in pixel-space) on ImageNet 64x64. Distilled 256 sampling steps.

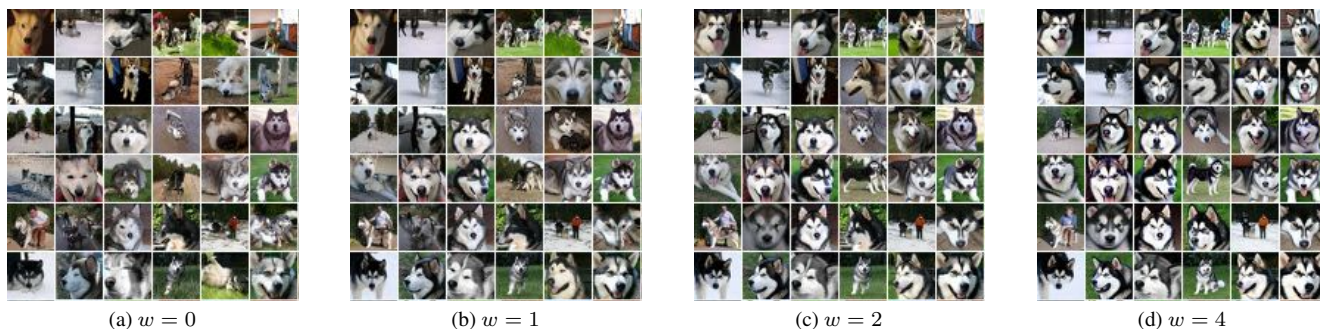


Figure 51. Ours (deterministic in pixel-space) on ImageNet 64x64. Distilled 256 sampling steps. Class-conditioned samples.

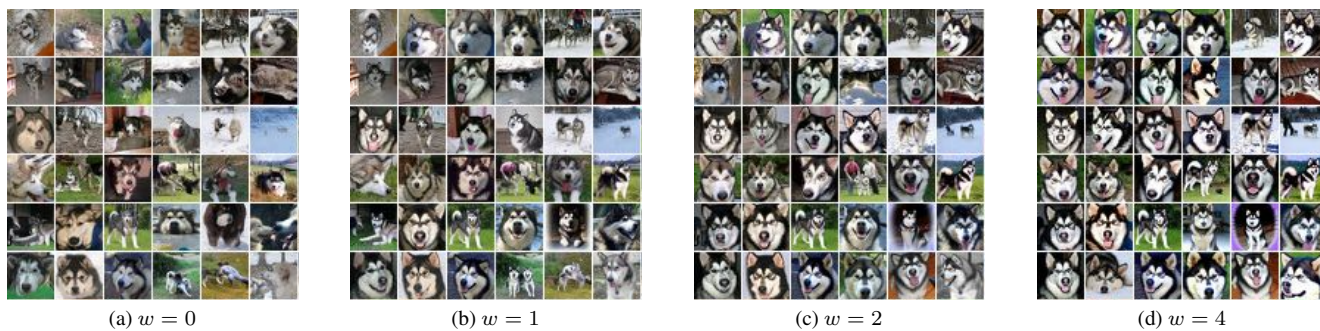


Figure 52. Ours (stochastic in pixel-space) on ImageNet 64x64. Distilled 256 sampling steps. Class-conditioned samples.

