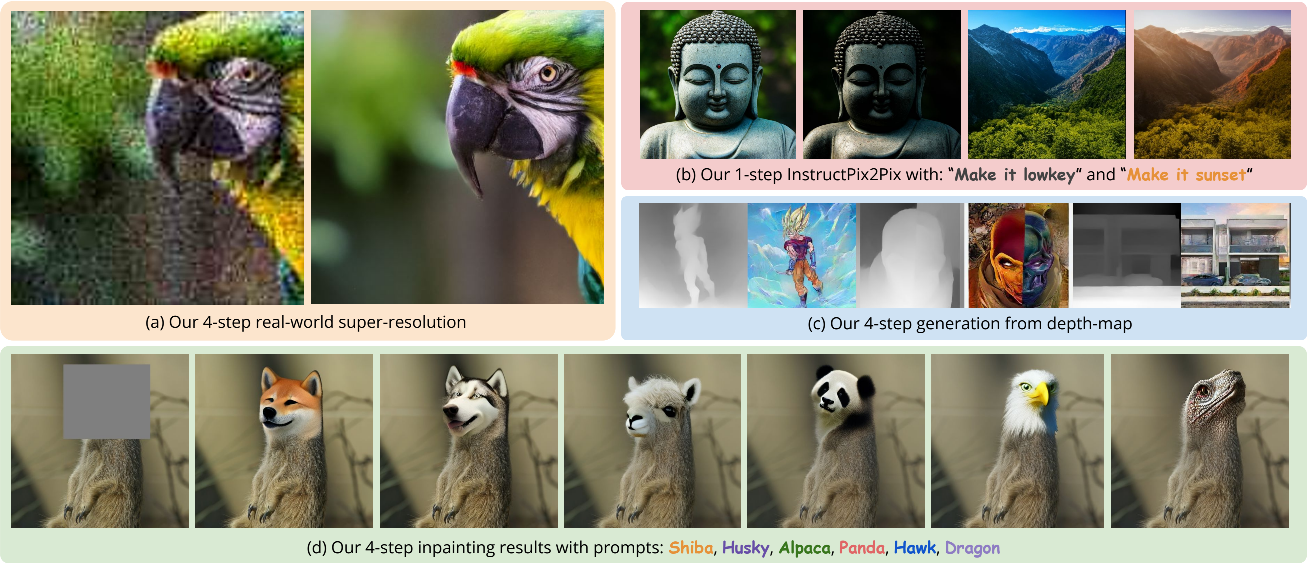
CoDi: Conditional Diffusion Distillation for Higher-Fidelity and Faster Image Generation

Kangfu Mei*[1,2], Mauricio Delbracio[1], Hossein Talebi[1],
Zhengzhong Tu[1], Vishal M. Patel[2], Peyman Milanfar[1]

[1] Google Research, [2] Johns Hopkins University

https://fast-codi.github.io


(a) Our 4-step real-world super-resolution

(b) Our 1-step InstructPix2Pix with: "Make it lowkey" and "Make it sunset"

(c) Our 4-step generation from depth-map

(d) Our 4-step inpainting results with prompts: Shiba, Husky, Alpaca, Panda, Hawk, Dragon

Figure 1. Our proposed CoDi efficiently distills a conditional diffusion model from an unconditional one, enabling rapid generation of high-quality images under various conditional settings. We demonstrate CoDi's capabilities through generated results across various tasks.

Abstract

Large generative diffusion models have revolutionized text-to-image generation and offer immense potential for conditional generation tasks such as image enhancement, restoration, editing, and compositing. However, their widespread adoption is hindered by the high computational cost, which limits their real-time application. To address this challenge, we introduce a novel method dubbed CoDi, that adapts a pre-trained latent diffusion model to accept additional image conditioning inputs while significantly reducing the sampling steps required to achieve high-quality results. Our method can leverage architectures such as ControlNet to incorporate conditioning inputs without compromising the model's prior knowledge gained during large scale pre-training. Additionally, a conditional consistency loss enforces consistent predictions across diffusion steps, effectively compelling the model to generate high-quality images with conditions in a few steps. Our conditional-task learning and distillation approach outperforms previous distillation methods, achieving a new state-of-the-art in producing high-quality images with very few steps (e.g., 1-4) across multiple tasks, including super-resolution, text-guided image editing, and depth-to-image generation.

1. Introduction

Text-to-image diffusion models [27, 29, 34] trained on large-scale data [15, 38] have significantly dominated generative tasks by delivering impressive high-quality and diverse results. A newly emerging trend is to use the prior of pre-trained text-to-image models such latent diffusion models (LDMs) [29] to guide the generated results with external image conditions for image-to-image transformation tasks such as image manipulation, enhancement, or super-resolution [22, 53]. Among these transformation processes, the diffusion prior introduced by pre-trained models is shown to be capable of greatly promoting the visual quality of the conditional image generation results [3, 16, 26, 31].

However, diffusion models heavily rely on an iterative refinement process [4, 33, 35, 43, 49] that often demands a substantial number of iterations, which can be challenging to accomplish efficiently. Their reliance on the number of iterations further increases for high-resolution image synthesis. For instance, in state-of-the-art text-to-image latent diffusion models [29], achieving optimal visual quality typically requires $20-200$ sampling steps (function evaluations), even with advanced sampling methods [10, 17]. The slow sampling time significantly impedes practical applications of the aforementioned conditional diffusion models.

Recent efforts to accelerate diffusion sampling predominantly employ distillation methods [21, 36, 44]. These methods achieve significantly faster sampling, completing the process in just $4-8$ steps, with only a marginal decrease in generative performance. Very recent works [14, 23] show that these strategies are even applicable for distilling pre-trained large-scale text-to-image diffusion models.

A very common application scenario is to incorporate new conditions into these distilled diffusion models, such as using low-resolution images for super-resoltion [35], or instruction-tuning for image editing [3], where the most straightforward way is to directly finetune the distilled text-to-image pre-trained model with new conditional data. An alternative common approach [23] is to first finetune the diffusion model with the new conditional data, then conducting distillation on the already-finetuned conditional model. While these two methods have been demonstrated to accelerate sampling, each has distinct disadvantages in terms of result quality and cross-task flexibility, as discussed below.

In this paper, we introduce a new algorithm for **Co**nditional **Di**stillation which we call **CoDi** for efficiently adding new controls into distilled models. Unlike previous distillation methods that rely on finetuning, our method directly distills a diffusion model from a text-to-image pre-training (*e.g*., StableDiffusion) and ends with a fully distilled conditional diffusion model. As depicted in Figure 1, our distilled model is capable of predicting high-quality results in just $1-4$ sampling steps.

By design, our method eliminates the need for the original text-to-image data [37, 38], a requirement in previous distillation methods (*i.e*., those that first distill the unconditional text-to-image model), thereby making our method more practical. Additionally, our formulation avoids sacrificing the diffusion prior in the pre-trained model during finetuning, a common drawback in the first stage of the finetuning-first procedure. Our extensive experiments show that our CoDi outperforms previous distillation methods in both visual quality and quantitative metrics, particularly when operating under the same sampling time.

Parameter-efficient distillation methods are a relatively understudied area. We demonstrate that our method also enables a new **P**arameter-**E**fficient distillation paradigm (**PE-CoDi**). It can transform an unconditional diffusion model to conditional tasks by incorporating a small number of additional learnable parameters. Specifically, our formulation allows for integration with various existing parameter-efficient tuning algorithms, *e.g*., ControlNet [53]. We show that our distillation process that integrates the ControlNet adapter can efficiently preserve the generative prior in pre-training while adapting the model to new conditioned data. This new paradigm significantly improves the practicality of different conditional tasks.

Our contributions are summarized as follows:

- We propose a new method for image and image-text conditioned generation. It can derive a conditional diffusion model from pretrained text-to-image LDMs for generating high-quality results in only a few sampling steps.
- The proposed method's efficiency and effectiveness arise from a non-trivial consistency between the model's predictions at different time steps. Enforcing this consistency through learning enables the simultaneous reduction of required sampling steps and the integration of new conditions into the model.
- We introduce the first parameter-efficient distillation mechanism that can produce compelling results in just a few steps, while requiring only a small number of additional parameters compared with the pretrained LDMs.

2. Related Work

Diffusion Distillation. To reduce the sampling time of diffusion models, Luhman et al. [21] proposed to learn a single-step student model from the output of the original (teacher) model using multiple sampling steps. However, this method requires to run the full inference with many sampling steps during training which make it poorly scalable. Inspired by this, Progressive Distillation [36] and its variants, including Guided Distillation [23] and SnapFusion [14], use a progressive learning scheme for improving the learning efficiency. A student model learns to predict the output of two steps of the teacher model in one step. Then, the teacher model is replaced by the student model, and the procedure is repeated to progressively distill the mode by halving the number of required steps. We demonstrate our method by comparing these methods on the conditional generation tasks. We note that strategies like classifier-free guidance distillation [14, 23], or the different adopted sampling techniques [51, 54], are orthogonal to our method, and they could be incorporated in our formulation. Even though some concurrent works [50, 52] find that tasks like super-resolution requires less sampling steps, we later show that distilling pre-trained diffusion models can still improve the performance in such restoration tasks.

Consistency Distillation. A Consistency Model is a single-step generative approach distilled from a pre-trained diffusion model [44]. The learning is achieved by enforcing a self-consistency in the predicted signal space. Based on this idea, following work [7, 11, 19, 41] have focus on improving the training techniques. However, learning consistency models for conditional generation has yet to be thoroughly studied. In this paper, we compare our method against a baseline approach that enforces self-consistency in an already fine-tuned conditional diffusion model. Our results demonstrate that our conditional distilled model outperforms the baseline approach, indicating the effectiveness of our proposed distillation strategy.

Diffusion Models Adaptations. Leveraging the knowledge of pre-trained models for new tasks, known as model adaptation, has gained significant traction in NLP and computer vision domains. This approach utilizes model adapters [9, 28, 30, 45] and HyperNetworks [1, 6] to effectively adapt pre-trained models to new domains and tasks. In the context of diffusion models, model adapters have been successfully employed to incorporate new conditions into pre-trained models [24, 53]. Our proposed method draws inspiration from these approaches and introduces a novel application of model adapters: distilling the sampling steps of diffusion models. Compared to fine-tuning the entire model [36], our method offers enhanced efficiency and flexibility. It enables the adaptation of multiple tasks using the same backbone model.

