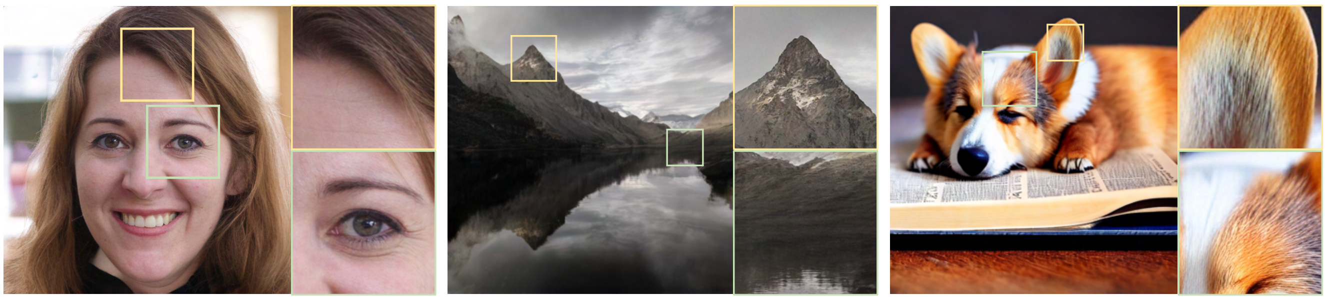
Image Neural Field Diffusion Models

Yinbo Chen[1] Oliver Wang[2] Richard Zhang[3] Eli Shechtman[3] Xiaolong Wang[1†] Michael Gharbi[3†]
[1]UC San Diego [2]Google Research [3]Adobe Research
https://yinboc.github.io/infd/



FFHQ
Resolution: 1024×1024

Mountains
Resolution: 2048×2048

"A cute corgi sleeping on a book, 4k"
Resolution: 2048×2048

Figure 1. **Generated samples from our image neural field diffusion models.** We show photorealistic high-resolution image generation by rendering generated image neural fields at 2K resolution for single domain models (left and middle), as well as general text-to-image models (right), with an efficient diffusion process on latent representation at only 64 × 64 resolution.

Abstract

Diffusion models have shown an impressive ability to model complex data distributions, with several key advantages over GANs, such as stable training, better coverage of the training distribution's modes, and the ability to solve inverse problems without extra training. However, most diffusion models learn the distribution of fixed-resolution images. We propose to learn the distribution of continuous images by training diffusion models on image neural fields, which can be rendered at any resolution, and show its advantages over fixed-resolution models. To achieve this, a key challenge is to obtain a latent space that represents photorealistic image neural fields. We propose a simple and effective method, inspired by several recent techniques but with key changes to make the image neural fields photorealistic. Our method can be used to convert existing latent diffusion autoencoders into image neural field autoencoders. We show that image neural field diffusion models can be trained using mixed-resolution image datasets, outperform fixed-resolution diffusion models followed by super-resolution models, and can solve inverse problems with conditions applied at different scales efficiently.

1. Introduction

Diffusion models [16, 34, 50] have recently become attractive alternatives to GANs. These likelihood-based models exhibit fewer artifacts, stable training, can model complex data distributions, do not suffer from mode collapse, and can solve inverse problems using the score function without extra training. Since diffusion typically requires many iterations at a fixed dimension, directly modeling the diffusion process in the pixel space [17, 40, 43] can be inefficient for high-resolution image synthesis. Latent diffusion models (LDMs) [41, 57] were proposed as a more efficient alternative. The key idea is to learn an autoencoder to map images to a latent representation from which the image can be decoded back, and train a diffusion model on the lower-dimensional latent representation. Despite their success, LDMs' latent space still represents images at fixed resolution (for example, 256 in LDM [41] and 512 in Stable Diffusion). To generate higher-resolution images (e.g., 2K), LDMs usually first generate a low-resolution image and upsample it using a separate super-resolution model.

In this work, we propose Image Neural Field Diffusion models (INFD). Our method is based on the latent diffusion framework, where we first learn a latent representation that represents an image neural field (which can be rendered at any resolution), then learn a diffusion model on this latent representation. A key challenge of our approach is to learn a latent space of photorealistic image neural fields where the diffusion model is applied. We propose a simple and effective method that can convert an existing autoencoder of latent diffusion models to a neural field autoencoder. We find that directly implementing an autoencoder with LIIF [8]

[†]Equal advising.

leads to blurred image details, and propose a Convolutional Local Image Function (CLIF), which can render the latent representation to photorealistic high-resolution images and the image content is consistent at different resolutions. Our neural field autoencoder is trained with L1 loss, perceptual loss [60], and GAN loss following LDM [41], and is supervised from multi-scale patches similar to AnyResGAN [6].

