Image Super-Resolution and Generation via Iterative Refinement

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, Mohammad Norouzi
Google Brain

Abstract

We present SR3, an approach to image Super-Resolution via Repeated Refinement. SR3 adapts denoising diffusion probabilistic models [13, 41] to conditional image generation, and performs super-resolution through a stochastic denoising process. Inference starts with pure Gaussian noise and iteratively refines the noisy output using a U-Net model trained on denoising. SR3 exhibits strong performance on super-resolution tasks at various magnification factors, on faces and natural images. We conduct human evaluation on a standard 8× face super-resolution task on CelebA-HQ, comparing with SOTA GAN methods. SR3 achieves a fool rate close to 50%, suggesting photo-realistic outputs, while GAN baselines do not exceed 34%. We further show the effectiveness of SR3 models in cascaded generation, where generative models are chained with super-resolution models, yielding competitive FID scores on ImageNet.

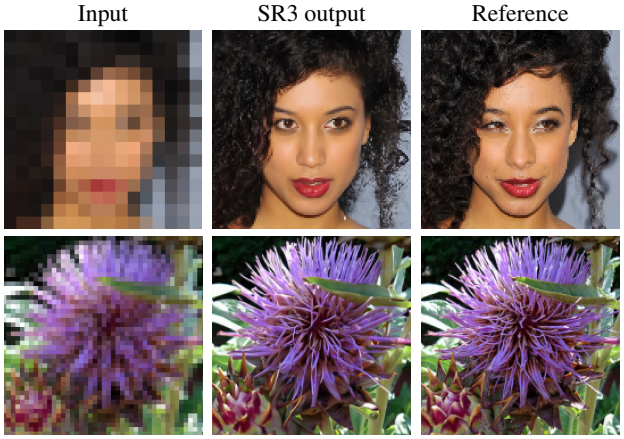

Figure 1: Two representative SR3 outputs: (top) 8× face super-resolution at 16×16→128×128 pixels (bottom) 4× natural image super-resolution at 64×64→256×256 pixels.

1. Introduction

Single-image super-resolution is the process of generating a high-resolution image that is consistent with an input low-resolution image. It falls under the broad family of image-to-image translation tasks, including colorization, in-painting, and de-blurring. Like many such inverse problems, image super-resolution is challenging because multiple output images may be consistent with a single input image, and the conditional distribution of output images given the input typically does not conform well to simple parametric distributions, *e.g.,* a multivariate Gaussian. Accordingly, while simple regression-based methods with feedforward convolutional nets may work for super-resolution at low magnification ratios, they often lack the high-fidelity details needed for high magnification ratios.

Deep generative models have seen success in learning complex empirical distributions of images (*e.g.,* [45, 49]). Autoregressive models [26, 27], variational autoencoders (VAEs) [19, 46], Normalizing Flows (NFs) [9, 18], and GANs [11, 15, 30] have shown convincing image generation results and have been applied to conditional tasks such as image super-resolution [6, 7, 20, 23, 28]. However, these approaches often suffer from various limitations; *e.g.,* autoregressive models are prohibitively expensive for high-resolution image generation, NFs and VAEs often yield sub-optimal sample quality, and GANs require carefully designed regularization and optimization tricks to tame optimization instability [2, 12] and mode collapse [24, 32].

We propose SR3 (Super-Resolution via Repeated Refinement), a new approach to conditional image generation, inspired by recent work on Denoising Diffusion Probabilistic Models (DDPM) [13, 40], and denoising score matching [13, 42]. SR3 works by learning to transform a standard normal distribution into an empirical data distribution through a sequence of refinement steps, resembling Langevin dynamics. The key is a U-Net architecture [36] that is trained with a denoising objective to iteratively remove various levels of noise from the output. We adapt DDPMs to *conditional* image generation by proposing a simple and effective modification to the U-Net architecture. In contrast to GANs that require inner-loop maximization, we minimize a well-defined loss function. Unlike autoregressive models, SR3 uses a constant number of inference steps regardless of output resolution.

SR3 works well across a range of magnification factors and input resolutions. SR3 models can also be cascaded,

e.g., going from 64×64 to 256×256, and then to 1024×1024. Cascading models allows one to independently train a few small models rather than a single large model with a high magnification factor. We find that chained models enable more efficient inference, since directly generating a high-resolution image requires more iterative refinement steps for the same quality. We also find that one can chain an unconditional generative model with SR3 models to unconditionally generate high-fidelity images. Unlike existing work that focuses on specific domains (*e.g.,* faces), we show that SR3 is effective on both faces and natural images.

Automated image quality scores like PSNR and SSIM do not reflect human preference well when the input resolution is low and the magnification ratio is large (*e.g.,* [6, 7, 23]). These quality scores often penalize synthetic high-frequency details, such as hair texture, because synthetic details do not perfectly align with the reference details. We resort to human evaluation to compare the quality of super-resolution methods. We adopt a 2-alternative forced-choice (2AFC) paradigm in which human subjects are shown a low-resolution input and are required to select between a model output and a ground truth image (*cf.* [54]). Based on this study, we calculate *fool rate* scores that capture both image quality and the consistency of model outputs with low-resolution inputs. Experiments demonstrate that SR3 achieves a significantly higher fool rate than SOTA GAN methods [6, 23] and a strong regression baseline.

Our key contributions are summarized as:

- We adapt denoising diffusion models to conditional image generation. Our method, *SR3*, is an approach to image super-resolution via iterative refinement.
- SR3 proves effective on face and natural image super-resolution at different magnification factors. On a standard 8× face super-resolution task, SR3 achieves a human fool rate close to 50%, outperforming FSRGAN [6] and PULSE [23] that achieve fool rates of at most 34%.
- We demonstrate unconditional and class-conditional generation by cascading a 64×64 image synthesis model with SR3 models to progressively generate 1024×1024 unconditional faces in 3 stages, and 256×256 class-conditional ImageNet samples in 2 stages. Our class conditional ImageNet samples attain competitive FID scores.

2. Conditional Denoising Diffusion Model

We are given a dataset of input-output image pairs, denoted $\mathcal{D} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^N$, which represent samples drawn from an unknown conditional distribution $p(\boldsymbol{y} \mid \boldsymbol{x})$. This is a one-to-many mapping in which many target images may be consistent with a single source image. We are interested in learning a parametric approximation to $p(\boldsymbol{y} \mid \boldsymbol{x})$ through a *stochastic* iterative refinement process that maps a source image $\boldsymbol{x}$ to a target image $\boldsymbol{y} \in \mathbb{R}^d$. We approach this problem by adapting the denoising diffusion probabilistic (DDPM) model of [13, 40] to *conditional* image generation.

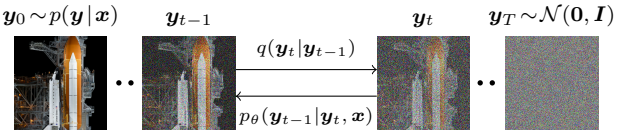

Figure 2: Depiction of the forward diffusion process q (left to right) which gradually adds Gaussian noise to the target image, and the reverse inference process p (right to left) which iteratively denoises the target image. (The input image $\boldsymbol{x}$ is not shown.)

The conditional DDPM model generates a target image $\boldsymbol{y}_0$ in T refinement steps. Starting with a pure noise image $\boldsymbol{y}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, the model iteratively refines the image through successive iterations $(\boldsymbol{y}_{T-1}, \boldsymbol{y}_{T-2}, \ldots, \boldsymbol{y}_0)$ according to learned, conditional transition distributions $p_\theta(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_t, \boldsymbol{x})$ such that $\boldsymbol{y}_0 \sim p(\boldsymbol{y}|\boldsymbol{x})$ (see Figure 2). The distributions of intermediate images in the inference chain are defined in terms of a *forward* diffusion process that gradually adds Gaussian noise to the signal via a fixed Markov chain, denoted $q(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1})$.

The goal of our model is to reverse the Gaussian diffusion process by iteratively recovering signal from noise through a reverse Markov chain. We learn this reverse chain using a neural denoising model f_θ that takes as input a source image and a noisy target image and estimates the noise. In what follows, we give an overview of the Gaussian diffusion process, and then discuss how the denoising model f_θ is trained and used for inference.