3. Background

Continuous-time VP diffusion model. A continuous-time variance-preserving (VP) diffusion model [8, 39] is a special case of diffusion models[1]. It has latent variables $\{\mathbf{z}_t | t \in [0, T]\}$ specified by a noise schedule comprising differentiable functions $\{\alpha_t, \sigma_t\}$ with $\sigma_t^2 = 1 - \alpha_t^2$. The clean data $\mathbf{x} \sim p_{\text{data}}$ is progressively perturbed in a (forward) Gaussian process as in the following Markovian structure:

$$q(\mathbf{z}_t | \mathbf{x}) = \mathcal{N}(\mathbf{z}_t; \alpha_t \mathbf{x}, \sigma_t^2 \mathbf{I}), \tag{1}$$

$$q(\mathbf{z}_t | \mathbf{z}_s) = \mathcal{N}(\mathbf{z}_t; \alpha_{t|s} \mathbf{z}_s, \sigma_{t|s}^2 \mathbf{I}), \tag{2}$$

where $0 \leq s < t \leq 1$ and $\alpha_{t|s}^2 = \alpha_t / \alpha_s$. Here the latent $\mathbf{z}_t$ is sampled from the combination of the clean data and random noise by using the reparameterization trick [13], which has $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$.

Deterministic sampling. The aforementioned diffusion process that starts from $\mathbf{z}_0 \sim p_{\text{data}}(\mathbf{x})$ and ends at $\mathbf{z}_T \sim \mathcal{N}(0, \mathbf{I})$ can be modeled as the solution of an stochastic differential equation (SDE) [43]. The SDE is formed by a vector-value function $f(\cdot, \cdot) : \mathbb{R}^d \to \mathbb{R}^d$, a scalar function $g(\cdot) : \mathbb{R} \to \mathbb{R}$, and the standard Wiener process $\mathbf{w}$ as:

$$\mathrm{d}\mathbf{z}_t = f(\mathbf{z}_t, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}. \tag{3}$$

The overall idea is that the reverse-time SDE that runs backwards in time, can generate samples of p_{data} from the prior distribution $\mathcal{N}(0, \mathbf{I})$. This reverse SDE is given by

$$\mathrm{d}\mathbf{z}_t = [f(\mathbf{z}_t, t) - g(t)^2 \nabla_{\mathbf{z}} \log p_t(\mathbf{z}_t)]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}, \tag{4}$$

where the $\bar{\mathbf{w}}$ is a also standard Wiener process in reversed time, and $\nabla_{\mathbf{z}} \log p_t(\mathbf{z}_t)$ is the score of the marginal distribution at time t. The score function can be estimated by training a score-based model $s_\theta(\mathbf{z}_t, t) \approx \nabla_z \log p_t(\mathbf{z}_t)$ with score-matching [42] or a denoising network $\hat{\mathbf{x}}_\theta(\mathbf{z}_t, t)$ [8]:

$$s_\theta(\mathbf{z}_t, t) := (\alpha_t \hat{\mathbf{x}}_\theta(\mathbf{z}_t, t) - \mathbf{z}_t) / \sigma_t^2. \tag{5}$$

Such backward SDE satisfies a special ordinary differential equation (ODE) that allows deterministic sampling given $\mathbf{z}_T \sim \mathcal{N}(0, \mathbf{I})$. This is known as the *probability flow* (PF) ODE [43] and is given by

$$\mathrm{d}\mathbf{z}_t = [f(\mathbf{z}_t, t) - \frac{1}{2} g^2(t) s_\theta(\mathbf{z}_t, t)]\mathrm{d}t, \tag{6}$$

where $f(\mathbf{z}_t, t) = \frac{\mathrm{d} \log \alpha_t}{\mathrm{d}t} \mathbf{z}_t$, $g^2(t) = \frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t} - 2 \frac{\mathrm{d} \log \alpha_t}{\mathrm{d}t} \sigma_t^2$ with respect to $\{\alpha_t, \sigma_t\}$ and t according to [12]. This ODE can be solved numerically with diffusion samplers like DDIM [40], where starting from $\hat{\mathbf{z}}_T \sim \mathcal{N}(0, \mathbf{I})$, we update for $s = t - \Delta t$:

$$\hat{\mathbf{z}}_s := \alpha_s \hat{\mathbf{x}}_\theta(\hat{\mathbf{z}}_t, t) + \sigma_s (\hat{\mathbf{z}}_t - \alpha_t \hat{\mathbf{x}}_\theta(\hat{\mathbf{z}}_t, t)) / \sigma_t, \tag{7}$$

till we reach $\hat{\mathbf{z}}_0$.

Diffusion models parametrizations. Leaving aside the aforementioned way of parametrizing diffusion models with a denoising network (signal prediction) or a score model (noise prediction equation 5), in this work, we adopt a parameterization that mixes both the score (or noise) and the signal prediction. Existing methods include either predicting the noise $\hat{\epsilon}_\theta(\mathbf{x}_t, t)$ and the signal $\hat{\mathbf{x}}_\theta(\mathbf{z}_t, t)$ separately using a single network [5], or predicting a combination of noise and signal by expressing them in a new term, like the velocity model $\hat{\mathbf{v}}_\theta(\mathbf{z}_t, t) \approx \alpha_t \epsilon - \sigma_t \mathbf{x}$ [36]. Note that one can derive an estimation of the signal and the noise from the velocity one,

$$\hat{\mathbf{x}} = \alpha_t \mathbf{z}_t - \sigma_t \hat{\mathbf{v}}_\theta(\mathbf{z}_t, t), \text{ and } \hat{\epsilon} = \alpha_t \hat{\mathbf{v}}_\theta(\mathbf{z}_t, t) + \sigma_t \mathbf{z}_t. \tag{8}$$

Similarly, DDIM update rule (equation 7) can be rewritten in terms of the velocity parametrization:

$$\hat{\mathbf{z}}_s := \alpha_s (\alpha_t \hat{\mathbf{z}}_t - \sigma_t \hat{\mathbf{v}}_\theta(\hat{\mathbf{z}}_t, t)) + \sigma_s (\alpha_t \hat{\mathbf{v}}_\theta(\hat{\mathbf{z}}_t, t) + \sigma_t \hat{\mathbf{z}}_t). \tag{9}$$

Self-consistency property. To accelerate inference, [44] introduced the idea of consistency models. Let $s_\theta(\cdot, t)$

[1] What we discussed based on the variance preserving (VP) form of SDE [43] is equivalent to most general diffusion models like Denoising Diffusion Probabilistic Models (DDPM) [8].

be a pre-trained diffusion model trained on data $\mathbf{x} \sim \mathcal{O}_{data}$. Then, a consistency function $f_\phi(\mathbf{z}_t, t)$ should satisfy that [44] where $f_\phi(\mathbf{x}, 0) = \mathbf{x}$ and

$$f_\phi(\mathbf{z}_t, t) = f_\phi(\mathbf{z}_{t'}, t'),\ \forall t, t' \in [0, T], \tag{10}$$

where $\{\mathbf{z}_t\}_{t\in[0,T]}$ is the solution trajectory of the probability flow ODE (PF-ODE) (equation 6). A boundary condition, *i.e.*, $f_\phi(\mathbf{x}, 0) = \mathbf{x}$ is parameterized with skip connections for ensuring continuous properties similar as done in previous works [2, 10, 44]:

$$F_\phi(\mathbf{z}_t, t) = c_{\text{skip}}(t)\mathbf{x} + c_{\text{out}}(t) f_\phi(\mathbf{z}_t, t), \tag{11}$$

where $c_{\text{skip}}(0) = 1$, $c_{\text{out}}(0) = 0$. In practice, $f_\phi(\mathbf{z}_t, t)$ is usually a denoising network that is distilled from a pre-trained diffusion model. We later show that we can replace the frozen PF-ODE with the distillation network and thus fit the PF-ODE for new conditional data during distillation.

4. Method

4.1. From Unconditional to Conditional

In order to utilize the image generation prior encapsulated by the pre-trained unconditional[2] diffusion model, we first propose to adapt the unconditional diffusion model into a conditional version for the conditional data $(\mathbf{x}, c) \sim p_{\text{data}}$. Similar to the zero initialization technique used by controllable generation [25, 53], our method adapts the unconditional pre-trained architecture by using an additional conditional encoder.

To elaborate, we take the widely used U-Net as the diffusion network. Let us introduce the conditional-module by duplicating the encoder layers of the pretrained network. Then, let $\boldsymbol{h}_\theta(\cdot)$ be the encoder features of the pretrained network, and $\boldsymbol{h}_\eta(\cdot)$ be the features on the additional conditional encoder. We define the new encoder features of the adapted model by

$$\boldsymbol{h}_\theta(\mathbf{z}_t)' = (1 - \mu)\boldsymbol{h}_\theta(\mathbf{z}_t) + \mu \boldsymbol{h}_\eta(c), \tag{12}$$

where μ is a learnable scalar parameter, initialized to $\mu = 0$. Starting from this zero initialization, we can adapt the unconditional architecture into a conditional one. Thus, our conditional diffusion model $\hat{\mathbf{w}}_\theta(\mathbf{z}_t, c, t)$ is the result of adapting the pre-trained unconditional diffusion model $\hat{\mathbf{v}}_\theta(\mathbf{z}_t, t)$ with the conditional features $\boldsymbol{h}_\eta(c)$.

4.2. A New Conditional Diffusion Consistency

Our core idea is to optimize the adapted conditional diffusion model $\hat{\mathbf{w}}_\theta(\mathbf{z}_t, c, t)$ from $\hat{\mathbf{v}}_\theta(\mathbf{z}_t, t)$, so it satisfies a conditional diffusion consistency property:

$$\hat{\mathbf{w}}_\theta(\mathbf{z}_t, c, t) = \hat{\mathbf{w}}_\theta(\hat{\mathbf{z}}_s, c, s),\ \forall t, s \in [0, T], \tag{13}$$

where the $\hat{\mathbf{z}}_s$ belong to the probability flow ODE (equation 6) of the adapted model. Note that this consistency property differs from the one in consistency models [44] in the probability flow ODE model used for sampling $\hat{\mathbf{z}}_s$ and the consistency loss space. To motivate this formulation, let us introduce the following general remark.

Remark 1. *If a diffusion model, parameterized by $\hat{\mathbf{v}}_\theta(\mathbf{z}_t, t)$, satisfies the self-consistency property (equation 10) on the noise prediction $\hat{\epsilon}_\theta(\mathbf{z}_t, t) = \alpha_t \hat{\mathbf{v}}_\theta(\mathbf{z}_t, t) + \sigma_t \mathbf{z}_t$, then it also satisfies the self-consistency property on the signal prediction $\hat{\mathbf{x}}_\theta(\mathbf{z}_t, t) = \alpha_t \mathbf{z}_t - \sigma_t \hat{\mathbf{v}}_\theta(\mathbf{z}_t, t)$.*

The proof is a direct consequence of change of variables from noise into signal and is given in Appendix. Based on this general remark, we claim that we can optimize the conditional diffusion model $\hat{\mathbf{w}}_\theta(\mathbf{z}_t, c, t)$ to jointly learn to enforce the self-consistency property on the noise prediction $\hat{\epsilon}_\theta(\mathbf{z}_t, c, t)$ and the new conditional generation $(\mathbf{x}, c) \sim p_{\text{data}}$ with the signal prediction $\hat{\mathbf{x}}_\theta(\mathbf{z}_t, c, t)$. We then impose the boundary condition for consistency distillation by parameterizing the noise prediction $\hat{\epsilon}_\theta(\mathbf{z}_t, c, t)$ with the same skip connections of equation 17.