We show that image neural field diffusion models have several key advantages over fixed-resolution diffusion models: (i) They can be built from mixed-resolution datasets without resizing images. The neural field decoder can render latent representation at any resolution and from patches, which can take supervision from ground-truth images at arbitrary high resolution without decoding the whole image. (ii) The same latent representation can be supervised with GAN loss from fixed-resolution patches at different scales, with the content consistency across scales, the multi-scale supervision helps high-resolution generation even if all ground-truth images are at a fixed high resolution. (iii) It does not require an extra Super-Resolution (SR) model for high-resolution generation. Besides the advantage of simplicity, since diffusion-generated low-resolution images do not have high-resolution ground truth, separate SR models are typically trained on real images, while the domain gap between real and generated images could significantly harm the performance of the SR model. (iv) Image neural field diffusion models learn a resolution-agnostic image prior. Therefore, it can be used to solve inverse problems with a set of conditions defined at different scales efficiently.

In summary, our main contributions are:

- An image neural field autoencoder that can learn representations from mixed-resolution datasets and renders scale-consistent and photorealistic images.
- A method to build diffusion models on mixed-resolution datasets and synthesize high-resolution images without extra SR models. Image synthesis is up to 2K resolution with an efficient latent diffusion process at only 64×64 resolution (see samples in Figure 1).
- A framework to solve inverse problems with conditions applied at different scales of the same image.

2. Related Work

Diffusion Models. Diffusion models are first proposed by Sohl-Dickstein et. al. [50]. They were recently connected to score-based generative models [51–53] and have been greatly improved [16, 34] for architectures and other training details, achieving state-of-the-art results on both unconditional and conditional image synthesis [11, 17, 33, 42, 44]. Compared to prior GAN-based methods [14, 21, 22, 38, 62], diffusion models have shown nice properties such as stable training, not suffering from mode collapse, and can perform image-to-image translation [28, 29] or be used to solve inverse problems [54], even with an unconditional model. One of the main drawbacks of current diffusion models is slow inference speed, as it relies on iterative reverse diffusion steps. While this can be remedied with faster sampling methods [23, 27, 46], performing the diffusion process in the pixel space of high-resolution images remains computationally expensive.

Our work is most closely related to latent diffusion models, which learn to map images to latent representation and train diffusion model on the latent space [41, 47, 57]. A key design in this direction is to choose the latent representation where the diffusion model is learned. Different from prior works that learn an autoencoder, where the latent representation corresponds to a fixed resolution image, we design a decoder and a renderer to learn a latent space that represents image neural fields. Since our latent space represents image neural fields, our autoencoder can learn a representation from high-resolution images in varied sizes with multi-scale patches, and can synthesize images at high resolution without relying on extra super-resolution models. Our method follows LDM [41], which trains in two stages.

Neural Fields for Image Synthesis. Neural field is also known as Implicit Neural Representations (INR), which represents signals as coordinate-based neural networks. It serves as a compact and powerful differentiable representation and achieves state-of-the-art results mainly for representing 3D shapes [9, 30, 36] and scenes [2, 5, 19, 31, 37, 45, 48]. Applications of neural fields for images are explored in early works [32, 55] and proposed for more applications such as image super-resolution [8] and image synthesis [1, 49]. Several recent works [6, 22] relax the pixel-independent assumption and perform convolutions on the coordinate map to render the output for image synthesis.

The idea of using neural fields for training with any-resolution images is explored in LIIF [8] for autoencoding with an L1 loss, AnyResGAN [6] and ScaleParty [35] for GANs with adversarial loss. Our work aims at building the resolution-agnostic learning framework for diffusion models, which is a different model family of generative models.

Image Super-Resolution. Image Super-Resolution [7, 12, 24, 26, 56, 58, 61] (SR) aims at upsampling a low-resolution image to higher resolution. Many recent works [8, 18, 25, 59] explore Arbitrary-Scale SR (ASSR) with a single network. While they are related to our method, the differences include: (i) Instead of learning an autoencoder and an extra SR model, our implementation can be viewed as making a bottleneck in a single ASSR model and training diffusion models on the bottleneck. (ii) Our decoder is decoding from latent space to RGB space while super-resolution is upsampling from RGB space to RGB space. (iii) Our input can potentially have any higher resolution information (e.g. crops from high resolution), we choose it as a fixed-low-resolution image for efficiency.