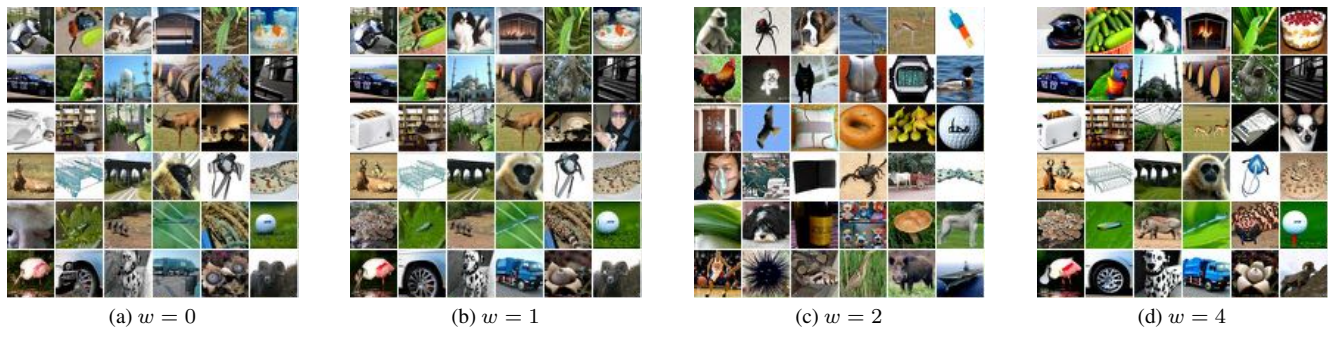


Figure 53. Ours (deterministic in pixel-space) on ImageNet 64x64. Distilled 8 sampling step.

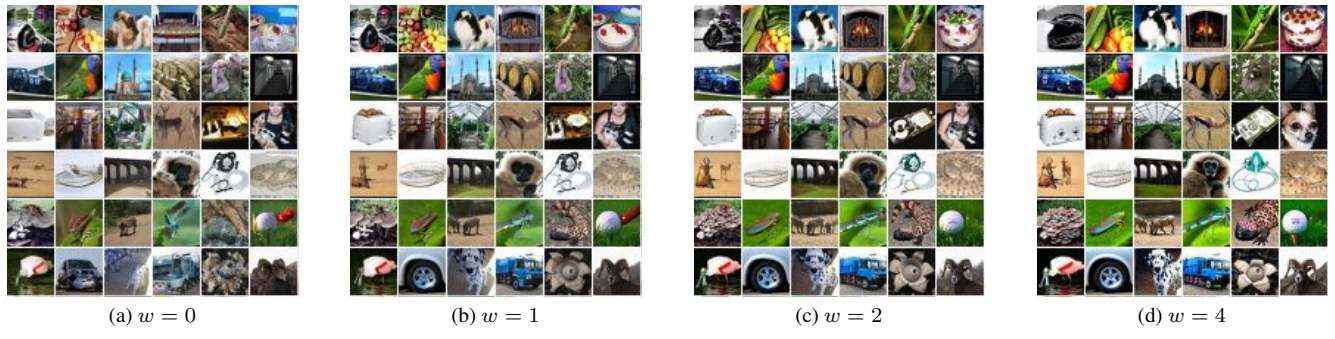


Figure 54. Ours (stochastic in pixel-space) on ImageNet 64x64. Distilled 8 sampling step.

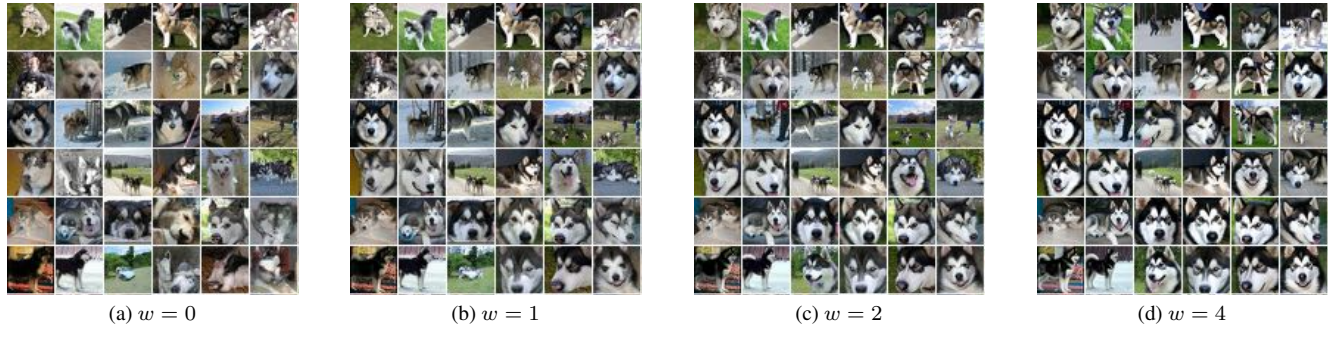


Figure 55. Ours (deterministic in pixel-space) on ImageNet 64x64. Distilled 8 sampling step. Class-conditioned samples.