Prediction of $\hat{\mathbf{z}}_s$. In the distillation process given by equation 15, the latent variable $\hat{\mathbf{z}}_s$ is achieved by running one step of a numerical ODE solver. Consistency models [44] solve the ODE using the Euler solver, while progressive distillation [36] and guided distillation [23] run two steps using the DDIM sampler (equation 7).

We propose an alternative prediction for $\hat{\mathbf{z}}_s$ that leverages the adapted diffusion model, $\hat{\mathbf{x}}_\theta(\mathbf{z}_t, c, t)$, as opposed to the conventional frozen pretraining one. We then sample $\hat{\mathbf{z}}_s$ in the adapted diffusion model PF-ODE by

$$\hat{\mathbf{z}}_s = \alpha_s \hat{\mathbf{x}}_\theta(\mathbf{z}_t, c, t) + \sigma_s \epsilon, \text{ with } \mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon, \tag{14}$$

and $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. This novel formulation effectively harmonizes the conflicting optimization directions between consistency distillation from pretrained data and conditional guidance from conditional data.

Training scheme. Inspired by consistency models [44], we use the exponential moving averaged parameters θ^- as the target network for stabilize training. Then, we can minimize the following training loss for conditional distillation:

$$\mathcal{L}(\theta) := \mathbb{E}[d_\epsilon(\hat{\epsilon}_{\theta^-}(\hat{\mathbf{z}}_s, s, c), \hat{\epsilon}_\theta(\mathbf{z}_t, t, c))) + d_{\mathbf{x}}(\mathbf{x}, \hat{\mathbf{x}}_\theta(\mathbf{z}_t, t, c))]. \tag{15}$$

where $d_\epsilon(\cdot, \cdot)$ and $d_{\mathbf{x}}(\cdot, \cdot)$ are two distance functions to measure difference in the noise space and in the signal space respectively. Note that the total loss is a balance between the conditional guidance given by $d_{\mathbf{x}}$, and the noise self-consistency property given by d_ϵ.

The overall conditional distillation algorithm is presented in Appendix. In the following, we will detail how

[2] The discussed unconditional models include text-conditioned image generation models, *e.g.*, StableDiffusion [29] and Imagen [34], which are only conditioned on text prompts.

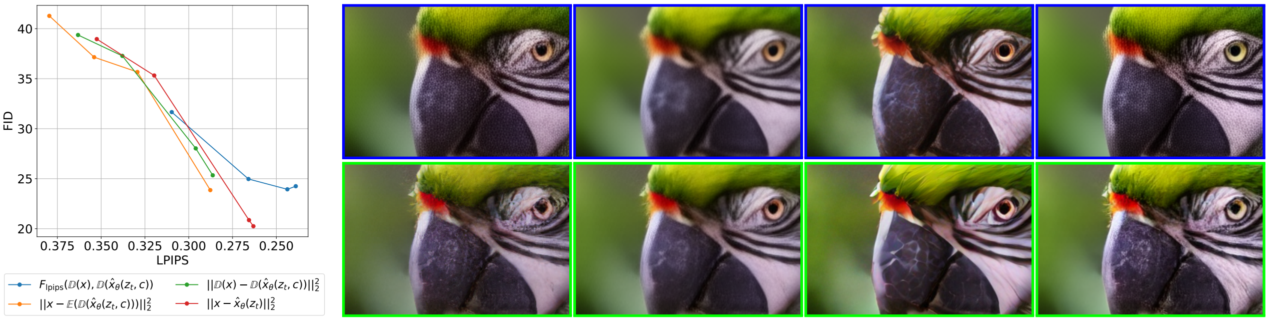


Figure 2. Sampled results between distilled models learned with alternative conditional guidance. Left curves shows the quantitative performance between the LPIPS and FID in $\{1, 2, 4, 8\}$ steps. Right part show the visual results where each result comes from the 1 sampling step (top) or 4 sampling steps (bottom). The distance function from the left to right is $\|\mathbf{x} - \mathbb{E}(\mathbb{D}(\hat{\mathbf{x}}_\theta(\mathbf{z}_t, c)))\|_2^2$, $\|\mathbb{D}(\mathbf{x}) - \mathbb{D}(\hat{\mathbf{x}}_\theta(\mathbf{z}_t, c))\|_2^2$, $\mathcal{F}_{\text{lpips}}(\mathbb{D}(\mathbf{x}), \mathbb{D}(\hat{\mathbf{x}}_\theta(\mathbf{z}_t, c))$, and our default $\|\mathbf{x} - \hat{\mathbf{x}}_\theta(\mathbf{z}_t)\|_2^2$, respectively.

we sample $\hat{\mathbf{z}}_s$ and discuss other relevant hyperparameters in our method (e.g., $d_\mathbf{x}$).

4.3. Effects of Different Conditional Guidance

To finetune the adapted diffusion model with the new conditional data, our conditional diffusion distillation loss in equation 15 penalizes the difference between the predicted signal $\hat{\mathbf{x}}_\theta(\mathbf{z}_t, c, t)$ and the corresponding image $\mathbf{x}$ with a distance function $d_\mathbf{x}(\cdot, \cdot)$ for distillation learning.

Here we investigate the impact of the distance function $d_\mathbf{x}(\cdot, \cdot)$ in the conditional guidance. According to both qualitative and quantitative results, shown in Figure 2, different distance functions lead to different behaviours when doing multi-step sampling (inference). If $d_\mathbf{x} = \| \cdot \|^2$ in the pixel space or the encoded space, *i.e.*, $\|\mathbf{x} - \mathbb{E}(\mathbb{D}(\hat{\mathbf{x}}_\theta(\mathbf{z}_t, c, t)))\|_2^2$ and $\|\mathbb{D}(\mathbf{x}) - \mathbb{D}(\hat{\mathbf{x}}_\theta(\mathbf{z}_t, c, t))\|_2^2$, multi-step sampling leads to more smooth and blurry results. If instead we adopt a perceptual distance in the pixel space, *i.e.*, $\mathcal{F}_{\text{lpips}}(\mathbb{D}(\mathbf{x}), \mathbb{D}(\hat{\mathbf{x}}_\theta(\mathbf{z}_t, c, t)))$, the iterative refinement in the multi-step sampling leads to over-saturated results. Overall, by default we adopted the ℓ_2 distance in the latent space since it leads to better visual quality and achieve the optimal FID with 4 sampling steps in Figure 2.

4.4. Parameter-Efficient Conditional Distillation

Our method offers the flexibility to selectively update parameters pertinent to distillation and conditional finetuning, leaving the remaining parameters frozen. This leads us to introduce a new fashion of parameter-efficient conditional distillation, aiming at unifying the distillation process across commonly-used parameter-efficient diffusion model finetuning, including ControlNet [53], T2I-Adapter [24], etc. We highlight the ControlNet architecture illustrated in Figure 3 as an example. This model duplicates the encoder part of the denoising network, highlighted in the green blocks, as the condition-related parameters. Our method can then optimizes the conditional guidance and the consistency by only updating the duplicated encoder.

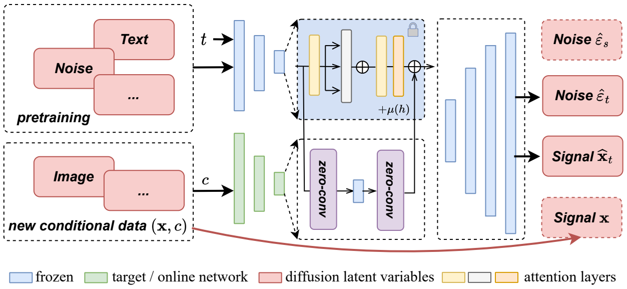

Text
Noise
...
pretraining
Image
...
new conditional data $(\mathbf{x}, c)$
t
c
$+\mu(h)$
zero-conv
zero-conv
Noise $\hat{\varepsilon}_s$
Noise $\hat{\varepsilon}_t$
Signal $\hat{\mathbf{x}}_t$
Signal $\mathbf{x}$
frozen
target / online network
diffusion latent variables
attention layers

Figure 3. Network architecture illustration of our parameter-efficient conditional distillation framework.

	CM-I	CM-II	GD-I	GD-II	Ours
stage-1	distill	finetune	distill	finetune	conditional distill
stage-2	finetune	distill	finetune	distill	n.a.
	✗	✓	✓	✓	✓

Table 1. We compare previous distillation methods by applying them to a T2I LDMs and then finetuning the distilled models (CM-X), and also distillation methods by directly applying them into the finetuned LDMs (GD-X). Since fine-tuning a distilled consistency model within the existing diffusion loss framework is not feasible, we excluded it from our comparison.

5. Experiments

We demonstrate the efficacy of our method on representative conditional generation tasks, including, real-world super-resolution [48], depth-to-image generation [53], and instructed image editing [3]. We utilize a pre-trained text-to-image latent diffusion models[3] and conduct conditional distillation directly from the model. Each of the compared methods, including the text-to-image pretraining, was independently trained for 8 days on 64 TPU-v4 pods.

[3] We base our work on a version of Latent Diffusion Model trained on internal text-to-image data. It is comparable with StableDiffusion v1.4.

