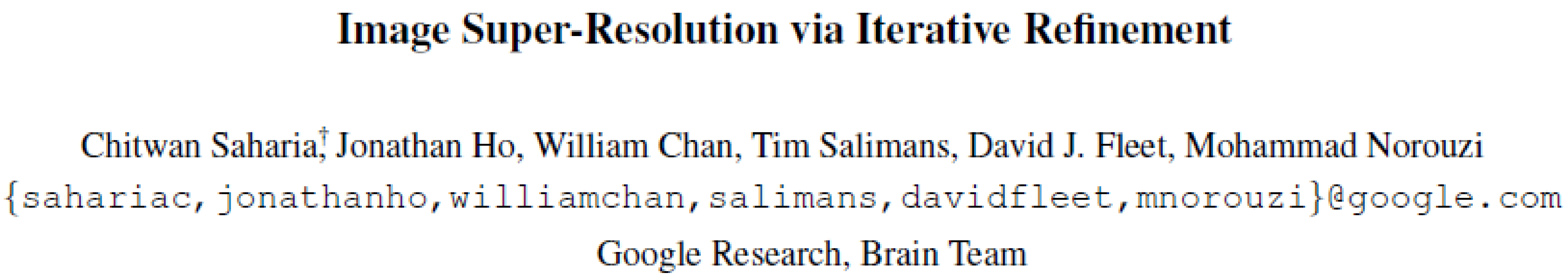


Image Super-Resolution via Iterative Refinement

Chitwan Saharia[†], Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, Mohammad Norouzi
{sahariac,jonathanho,williamchan,salimans,davidfleet,mnorouzi}@google.com
Google Research, Brain Team

2022.06.27(월)
남궁영수

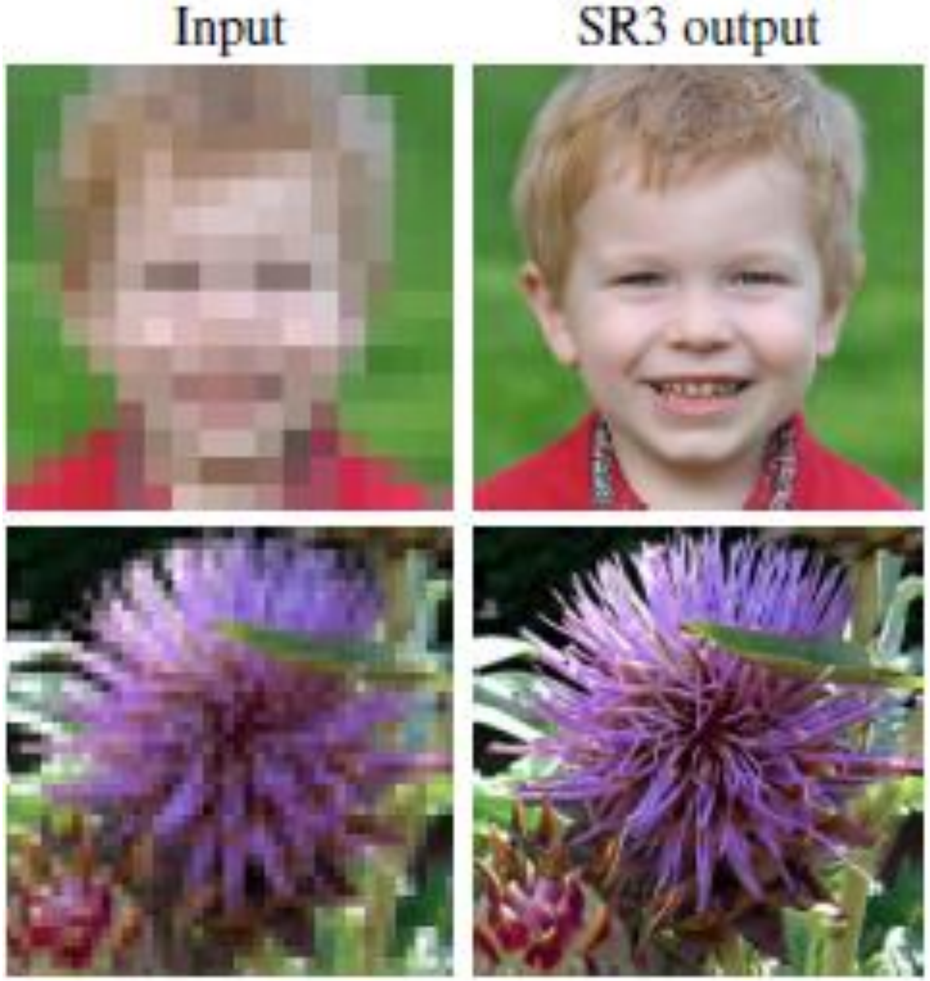
Contents

- 1. Introduction
- 2. Key Contribution
- 3. Conditional Denoising Diffusion Model
- 4. Experiments
- 5. Discussion and Conclusion

Introduction

- Single-image super-resolution

- It is a process of generating a high-resolution image that is consistent with an input low-resolution image.
- It is inverse problem
- It is challenging because multiple output images may be consistent with a single input image.



Introduction

- Generative Model

- Autoregressive model :

- VAEs :

- Flows :

- GAN :

Introduction

- Generative Model

- Autoregressive model : expensive for high-resolution image generation

- VAEs : often yield sub-optimal sample quality

- Flows : often yield sub-optimal sample quality

- GAN : require carefully designed regularization and optimization trick to tame optimization instability and mode collapse

Introduction

- SR3

- Inspired by Denoising Diffusion Probabilistic Models(DDPM), and denoising score matching.

- Works by learning to transform a standard normal distribution into an empirical data distribution through a sequence of refinement steps.