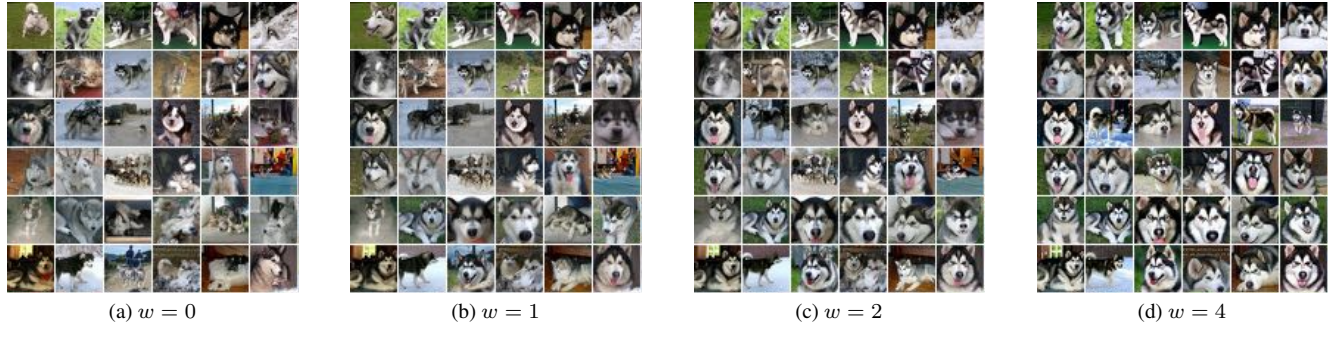


Figure 56. Ours (stochastic in pixel-space). Distilled 8 sampling step. Class-conditioned samples.

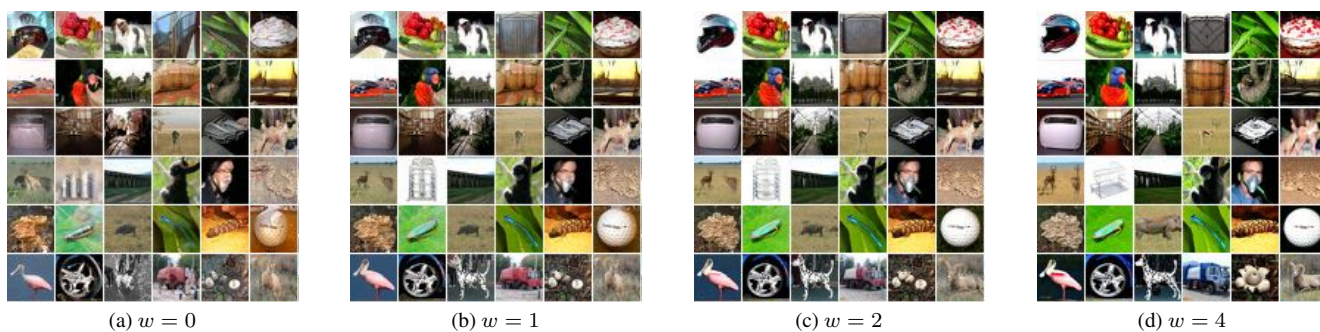


Figure 57. Ours (deterministic in pixel-space) on ImageNet 64x64. Distilled 2 sampling steps.

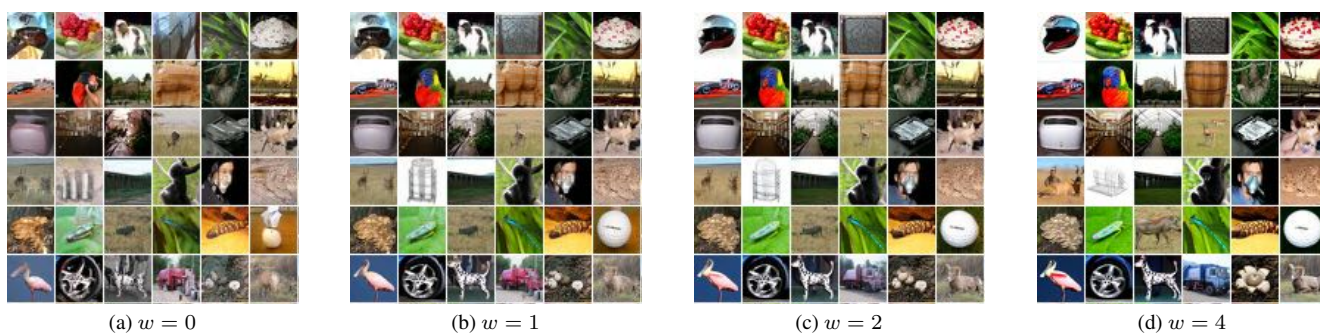


Figure 58. Ours (stochastic in pixel-space) on ImageNet 64x64. Distilled 2 sampling steps.