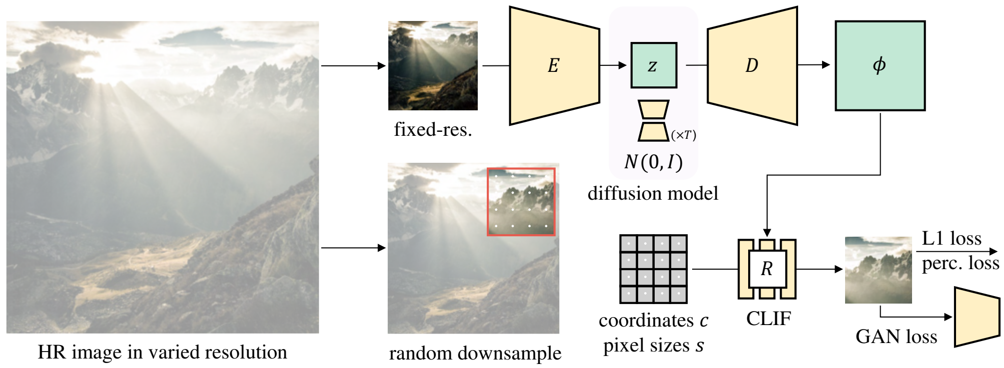


Figure 2. **Method overview.** Given a training image at an arbitrary resolution, we first downsample it to a fixed resolution and pass it into the encoder E to get a latent representation z. A decoder D then takes z as input and produces a feature map ϕ that drives a neural field renderer R, which can render images by querying with the appropriate grid of pixel coordinates c and pixel sizes s. The autoencoder is trained on crops from a randomly downsampled image ground truth, generating image crops at the corresponding coordinates. At test time, a diffusion model generates a latent representation z, which is then decoded and used to render a high-resolution image.

IDM [13] uses a diffusion model for ASSR where the output is at a medium resolution.

3. Preliminaries

Our algorithm builds on diffusion models and neural fields. We introduce the core concepts and notation below.

Diffusion Models. Given a sample x_0 from a data distribution $q(x_0)$, forward diffusion progressively destroys the information in x_0 in T steps, $x_{t-1} \mapsto x_t$, each adding some small Gaussian noise. The process can be concisely rewritten as $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$, and $\bar{\alpha}_{0\ldots T-1}$ gradually decreases from 1 to 0. The final distribution is approximately normal, $q(x_T) \sim \mathcal{N}(0, I)$. A diffusion model learns to reverse this diffusion process. Once trained, new samples can be generated by first sampling $x_T \sim q(x_T)$ and reversing each step of the diffusion process using a learned transition probability $p_\theta(x_{t-1}|x_t)$, parameterized by a neural network. Many prior works on image generation are based on maximizing a reweighted variational lower-bound of $p_\theta(x_0)$, which is shown by [16] to lead to the following training objective:

$$\mathcal{L} = \mathbb{E}_{x_0,t,\epsilon}\left[||\epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t) - \epsilon||^2\right], \quad (1)$$

where t is sampled in $\{0, \ldots, T-1\}$, and ϵ_θ is the network trained to reverse the diffusion process.

Neural Fields. Neural Fields represent a signal using a coordinate-based neural network. For example, a neural field can represent an image as a function $c = f(x; \phi)$, where $x \in [-1, 1]^2$ are the spatial coordinates in the image domain, $c \in \mathbb{R}^3$ is the RGB color at the corresponding continuous coordinate, and ϕ denotes the parameters of the neural field f. Since x can take continuous values, the RGB value can be decoded at arbitrary coordinates. Accordingly, an image neural field can be rendered at arbitrary resolution by sampling the corresponding pixel coordinates.

4. Method

Similar to LDM [41], our approach has two stages. First, we train an autoencoder that converts images to latent representations of 2D neural fields (§ 4.1), which can be rendered to images at any given resolution (§ 4.2). Second, we train a diffusion model to generate samples from this latent space (§ 4.3). Figure 2 illustrates our pipeline.

4.1. Image Neural Field Autoencoder

In the first stage, we seek to convert every image in our training set into a photorealistic image neural field. We do this by training an autoencoder, made of an encoder E, a decoder D, and a neural field renderer R. The encoder maps an RGB input image I to a latent code $z = E(I)$, which is decoded by the decoder to a feature tensor $\phi = D(z)$ used by the neural field renderer to produce the final image.

Patch-wise decoding. For training efficiency, we want to avoid decoding the whole image, because the ground truth can be at a very high resolution. We take advantage of the coordinate-based decoding property of neural fields, to train with constant-size crops from mixed-resolution data, which is amenable to batching. Specifically, we crop a random patch p_{GT} at a fixed $P \times P$ resolution (the red box in Figure 2) from a *randomly downsampled* ground-truth. Since p_{GT} is a fixed-size patch, downsampling the global ground truth lets the patch p_{GT} cover regions at varying scales of the image. This provides supervision at multiple scales to the latent representation, from local details to global structure. We discuss this further in § 5.2. Let c denote the coordinates of pixel centers in the patch within the image, and s denote their pixel sizes relative to the whole ground-truth image. Our renderer R takes as input the features $\phi = D(z)$ decoded from the latent representation, and the coordinates and pixel sizes c, s to synthesize an output patch $p = R(c, s; \phi)$.