2.1. Gaussian Diffusion Process

Following [13, 40], we first define a *forward* Markovian diffusion process q that gradually adds Gaussian noise to a high-resolution image $\boldsymbol{y}_0$ over T iterations:

$$q(\boldsymbol{y}_{1:T} \mid \boldsymbol{y}_0) = \prod_{t=1}^{T} q(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}), \quad (1)$$

$$q(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}) = \mathcal{N}(\boldsymbol{y}_t \mid \sqrt{\alpha_t}\,\boldsymbol{y}_{t-1}, (1-\alpha_t)\boldsymbol{I}), \quad (2)$$

where the scalar parameters $\alpha_{1:T}$ are hyper-parameters, subject to $0 < \alpha_t < 1$, which determine the variance of the noise added at each iteration. Note that $\boldsymbol{y}_{t-1}$ is attenuated by $\sqrt{\alpha_t}$ to ensure that the variance of the random variables remains bounded as $t \to \infty$. For instance, if the variance of $\boldsymbol{y}_{t-1}$ is 1, then the variance of $\boldsymbol{y}_t$ is also 1.

Importantly, one can characterize the distribution of $\boldsymbol{y}_t$ given $\boldsymbol{y}_0$ by marginalizing out the intermediate steps as

$$q(\boldsymbol{y}_t \mid \boldsymbol{y}_0) = \mathcal{N}(\boldsymbol{y}_t \mid \sqrt{\gamma_t}\,\boldsymbol{y}_0, (1-\gamma_t)\boldsymbol{I}), \quad (3)$$

where $\gamma_t = \prod_{i=1}^{t} \alpha_i$. Furthermore, with some algebraic manipulation and completing the square, one can derive the

Algorithm 1 Training a denoising model f_θ

repeat
  $(\boldsymbol{x}, \boldsymbol{y}_0) \sim p(\boldsymbol{x}, \boldsymbol{y})$
  $\gamma \sim p(\gamma)$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take a gradient descent step on
    $\nabla_\theta \left\| f_\theta(\boldsymbol{x}, \sqrt{\gamma}\boldsymbol{y}_0 + \sqrt{1-\gamma}\boldsymbol{\epsilon}, \gamma) - \boldsymbol{\epsilon} \right\|_p^p$
until converged

Algorithm 2 Inference in T iterative refinement steps

$\boldsymbol{y}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
  $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\boldsymbol{z} = \mathbf{0}$
  $\boldsymbol{y}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{y}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t) \right) + \sqrt{1-\alpha_t}\boldsymbol{z}$
end for
return $\boldsymbol{y}_0$

posterior distribution of $\boldsymbol{y}_{t-1}$ given $(\boldsymbol{y}_0, \boldsymbol{y}_t)$ as

$$
\begin{aligned}
q(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_0, \boldsymbol{y}_t) &= \mathcal{N}(\boldsymbol{y}_{t-1} \mid \boldsymbol{\mu}, \sigma^2 \boldsymbol{I}) \\
\boldsymbol{\mu} &= \frac{\sqrt{\gamma_{t-1}}\,(1-\alpha_t)}{1-\gamma_t}\,\boldsymbol{y}_0 + \frac{\sqrt{\alpha_t}\,(1-\gamma_{t-1})}{1-\gamma_t}\boldsymbol{y}_t \\
\sigma^2 &= \frac{1-\gamma_t}{(1-\gamma_{t-1})(1-\alpha_t)}\,.
\end{aligned} \tag{4}
$$

This posterior distribution is helpful when parameterizing the reverse chain and formulating a variational lower bound on the log-likelihood of the reverse chain. We next discuss how one can learn a neural network to reverse this Gaussian diffusion process.

2.2. Optimizing the Denoising Model

To help reverse the diffusion process we optimize a neural denoising model f_θ that takes as input a noisy image $\widetilde{\boldsymbol{y}}$,

$$
\widetilde{\boldsymbol{y}} = \sqrt{\gamma}\,\boldsymbol{y}_0 + \sqrt{1-\gamma}\,\boldsymbol{\epsilon}\,, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})\,, \tag{5}
$$

and aims to recover the noiseless image $\boldsymbol{y}_0$. This definition of a noisy image is compatible with the marginal distribution of noisy images at different steps of the forward diffusion process in (3). To make the denoising model aware of the level of noise, we provide γ as an additional input to f_θ, as suggested by prior work [13, 42].

The denoising model $f_\theta(\boldsymbol{x}, \widetilde{\boldsymbol{y}}, \gamma)$ takes as input a source image $\boldsymbol{x}$, a noisy target image $\widetilde{\boldsymbol{y}}$, and the sufficient statistics for the variance of the noise γ, and is trained to predict the noise vector $\boldsymbol{\epsilon}$. The proposed objective function for training f_θ is expressed as

$$
\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})}\mathbb{E}_{\boldsymbol{\epsilon},\gamma}\left\| f_\theta(\boldsymbol{x}, \underbrace{\sqrt{\gamma}\,\boldsymbol{y}_0 + \sqrt{1-\gamma}\,\boldsymbol{\epsilon}}_{\widetilde{\boldsymbol{y}}}, \gamma) - \boldsymbol{\epsilon} \right\|_p^p, \tag{6}
$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, $(\boldsymbol{x}, \boldsymbol{y})$ is sampled from the training dataset, $p \in \{1, 2\}$, and $\gamma \sim p(\gamma)$. The distribution of γ has a big impact on the quality of the model and the generated outputs. We discuss our choice of $p(\gamma)$ in Section 2.4.

Instead of regressing the output of f_θ to $\boldsymbol{\epsilon}$, as in (6), one can also regress the output of f_θ to $\boldsymbol{y}_0$. Given γ and $\widetilde{\boldsymbol{y}}$, the values of $\boldsymbol{\epsilon}$ and $\boldsymbol{y}_0$ can be derived from each other deterministically, but changing the regression target has an impact on the scale of the loss function. We expect both of these variants to work reasonably well if $p(\gamma)$ is modified to account for the scale of the loss function. Further investigation of the loss function used for training the denoising model is an interesting avenue for future research in this area.

2.3. Inference via Iterative Refinement

Inference under our model is defined as a *reverse* Markovian process, which goes in the reverse direction of the forward diffusion process, starting from Gaussian noise $\boldsymbol{y}_T$:

$$
\begin{aligned}
p_\theta(\boldsymbol{y}_{0:T}|\boldsymbol{x}) &= p(\boldsymbol{y}_T) \textstyle\prod_{t=1}^{T} p_\theta(\boldsymbol{y}_{t-1}|\boldsymbol{y}_t, \boldsymbol{x}) && (7)\\
p_\theta(\boldsymbol{y}_T) &= \mathcal{N}(\boldsymbol{y}_T \mid \mathbf{0}, \boldsymbol{I}) && (8)\\
p_\theta(\boldsymbol{y}_{t-1}|\boldsymbol{y}_t, \boldsymbol{x}) &= \mathcal{N}(\boldsymbol{y}_{t-1} \mid \mu_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t), \sigma_t^2 \boldsymbol{I})\,. && (9)
\end{aligned}
$$

We define the inference process in terms of isotropic Gaussian conditional distributions, $p_\theta(\boldsymbol{y}_{t-1}|\boldsymbol{y}_t, \boldsymbol{x})$, which are learned. If the noise variance of the forward process steps are set as small as possible, *i.e.,* $\alpha_{1:T} \approx 1$, the optimal reverse process $p(\boldsymbol{y}_{t-1}|\boldsymbol{y}_t, \boldsymbol{x})$ will be approximately Gaussian [40]. Accordingly, our choice of Gaussian conditionals in the inference process (9) can provide a reasonable fit to the true reverse process. Meanwhile, $1-\gamma_T$ should be large enough so that $\boldsymbol{y}_T$ is approximately distributed according to the prior $p(\boldsymbol{y}_T) = \mathcal{N}(\boldsymbol{y}_T|\mathbf{0}, \boldsymbol{I})$, allowing the sampling process to start at pure Gaussian noise.

Recall that the denoising model f_θ is trained to estimate $\boldsymbol{\epsilon}$, given any noisy image $\widetilde{\boldsymbol{y}}$ including $\boldsymbol{y}_t$. Accordingly, given $\boldsymbol{y}_t$, we approximate $\boldsymbol{y}_0$ by rearranging the terms in (5) as

$$
\hat{\boldsymbol{y}}_0 = \frac{1}{\sqrt{\gamma_t}} \left(\boldsymbol{y}_t - \sqrt{1-\gamma_t}\, f_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t) \right)\,. \tag{10}
$$

Following the formulation of [13], we substitute our estimate $\hat{\boldsymbol{y}}_0$ into the posterior distribution of $q(\boldsymbol{y}_{t-1}|\boldsymbol{y}_0, \boldsymbol{y}_t)$ in (4) to parameterize the mean of $p_\theta(\boldsymbol{y}_{t-1}|\boldsymbol{y}_t, \boldsymbol{x})$ as

$$
\mu_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t) = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{y}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t) \right)\,, \tag{11}
$$

and we set the variance of $p_\theta(\boldsymbol{y}_{t-1}|\boldsymbol{y}_t, \boldsymbol{x})$ to $(1-\alpha_t)$, a default given by the variance of the forward process [13].