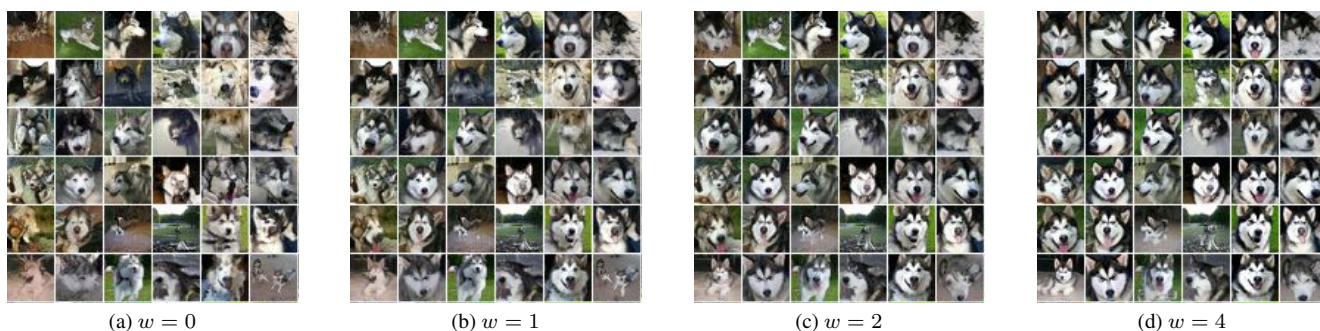


Figure 59. Ours (deterministic in pixel-space) on ImageNet 64x64. Distilled 2 sampling steps. Class-conditioned samples.

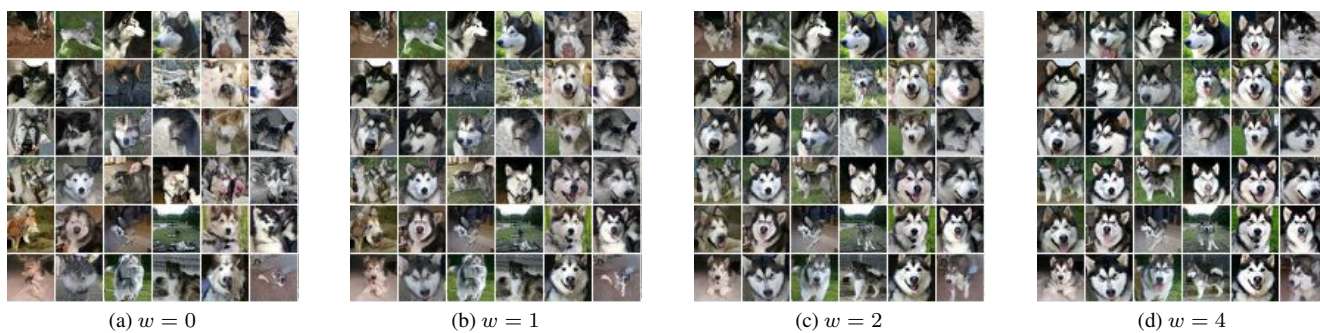


Figure 60. Ours (stochastic in pixel-space) on ImageNet 64x64. Distilled 2 sampling steps. Class-conditioned samples.