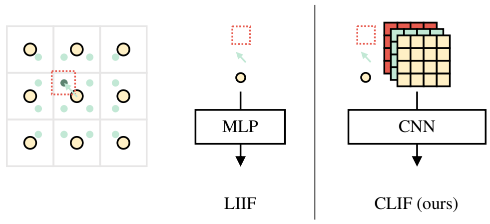


Figure 3. **Convolutional Local Image Function (CLIF).** Given a feature map ϕ (yellow dots), for each query point x (green dot), we fetch the nearest feature vector, along with the relative coordinates and the pixel size. The grid of query information is then passed into a convolutional network (right) that renders an RGB grid. Different than the pointwise function LIIF, CLIF has a higher generation capacity and is learned to be still scale-consistent.

Training objective. We compare the synthesized patch to the ground truth using a sum of an L_1 loss, a perceptual loss $\mathcal{L}_{\text{perc}}$ [60], and an adversarial loss $\mathcal{L}_{\text{GAN}}$ [14]. The discriminator is simultaneously trained to distinguish between the distributions of p and p_{GT}. Thus, we minimize the following objective to train E, D, and R:

$$\mathcal{L}_{AE} = ||p - p_{GT}||_1 + \mathcal{L}_{\text{perc}}(p, p_{GT}) + \mathcal{L}_{\text{GAN}}(p). \quad (2)$$

Our implementation follows the autoencoder architecture of LDM [41], with the same encoder and decoder (removed the last layer) architectures to facilitate comparisons. Since training images have arbitrary resolutions, we resample the encoder's input to a fixed resolution 256×256. Note that despite this downsampling, we still train against mixed-resolution references. The encoder and decoder are with a spatial downsampling/upsampling rate of 4 correspondingly, therefore the latent representation is in 64×64. A vector-quantization (VQ) layer is prepended to the first layer of the decoder to regularize the latent space. ϕ is 256×256 with 128 channels. We set patch size $P = 256$.

4.2. Neural Field Renderer

Our renderer R, shown in Figure 3, is a neural field coordinate-based decoder, which we dubbed Convolutional Local Image Function (CLIF). To decode an image patch with CLIF, each query point c (green dot) fetches the spatially nearest feature vector from the feature map ϕ (yellow dots). We concatenate the nearest feature vector with the query coordinates c and pixel size s, then process the grid of query information using a convolutional network to output an RGB image. Intuitively, the concatenated feature is the field information at a point. By changing the query coordinates and pixel sizes, we can decode images at any resolution. LIIF [8] decodes with similar information, but it uses a pointwise function. We found this limits LIIF's ability to produce realistic high-frequency details (see supplementary material). CLIF remedies this issue by exploiting more local feature context. Our CLIF renderer is learned to be scale-consistent, i.e., details are consistent when decoding at different resolutions (see supplementary material).

4.3. Latent diffusion

Once the autoencoder is trained, we map every image I in the training dataset to its latent representation z, and train a diffusion model by optimizing the DDPM [16] objective

$$\mathcal{L}_{DM} = \mathbb{E}_{z\sim E(I),t,\epsilon}\left[||\epsilon_\theta(\sqrt{\bar{\alpha}_t}z+\sqrt{1-\bar{\alpha}_t}\epsilon, t)-\epsilon||^2\right], \quad (3)$$

using the distribution over z induced by the encoder. After training, the encoder can be discarded. The diffusion model generates a latent representation z, which is then decoded to $\phi = D(z)$ and rendered at a resolution specified by the pixel coordinates $R(c, s; \phi)$ as described next.

4.4. Patchwise Image Generation

Despite being trained with small patches for efficiency, our method can generate high-resolution images. For this, we first generate a global feature map ϕ from a sampled z, then generate sub-tiles of a large image by querying the renderer at the corresponding coordinates, as described in § 4.2. To avoid discontinuities at tile boundaries, we expand the query region for each tile by a fixed padding size larger than CLIF's receptive field (8 in our experiments). We then crop the output tiles by the same amount and assemble the tiles into a seamless composite. Our renderer is fully convolutional, it can also generate the image at once, as long as memory is sufficient to hold intermediate buffers.

5. Experiments

We evaluate our method on several datasets and compare it to LDM [41] with super-resolution models (§ 5.1). Sections § 5.2 and § 5.3 presents model ablations. We show results in solving multi-scale inverse problems in § 5.5, and text-to-image generation in § 5.6.

Data. The FFHQ [20] used in LDM [41] contains 70K high-resolution images (1024×1024) and several baselines use it for comparison. When comparing to LDM, we follow their training and validation split: 60K images for training and 10K for validation. Since our method is flexible and not limited to a fixed-resolution dataset, we also follow a controlled setting in prior work [6], which constructs a varied-resolution dataset from FFHQ by constructing and merging three sets: (i) all images at 256 low-resolution; (ii) a subset of 5K samples in varied-resolution from 512 to 1024; (iii) a subset of 1K images at 1024 resolution. We denote this setting as FFHQ 6K-Mix. Besides FFHQ, we evaluate our method on the Mountains dataset [6], which contains a low-resolution subset, with about 500K samples around 1024 resolution, and a high-resolution subset, with about 9K images at resolutions beyond 2048. Figure 4 shows examples of our generated results on these datasets.