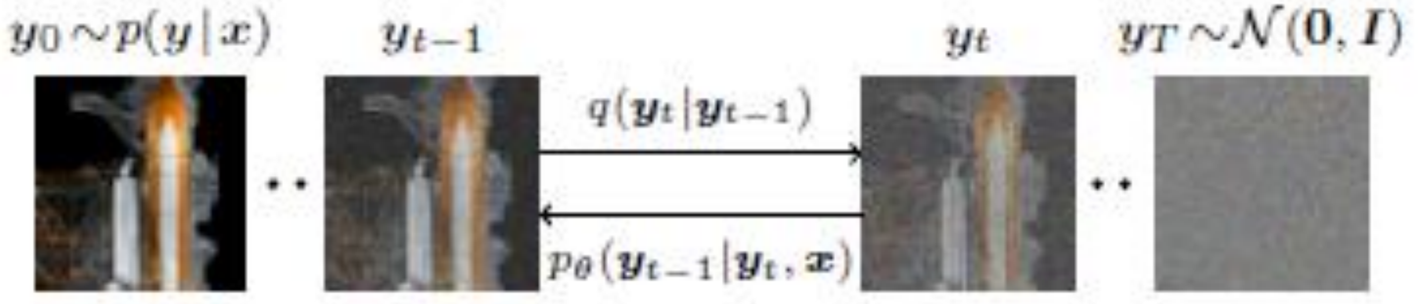


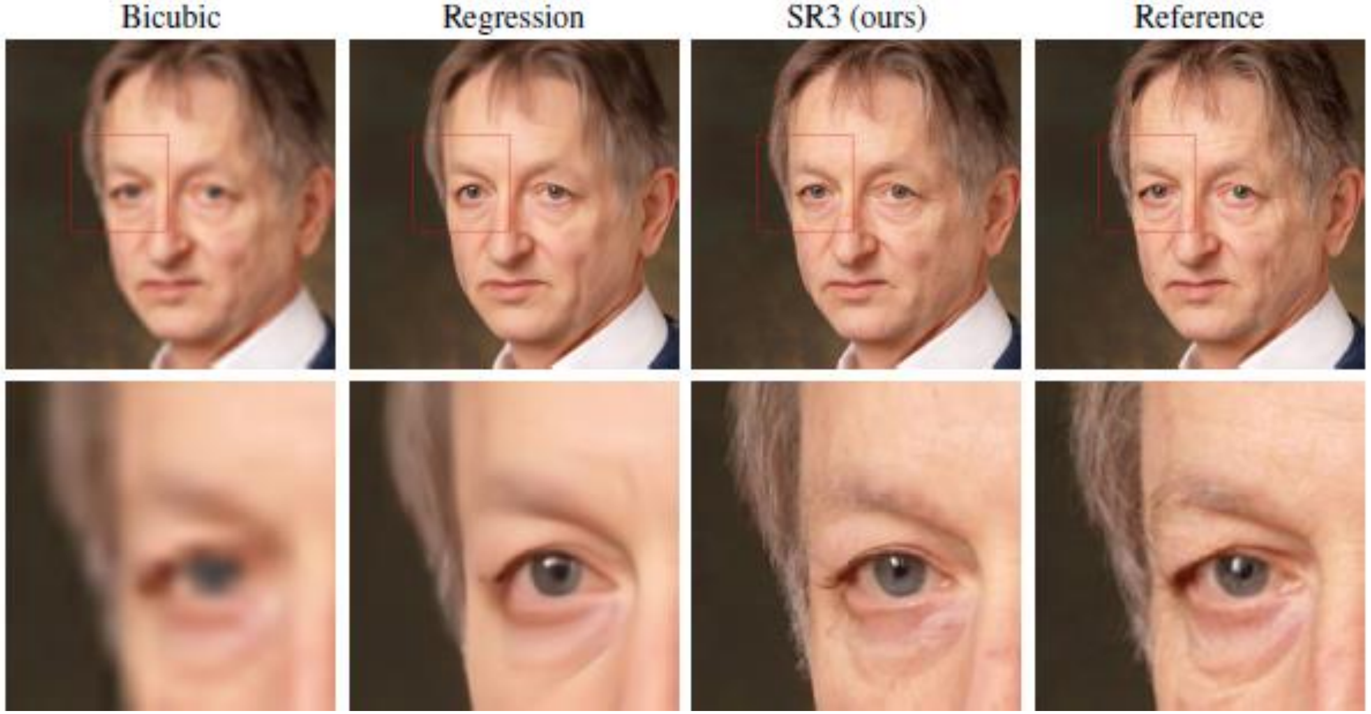
Figure 2: The forward diffusion process q (left to right) gradually adds Gaussian noise to the target image. The reverse inference process p (right to left) iteratively denoises the target image conditioned on a source image x. Source image x is not shown here.

Introduction

- Fool rate

- PSNR and SSIM don’t reflect human preference well when the input resolution is low and the magnification ratio is large

- PSNR and SSIM often penalize synthetic high-frequency details.

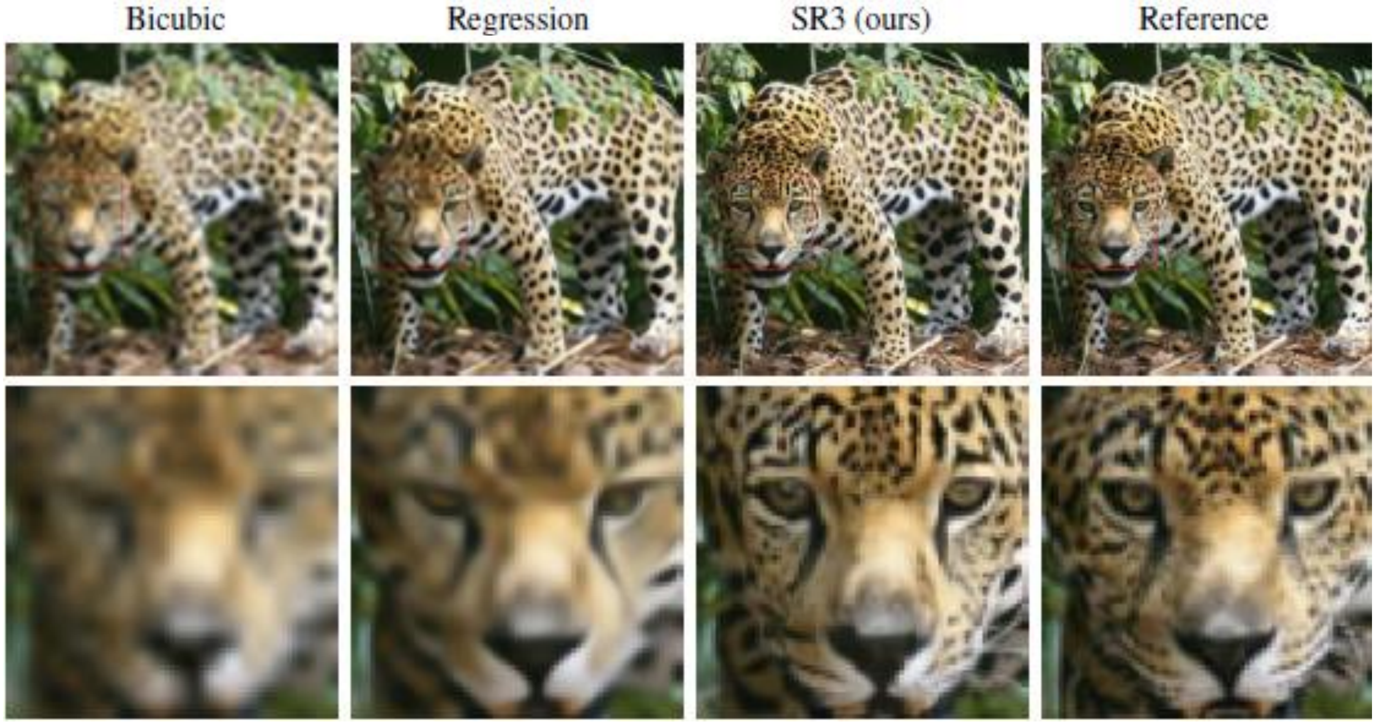


Introduction

- Fool rate

- PSNR and SSIM don't reflect human preference well when the input resolution is low and the magnification ratio is large

- PSNR and SSIM often penalize synthetic high-frequency details.



Introduction

- Fool rate

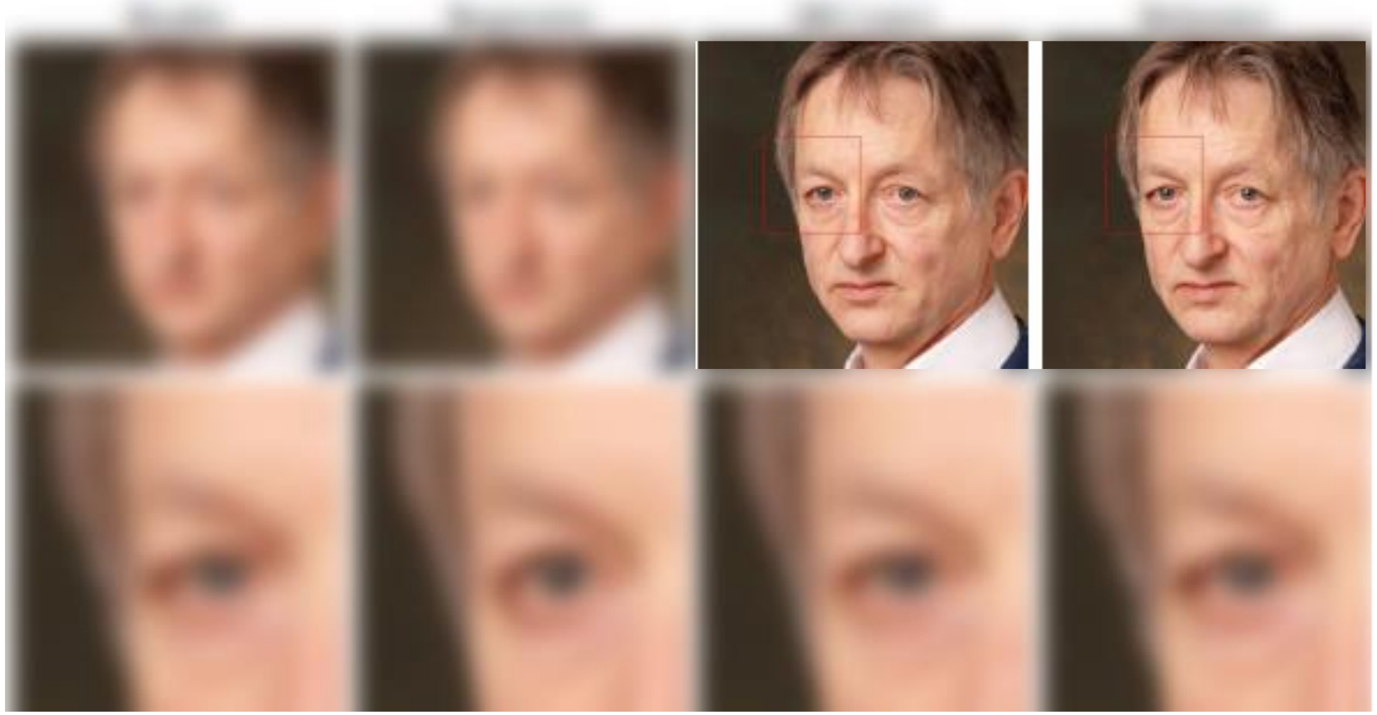
- So we resort to human evaluation.
- Human subjects are shown a low-resolution input and are required to select between a model output and a ground truth image.