Following this parameterization, each iteration of iterative refinement under our model takes the form,

$$
\boldsymbol{y}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{y}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t) \right) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t\,,
$$

where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. This resembles one step of Langevin dynamics with f_θ providing an estimate of the gradient of the data log-density. We justify the choice of the training objective in (6) for the probabilistic model outlined in (9) from a variational lower bound perspective and a denoising score-matching perspective in Appendix C.

2.4. SR3 Model Architecture and Noise Schedule

The SR3 architecture is similar to the U-Net found in DDPM [13], with modifications adapted from [44]; we replace the original DDPM residual blocks with residual blocks from BigGAN [3], and we re-scale skip connections by $\frac{1}{\sqrt{2}}$. We also increase the number of residual blocks, and the channel multipliers at different resolutions (see Appendix B for details). To condition the model on the input $\boldsymbol{x}$, we up-sample the low-resolution image to the target resolution using bicubic interpolation. The result is concatenated with $\boldsymbol{y}_t$ along the channel dimension. We experimented with more sophisticated methods of conditioning, such as using FiLM [29], but we found that the simple concatenation yielded similar generation quality.

For our training noise schedule, we follow [5], and use a piece wise distribution for γ, $p(\gamma) = \sum_{t=1}^{T} \frac{1}{T} U(\gamma_{t-1}, \gamma_t)$. Specifically, during training, we first uniformly sample a time step $t \sim \{0, ..., T\}$ followed by sampling $\gamma \sim U(\gamma_{t-1}, \gamma_t)$. We set $T = 2000$ in all our experiments.

Prior work of diffusion models [13, 44] require 1-2k diffusion steps during inference, making generation slow for large target resolution tasks. We adapt techniques from [5] to enable more efficient inference. Our model conditions on γ directly (vs t as in [13]), which allows us flexibility in choosing number of diffusion steps, and the noise schedule during inference. This has been demonstrated to work well for speech synthesis [5], but has not been explored for images. For efficient inference we set the maximum inference budget to 100 diffusion steps, and hyper-parameter search over the inference noise schedule. This search is inexpensive as we only need to train the model once [5]. We use FID on held out data to choose the best noise schedule, as we found PSNR did not correlate well with image quality.

3. Related Work

SR3 is inspired by recent work on deep generative models and recent learning-based approaches to super-resolution.

Generative Models. Autoregressive models (ARs) [47, 38] can model exact data log likelihood, capturing rich distributions. However, their sequential generation of pixels is expensive, limiting application to low-resolution images. Normalizing flows [34, 9, 18] improve on sampling speed while modelling the exact data likelihood, but the need for invertible parameterized transformations with a tractable Jacobian determinant limits their expressiveness. VAEs [19, 35] offer fast sampling, but tend to underperform GANs and ARs in image quality [46]. Generative Adversarial Networks (GANs) [11] are popular for class conditional image generation and super-resolution. Nevertheless, the inner-outer loop optimization often requires tricks to stabilize training [2, 12], and conditional tasks like super-resolution usually require an auxiliary consistency-based loss to avoid mode collapse [20]. Cascades of GAN models have been used to generate higher resolution images [8].

Score matching [14] models the gradient of the data log-density with respect to the image. Score matching on noisy data, called denoising score matching [50], is equivalent to training a denoising autoencoder, and to DDPMs [13]. Denoising score matching over multiple noise scales with Langevin dynamics sampling from the learned score functions has recently been shown to be effective for high quality unconditional image generation [42, 13]. These models have also been generalized to continuous time [44]. Denoising score matching and diffusion models have also found success in shape generation [4], and speech synthesis [5]. We extend this method to super-resolution, with a simple learning objective, a constant number of inference generation steps, and high quality generation.

Super-Resolution. Much of the early work on super-resolution is regression based and trained with an MSE loss [1, 6, 10, 17]. As such, they effectively estimate the posterior mean, yielding blurry images when the posterior is multi-modal [20, 23]. Our regression baseline defined below is also a one-step regression model trained with MSE (cf. [1, 17]), but with a large U-Net architecture. SR3, by comparison, relies on a series of iterative refinement steps, each of which is trained with a regression loss. This difference permits our iterative approach to capture richer distributions. Further, rather than estimating the posterior mean, SR3 generates samples from the target posterior.

Autoregressive models have been used successfully for super-resolution and cascaded up-sampling [7, 22, 48, 28]. Nevertheless, the expensive of inference limits their applicability to low-resolution images. SR3 can generate high-resolution images, e.g., 1024×1024, but with a constant number of refinement steps (often no more than 100).

Normalizing flows have been used for super-resolution with a multi-scale approach [53]. They are capable of generating 1024×1024 images due in part to their efficient inference process. But SR3 uses a series of reverse diffusion steps to transform a Gaussian distribution to an image distribution while flows require a deep and invertible network.

GAN-based super-resolution methods have also found considerable success [15, 20, 23, 52]. FSRGAN [6] and PULSE [23] in particular have demonstrated high quality face super-resolution results. However, many such methods tend to focus specifically on faces [6, 23], while in our work we demonstrate the effectiveness of SR3 on both faces and large-scale natural image datasets.

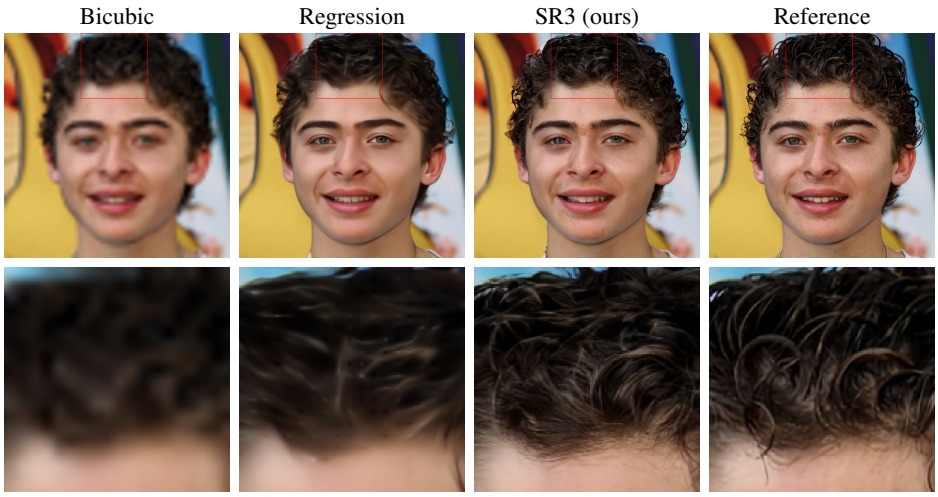

Figure 3: Results of a SR3 model (64×64 → 512×512), trained on FFHQ, and applied to a test image from CelebA-HQ.

4. Experiments

We assess the effectiveness of SR3 models in super-resolution on faces, natural images, and synthetic images obtained from a low-resolution generative model. The latter enables high-resolution image synthesis using model cascades. We compare SR3 with recent methods such as FSRGAN [6] and PULSE [23] using human evaluation, and report FID for various tasks. We also compare to a regression baseline model that shares the same architecture as SR3, but is trained with a MSE loss. Our experiments include:

- Face super-resolution at 16×16 → 128×128 and 64×64 → 512×512 trained on FFHQ and evaluated on CelebA-HQ.
- Natural image super-resolution at $64 \times 64 \rightarrow 256 \times 256$ pixels on ImageNet [37].
- Unconditional 1024×1024 face generation by a cascade of 3 models, and class-conditional 256×256 ImageNet image generation by a cascade of 2 models.

Datasets: We follow previous work [23], training face super-resolution models on Flickr-Faces-HQ (FFHQ) [16] and evaluating on CelebA-HQ [15]. For natural image super-resolution, we train on ImageNet 1K [37] and use the dev split for evaluation. We train unconditional face and class-conditional ImageNet generative models using DDPM on the same datasets discussed above. For training and testing, we use low-resolution images that are downsampled using bicubic interpolation with anti-aliasing enabled. For ImageNet, we discard images where the shorter side is less than the target resolution. We use the largest central crop like [3], which is then resized to the target resolution using area resampling as our high resolution image.

Training Details: We train all of our SR3 and regression models for 1M training steps with a batch size of 256. We choose a checkpoint for the regression baseline based on peak-PSNR on the held out set. We do not perform any checkpoint selection on SR3 models and simply select the latest checkpoint. Consistent with [13], we use the Adam Optimizer with a linear warmup schedule over 10k training steps, followed by a fixed learning rate of 1e-4 for SR3 models and 1e-5 for regression models. We use 625M parameters for our $64 \times 64 \rightarrow \{256 \times 256, 512 \times 512\}$ models, 550M parameters for the 16×16 → 128×128 models, and 150M parameters for 256×256 → 1024×1024 model. We use a dropout rate of 0.2 for 16×16 → 128×128 models super-resolution, but otherwise, we do not use dropout. (See Appendix B for task specific architectural details.)