1024×1024 512×512

Figure 4. Generated samples from our method on FFHQ and Mountains dataset.

pFID metric. Standard FID evaluation first resizes images to 299, which evaluates the global structure of images regardless of their original resolution, but is insufficient to measure the quality of details from high-resolution generators. Patch-FID [6] (pFID) addresses this limitation. It computes the FID between fixed-resolution patches cropped from arbitrary resolution ground-truth and generated images, thus evaluating local details in synthesized images.

The original pFID reports the metric on random crops from images at varying resolutions. Since our practical focus is on high-resolution synthesis, we evaluate pFID between patches cropped from fixed-high-resolution images. To further disentangle the evaluation for image details at different scales, we separately report pFID for patches at different resolutions. P/1K will denote the FID between random crops at resolution P from ground truth and synthesized images at resolution 1024. For example, pFID-256/1K evaluates the local details, pFID-1K/1K evaluates the global structure and is the same as standard FID applied to 1024-resolution images. We generate 50K samples to compute FID in most experiments. In some experiments, we use 5K samples if it suffices to observe the performance gap (specified after FID@ in Tables).

5.1. Comparison to LDM

LDM [41] trains an autoencoder for images at 256×256 resolution and learns a diffusion model on its latent space. For a fair comparison, we used the same encoder and decoder architectures as LDM. Our method only adds a lightweight CLIF renderer, which contains 2 convolution layers and 2 ResNet [15] blocks.

Even though LDM's denoiser is fully convolutional and can generate high-resolution images simply by diffusing from a larger noise map, it is known that this approach generates repetitive patterns (e.g., distorted faces with duplicate features), we show examples in the supplementary material. As a result, we follow the standard approach to generate high-resolution images with LDM by running an independent super-resolution model on its output. We combine LDM with several recent state-of-the-art arbitrary-scale super-resolution methods: LIIF [8], ITSRN [59], LTE [25], which allow for inference at continuous upsampling scales, and Real-ESRGAN [58], a super-resolution model for a fixed upsampling scale that has state-of-the-art perceptual quality. We report qualitative results in Figure 5, and the pFID in Table 1 on FFHQ. As Real-ESRGAN shows the most competitive results, we also compare to it on Mountains dataset in Table 2.

We find that for standard FID at 256 resolution (i.e. pFID 256/256), the image neural field diffusion model is competitive with the original LDM, which can only generate images at 256. Our retraining of LDM reaches a slightly worse FID than the officially reported scores, which we attribute to implementation differences and training variance. At all higher resolutions, our method outperforms LDM followed by super-resolution, which is consistent with our qualitative observation and suggests we achieve better image quality for high-resolution detail at different scales.

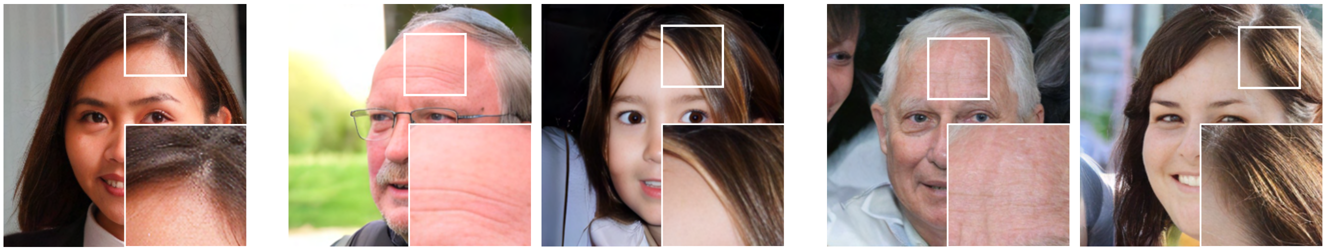


Figure 5. Qualitative comparison with LDM followed by super-resolution on FFHQ. LIIF shows noise in details, while Real-ESRGAN tends to be smooth and results lack rich details. Our approach generates images with more realistic details.

Model	#Params	Coordinate-based Decoder	pFID@50K			
			256/256	256/1K	512/1K	1K/1K
LDM [41]	33.0M		4.98	-	-	-
LDM + LIIF [8]	33.0M + 22.3M	✓	-	56.65	17.83	8.97
LDM + ITSRN-RDN [59]	33.0M + 22.6M	✓	-	51.73	17.94	8.02
LDM + SwinIR-LTE [25]	33.0M + 12.1M	✓	-	52.62	17.72	9.09
LDM + Real-ESRGAN [58]	33.0M + 16.7M		-	18.38	18.46	16.04
LDM (our reimplementation)	33.0M		6.02	-	-	-
INFD (ours)	35.7M	✓	5.34	**8.07**	**6.64**	**5.57**

Table 1. Comparison to Latent Diffusion Model (LDM) with super-resolution models for high-resolution image synthesis on FFHQ dataset. #Params counts for the decoder, and the renderer or super-resolution model if it exists, which are used in image generation.