Low-resolution input

Introduction

- Fool rate

- So we resort to human evaluation.
- Human subjects are shown a low-resolution input and are required to select between a model output and a ground truth image.



Select between
Model output and
ground truth

Key Contribution

- adapt denoising diffusion models to conditional image generation.

- prove effective on face and natural image super-resolution at different magnification factors. On standard 8x face super-resolution task, SR3 achieves a human fool rate close to 50%, outperforming PSRGAN and PULSE.

- demonstrate unconditional and class-conditional generation by cascading a 64x64 image synthesis model with SR3 models.

Conditional Denoising Diffusion Model

dataset

Unknown conditional distribution

$$\begin{tikzcd}[column sep=huge] \mathcal{D} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{N} & p(\boldsymbol{y} \,|\, \boldsymbol{x}) \arrow[l, "\text{Sample from}"'] \end{tikzcd}$$

Conditional Denoising Diffusion Model

dataset

Unknown conditional distribution

$$\mathcal{D} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{N} \xleftarrow{\text{Sample from}} p(\boldsymbol{y} \,|\, \boldsymbol{x})$$

$\boldsymbol{x}$

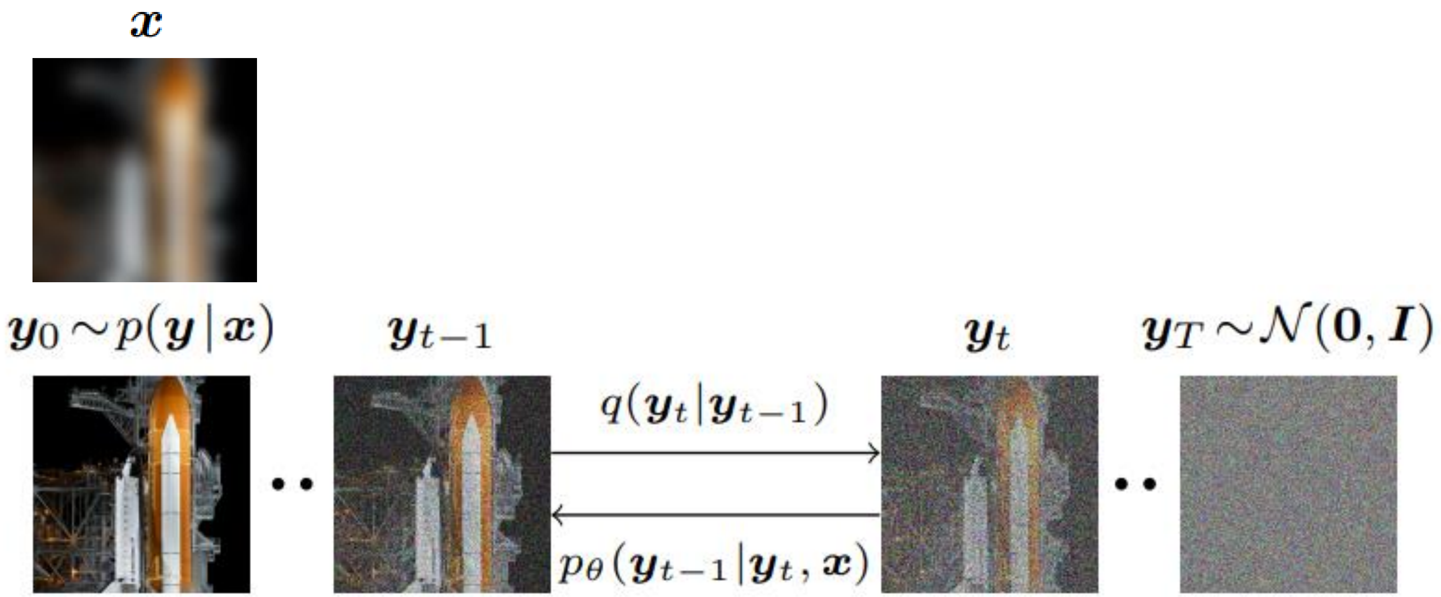
$\boldsymbol{y}_0 \sim p(\boldsymbol{y} \,|\, \boldsymbol{x})$

Conditional Denoising Diffusion Model

dataset

Unknown conditional distribution

$$\mathcal{D} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{N} \xleftarrow{\text{Sample from}} p(\boldsymbol{y} \,|\, \boldsymbol{x})$$



$\boldsymbol{x}$
$\boldsymbol{y}_0 \sim p(\boldsymbol{y} \,|\, \boldsymbol{x})$
$\boldsymbol{y}_{t-1}$
$q(\boldsymbol{y}_t|\boldsymbol{y}_{t-1})$
$p_\theta(\boldsymbol{y}_{t-1}|\boldsymbol{y}_t, \boldsymbol{x})$
$\boldsymbol{y}_t$
$\boldsymbol{y}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

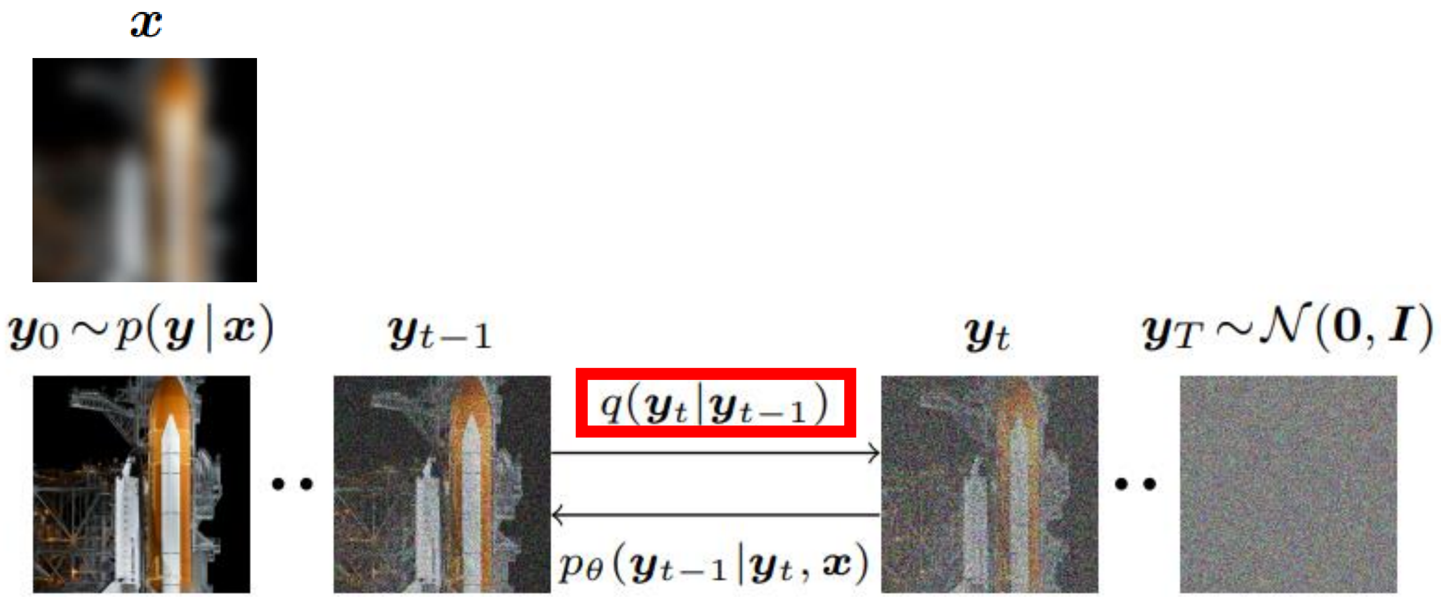
Figure 2: The forward diffusion process q (left to right) gradually adds Gaussian noise to the target image. The reverse inference process p (right to left) iteratively denoises the target image conditioned on a source image $\boldsymbol{x}$. Source image $\boldsymbol{x}$ is not shown here.

Conditional Denoising Diffusion Model

dataset

Unknown conditional distribution

$$\mathcal{D} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{N} \xleftarrow{\text{Sample from}} p(\boldsymbol{y} \,|\, \boldsymbol{x})$$



$\boldsymbol{x}$
$\boldsymbol{y}_0 \sim p(\boldsymbol{y} \,|\, \boldsymbol{x})$
$\boldsymbol{y}_{t-1}$
$\boldsymbol{y}_t$
$\boldsymbol{y}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
$q(\boldsymbol{y}_t | \boldsymbol{y}_{t-1})$
$p_\theta(\boldsymbol{y}_{t-1} | \boldsymbol{y}_t, \boldsymbol{x})$

Figure 2: The forward diffusion process q (left to right) gradually adds Gaussian noise to the target image. The reverse inference process p (right to left) iteratively denoises the target image conditioned on a source image $\boldsymbol{x}$. Source image $\boldsymbol{x}$ is not shown here.