4.1. Qualitative Results

Face Images: Figure 3 shows the output of a face super-resolution model (64×64 → 512×512) on a test image. The bottom row shows a zoomed in view of a patch (red outline) from the full image. With the 8× magnification factor one can clearly see the detailed structure inferred. Note that, because of the large magnification factor, there are many plausible outputs, consequently we do not expect the output to exactly match the reference image. This is evident in the hair region in the bottom row.

Natural Images: Figure 4 gives examples of super-resolution ntural images for 64×64 → 256×256 on the ImageNet dev set. The baseline Regression model generates images that are faithful to the inputs, but blurry and lack detail. By comparison, SR3 produces sharp images with more detail; this is most evident in the zoomed-in patches. For more samples see the supplementary material.

4.2. Benchmark Comparison

Evaluations metrics: Previous work [6, 7, 23] observed that conventional automated evaluation measures like peak-SNR and SSIM [51] do not correlate well with human perception when the input resolution is low and the magnification factor is large. This is not surprising because these metrics tend to penalize any synthetic high-frequency detail that

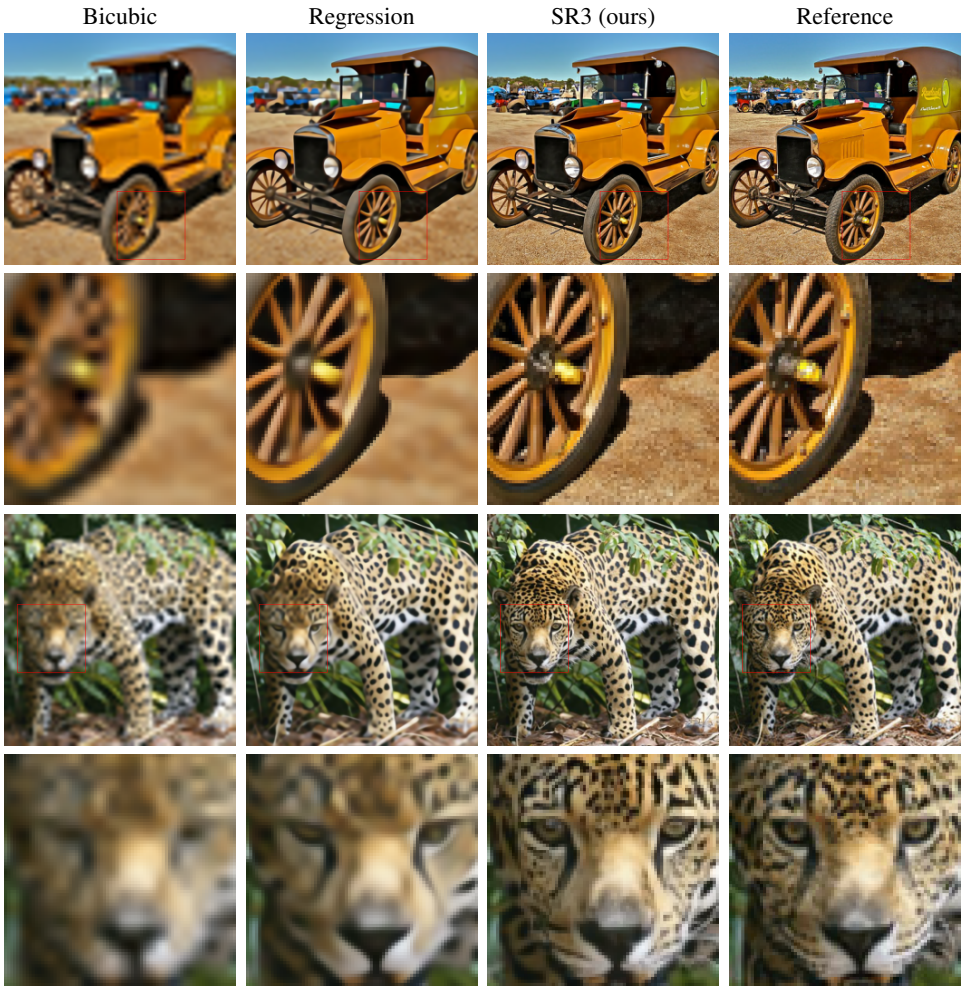

Figure 4: Results of a SR3 model (64×64 → 256×256), trained on ImageNet and evaluated on two ImageNet test images.

Model	FID	IS	PSNR	SSIM
Reference	1.9	240.8	-	-
Regression	15.2	121.1	**27.9**	**0.801**
SR3	**5.2**	**180.1**	26.4	0.762

Table 1: Performance comparison between SR3 and Regression baseline on natural image super-resolution using standard metrics computed on the ImageNet validation set.

is not perfectly aligned with the target image. Since generating perfectly aligned high-frequency details, *e.g.,* the exact same hair strands in Figure 3 and identical leopard spots in Figure 4, is almost impossible, PSNR and SSIM tend to prefer MSE regression-based techniques that are extremely conservative with high-frequency details. Table 1 confirms that for ImageNet super-resolution (64×64 → 256×256), the outputs of SR3 achieve higher sample quality scores (FID and IS), but worse PSNR and SSIM than regression.

In this work, we are interested in photo-realistic super-resolution with large magnification factors. Accordingly, we resort to direct human evaluation. While mean opinion score (MOS) is commonly used to measure image quality in this context, forced choice pairwise comparison has been found to be a more reliable method for such subjective quality assessments [21]. Furthermore, standard MOS studies do not capture consistency between low-resolution inputs and high-resolution outputs.

Human Evaluation (2AFC): We use a 2-alternative forced-choice (2AFC) paradigm to measure how well humans can discriminate true images from those generated from a model. In Task-1 subjects were shown a low resolution input in between two high-resolution images, one being the real image (ground truth), and the other generated from the model. Subjects were asked *"Which of the two images is a better high quality version of the low resolution image in the middle?"* This task takes into account both image quality and consistency with the low resolution input. Task-2 is similar to Task-1, except that the low-resolution image was not shown, so subjects only had to select the image that was more photo-realistic. They were asked *"Which*

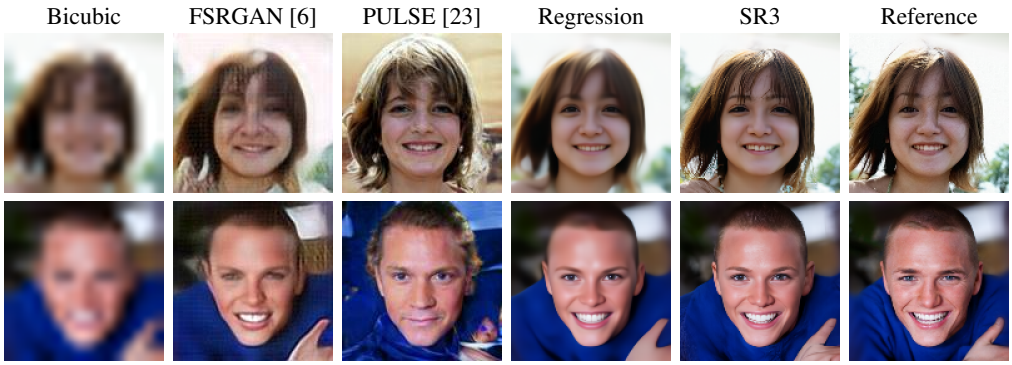

Figure 5: Comparison of different methods on the $16\times16 \rightarrow 128\times128$ face super-resolution task.

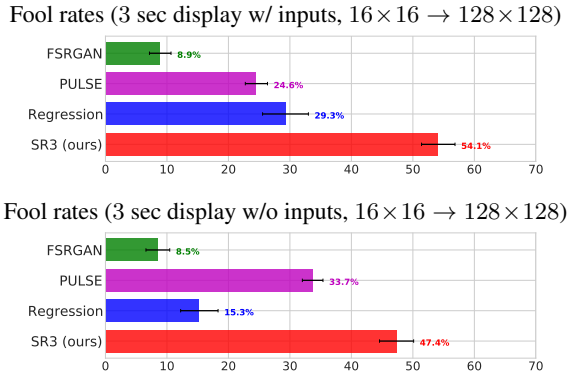

Figure 6: Face super-resolution human fool rates (higher is better, photo-realistic samples yield a fool rate of 50%). Outputs of 4 models are compared against ground truth. (top) Subjects are shown low-resolution inputs. (bottom) Inputs are not shown.

image would you guess is from a camera?" Subjects viewed images for 3 seconds before responding, in both tasks.