We hypothesize that the main issues for extra super-resolution models are that: (i) LDM with an extra super-resolution model can be viewed as first decoding the latent representation to RGB space, then upsample to another RGB space, where the first RGB space becomes a bottleneck that contains much less information than feature space; (ii) the artifacts generated by LDM, even inconspicuous, can cause a domain shift to the input of the super-resolution models, which could significantly degrade their performance. The super-resolution models can not be directly trained to upsample the generated low-resolution images since no paired high-resolution ground truth is available. In our method, the domain shift from real to generated samples happens in the latent space (with VQ or KL regularization). We hypothesize that latent space is much more robust than RGB space to the domain shift. We observe that when replacing the latent space with RGB space, the generated images become overly smooth similar to LDM with Real-ESRGAN super-resolution, which is consistent with our hypothesis (see supplementary material).

Model	pFID@50K		
	256/1K	512/1K	1K/1K
LDM + Real-ESRGAN [58]	17.36	15.11	10.39
INFD (ours)	**7.53**	**6.84**	**5.13**

Table 2. Comparison to Latent Diffusion Model (LDM) with super-resolution models on Mountains dataset.

5.2. Effect of scale-varied training

Randomly downsampling the global image before extracting fixed-resolution training patches would make patches cover all scales. With the scale consistency of CLIF, an image neural field is supervised to be realistic in all scales via the perceptual and GAN losses. We conduct an ablation on FFHQ to evaluate the impact of this random downsampling strategy. Specifically, we disable it during training, keeping all ground-truth images at 1024 resolution. The results are shown in Table 3. Without random downsampling, the pFID is worse especially for 512/1K and 1K/1K, suggesting that random downsampling of the ground truth improves quality, even if we only aim at generating images at 1024 high resolution, because it helps supervise every single image to be realistic at all scales. For Mountains dataset, we observe that there will be obvious artifacts without random downsampling (see supplementary material). This contrasts with the observation in LIIF [8], which found that random downsampling hurts performance at a fixed highest scale when it is only using an L1 loss.

5.3. Training with limited high-resolution images

While previous experiments use high-resolution training images, a key advantage of our method is that it can learn from mixed-resolution datasets and still generate

Downsample	pFID@50K		
	256/1K	512/1K	1K/1K
Fix 1024	8.19	6.82	6.04
256–1024	**8.07**	**6.64**	**5.57**
Δ	0.12	0.18	0.47

Table 3. FFHQ dataset. Random downsampling during training improves image generation quality even for fixed high resolution.

Data	pFID@50K		
	256/1K	512/1K	1K/1K
All HR	8.07	6.64	5.57
6K-Mix	16.27	10.8	11.3
6K-Mix, bal.	12.41	7.74	6.9

Table 4. FFHQ dataset. Our method can learn from mixed-resolution images with a limited number of full-resolution images.

high-resolution outputs, even when the number of high-resolution images is limited. We quantify this in Table 4 using FFHQ 6K-Mix, where most images are at low resolution 256, 5K images are at 512–1024, and 1K images are at 1024. Training with the mixed-resolution dataset as-is (also with random downsampling) already yields a model that performs decently at 1024, but with worse details than the model trained with all images at 1024. We suspect that the model is optimized for too few steps using images at 1024, since the number of high-resolution images is small and they go through random downsampling. We balance the training resolution by sampling images from the 5K + 1K high-resolution subset with a probability of 0.5. This largely closes the performance gap (6K-Mix, bal.) with the model train on high-resolution images only.

5.4. Image generation beyond 1024

We explore going beyond the 1024 resolution and train our model on a collected dataset of faces at varied resolutions between 1024 to 2048, and on Mountains dataset including a higher-resolution subset (a generated sample is shown in Figure 1). We observe that our method can be effectively applied to resolutions up to 2K. More samples and details are in the supplementary material.

5.5. Inverse problems with conditions at any scale

Our method builds a diffusion model on a latent space that represents image neural fields. A key property of image neural fields is that they can be efficiently rendered for any sub-region at any given resolution without decoding the whole image at full resolution. With the image prior learned by diffusion models, it enables efficiently solving inverse problems where conditions can be defined on any scale of the image based on coordinates. We take zero-shot any-scale layout-to-image generation as an example. It uses

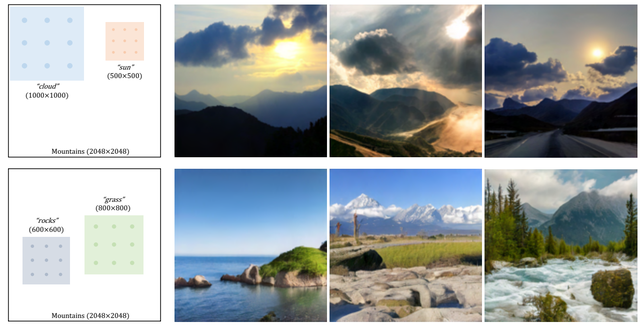


Figure 6. **Solving inverse problems with multi-scale conditions per image.** We can solve for an image that satisfies multi-scale conditions, defined as square regions and a text prompt (left). For this, we decode the corresponding region and pass it to a pre-trained CLIP [39] model operating at fixed-resolution (224 × 224), and maximize the CLIP similarity to the corresponding text prompt. This enables layout-to-image generation without extra training. We show generated solutions on the right.