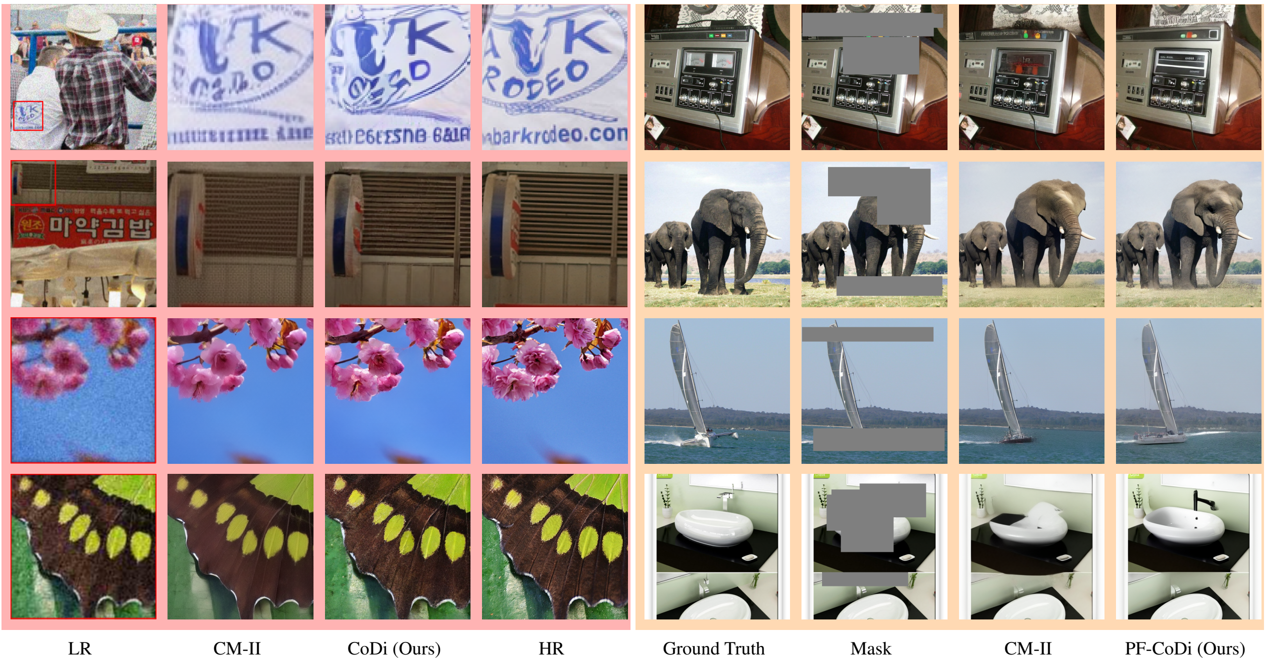


Figure 4. We show the results sampled in 4 steps by different models. Samples generated according to the low-resolution images (left) and masks (right) respectively. Please see our supplement for many more examples such as visual comparisons with the other methods.

Super-resolution (DF2K)

Sampling Steps	Methods	FID ↓	LPIPS ↓
1 step	RealESRGAN [48]	37.640	0.3112
200 steps	StableSR [46]	24.440	0.3114
4 steps	DiffIR [50]	31.719	0.3088
4 steps	ControlNet [53]	34.56	0.3381
250 steps	LDMs [29]	19.200	0.2639
50 steps	LDMs [29]	19.231	0.2603
20 steps	LDMs [29]	20.510	0.2627
8 steps	LDMs [29]	24.493	0.2789
6 steps	LDMs [29]	26.338	0.2873
4 steps	LDMs [29]	29.266	0.3014
4 steps	+ DPM Solve [17]	28.936	0.3077
4 steps	+ DPM Solver++ [18]	28.937	0.3073
	GD-I [23]	27.806	0.3202
	GD-II [23]	23.675	0.2796
	CM-II (frozen) [44]	28.088	0.3192
	CM-II [44]	27.810	0.3172
4 steps	**PE-CoDi** (Ours)	25.214	0.2941
	CoDi (Ours)	19.637	0.2656

Inpainting (ImageNet)

Sampling Steps	Methods	FID	LPIPS
1000 steps	Palette [33]	13.151	-
250 steps	Repaint [20]	-	0.2827
50 steps	ControlNet [53]	14.895	0.2260
4 steps	ControlNet [29]	20.205	0.2635
	+ DPM Solver++ [18]	19.941	0.2644
	CM-II [44]	17.710	0.2580
	GD-II [23]	15.95	0.2452
4 steps	**PE-CoDi** (Ours)	14.700	0.2231

Text-guided Depth-to-image (WebLI)

Sampling Steps	Methods	FID	CLIP
250 steps	ControlNet [53]	20.884	0.2910
4 steps	ControlNet [53]	29.780	0.2854
	+ DPM Solver++ [18]	32.208	0.2834
	CM-II [44]	27.640	0.2869
	GD-II [23]	26.51	0.2870
4 steps	**PE-CoDi** (Ours)	23.047	0.2874

Table 2. Quantitative performance comparisons between the baselines and our methods. Our model can achieve comparable performance in 4 steps than models sampled in 250 steps. The 4-step sampling results of our parameters-efficient distillation (PE-CoDi) is comparable with the original 8-step sampling results, while PE-CoDi doesn't sacrifice the original generative performance with frozen backbone.

5.1. Results

Baselines. We compare our method with two previous SOTA diffusion distillation methods, *i.e.*, consistency models (CM) [44] and guided-distillation (GD) [23]. We implement CM with ControlNet without freezing denoising U-Net, which leads to the same network architecture and the same number of parameters as ours. For completeness, we consider two different ways of applying the tested distillation techniques, by first making the model conditional (fine-tuning first), or by first distilling the model and then making it conditional (distill first). A summary of the tested

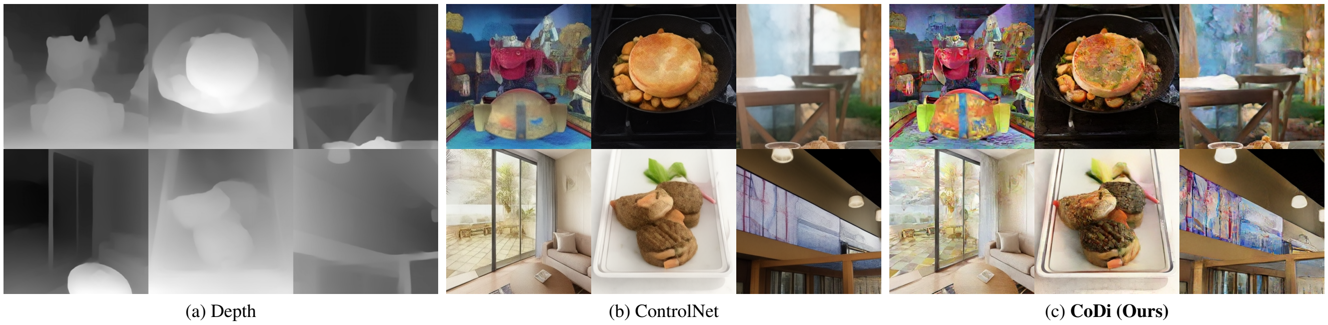


Figure 5. Samples generated according to the depth image (left) from ControlNet sampled in 4 steps (middle), and ours from the unconditional pretraining sampled in 4 steps (right). Please see our supplement for many more examples.

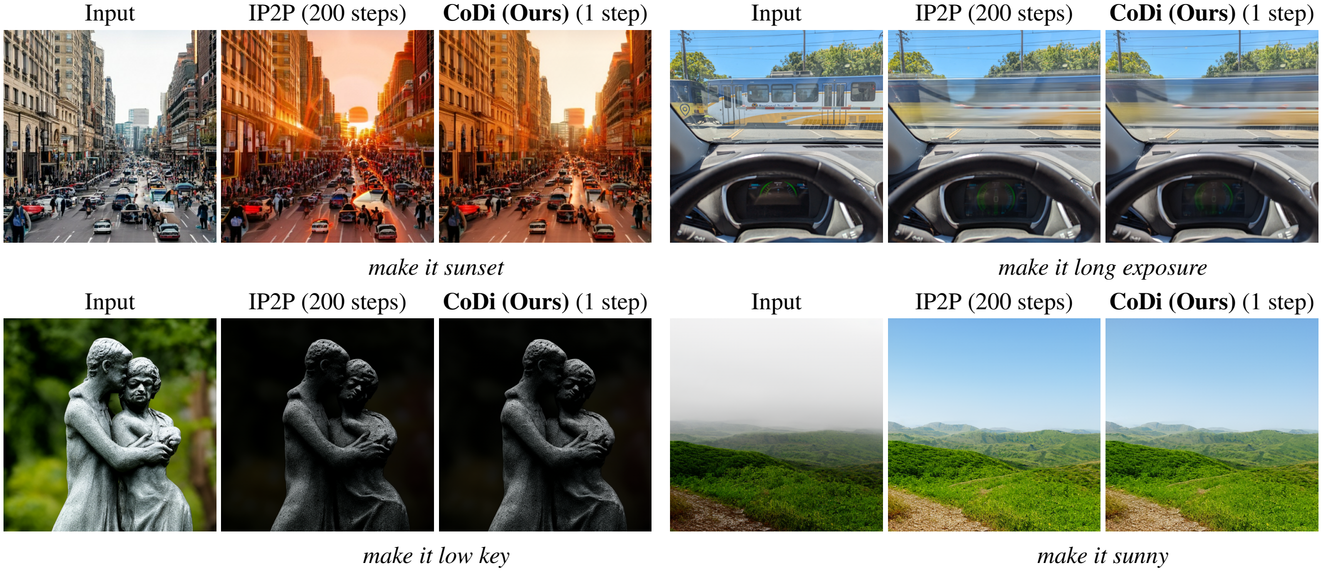


Figure 6. Generated edited image according to the input image and the instruction (bottom) from Instructed Pix2Pix (IP2P) sampled in 200 steps and ours sampled in 1 step. Please see our supplement for many more examples.

configurations is shown in Table 1. Additionally, we compare our method to recently introduced fast ODE solvers, including DPM-Solver [17] and DPM-Solver++ [18].

Real-world super-resolution. We evaluate our method on the challenging real-world super-resolution task, where the degradation is simulated using Real-ESRGAN pipeline [47]. Following StablSR [46], we compare all methods on 3,000 randomly degraded image pairs. The quantitative performance is shown in Table 2. The results demonstrate that our distilled method leads to a significant better performance than other distillation techniques. Our method achieves better results than fine-tuned diffusion models that requires 50× more sampling setps. Compared with the distilled model by applying the guided-distillation, our model outperforms it both quantitatively and qualitatively. The visual comparison presented in Figure. 4 also demonstrates the superiority of our method.

Inpainting. Similar to the above super-resolution comparisons, we demonstrate our method on the inpainting task that conditioned on the masked image, as the quantitative performance shown in Table 2. Similar to Palette [33], we apply random masks into ImageNet data [32] for both training and testing. Note that we conduct experiments on the up-scaled images in a 512×512 resolution, which is different than Palette in 256×256 resolution. Even though we evaluate their results in the same resoltuion, their number can only be used for reference.

Depth-to-image generation. In order to demonstrate the generality of our method on less informative conditions, we apply our method in depth-to-image generation. The task is usually conducted in parameter-efficient diffusion model finetuning [24, 53], which can demonstrate the capability of utilizing text-to-image generation priors. As Figure 5 illustrated, our distilled model from the unconditional pretraining can effectively utilize the less informative conditions and generate matched images with more details.

Instructed image editing. To demonstrate our conditional distillation capability on text-to-image generation, here we apply our method on text-instructed image editing data [3] and compare our conditional distilled model with the In-

Methods	Params	FID	LPIPS
LDMs	865M	29.266	0.3014
+ ControlNet	1.22B	28.951	0.3049
PE-CoDi (Ours)	364M	25.214	0.2941
CoDi (Ours)	1.22B	19.637	0.2656
- distilling PF-ODE	1.22B	20.307	0.2733
- noise-consistency	1.22B	25.728	0.3252

Table 3. Impact of the network architecture and conditional distillation process, where all methods are using the same 4-step sampling.