The subject *fool rate* is the fraction of trials on which a subject selects the model output over ground truth. Our fool rates for each model are based on 50 subjects, each of whom were shown 50 of the 100 images in the test set. Figure 6 shows the fool rates for Task-1 (top), and for Task-2 (bottom). In both experiments, the fool rate of SR3 is close to 50%, indicating that SR3 produces images that are both photo-realistic and faithful to the low-resolution inputs.

The fool rates for FSRGAN and PULSE in Task-1 are lower than the Regression baseline and SR3. We speculate that the PULSE optimization has failed to converge to high resolution images sufficiently close to the inputs. Indeed, when asked solely about image quality in Task-2 (Fig. 6 (bottom)), the PULSE fool rate increases significantly.

The fool rate for the Regression baseline is lower in Task-2 (Fig. 6 (bottom)) than Task-1. The regression model tends to generate images that are blurry, but nevertheless faithful to the low resolution input. We speculate that in Task-1, given the inputs, subjects are influenced by consistency, while in Task-2, ignoring consistency, they instead focus on image sharpness.

We conduct similar human evaluation studies on natural images comparing SR3 and the regression baseline on ImageNet, which demonstrate that SR3 consistently achieves a higher fool rate (see Appendix A.1).

To further appreciate the experimental results it is useful to visually compare outputs of different models on the same inputs, as in Figure 5. FSRGAN exhibits distortion in face region and lacks detail in the hair (e.g., top row). It also produces some unusual artifacts in the background (e.g., the hand in the bottom row). PULSE often produces images that differ significantly from the input image, both in the shape of the face and the background, presumably due to failure of the optimization to find a sufficiently good minima. As noted above, our Regression baseline produces results consistent to the input, however they are typically quite blurry. By comparison, the SR3 results are consistent with the input and contain more detailed image structure.

4.3. Cascaded High-Resolution Image Synthesis

We study *cascaded* image generation, where SR3 models at different scales are chained together with unconditional generative models, enabling high-resolution image synthesis. Cascaded generation allows one to train different models in parallel, and each model in the cascade solves a simpler task, requiring fewer parameters and less computation for training. Inference with cascaded models is also more efficient, especially for iterative refinement models. With cascaded generation we found it effective to use more refinement steps at low-resolutions, and fewer steps at higher resolutions. This was much more efficient than generating directly at high resolution without sacrificing image quality.

We train a DDPM [13] model for unconditional 64×64 face generation. Samples from this model are then fed to two $4\times$ SR3 models, up-sampling to 256^2 and then to 1024^2 pixels. Synthetic high-resolution face samples are shown in Figure 7. In addition, we train a class-conditional 64×64 DDPM model on ImageNet using techniques from [25], and we pass its samples to a $4\times$ SR3 model yielding 256^2 pixels. The $4\times$ SR3 model is not conditioned on the class label. See Figure 8 for representative samples.

Figure 7: Synthetic 1024×1024 faces images. We first sample from an unconditional 64×64 diffusion model, then pass the samples through two 4× SR3 models, *i.e.,* 64×64 → 256×256 → 1024×1024. See supplementary material for more samples.

Figure 8: Synthetic 256×256 ImageNet images. We first draw a random label, then sample a 64×64 image from a class-conditional diffusion model, and apply a 4× SR3 model to obtain 256×256 images. See supplementary material for more samples.

Model	FID
Prior Work	
VQ-VAE-2 [33]	38.1
BigGAN (Truncation 0.5) [3]	14.4
BigGAN (Truncation 1.5) [3]	11.8
Our Work	
SR3 (Two Stage)	12.6

Table 2: FID scores for class-conditional 256×256 ImageNet.

Table 2 reports FID scores for the resulting class-conditional ImageNet samples. Our 2-stage model improves on VQ-VAE-2 [33], and marginally underperforms BigGAN [3] at a truncation factor of 1.5. Unlike BigGAN, our diffusion models do not provide a knob to control sample quality *vs.* sample diversity, and finding ways to do so is interesting avenue for future research. Nichol and Dhariwal [25] concurrently trained cascaded generation models using super-resolution conditioned on class labels (our super-resolution is not conditioned on class labels), and observed a similar trend in FID scores. The effectiveness of cascaded image generation indicates that SR3 models are robust to the precise distribution of inputs (*i.e.,* the specific form of anti-aliasing and downsampling).

5. Discussion and Conclusion

Bias is an important problem in all generative models. SR3 is no different, and suffers from bias issues. While in theory, our log-likelihood based objective is mode covering (*e.g.,* unlike some GAN-based objectives), we believe it is likely our diffusion-based models drop modes. We observed some evidence of mode dropping, the model consistently generates nearly the same image output during sampling (when conditioned on the same input). We also observed the model to generate very continuous skin texture in face super-resolution, dropping moles, pimples and piercings found in the reference. SR3 should not be used for any real world super-resolution tasks, until these biases are thoroughly understood and mitigated.

In conclusion, SR3 is an approach to image super-resolution via iterative refinement. SR3 can be used in a cascaded fashion to generate high resolution super-resolution images, as well as unconditional samples when cascaded with a unconditional model. We demonstrate SR3 on face and natural image super-resolution at high resolution and high magnification ratios (*e.g.,* 64×64→256×256 and 256×256→1024×1024). SR3 achieves a human fool rate close to 50%, suggesting photo-realistic outputs.

References

[1] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Image Super-resolution via Progressive Cascading Residual Network. In *CVPR*, 2018.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. In *arXiv*, 2017.

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[4] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning Gradient Fields for Shape Generation. In *ECCV*, 2020.

[5] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, and William Chan. WaveGrad: Estimating Gradients for Waveform Generation. In *ICLR*, 2021.

[6] Yu Chen, Ying Tai, Xiaoming Liu, Chunhua Shen, and Jian Yang. Fsrnet: End-to-end learning face super-resolution with facial priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2492–2501, 2018.

[7] Ryan Dahl, Mohammad Norouzi, and Jonathon Shlens. Pixel recursive super resolution. In *ICCV*, 2017.

[8] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. In *NIPS*, 2015.

[9] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. *arXiv:1605.08803*, 2016.

[10] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[11] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *NIPS*, 2014.

[12] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[14] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *JMLR*, 6(4), 2005.

[15] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.

[16] Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In *CVPR*, 2019.

[17] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016.

[18] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative Flow with Invertible 1x1 Convolutions. In *NIPS*, 2018.

[19] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. In *ICLR*, 2013.

[20] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *ICCV*, 2017.

[21] Rafał K Mantiuk, Anna Tomaszewska, and Radosław Mantiuk. Comparison of four subjective methods for image quality assessment. In *Computer graphics forum*, volume 31, pages 2478–2491. Wiley Online Library, 2012.

[22] Jacob Menick and Nal Kalchbrenner. Generating High Fidelity Images with Subscale Pixel Networks and Multidimensional Upscaling. In *ICLR*, 2019.

[23] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *CVPR*, 2020.

[24] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

[25] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *arXiv preprint arXiv:2102.09672*, 2021.

[26] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. *arXiv preprint arXiv:1609.03499*, 2016.

[27] Aäron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional Image Generation with PixelCNN Decoders. In *NIPS*, 2016.

[28] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, 2018.

[29] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning with a General Conditioning Layer . In *AAAI*, 2018.

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

[31] Martin Raphan and Eero P Simoncelli. Least squares estimation without priors or supervision. *Neural computation*, 23(2):374–420, 2011.

[32] Suman Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. *arXiv preprint arXiv:1905.10887*, 2019.

[33] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *arXiv preprint arXiv:1906.00446*, 2019.

[34] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538. PMLR, 2015.

[35] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015.

[37] 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*, 115(3):211–252, 2015.

[38] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications. In *International Conference on Learning Representations*, 2017.

[39] Saeed Saremi, Arash Mehrjou, Bernhard Schölkopf, and Aapo Hyvärinen. Deep Energy Estimator Networks. *arXiv preprint arXiv:1805.08306*, 2018.

[40] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pages 2256–2265. PMLR, 2015.

[41] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *ICML*, 2015.

[42] Yang Song and Stefano Ermon. Generative Modeling by Estimating Gradients of the Data Distribution. In *NeurIPS*, 2019.

[43] Yang Song and Stefano Ermon. Improved Techniques for Training Score-Based Generative Models. *arXiv preprint arXiv:2006.09011*, 2020.

[44] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *ICLR*, 2021.