Conditional Denoising Diffusion Model

dataset

Unknown conditional distribution

Sample from

$$\mathcal{D} \doteq \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{N} \longleftarrow p(\boldsymbol{y} \,|\, \boldsymbol{x})$$

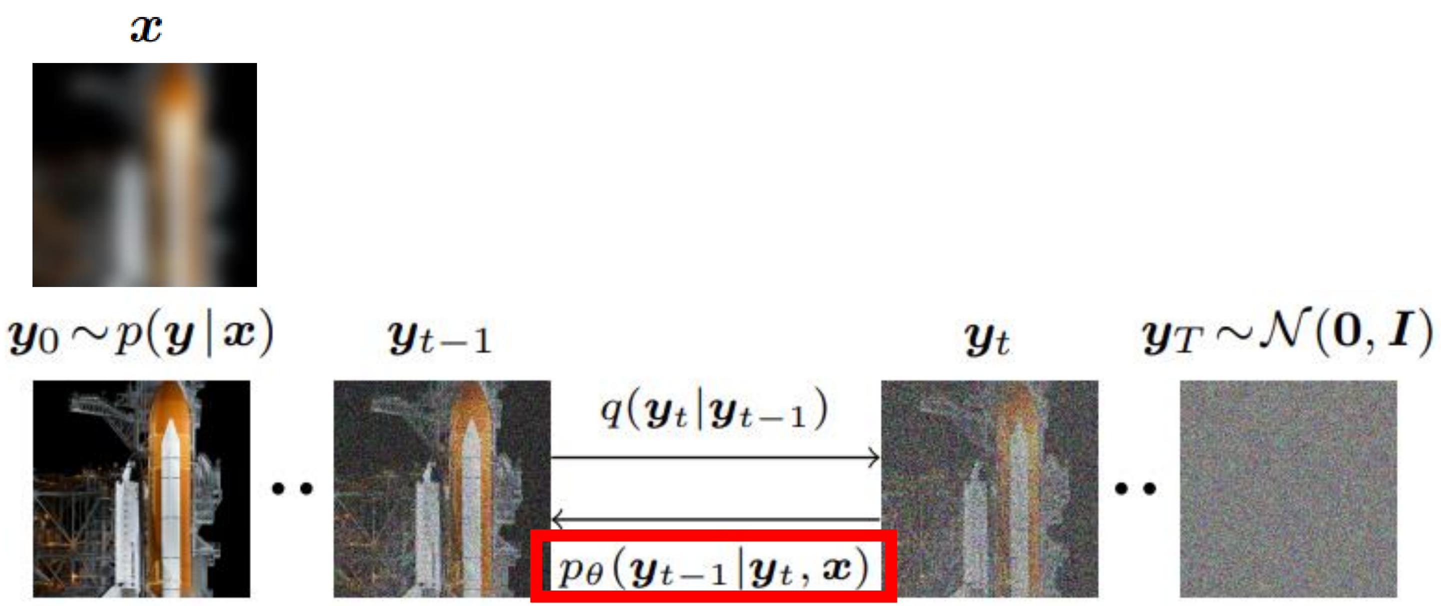
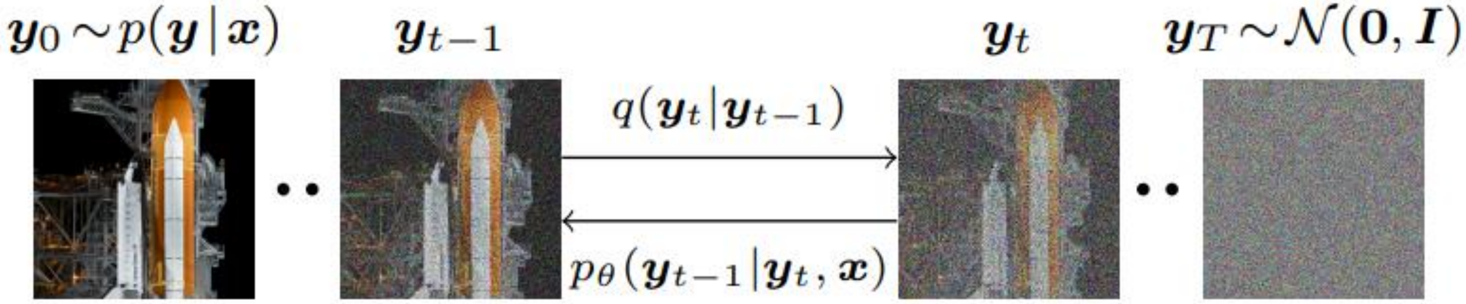
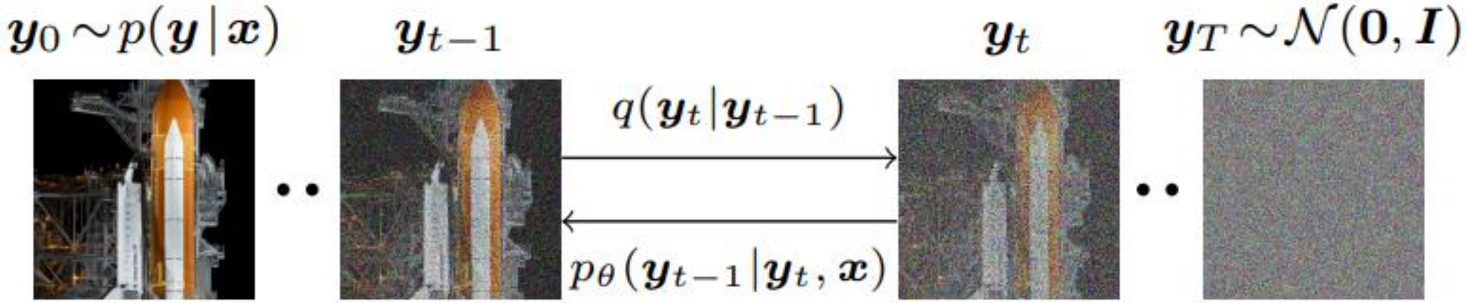

$\boldsymbol{x}$
$\boldsymbol{y}_0 \sim p(\boldsymbol{y} \,|\, \boldsymbol{x})$
$\boldsymbol{y}_{t-1}$
$\boldsymbol{y}_t$
$\boldsymbol{y}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
$q(\boldsymbol{y}_t | \boldsymbol{y}_{t-1})$
$p_\theta(\boldsymbol{y}_{t-1} | \boldsymbol{y}_t, \boldsymbol{x})$

Figure 2: The forward diffusion process q (left to right) gradually adds Gaussian noise to the target image. The reverse inference process p (right to left) iteratively denoises the target image conditioned on a source image $\boldsymbol{x}$. Source image $\boldsymbol{x}$ is not shown here.

Conditional Denoising Diffusion Model


$\boldsymbol{y}_0 \sim p(\boldsymbol{y} \mid \boldsymbol{x})$
$\boldsymbol{y}_{t-1}$
$q(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1})$
$p_\theta(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_t, \boldsymbol{x})$
$\boldsymbol{y}_t$
$\boldsymbol{y}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

Conditional Denoising Diffusion Model

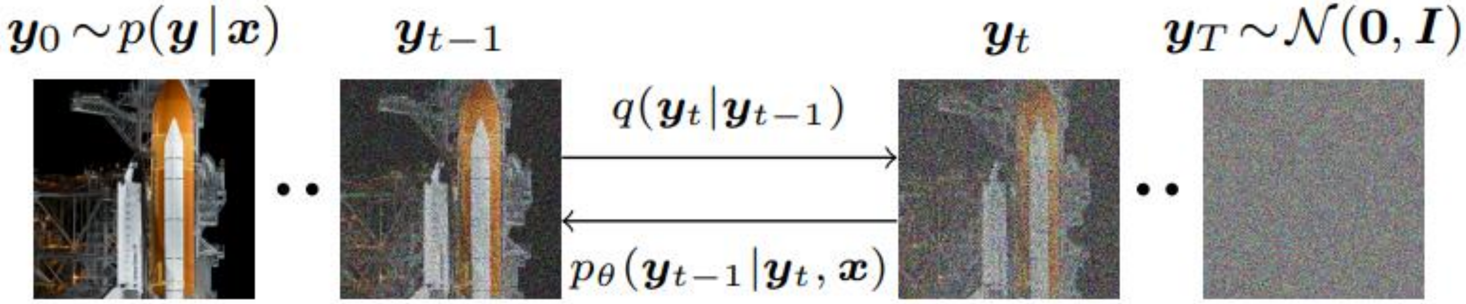

$\boldsymbol{y}_0 \sim p(\boldsymbol{y} \,|\, \boldsymbol{x})$
$\boldsymbol{y}_{t-1}$
$q(\boldsymbol{y}_t|\boldsymbol{y}_{t-1})$
$p_\theta(\boldsymbol{y}_{t-1}|\boldsymbol{y}_t, \boldsymbol{x})$
$\boldsymbol{y}_t$
$\boldsymbol{y}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