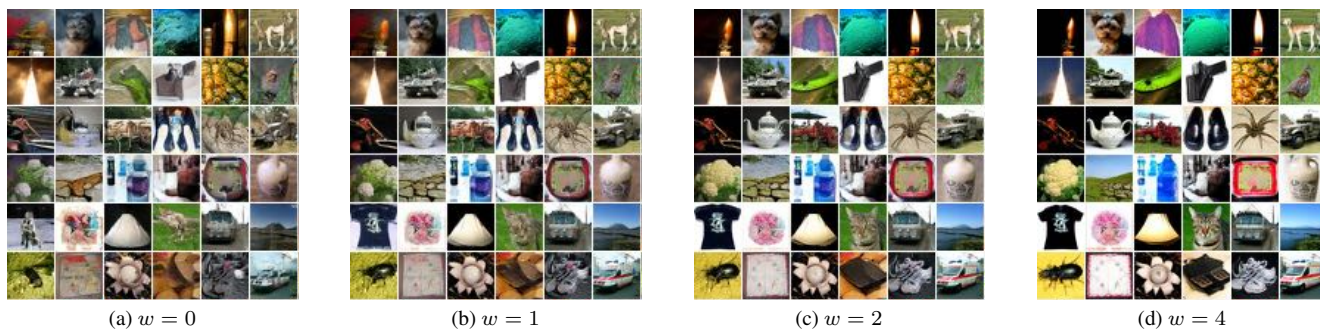


Figure 61. Ours (deterministic in pixel-space) on ImageNet 64x64. Distilled 1 sampling step.

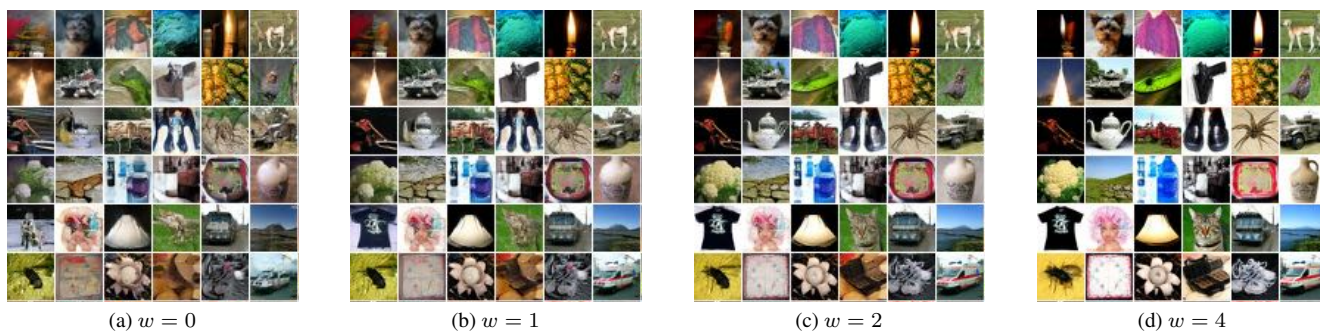


Figure 62. Ours (stochastic in pixel-space) on ImageNet 64x64. Distilled 1 sampling step.

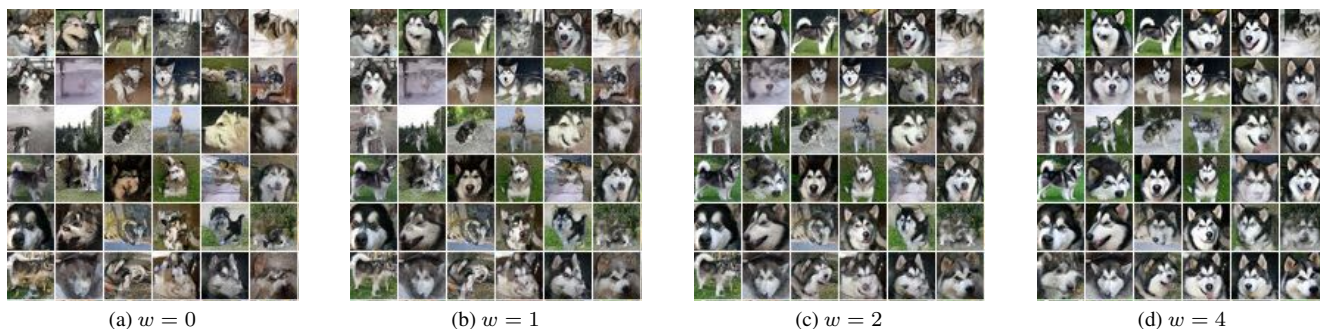


Figure 63. Ours (deterministic in pixel-space) on ImageNet 64x64. Distilled 1 sampling step. Class-conditioned samples.



Figure 64. Ours (stochastic in pixel-space) on ImageNet 64x64. Distilled 1 sampling step. Class-conditioned samples.