[45] Ilya Sutskever, Oriol Vinyals, and Quoc Le. Sequence to Sequence Learning with Neural Networks. In *NIPS*, 2014.

[46] Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. In *NeurIPS*, 2020.

[47] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *International Conference on Machine Learning*, 2016.

[48] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with PixelCNN decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798, 2016.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *NIPS*, 2017.

[50] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

[51] Zhou Wang, A.C. Bovik, H.R. Sheikh., and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[52] Lingbo Yang, Chang Liu, Pan Wang, Shanshe Wang, Peiran Ren, Siwei Ma, and Wen Gao. HiFaceGAN: Face Renovation via Collaborative Suppression and Replenishment. In *arXiv*, 2020.

[53] Jason J. Yu, Konstantinos G. Derpanis, and Marcus A. Brubaker. Wavelet Flow: Fast Training of High Resolution Normalizing Flows. In *arXiv*, 2020.

[54] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.

A. Additional Results

This appendix includes more experimental results, more details about the formulation of the loss, and more examples of SR3 on faces, natural images, and on samples from unconditional generative models. We begin with quantitative experiments on the evaluation of ImageNet super resolution models using human subjects. This is followed by qualitative analysis of images with best and worst fool rates for PULSE [23] and SR3. We then show more examples of several models, augmenting the results shown in Figures 3, 4, 5, 7 and 8 in the main body of the paper.

A.1. Human Evaluation on ImageNet Super-Resolution

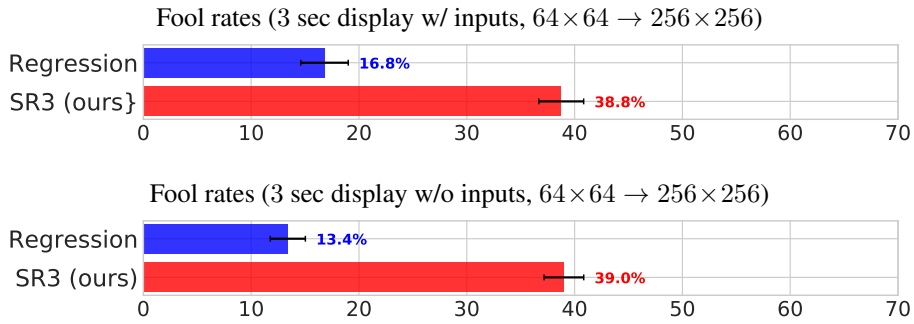

Figure A.1: ImageNet super-resolution fool rates (higher is better, photo-realistic samples yield a fool rate of 50%). SR3 and Regression outputs are compared against ground truth. (top) Subjects are shown low-resolution inputs. (bottom) Inputs are not shown.

Quantitative human evaluation is used to measure the quality of images generated by SR3 and other models. Sec. 4.2 reports results on face images. Here we report similar experiments for models applied to natural images. Models were trained on ImageNet for a $64\times64 \rightarrow 256\times256$ super-resolution task (for image samples, see Figs. 4 and A.6). 2AFC tasks like those in Sec. 4.2 were run on ImageNet test images. In Task-1 subjects were shown a low resolution input in between two high-resolution images, one being the real image and the other generated from the model. Subjects were asked *"Which of the two images is a better high quality version of the low resolution image in the middle?"* Task-2 is similar except the low-resolution image was not shown, so subjects only had to select the image that was more photo-realistic. They were asked *"Which image would you guess is from a camera?"* Subjects viewed images for 3 seconds before responding. For this experiment we only compare SR3 to our baseline Regression model, which performed well with faces compared to FSRGAN and PULSE (since FSRGAN is face specific, and we do not have a PULSE implementation).

Figure A.1 shows the results for Task-1 (top) and task-2 (bottom). The fool rate indicates the fraction of times that subjects selected the model output as the real image. In both tasks with natural images, SR3 achieves a human subject fool rate is close to 40%. Like the face image experiments in Fig. 6, here again we find that the Regression baseline yields a lower fool rate in Task-2, where the low resolution image is not shown. Again we speculate that this is a result of a somewhat simpler task (looking at 2 rather than 3 images), and the fact that subjects can focus solely on image artifacts, such as blurriness, without having to worry about consistency between model output and the low resolution input.

A.2. Images with the Lowest and Highest Fool Rates

In interpreting the fool rate results in Figure 6, it is interesting to inspect those images that maximize the fool rates for a given technique, as well as those images that minimize the fool rate. This provides insight into the nature of the problems that models exhibit, as well as cases in which the model outputs are good enough to regularly fool people.

In Figure A.2 we display the images with the lowest fool rates generated by PULSE [23] and SR3 for both Task-1 (the conditional task), and Task-2, (the unconditional task). In order to be consistent with our human study interface, we show the corresponding low resolution image only for Task-1. Notice that images from PULSE for which the fool rate is low have obvious distortions, and the fool rates are lower than 10% for both tasks. For SR3, by comparison, the images with the lowest fool rates are still reasonably good, with much higher fool rates of 14% and 19% in Task-1, and 21% and 26% in Task-2.

Figure A.3 shows images that best fool human subjects. In this case, it is interesting to note that the best fool rates for SR3 are 84% and 88%. The corresponding original images are somewhat noisy, and as a consequence, many subjects refer the SR3 outputs.

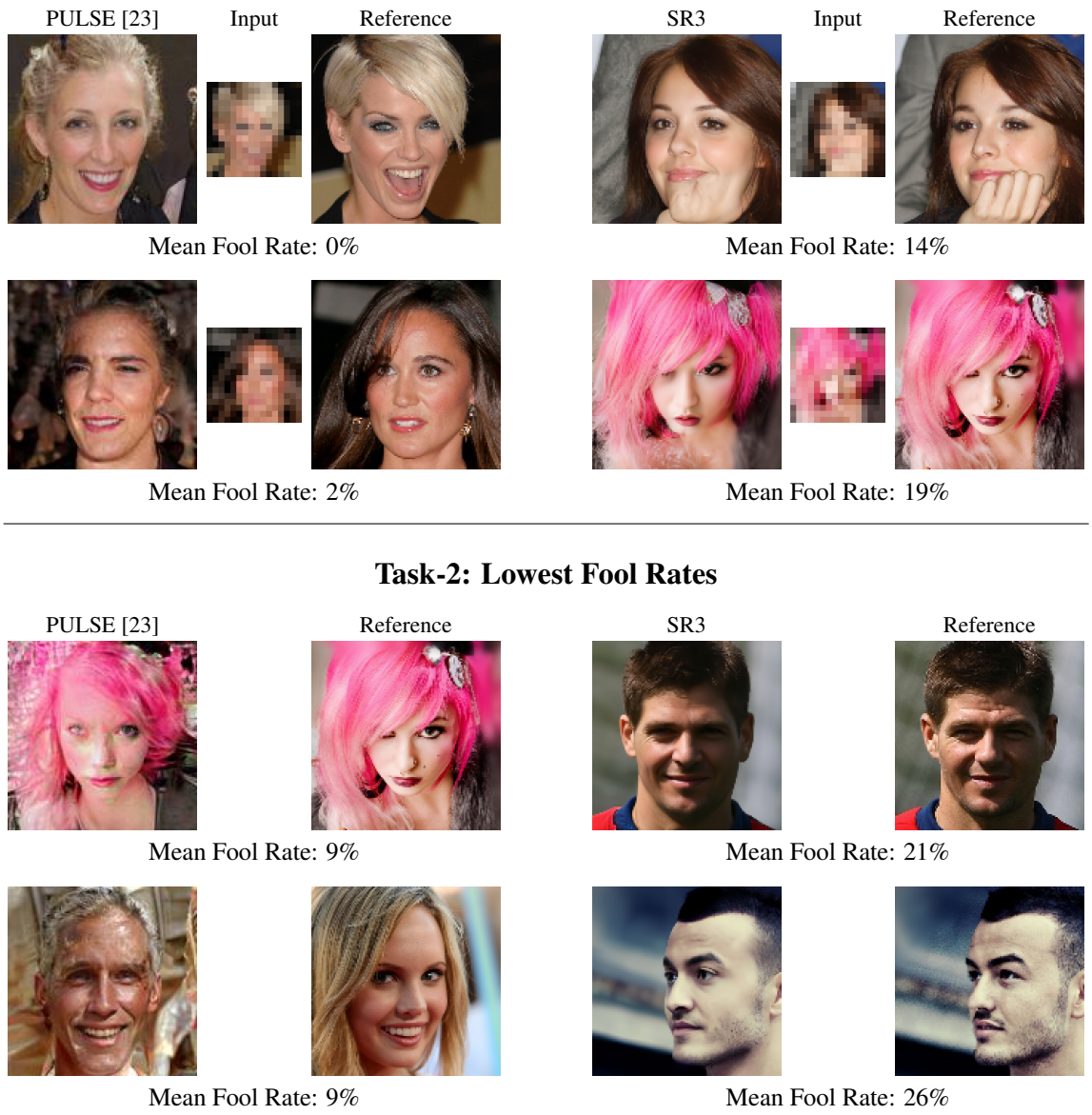

Figure A.2: The images with lowest fool rates, PULSE on the left and SR3 on the right. The first 2 rows correspond to the conditional human evaluation (Task-1), and the last 2 rows correspond to the unconditional human evaluation (Task-2). The image layout mimics the original interface of our human evaluation study. For Task-1, we include the low resolution input image, and for Task-2, we omit the low resolution image to be consistent with the human evaluation setting. We also report the Mean Fool Rate for each of the image pairs right below.