2.1. Gaussian Diffusion Process

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

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

Conditional Denoising Diffusion Model


$\boldsymbol{y}_0 \sim p(\boldsymbol{y} \mid \boldsymbol{x})$
$\boldsymbol{y}_{t-1}$
$q(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1})$
$\boldsymbol{y}_t$
$\boldsymbol{y}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
$p_\theta(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_t, \boldsymbol{x})$

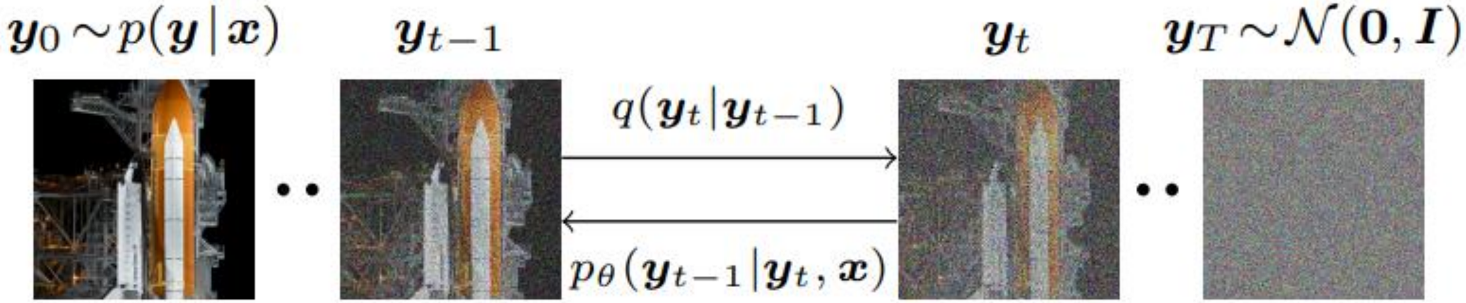
2.1. Gaussian Diffusion Process

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

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

$$q(\boldsymbol{y}_t \mid \boldsymbol{y}_0) = \mathcal{N}(\boldsymbol{y}_t \mid \sqrt{\gamma_t}\, \boldsymbol{y}_0, (1-\gamma_t)\boldsymbol{I}) \,, \quad (3) \quad \text{where } \gamma_t = \prod_{i=1}^{t} \alpha_i$$

Conditional Denoising Diffusion Model


$\boldsymbol{y}_0 \sim p(\boldsymbol{y} \,|\, \boldsymbol{x})$
$\boldsymbol{y}_{t-1}$
$\boldsymbol{y}_t$
$\boldsymbol{y}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
$q(\boldsymbol{y}_t | \boldsymbol{y}_{t-1})$
$p_\theta(\boldsymbol{y}_{t-1} | \boldsymbol{y}_t, \boldsymbol{x})$

2.1. Gaussian Diffusion Process

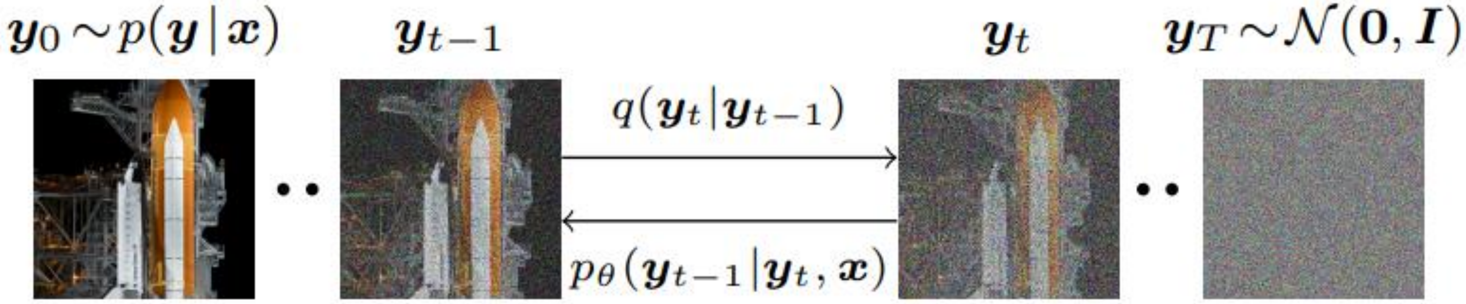
$$q(\boldsymbol{y}_{1:T} \mid \boldsymbol{y}_0) = \prod_{t=1}^{T} q(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}) \,, \qquad (1)$$

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

$$q(\boldsymbol{y}_t \mid \boldsymbol{y}_0) = \mathcal{N}(\boldsymbol{y}_t \mid \sqrt{\gamma_t}\, \boldsymbol{y}_0, (1-\gamma_t)\boldsymbol{I}) \,, \qquad (3) \quad \text{where } \gamma_t = \prod_{i=1}^{t} \alpha_i$$

With some algebraic manipulation, one can derive the posterior distribution of $\boldsymbol{y}_{t-1}$ given $(\boldsymbol{y}_0, \boldsymbol{y}_t)$

Conditional Denoising Diffusion Model



2.1. Gaussian Diffusion Process

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

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

$$q(\boldsymbol{y}_t \mid \boldsymbol{y}_0) = \mathcal{N}(\boldsymbol{y}_t \mid \sqrt{\gamma_t}\, \boldsymbol{y}_0, (1-\gamma_t)\boldsymbol{I}), \tag{3}$$ where $\gamma_t = \prod_{i=1}^{t} \alpha_i$

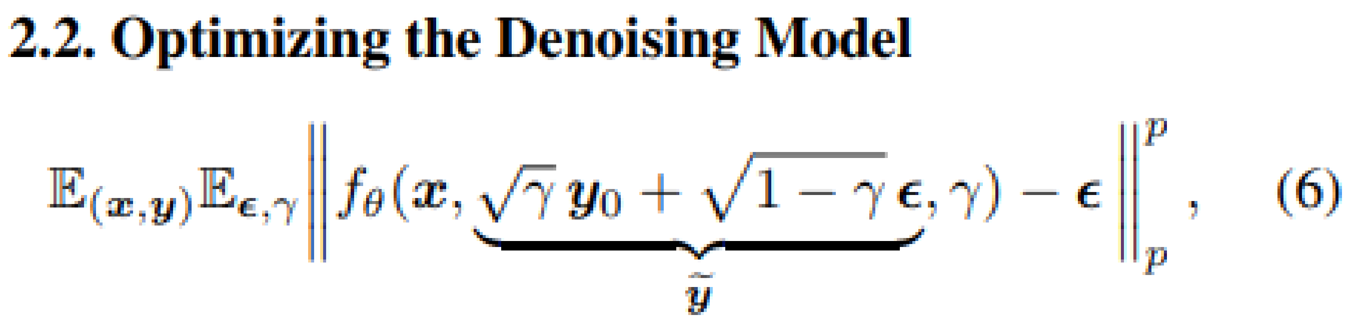
With some algebraic manipulation, one can derive the posterior distribution of $\boldsymbol{y}_{t-1}$ given $(\boldsymbol{y}_0, \boldsymbol{y}_t)$

$$q(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_0, \boldsymbol{y}_t) = \mathcal{N}(\boldsymbol{y}_{t-1} \mid \boldsymbol{\mu}, \sigma^2 \boldsymbol{I})$$

$$\boldsymbol{\mu} = \frac{\sqrt{\gamma_{t-1}}\,(1-\alpha_t)}{1-\gamma_t}\, \boldsymbol{y}_0 + \frac{\sqrt{\alpha_t}\,(1-\gamma_{t-1})}{1-\gamma_t}\, \boldsymbol{y}_t \tag{4}$$

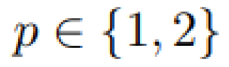
$$\sigma^2 = \frac{(1-\gamma_{t-1})(1-\alpha_t)}{1-\gamma_t}\,.$$