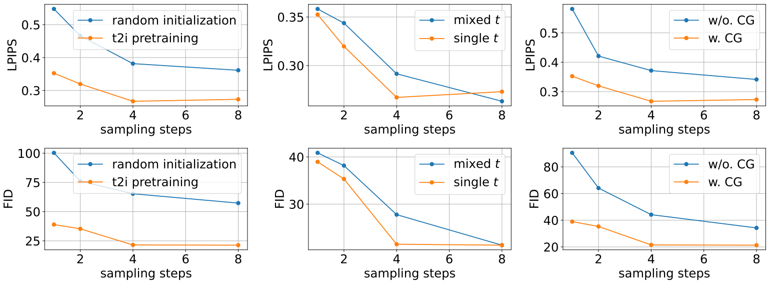


Figure 7. Ablations between alternative settings of our method.

structPix2Pix (IP2P) model. As the results shown in Figure 6, our single-step sampling result can achieve comparable visual quality to 200 steps of the IP2P model. We experimentally find only small visual difference between the results from our single-step sampling and the 200 steps sampling. We believe this suggests that the effect of the conditional guidance on distillation correlates with the similarity between the conditions and the target data, further demonstrating the effectiveness of our method.

5.2. Ablations

Here we compare the performance of the aforementioned designs in our conditional distillation framework. Specifically we focus on the representative conditional generation task *i.e.*, real-world super-resolution [48] that conditions on the low-resolution, noisy, blurry images.

Network architecture and distillation process. To eliminate the impact of the architecture change, we compare our method with a baseline given by adding a ControlNet module trained on super-resolution without freezing the UNet. As Table 3 shows, simply adopting a ControlNet module for super-resolution has negligible impact on the performance. To evaluate the proposed conditional diffusion consistency, we removed the noise consistency term (equation 15) and employed the training model in the PF-ODE instead of the frozen one as used in [44] formulation. As shown in Table 3, adopting the distillation model PF-ODE and noise-space consistency have positive effects on the final results. These comparisons demonstrate the superiority of our method without network architecture effects.

Pretraining. To validate the effectiveness of leveraging pretraining in our model, we compare the results of random initialization with initialization from the pre-trained text-to-image model. As shown in Figure 7, our method outperforms the random initialized counterpart by a large margin, thereby confirming that our strategy indeed utilizes the advantages of pretraining during distillation instead of simply learning from scratch.

Sampling of $\mathbf{z}_t$. We empirically show that the way of sampling $\mathbf{z}_t$ plays a crucial role in the distillation learning process. Compared with the previous protocol [23, 36] that samples $\mathbf{z}_t$ in different time t in a single batch, we show that using a consistent time t across different samples in a single batch leads to a better performance in our targeted 1-4 steps. As the comparisons shown in Figure 7, the model trained with a single time t (in a single batch) achieves better performance in both the visual quality (*i.e.*, FID) and the accuracy (*i.e.*, LPIPS) when the number of evaluations is increasing during inference.

Conditional guidance. In order to demonstrate the importance of our proposed conditional guidance (CG) for distillation, which is claimed to be capable of regularizing the distillation process during training, we conduct comparisons between the setting of using the conditional guidance as $r = \|\mathbf{x} - \hat{\mathbf{x}}_\theta(\mathbf{z}_t, c)\|_2^2$ and not using as $r = 0$. As the result shown in Figure 7, the conditional guidance improves both the fidelity of the generated results and visual quality. We further observed that the distillation process will converge toward over-saturated direction without CG, which thus lower the FID metric. In contrast, our model avoids such a local minimum by using the proposed guidance loss.

6. Conclusion

We introduce a new framework for distilling an unconditional diffusion model into a conditional one that allows sampling with very few steps. To the best of our knowledge, this is the first method that distills the conditional diffusion model from the unconditional pretraining in a single stage. Compared with previous two-stage distillation and finetuning techniques, our method leads to better quality given the same number of (very few) sampling steps. Our method also enables a new parameter-efficient distillation that allows different distilled models, trained for different tasks, to share most of their parameters. Only a few additional parameters are needed for each different conditional generation task. We believe the method can serve as a strong practical approach for accelerating large-scale conditional diffusion models.

References

[1] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, 2022. 3

[2] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *ArXiv preprint*, 2022. 4

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2, 5, 7

[4] Mauricio Delbracio and Peyman Milanfar. Inversion by direct iteration: An alternative to denoising diffusion for image restoration. *Transactions on Machine Learning Research*, 2023. Featured Certification. 2

[5] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021. 3

[6] Tan M. Dinh, Anh Tuan Tran, Rang Nguyen, and Binh-Son Hua. Hyperinverter: Improving stylegan inversion via hypernetwork. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, 2022. 3

[7] Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Lingjie Liu, and Joshua M Susskind. Boot: Data-free distillation of denoising diffusion models with bootstrapping. In *ICML 2023 Workshop on Structured Probabilistic Inference {\&} Generative Modeling*, 2023. 3

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 3

[9] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, 2019. 3

[10] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 2022. 2, 4, 13

[11] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *ArXiv preprint*, 2023. 3

[12] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 2021. 3

[13] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. 3

[14] Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. *NeurIPS*, 2023. 2, 12

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1

[16] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 2

[17] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 2022. 2, 6, 7

[18] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *ArXiv preprint*, 2022. 6, 7

[19] Haoye Lu, Yiwei Lu, Dihong Jiang, Spencer Ryan Szabados, Sun Sun, and Yaoliang Yu. Cm-gan: Stabilizing gan training with consistency models. In *ICML 2023 Workshop on Structured Probabilistic Inference {\&} Generative Modeling*, 2023. 3

[20] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and L Repaint Van Gool. Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*. 6

[21] Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *ArXiv preprint*, 2021. 2

[22] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022. 1

[23] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2, 4, 6, 8

[24] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *ArXiv preprint*, 2023. 3, 5, 7

[25] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, 2021. 4

[26] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 2

[27] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv preprint*, 2022. 1

[28] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018. 3

[29] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022. 1, 2, 4, 6

[30] Amir Rosenfeld and John K Tsotsos. Incremental learning through deep adaptation. *IEEE transactions on pattern analysis and machine intelligence*, (3), 2018. 3

[31] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 2015. 7

[33] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, 2022. 2, 6, 7

[34] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 2022. 1, 4

[35] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (4), 2022. 2

[36] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022. 2, 3, 4, 8

[37] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *ArXiv preprint*, 2021. 2

[38] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 2022. 1, 2

[39] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, 2015. 3

[40] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021. 3

[41] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *ArXiv preprint*, 2023. 3

[42] Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, 2019. 3

[43] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021. 2, 3

[44] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *ICML*, 2023. 2, 3, 4, 6, 8, 13

[45] Asa Cooper Stickland and Iain Murray. BERT and pals: Projected attention layers for efficient adaptation in multi-task learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, 2019. 3

[46] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. *ArXiv preprint*, 2023. 6, 7

[47] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *IEEE/CVF International Conference on Computer Vision Workshops, ICCVW 2021, Montreal, BC, Canada, October 11-17, 2021*, 2021. 7

[48] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Realesrgan: Training real-world blind super-resolution with pure synthetic data supplementary material. *Computer Vision Foundation open access*, 2022. 5, 6, 8

[49] Jay Whang, Mauricio Delbracio, Hossein Talebi, Chitwan Saharia, Alexandros G Dimakis, and Peyman Milanfar. Deblurring via stochastic refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[50] Bin Xia, Yulun Zhang, Shiyin Wang, Yitong Wang, Xinglong Wu, Yapeng Tian, Wenming Yang, and Luc Van Gool. Diffir: Efficient diffusion model for image restoration. *ICCV*, 2023. 2, 6

[51] Yilun Xu, Mingyang Deng, Xiang Cheng, Yonglong Tian, Ziming Liu, and Tommi Jaakkola. Restart sampling for improving generative processes. *ArXiv preprint*, 2023. 2

[52] Zongsheng Yue, Jianyi Wang, and Chen Change Loy. Resshift: Efficient diffusion model for image super-resolution by residual shifting. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2

[53] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 1, 2, 3, 4, 5, 6, 7

[54] Hongkai Zheng, Weili Nie, Arash Vahdat, Kamyar Azizzadenesheli, and Anima Anandkumar. Fast sampling of diffusion models via operator learning. In *International Conference on Machine Learning*. PMLR, 2023. 2

A. Discussion

Limitations. We have shown image conditions benefit our distillation learning. However, the distillation learning depends on the adapter architecture that introduces additional computation in our current framework. As a future work, we would like to explore lightweight network architectures [14] in our distillation technique to further reduce the inference latency. Nevertheless, CoDI's significantly reduced sampling steps lead to lower latency. See the following table (measured in TPUv5) for a detailed comparison:

Method	CoDi (4step)	ControlNet (4step)	LDMs (4step)	LDMs (50step)
Latency (ms)	107 ± 3	107 ± 3	103 ± 2	977 ± 1

Ethics statement. The diffusion distillation technique introduce in this work holds the promise of significantly enhancing the practicality of diffusion models in everyday applications such as consumer photography and artistic creation. While we are excited about the possibilities this model offers, we are also acutely aware of the possible risks and challenges associated with its deployment. Our model's ability to generate realistic scenes could be misused for generating deceptive content. We encourage the research community and practitioners to prioritize privacy-preserving practices when using our method.

B. Proofs

B.1. Notations

We use $\hat{\mathbf{v}}_\theta(\cdot,\cdot)$ to denote a pre-trained diffusion model that learns the unconditional data distribution $\mathbf{x} \sim p_{\text{data}}$ with parameters θ. The signal prediction and the noise prediction transformed by equation 8 are denoted by $\hat{\mathbf{x}}_\theta(\cdot,\cdot)$ and $\hat{\epsilon}_\theta(\cdot,\cdot)$, and they share the same parameters θ with $\hat{\mathbf{v}}_\theta(\cdot,\cdot)$.