CLIP [39] similarity as the constraint for image generation with semantic bounding boxes at arbitrary scales.

Specifically, we take a pre-trained CLIP model, which takes 224×224 fixed-resolution images as inputs. Given a layout and our image neural field diffusion model (unconditional), in each diffusion step, let z denote the current denoised latent representation, for each semantic bounding box i in the input layout we render z for the corresponding sub-region to a patch at the resolution 224×224, i.e. patch $p_i = R \circ D(z; c_i, s_i)$, with our decoder D and renderer R, where c_i, s_i denote the coordinates and scale of a 224×224 pixel grid of bounding box i. The clip similarity loss $l_i = \text{CLIP}(p_i, T_i)$ is computed between patch p_i and the given text T_i. The gradients $\frac{\partial l_i}{\partial z}$ are back-propagated and then used to modify the diffusion score for each diffusion step. We follow the techniques in DPS [10] as the inverse problem solver. Figure 6 shows the results of using our mountains model (2K resolution). Note that without image neural field diffusion models, a fixed-resolution diffusion model needs to decode the whole region at full resolution (e.g. 1000 × 1000 "cloud" region in a 2048 × 2048 canvas) before passing it as 224 × 224 input to CLIP, which would have very intensive computation and memory cost.

5.6. Qualitative results for text to image generation

We explore a preliminary application of our method for text-to-image generation by finetuning a pre-trained Stable Diffusion model [41]. Because of the high computational cost of training Stable Diffusion, we freeze the encoder and only finetune the decoder from publicly available pre-trained weights on a high-resolution subset of LAION-5B, which contains samples at resolutions higher than 2K. Our renderer has the same architecture as in previous experiments and is jointly trained from scratch. We show some qualitative samples from our fine-tuned model and compare them

"A cute corgi sleeping on a book, 4k" *"A huge mushroom in the jungle, 4k"* *"A Shiba Inu dog wearing sunglasses, 4k"*

Figure 7. Samples of our method finetuned from Stable Diffusion (LDM) compared to Stable Diffusion with an extra super-resolution model. Our approach yields finer, high-frequency textures.

to upsampling the Stable Diffusion's 512×512 output using Real-ESRGAN in Figure 7. We observe that the comparison is similar to the experiments on FFHQ and Mountains: our method generates more details than the Real-ESRGAN applied to Stable Diffusion's outputs. More samples and details are in the supplementary material.

6. Discussion

Comparison to GANs with any-resolution learning. In this work, we propose a method to build diffusion models on a resolution-agnostic space and show its applications. The any-resolution learning framework has also been developed for different generative model families in prior works, for example, AnyresGAN [6]. We observe that the comparison between image neural field diffusion models and AnyresGAN matches the comparison between typical fixed-resolution diffusion models and GANs. A detailed comparison is in the supplementary material. In summary, we observe that GANs are still state-of-the-art on FID value (which is not always consistent with image quality [3, 4]) for the single-class generation, while image neural field diffusion model has better visual quality and diversity than AnyRes-GAN. Image neural field diffusion models are also free of the artifacts that commonly affect AnyRes-GAN's outputs, such as black dots and grid-like patterns. Finally, image neural field diffusion models can be directly applied for text-to-image synthesis, which remains a challenge for any-resolution GANs. Solving inverse problems based on the score functions of diffusion models is also not yet available in GANs.

Limitations. Our current method assumes that the training data are scale-consistent, i.e., low-resolution images follow the same distribution as downsampled high-resolution images (see supplementary material). This assumption is violated by datasets that contain a low-resolution subset with noisy, compressed images, and a high-resolution subset with clean images (e.g., Birds, Churches datasets [6]).

Due to limited resources, our text-to-image synthesis model only fine-tunes an existing Stable Diffusion checkpoint on a small subset of the LAION dataset containing high-resolution images. This causes two issues. First, the training set of the pre-trained model includes noisy images, but our high-resolution fine-tuning dataset only contains clean images; this violates our scale consistency assumption. As a result, we found that our model requires extra prompts such as "4k" to generate detailed high-resolution images. Second, our fine-tuning LAION subset does not cover all possible object categories, so our model may not perform optimally on some out-of-distribution objects. Training from scratch on the full LAION dataset might resolve these limitations. Researching efficient any-resolution encoders is also a promising avenue for future work.