Conditional Denoising Diffusion Model



2.2. Optimizing the Denoising Model

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



$$p \in \{1, 2\}$$

Conditional Denoising Diffusion Model

2.2. Optimizing the Denoising Model

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

$p \in \{1, 2\}$

Input : $\boldsymbol{x}$

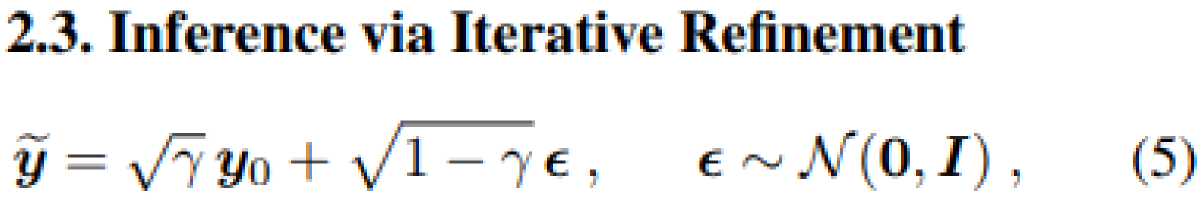
$$\widetilde{\boldsymbol{y}} = \sqrt{\gamma}\,\boldsymbol{y}_0 + \sqrt{1-\gamma}\,\boldsymbol{\epsilon}\,, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})\,, \qquad (5)$$

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

$\gamma \sim p(\gamma)$ (sampling method : sample t~{0,…,T} and sample γ ~$U(\gamma_{t-1}, \gamma_t)$)
(scaling factor for noise)