B.2. Self-consistency in Noise Prediction

Remark. *If a diffusion model, parameterized by $\hat{\mathbf{v}}_\theta(\mathbf{z}_t, t)$, satisfies the self-consistency property on the noise prediction $\hat{\epsilon}_\theta(\mathbf{z}_t, t) = \alpha_t \hat{\mathbf{v}}_\theta(\mathbf{z}_t, t) + \sigma_t \mathbf{z}_t$, then it also satisfies the self-consistency property on the signal prediction $\hat{\mathbf{x}}_\theta(\mathbf{z}_t, t) = \alpha_t \mathbf{z}_t - \sigma_t \hat{\mathbf{v}}_\theta(\mathbf{z}_t, t)$.*

Proof. The diffusion model that satisfies the self-consistency in the noise prediction implies:

$$\begin{aligned}
\hat{\epsilon}_\theta(\mathbf{z}_{t'}, t') &= \hat{\epsilon}_\theta(\mathbf{z}_t, t), \\
\alpha_{t'} \hat{\mathbf{v}}_\theta(\mathbf{z}_{t'}, t') + \sigma_{t'} \mathbf{z}_{t'} &= \alpha_t \hat{\mathbf{v}}_\theta(\mathbf{z}_t, t) + \sigma_t \mathbf{z}_t, \\
\hat{\mathbf{v}}_\theta(\mathbf{z}_{t'}, t') &= \frac{\alpha_t \hat{\mathbf{v}}_\theta(\mathbf{z}_t, t) + \sigma_t \mathbf{z}_t - \sigma_{t'} \mathbf{z}_{t'}}{\alpha_{t'}},
\end{aligned} \tag{16}$$

Based on the above equivalence, the transformation between the signal prediction $\mathbf{x}_\theta(\mathbf{z}_{t'}, t')$ and $\mathbf{x}_\theta(\mathbf{z}_t, t)$ by using the update ruler in equation 7 and the reparameterization trick is:

$$\begin{aligned}
\mathbf{x}_\theta(\mathbf{z}_{t'}, t') &= \alpha_{t'} \mathbf{z}_{t'} - \sigma_{t'} \hat{\mathbf{v}}_\theta(\mathbf{z}_{t'}, t') \\
&= \alpha_{t'} \mathbf{z}_{t'} - \sigma_{t'} \frac{\alpha_t \hat{\mathbf{v}}_\theta(\mathbf{z}_t, t) + \sigma_t \mathbf{z}_t - \sigma_{t'} \mathbf{z}_{t'}}{\alpha_{t'}} && \text{// integrating equation 16} \\
&= \frac{\alpha_{t'}^2 \mathbf{z}_{t'} - \sigma_{t'} \alpha_t \hat{\mathbf{v}}_\theta(\mathbf{z}_t, t) - \sigma_{t'} \sigma_t \mathbf{z}_t + \sigma_{t'}^2 \mathbf{z}_{t'}}{\alpha_{t'}} \\
&= \frac{(1 - \sigma_{t'}^2) \mathbf{z}_{t'} - \sigma_{t'} \alpha_t \hat{\mathbf{v}}_\theta(\mathbf{z}_t, t) - \sigma_{t'} \sigma_t \mathbf{z}_t + \sigma_{t'}^2 \mathbf{z}_{t'}}{\alpha_{t'}} \\
&= \frac{\mathbf{z}_{t'} - \sigma_{t'} (\alpha_t \hat{\mathbf{v}}_\theta(\mathbf{z}_t, t) + \sigma_t \mathbf{z}_t)}{\alpha_{t'}} \\
&= \frac{\mathbf{z}_{t'} - \sigma_{t'} (\hat{\epsilon}_\theta(\mathbf{z}_t, t))}{\alpha_{t'}} && \text{// transformed with equation 8} \\
&= \frac{\alpha_{t'} \mathbf{x}_\theta(\mathbf{z}_t, t) + \sigma_{t'} \hat{\epsilon}_\theta(\mathbf{z}_t, t) - \sigma_{t'} (\hat{\epsilon}_\theta(\mathbf{z}_t, t))}{\alpha_{t'}} && \text{// update ruler equation 9 of DDIM} \\
&= \mathbf{x}_\theta(\mathbf{z}_t, t).
\end{aligned}$$

The derived equivalence shows that enforcing the self-consistency in the noise prediction, which is implemented by learning to minimize our distillation loss in equation 15, enforces the self-consistency in the signal prediction and can distill the pre-trained diffusion model. □

C. Difference between Consisntecy Models

Algorithm 1 Conditional Diffusion Distillation (CDD)

```
Input: conditional data $(\mathbf{x}, c) \sim p_{\text{data}}$, adapted diffusion model $\hat{\mathbf{w}}_\theta(\mathbf{z}_t, c, t)$, learning rate $\eta$, distance functions $d_\epsilon(\cdot,\cdot)$
and $d_{\mathbf{x}}(\cdot,\cdot)$, and EMA $\gamma$
$\boldsymbol{\theta}^- \leftarrow \boldsymbol{\theta}$                                  // target network initlization
repeat
    Sample $(\mathbf{x}, c) \sim p_{\text{data}}$ and $t \sim [\Delta t, T]$              // empirically $\Delta t = 1$
    Sample $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
    $s \leftarrow t - \Delta t$
    Sample $\mathbf{z}_t \leftarrow \alpha_t \mathbf{x} + \sigma_t \epsilon$
    - $\hat{\mathbf{x}}_t \leftarrow \alpha_t \mathbf{z}_t - \sigma_t \Phi(\mathbf{z}_t, c, t)$
    - $\hat{\epsilon}_t \leftarrow \alpha_t \Phi(\mathbf{z}_t, c, t) + \sigma_t \mathbf{z}_t$
    + $\hat{\mathbf{x}}_t \leftarrow \alpha_t \mathbf{z}_t - \sigma_t \hat{\mathbf{w}}_\theta(\mathbf{z}_t, c, t)$      // signal prediction in equation 8
    + $\hat{\epsilon}_t \leftarrow \alpha_t \hat{\mathbf{w}}_\theta(\mathbf{z}_t, c, t) + \sigma_t \mathbf{z}_t$      // noise prediction in equation 8
    $\hat{\mathbf{z}}_s \leftarrow \alpha_s \hat{\mathbf{x}}_t + \sigma_s \hat{\epsilon}_t$                        // update rule in equation 9
    - $\hat{x}'_t \leftarrow \alpha_t \mathbf{w}_\theta(\mathbf{z}_t, c, t) + \sigma_t \mathbf{z}_t$
    - $\hat{x}'_s \leftarrow \alpha_t \mathbf{w}_{\theta^-}(\hat{\mathbf{z}}_s, c, s) + \sigma_s \hat{\mathbf{z}}_s$
    + $\hat{\epsilon}_s \leftarrow \alpha_s \mathbf{w}_{\theta^-}(\hat{\mathbf{z}}_s, c, t) + \sigma_s \hat{\mathbf{z}}_s$      // noise prediction in equation 8
    - $\mathcal{L}(\theta, \theta^-) \leftarrow d_{\mathbf{x}}(\hat{x}'_t, \hat{x}'_s)$
    + $\mathcal{L}(\theta, \theta^-) \leftarrow d_\epsilon(\hat{\epsilon}_t, \hat{\epsilon}_s) + d_{\mathbf{x}}(\mathbf{x}, \hat{\mathbf{x}}_t)$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-)$
    $\boldsymbol{\theta}^- \leftarrow \text{stopgrad}(\gamma \boldsymbol{\theta}^- + (1-\gamma)\boldsymbol{\theta})$          // exponential moving average
until convergence
```

D. Implementation Details

Skip Connections. We implement the skip connections as follows, which is same as the consistency models [44] and EDMs [10] for satisfying the boundary condition but f_ϕ could be either the signal prediction or noise prediction:

$$f'_\phi(\mathbf{z}_t, t) = c_{\text{skip}}(t)\mathbf{x} + c_{\text{out}}(t) f_\phi(\mathbf{z}_t, t), \tag{17}$$

where

$$c_{\text{skip}}(t) = \frac{\sigma_{\text{data}}}{t^2 + \sigma^2_{\text{data}}}, c_{\text{out}}(t) = \frac{\sigma_{\text{data}} t}{\sqrt{t^2 + \sigma^2_{\text{data}}}}. \tag{18}$$

We use $\sigma_{\text{data}} = 0.5$.

E. Sampling Process Visualization

In order to provide a comprehensive understanding about the sampling process of our distilled model, as well as the difference between ours and the finetuned conditional diffusion model, here we visualize their predicted clean image $\hat{\mathbf{x}}_0$ at each sampling steps in equation 8.

As the results shown in Figure 8, we can find that our method constantly adds more details into the predicted $\hat{\mathbf{x}}_0$ when samples more steps. In contrast, such a constanly refinement is less visible in the results of the finetuned undistilled model. The different demonstrate that our method indeed can reduce the sampling time by learning to replicate the iterative refinement effects.

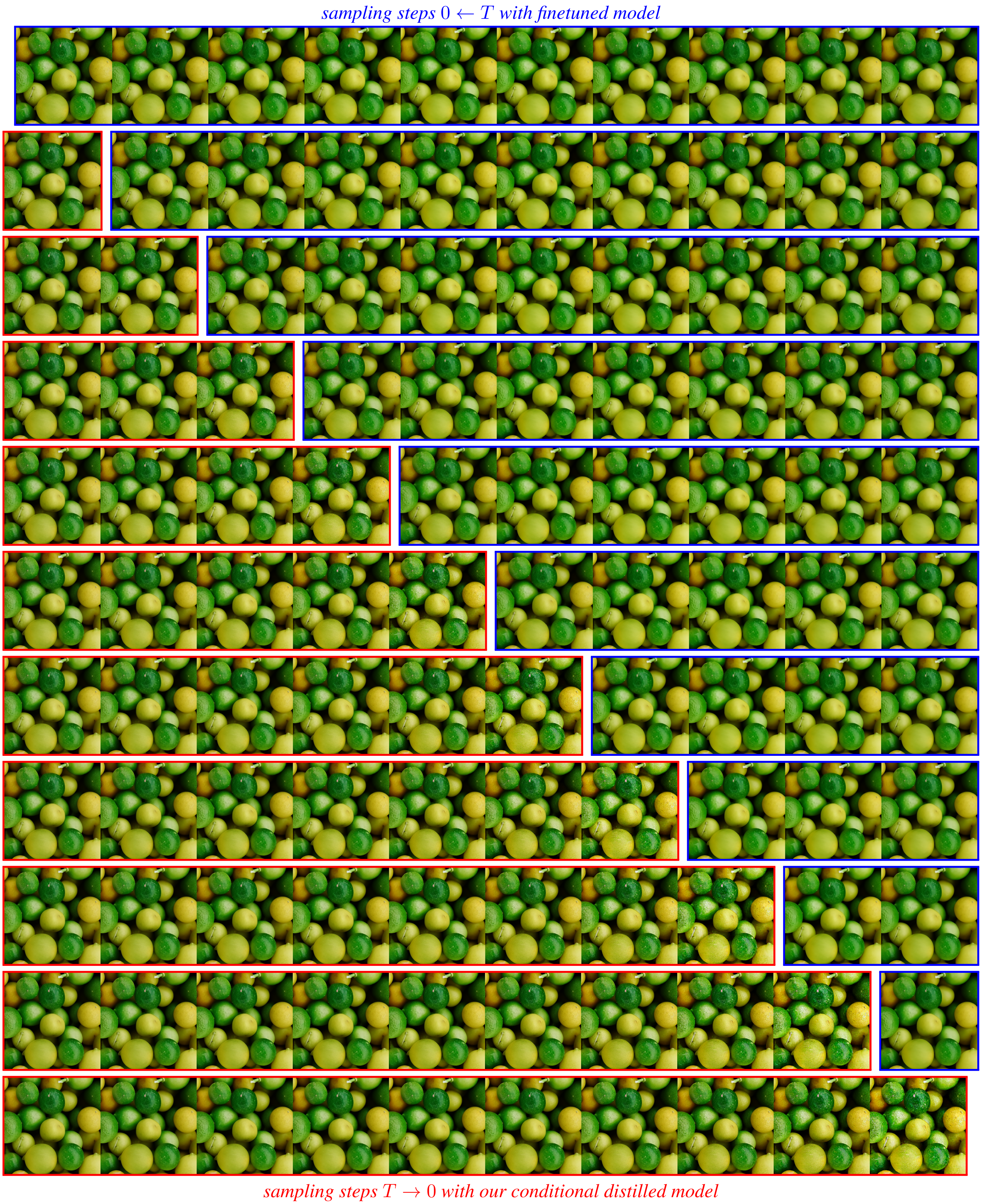

sampling steps $0 \leftarrow T$ with finetuned model
sampling steps $T \rightarrow 0$ with our conditional distilled model

Figure 8. Sampling process visualization of the distilled model by using our conditional diffusion distillation and the finetuned conditional diffusion model. The results belong to the same row come from the predicted $\hat{\mathbf{x}}_0$ at different time of the same sampling process, while different row denotes different sampling process that uses different the total number of the sampling time, which are increased from $T = 0$ into $T = 10$ and decreased from $T = 10$ into $T = 0$, respectively.