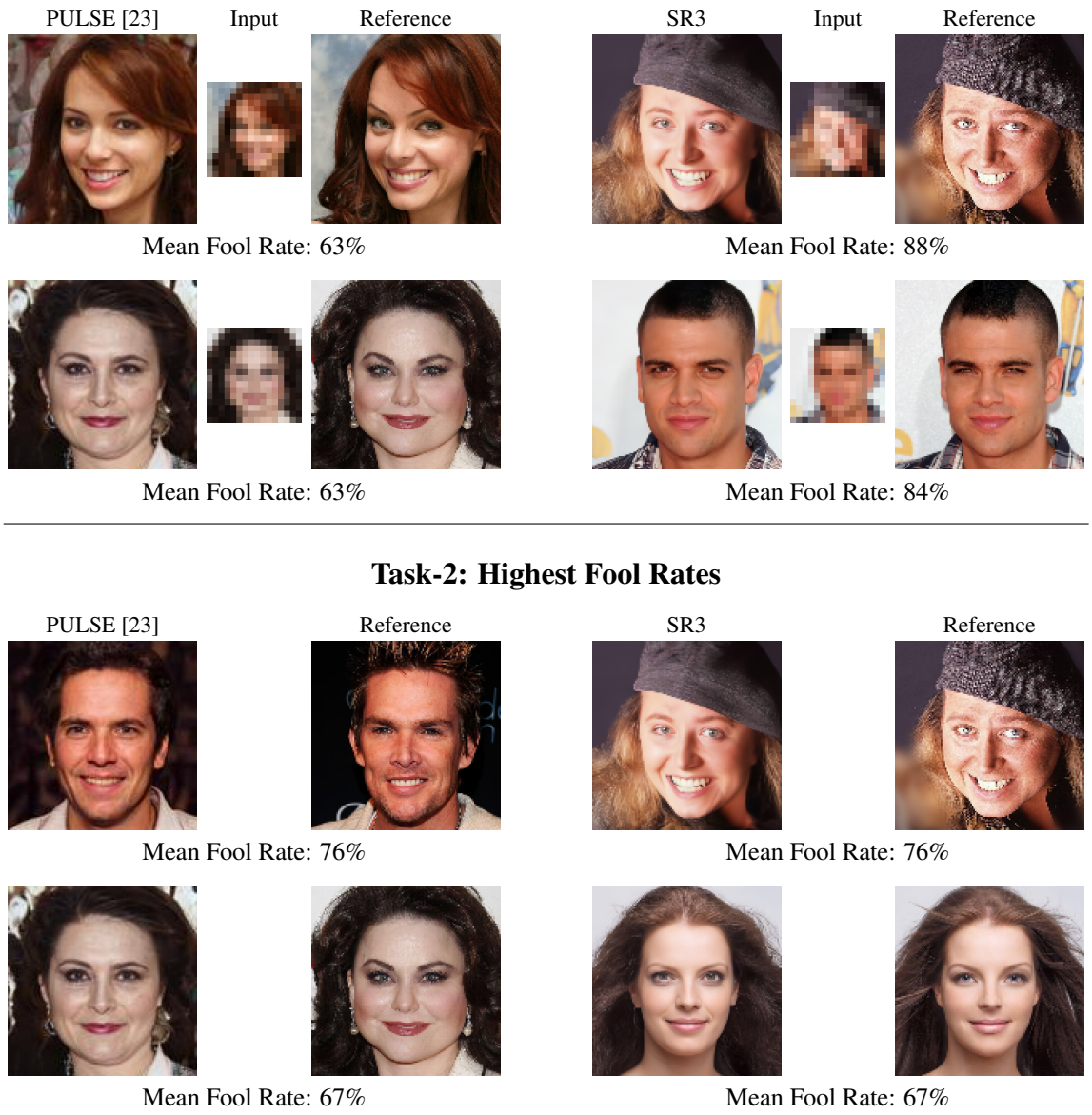

Figure A.3: The images with highest fool rates, PULSE on the left and SR3 on the right. The first 2 rows correspond to the conditional human evaluation (Task-1), and the last 2 rows correspond to the unconditional human evaluation (Task-2). The image layout mimics the original interface of our human evaluation study. For Task-1, we include the low resolution input image, and for Task-2, we omit the low resolution image to be consistent with the human evaluation setting. We also report the Mean Fool Rate for each of the image pairs right below.

A.3. More Samples of SR3 Outputs

The next several pages provide more samples in addition to those presented in the main body of the paper, *i.e.,* Figures 3, 4, 5, 7 and 8.

Figure A.4: Additional results showing the comparison between different methods on the 16×16 → 128×128 face super-resolution task.

Figure A.5: Additional results of a SR3 model (64×64 → 512×512), trained on FFHQ, and applied to a test image from CelebA-HQ.

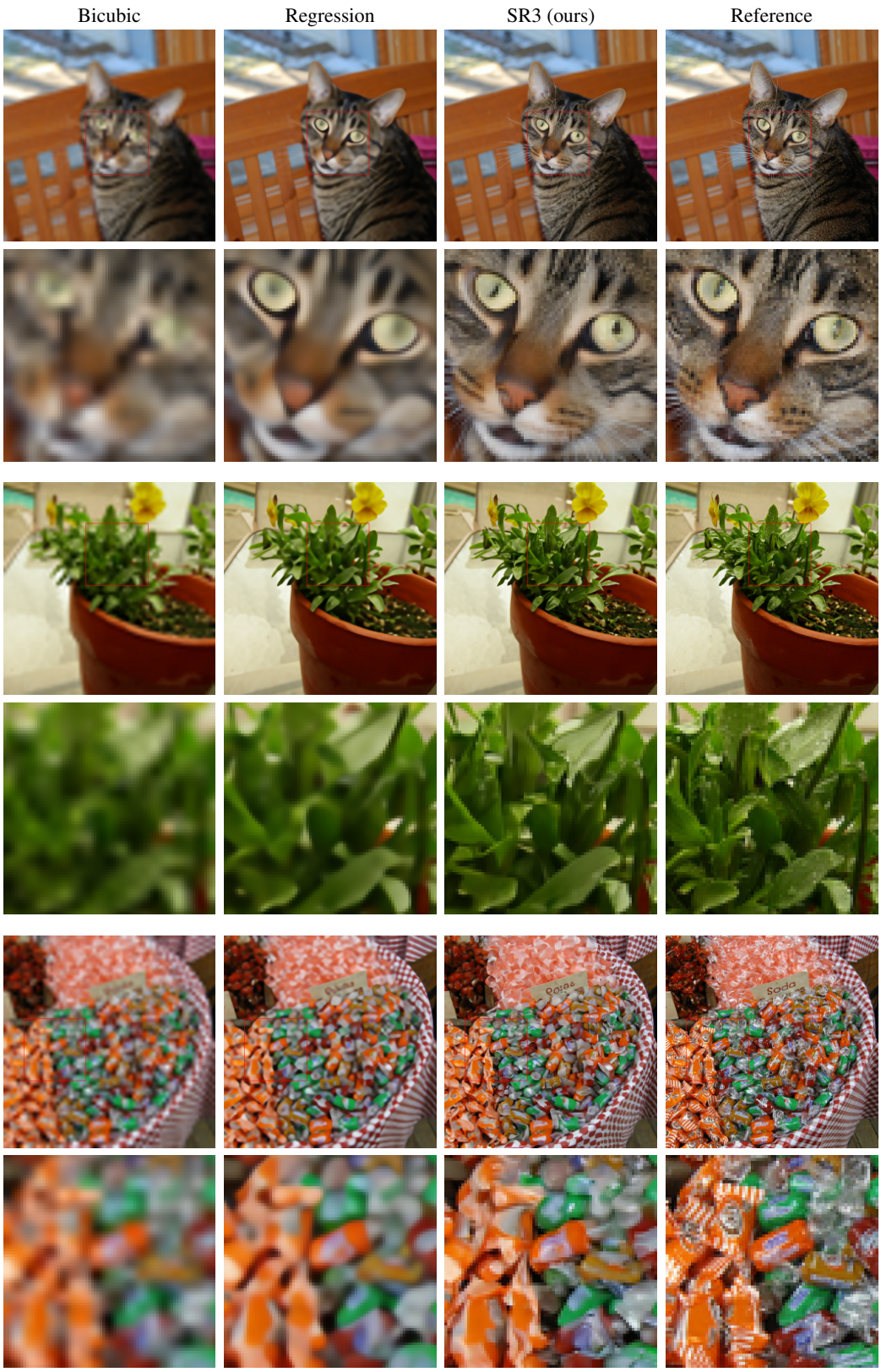

Figure A.6: Additional results of a SR3 model (64×64 → 256×256), trained on ImageNet and evaluated on two ImageNet test images.