7. Conclusion

We proposed image neural field diffusion models, the diffusion models on a resolution-agnostic latent space, and demonstrated its advantages over fixed-resolution models. We presented a simple yet effective framework as an implementation, which can be easily applied to convert from an existing latent diffusion model. Our method can build diffusion models from mixed-resolution datasets, achieving high-resolution synthesis without extra super-resolution models. The resolution-agnostic image prior learned by the diffusion model also enables solving inverse problems with conditions applied at different scales of the same image.

Acknowledgements. This work was supported, in part, by NSF CAREER Award IIS-2240014, and the CISCO Faculty Award.

References

[1] Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhenkov. Image generators with conditionally-independent pixel synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14278–14287, 2021. 2

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 2

[3] Ali Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41–65, 2019. 8

[4] Ali Borji. Pros and cons of gan evaluation measures: New developments. *Computer Vision and Image Understanding*, 215:103329, 2022. 8

[5] Rohan Chabra, Jan Eric Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. *arXiv preprint arXiv:2003.10983*, 2020. 2

[6] Lucy Chai, Michael Gharbi, Eli Shechtman, Phillip Isola, and Richard Zhang. Any-resolution training for high-resolution image synthesis. *arXiv preprint arXiv:2204.07156*, 2022. 2, 4, 5, 8

[7] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. Super-resolution through neighbor embedding. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, pages I–I. IEEE, 2004. 2

[8] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8628–8638, 2021. 1, 2, 4, 5, 6

[9] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 2

[10] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. 7

[11] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 2

[12] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pages 184–199. Springer, 2014. 2

[13] Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yanjing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen, and Baochang Zhang. Implicit diffusion models for continuous super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10021–10030, 2023. 3

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2, 4

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1, 2, 3, 4

[17] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23:47–1, 2022. 1, 2

[18] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1575–1584, 2019. 2

[19] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020. 2

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 4

[21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 2

[22] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021. 2

[23] Zhifeng Kong and Wei Ping. On fast sampling of diffusion probabilistic models. *arXiv preprint arXiv:2106.00132*, 2021. 2

[24] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017. 2

[25] Jaewon Lee and Kyong Hwan Jin. Local texture estimator for implicit representation function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1929–1938, 2022. 2, 5, 6

[26] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 2

[27] 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. *arXiv preprint arXiv:2206.00927*, 2022. 2

[28] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022. 2

[29] 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 *International Conference on Learning Representations*, 2021. 2

[30] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 2

[31] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *arXiv preprint arXiv:2003.08934*, 2020. 2

[32] Alexander Mordvintsev, Nicola Pezzotti, Ludwig Schubert, and Chris Olah. Differentiable image parameterizations. *Distill*, 3(7):e12, 2018. 2

[33] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 2

[34] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 1, 2

[35] Evangelos Ntavelis, Mohamad Shahbazi, Iason Kastanis, Radu Timofte, Martin Danelljan, and Luc Van Gool. Arbitrary-scale image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11533–11542, 2022. 2

[36] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 2

[37] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. *arXiv preprint arXiv:2003.04618*, 2020. 2

[38] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 7

[40] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1

[41] 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*, pages 10684–10695, 2022. 1, 2, 3, 4, 5, 6, 7

[42] 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*, pages 1–10, 2022. 2

[43] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 1

[44] 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*, 2022. 2

[45] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2304–2314, 2019. 2

[46] Robin San-Roman, Eliya Nachmani, and Lior Wolf. Noise estimation for generative diffusion models. *arXiv preprint arXiv:2104.02600*, 2021. 2

[47] Abhishek Sinha*, Jiaming Song*, Chenlin Meng, and Stefano Ermon. D2c: Diffusion-denoising models for few-shot conditional generation. *arXiv preprint arXiv:2106.06819*, 2021. 2

[48] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, pages 1121–1132, 2019. 2

[49] Ivan Skorokhodov, Savva Ignatyev, and Mohamed Elhoseiny. Adversarial generation of continuous images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10753–10764, 2021. 2

[50] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 1, 2

[51] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[52] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

[53] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based

generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 2

[54] Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. *arXiv preprint arXiv:2111.08005*, 2021. 2

[55] Kenneth O Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2):131–162, 2007. 2

[56] Radu Timofte, Vincent De Smet, and Luc Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *Proceedings of the IEEE international conference on computer vision*, pages 1920–1927, 2013. 2

[57] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021. 1, 2

[58] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1905–1914, 2021. 2, 5, 6

[59] Jingyu Yang, Sheng Shen, Huanjing Yue, and Kun Li. Implicit transformer network for screen content image continuous super-resolution. *Advances in Neural Information Processing Systems*, 34:13304–13315, 2021. 2, 5, 6

[60] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 2, 4

[61] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018. 2

[62] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 2