Output : estimation of $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

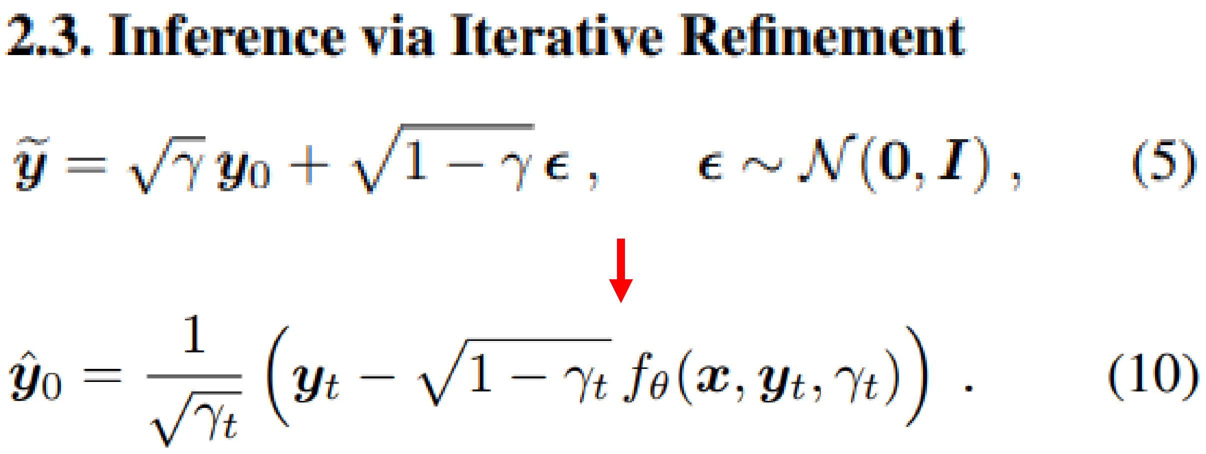
Conditional Denoising Diffusion Model



2.3. Inference via Iterative Refinement

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

Conditional Denoising Diffusion Model

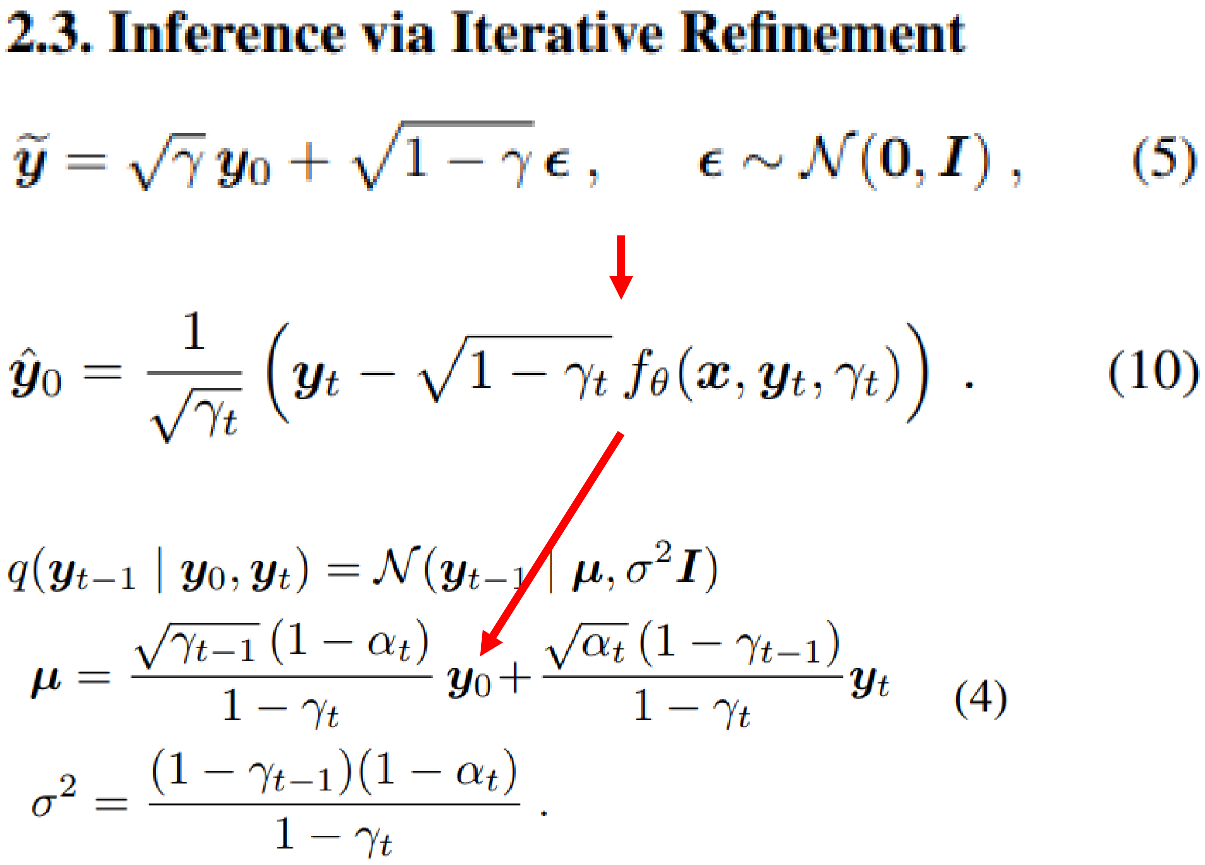


2.3. Inference via Iterative Refinement

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

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

Conditional Denoising Diffusion Model



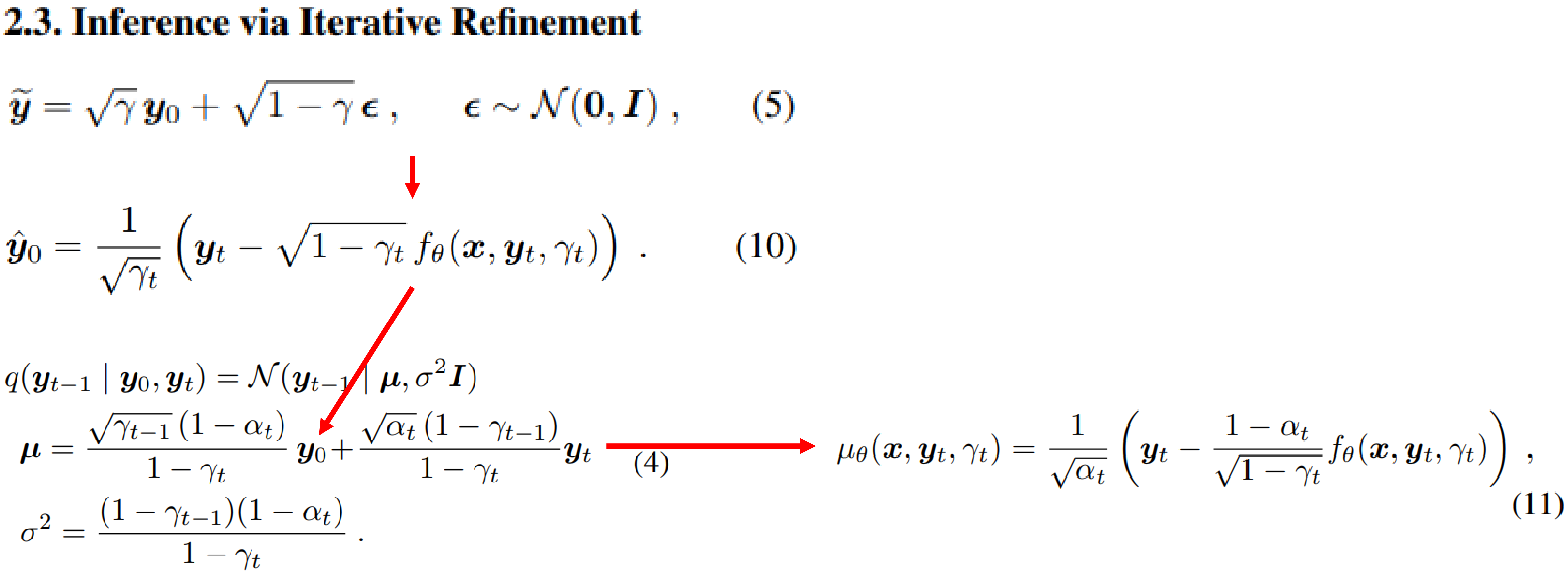
2.3. Inference via Iterative Refinement

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

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

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

Conditional Denoising Diffusion Model



2.3. Inference via Iterative Refinement

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

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

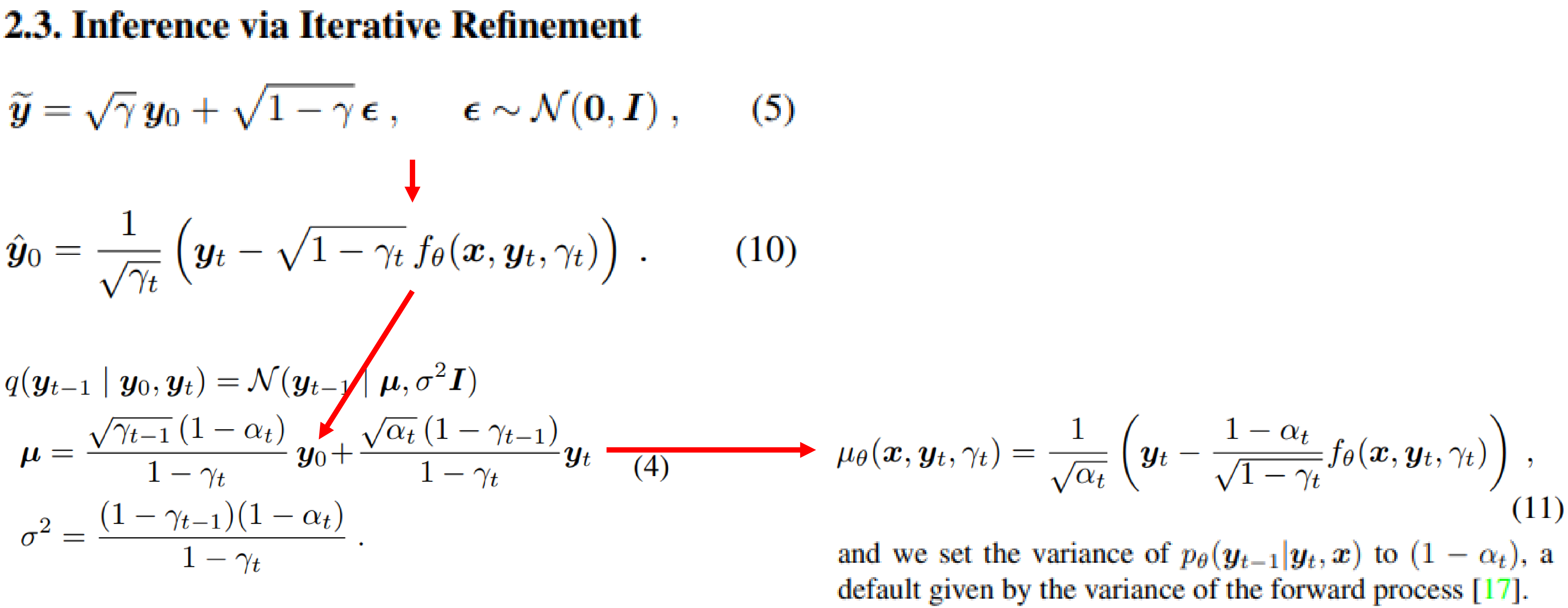
$$q(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_0, \boldsymbol{y}_t) = \mathcal{N}(\boldsymbol{y}_{t-1} \mid \boldsymbol{\mu}, \sigma^2\boldsymbol{I})$$

$$\boldsymbol{\mu} = \frac{\sqrt{\gamma_{t-1}}\,(1-\alpha_t)}{1-\gamma_t}\,\boldsymbol{y}_0 + \frac{\sqrt{\alpha_t}\,(1-\gamma_{t-1})}{1-\gamma_t}\boldsymbol{y}_t \tag{4}$$

$$\sigma^2 = \frac{(1-\gamma_{t-1})(1-\alpha_t)}{1-\gamma_t}\,.$$

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

Conditional Denoising Diffusion Model



2.3. Inference via Iterative Refinement

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

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

$$q(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_0, \boldsymbol{y}_t) = \mathcal{N}(\boldsymbol{y}_{t-1} \mid \boldsymbol{\mu}, \sigma^2 \boldsymbol{I})$$

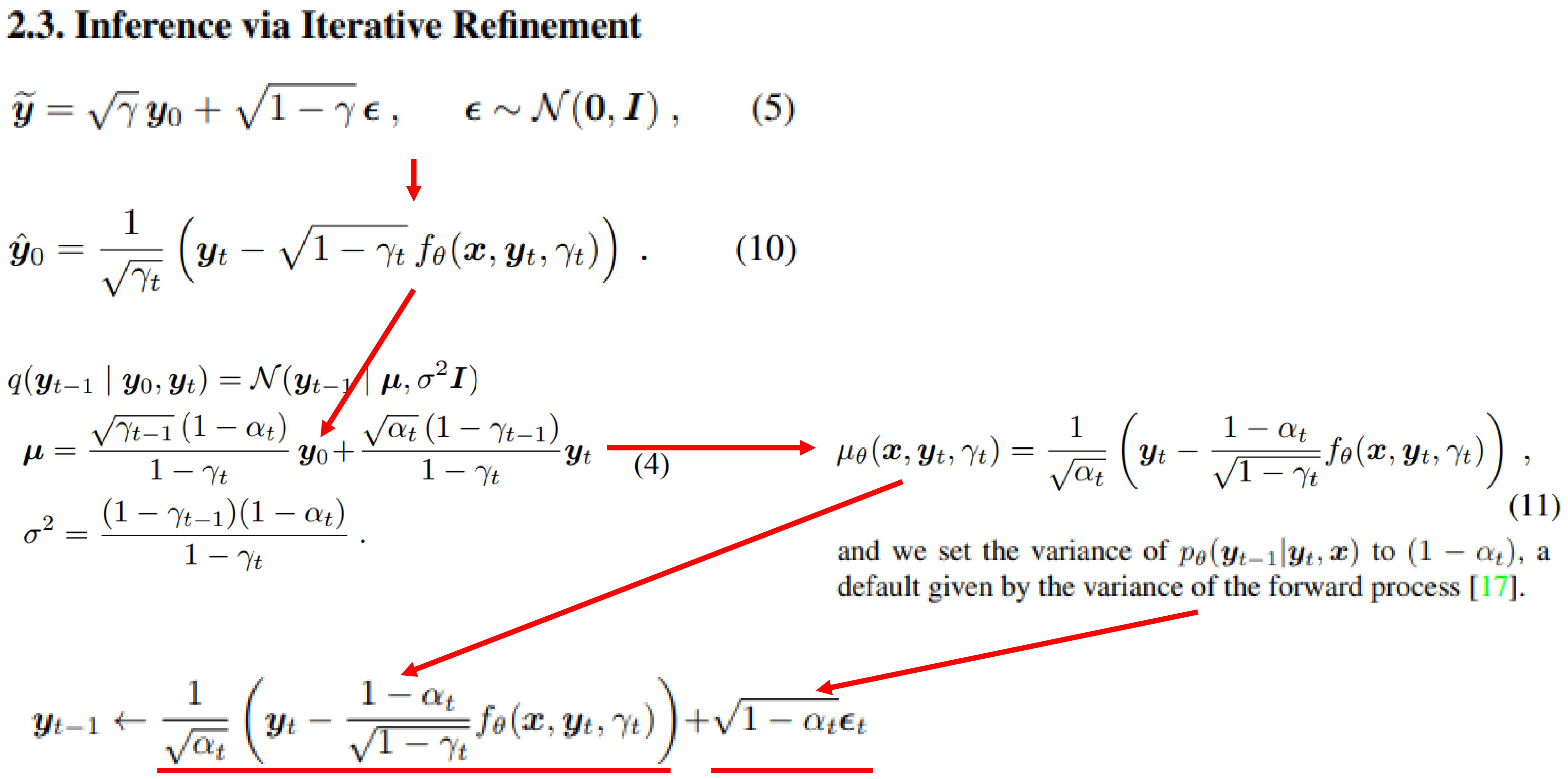
$$\boldsymbol{\mu} = \frac{\sqrt{\gamma_{t-1}}\,(1-\alpha_t)}{1-\gamma_t}\,\boldsymbol{y}_0 + \frac{\sqrt{\alpha_t}\,(1-\gamma_{t-1})}{1-\gamma_t}\boldsymbol{y}_t \qquad (4)$$

$$\sigma^2 = \frac{(1-\gamma_{t-1})(1-\alpha_t)}{1-\gamma_t}\,.$$

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

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 [17].