Figure A.7: Cascaded generation on faces using an unconditional model chained with two SR3 models.

Unconditional Face Samples 1024×1024

Figure A.8: Additional Synthetic 1024×1024 faces images. We first sample from an unconditional 64×64 diffusion model, then pass the samples through two 4× SR3 models, *i.e.,* 64×64 → 256×256 → 1024×1024.

Figure A.9: Additional Synthetic 1024×1024 faces images. We first sample from an unconditional 64×64 diffusion model, then pass the samples through two 4× SR3 models, *i.e.,* 64×64 → 256×256 → 1024×1024.

Figure A.10: Additional Synthetic 1024×1024 faces images. We first sample from an unconditional 64×64 diffusion model, then pass the samples through two 4× SR3 models, *i.e.,* 64×64 → 256×256 → 1024×1024.

Figure A.11: Additional Synthetic 256×256 ImageNet images. We first draw a random label, then sample a 64×64 image from a class-conditional diffusion model, and apply a 4× SR3 model to obtain 256×256 images.

Figure A.12: Classwise Synthetic 256×256 ImageNet images. Each row represents a specific ImageNet class. Classes from top to bottom - Goldfish, Indigo Bird, Red Fox, Monarch Butterfly, African Elephant, Balloon, Church, Fire Truck. For a given label, we sample a 64×64 image from a class-conditional diffusion model, and apply a 4× SR3 model to obtain 256×256 images.

B. Task Specific Architectural Details

Table B.1 summarizes the primary architecture details for each super-resolution task. For a particular task, we use the same architecture for both SR3 and Regression models. Figure B.1 describes our method of conditioning the diffusion model on the low resolution image. We first interpolate the low resolution image to the target high resolution, and then simply concatenate it with the input noisy high resolution image.

Task	Channel Dim	Depth Multipliers	# ResNet Blocks	# Parameters
$16 \times 16 \rightarrow 128 \times 128$	128	$\{1, 2, 4, 8, 8\}$	3	550M
$64 \times 64 \rightarrow 256 \times 256$	128	$\{1, 2, 4, 4, 8, 8\}$	3	625M
$64 \times 64 \rightarrow 512 \times 512$	64	$\{1, 2, 4, 8, 8, 16, 16\}$	3	625M
$256 \times 256 \rightarrow 1024 \times 1024$	16	$\{1, 2, 4, 8, 16, 32, 32, 32\}$	2	150M

Table B.1: Task specific architecture hyper-parameters for the U-Net model. Channel Dim is the dimension of the first U-Net layer, while the depth multipliers are the multipliers for subsequent resolutions.

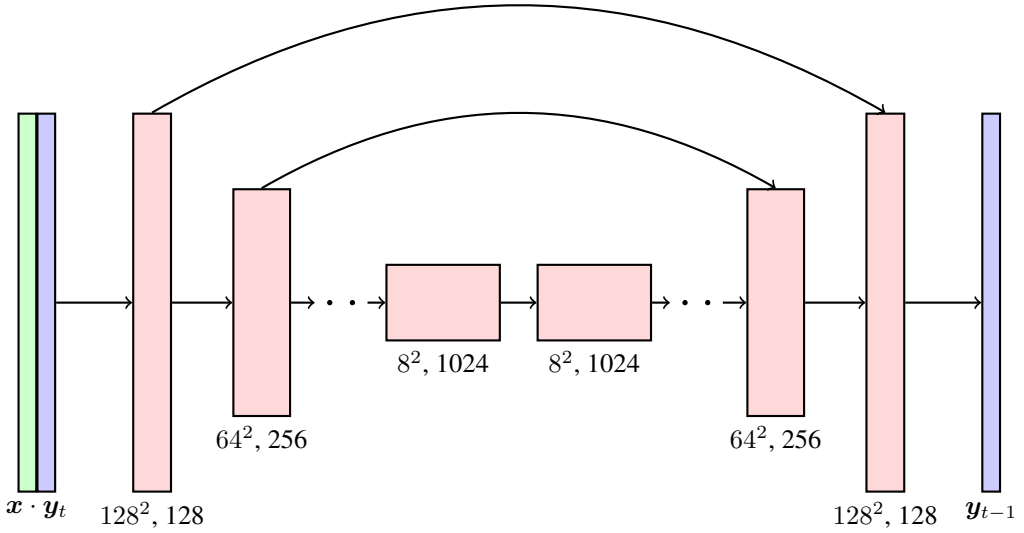

Figure B.1: Description of the U-Net architecture with skip connections. The low resolution input image $\boldsymbol{x}$ is interpolated to the target high resolution, and concatenated with the noisy high resolution image $\boldsymbol{y}_t$. We show the activation dimensions for the example task of $16\times16 \rightarrow 128\times128$ super resolution.

C. Justification of the Training Objective

C.1. A Variational Bound Perspective

Following Ho *et al.* [13], we justify the choice of the training objective in (6) for the probabilistic model outlined in (9) from a variational lower bound perspective. If the forward diffusion process is viewed as a fixed approximate posterior to the inference process, one can derive the following variational lower bound on the marginal log-likelihood:

$$\begin{aligned}&\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}_0)} \log p_\theta(\boldsymbol{y}_0|\boldsymbol{x}) \geq \\ &\quad \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}_0}\mathbb{E}_{q(\boldsymbol{y}_{1:T}|\boldsymbol{y}_0)}\left[\log p(\boldsymbol{y}_T) + \sum_{t\geq 1} \log \frac{p_\theta(\boldsymbol{y}_{t-1}|\boldsymbol{y}_t,\boldsymbol{x})}{q(\boldsymbol{y}_t|\boldsymbol{y}_{t-1})}\right].\end{aligned} \tag{12}$$

Given the particular parameterization of the inference process outlined above, one can show [13] that the negative variational lower bound can be expressed as the following simplified loss, up to a constant weighting of each term for each time step:

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}_0,\boldsymbol{\epsilon}} \sum_{t=1}^{T} \frac{1}{T} \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}, \sqrt{\gamma_t}\boldsymbol{y}_0 + \sqrt{1-\gamma_t}\boldsymbol{\epsilon}, \gamma_t) \right\|_2^2 \tag{13}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. Note that this objective function corresponds to L_2 norm in (6), and a characterization of $p(\gamma)$ in terms of a uniform distribution over $\{\gamma_1, \ldots, \gamma_T\}$.

C.2. A Denoising Score-Matching Perspective

Our approach is also linked to denoising score matching [14, 50, 31, 39] for training unnormalized energy functions for density estimation. These methods learn a parametric score function to approximate the gradient of the empirical data log-density. To make sure that the gradient of the data log-density is well-defined, one often replaces each data point with a Gaussian distribution with a small variance. Song and Ermon [43] advocate for the use of a Multi-scale Guassian mixture as the target density, where each data point is perturbed with different amounts of Guassian noise, so that Langevin dynamics starting from pure noise can still yield reasonable samples.

One can view our approach as a variant of denoising score matching in which the target density is given by a mixture of $q(\widetilde{\boldsymbol{y}}|\boldsymbol{y}_0,\gamma) = \mathcal{N}(\widetilde{\boldsymbol{y}} \,|\, \sqrt{\gamma}\boldsymbol{y}_0, 1-\gamma)$ for different values of $\boldsymbol{y}_0$ and γ. Accordingly, the gradient of data log-density is given by

$$\frac{\mathrm{d}\log q(\widetilde{\boldsymbol{y}} \mid \boldsymbol{y}_0,\gamma)}{\mathrm{d}\widetilde{\boldsymbol{y}}} \;=\; -\frac{\widetilde{\boldsymbol{y}} - \sqrt{\gamma}\boldsymbol{y}_0}{\sqrt{1-\gamma}} \;=\; -\boldsymbol{\epsilon}\,, \tag{14}$$

which is used as the regression target of our model.