F. Additional results

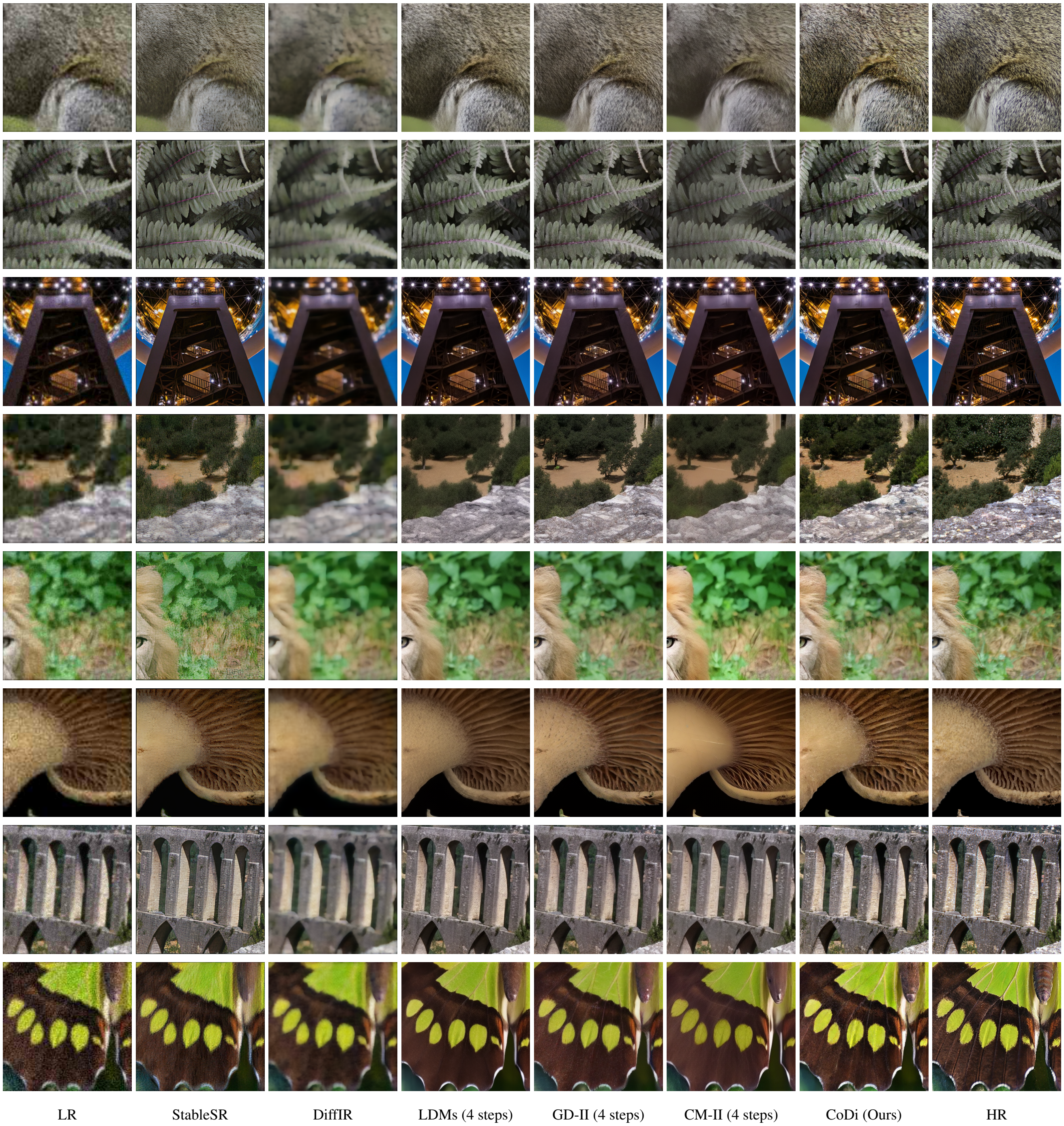
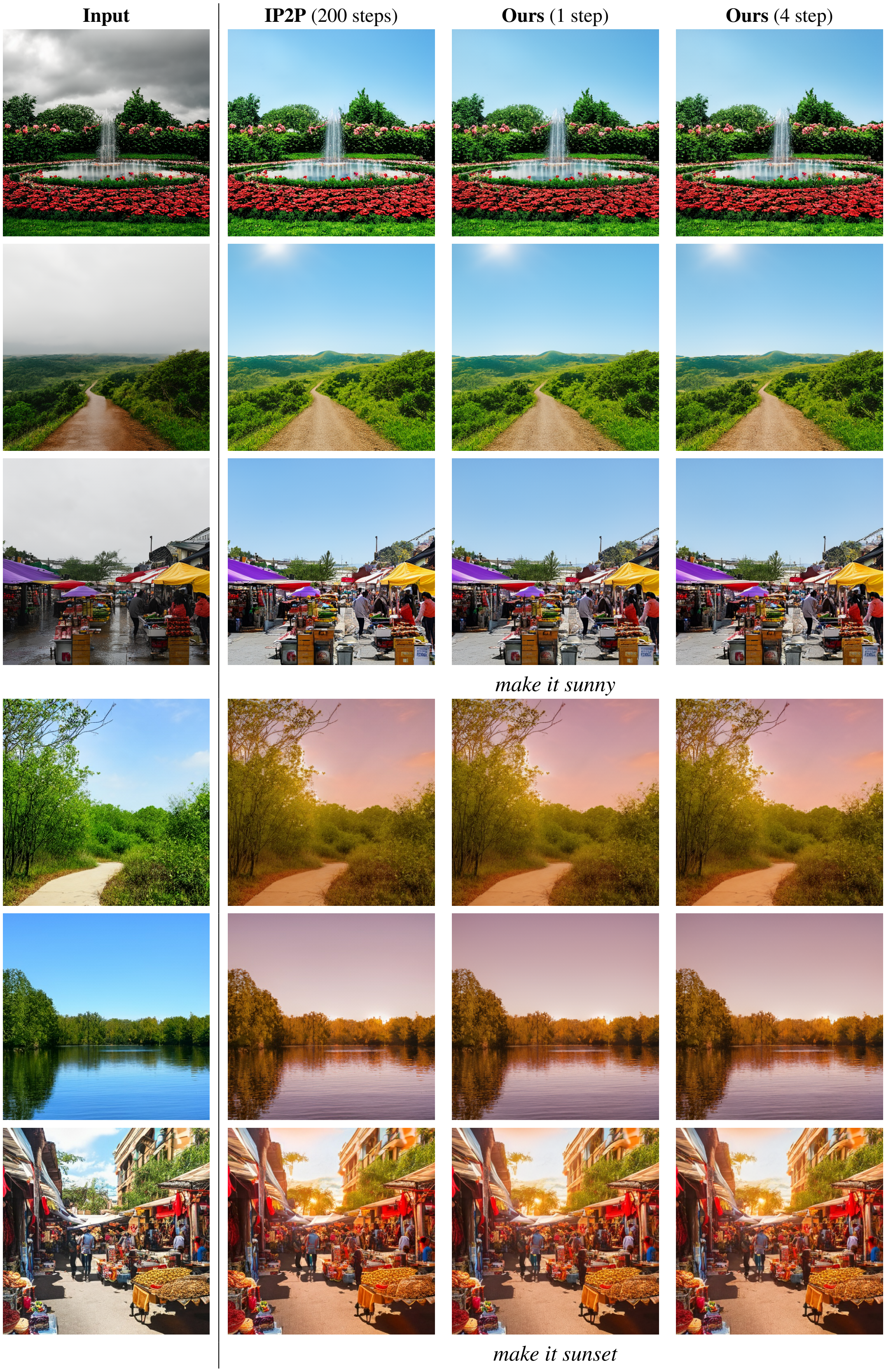


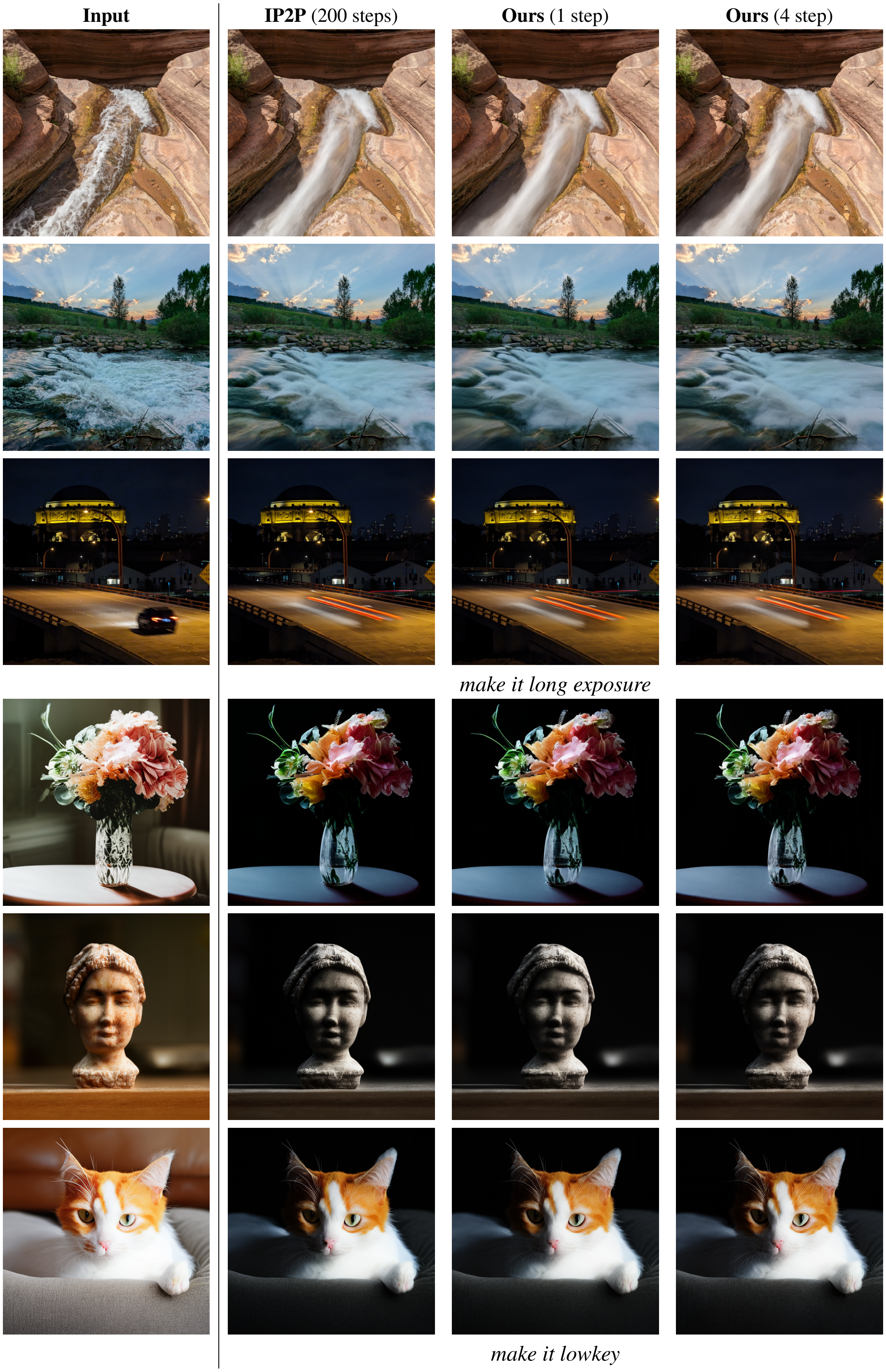
Figure 9. Visual comparisons of various diffusion-based methods on the simulated real-world super-resolution benchmark. The input of all methods is a 'Bicubic'-upsampled image.



make it sunny

make it sunset

Figure 10. Visual comparisons with the IP2P model and our conditional distilled model.



Input | **IP2P** (200 steps) | **Ours** (1 step) | **Ours** (4 step)

make it long exposure

make it lowkey

Figure 11. Visual comparisons with the IP2P model and our conditional distilled model.

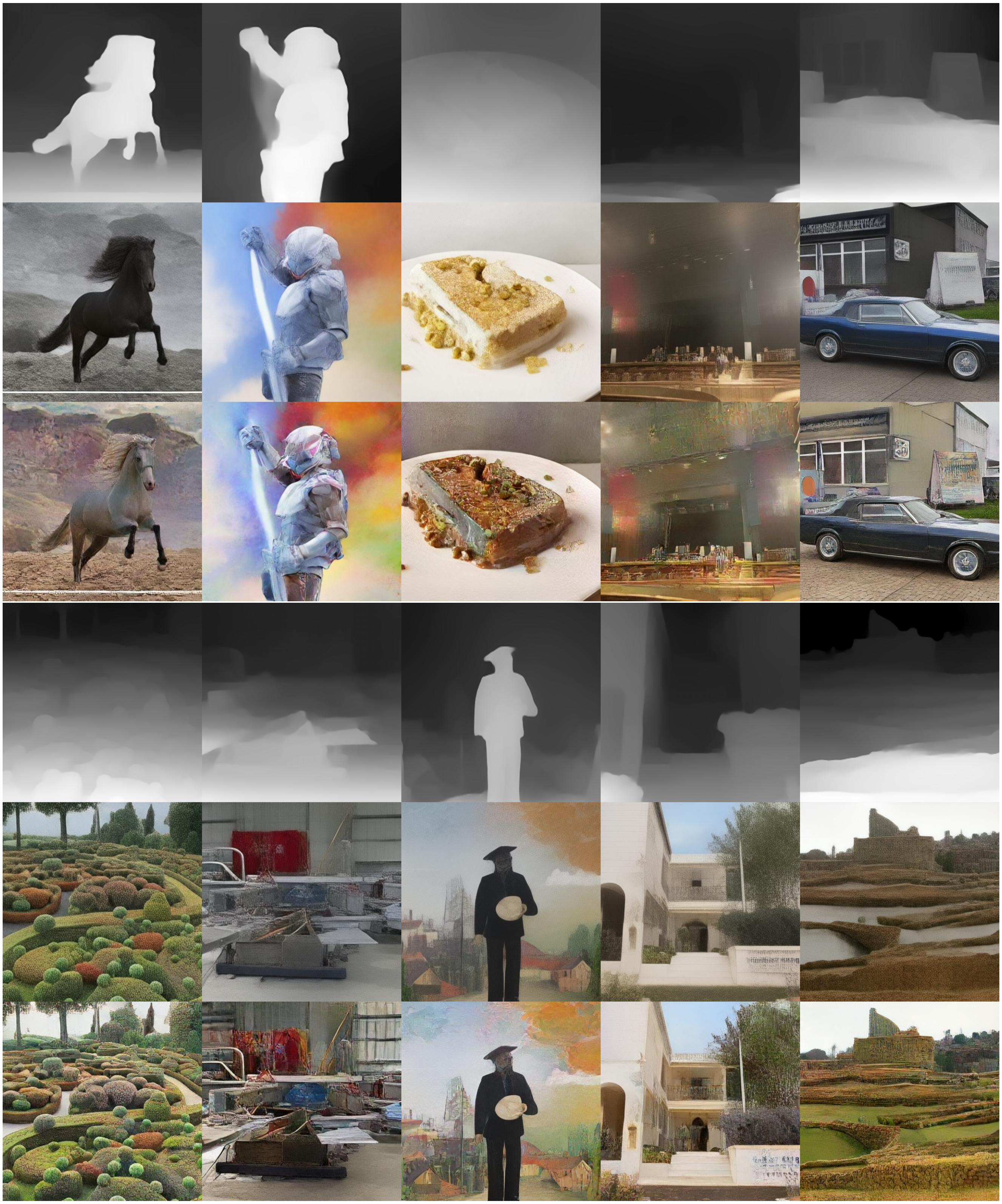


Figure 12. Visual comparisons of depth to image generation with the native ControlNet (central row of each item) and our conditional distilled model (bottom row of each item) in 4 sampling steps.

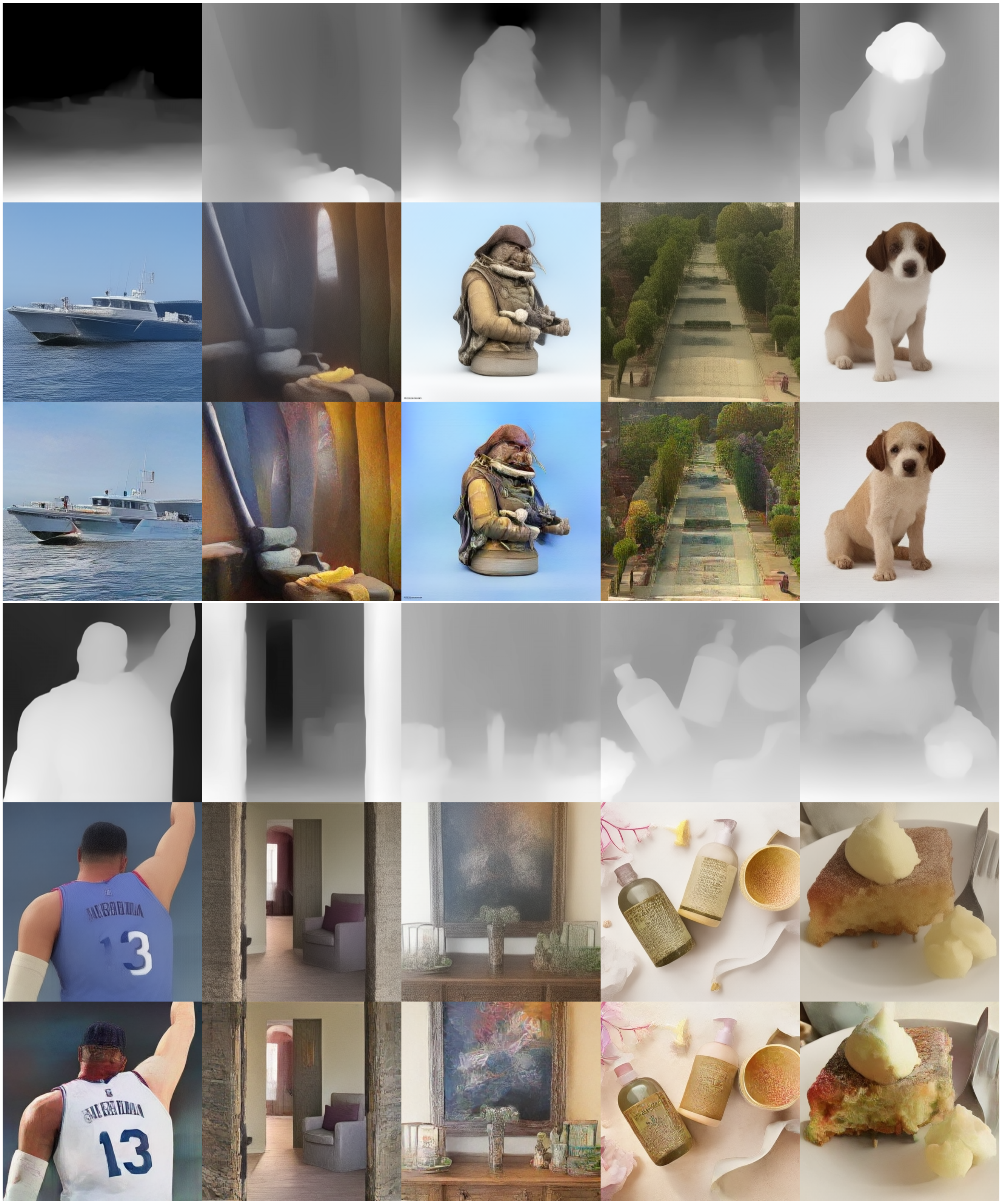


Figure 13. Visual comparisons of depth to image generation with the native ControlNet (central row of each item) and our conditional distilled model (bottom row of each item) in 4 sampling steps.