Conditional Denoising Diffusion Model



2.3. Inference via Iterative Refinement

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

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

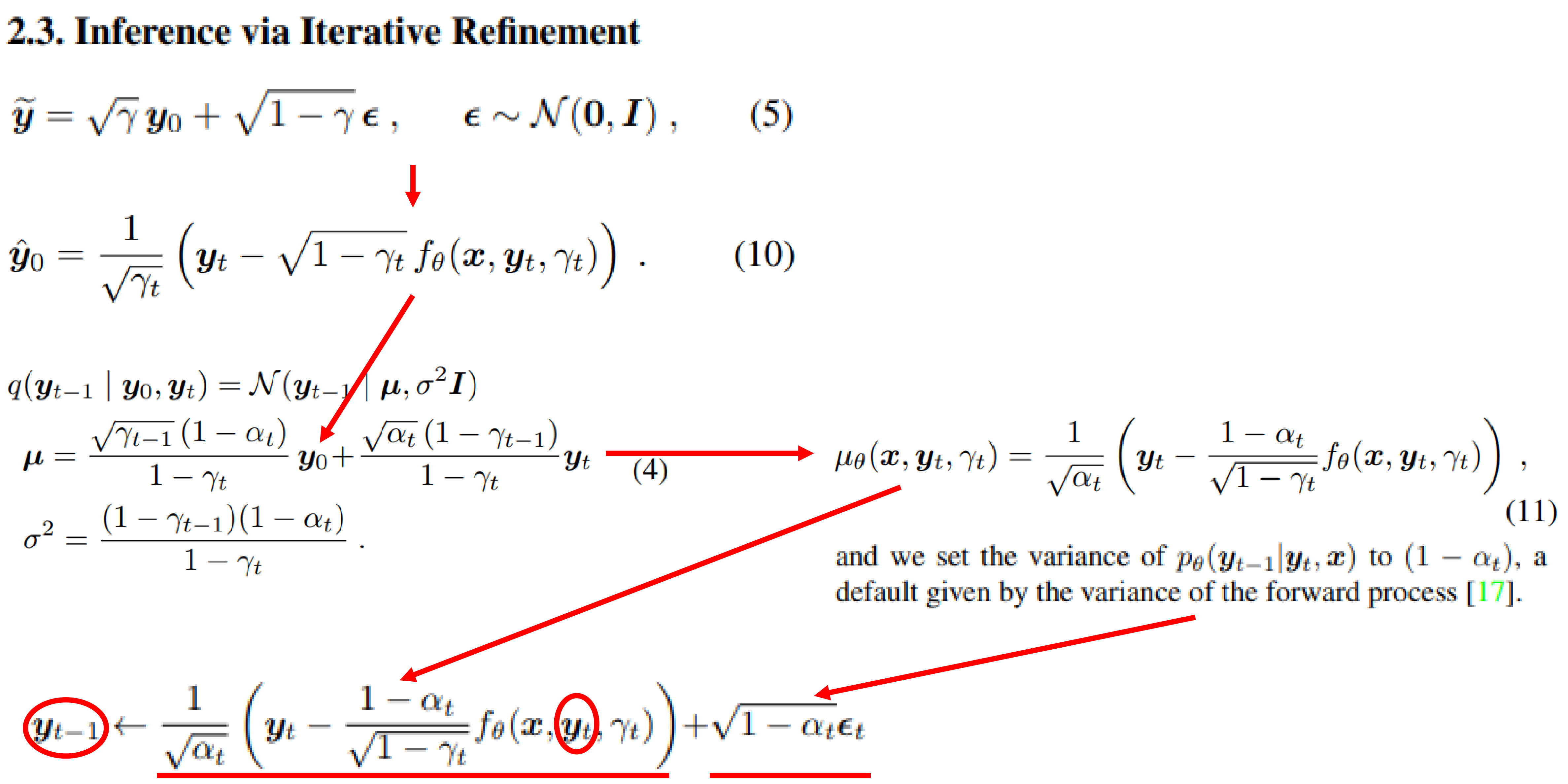
$$\begin{aligned} q(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_0, \boldsymbol{y}_t) &= \mathcal{N}(\boldsymbol{y}_{t-1} \mid \boldsymbol{\mu}, \sigma^2 \boldsymbol{I}) \\ \boldsymbol{\mu} &= \frac{\sqrt{\gamma_{t-1}}\,(1-\alpha_t)}{1-\gamma_t}\,\boldsymbol{y}_0 + \frac{\sqrt{\alpha_t}\,(1-\gamma_{t-1})}{1-\gamma_t}\,\boldsymbol{y}_t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

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

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 [17].

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

Conditional Denoising Diffusion Model



2.3. Inference via Iterative Refinement

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

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

$$q(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_0, \boldsymbol{y}_t) = \mathcal{N}(\boldsymbol{y}_{t-1} \mid \boldsymbol{\mu}, \sigma^2 \boldsymbol{I})$$

$$\boldsymbol{\mu} = \frac{\sqrt{\gamma_{t-1}}\,(1-\alpha_t)}{1-\gamma_t}\, \boldsymbol{y}_0 + \frac{\sqrt{\alpha_t}\,(1-\gamma_{t-1})}{1-\gamma_t}\, \boldsymbol{y}_t \tag{4}$$

$$\sigma^2 = \frac{(1-\gamma_{t-1})(1-\alpha_t)}{1-\gamma_t}\,.$$

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

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 [17].

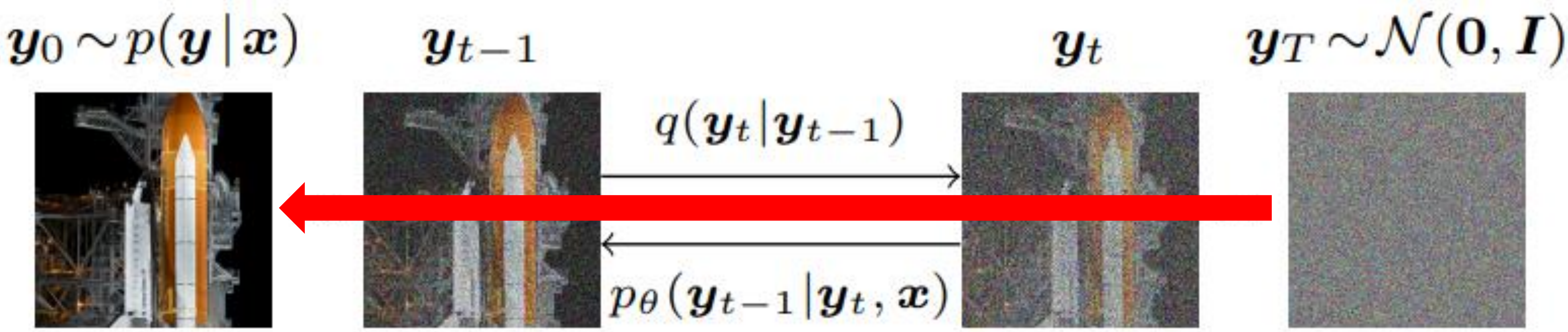
$$\boldsymbol{y}_{t-1} \leftarrow \underline{\frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{y}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}}\, f_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t) \right)} + \underline{\sqrt{1-\alpha_t}\,\boldsymbol{\epsilon}_t}$$

Conditional Denoising Diffusion Model

2.3. Inference via Iterative Refinement

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

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


$\boldsymbol{y}_0 \sim p(\boldsymbol{y} \mid \boldsymbol{x})$
$\boldsymbol{y}_{t-1}$
$\boldsymbol{y}_t$
$\boldsymbol{y}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
$q(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1})$
$p_\theta(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_t, \boldsymbol{x})$

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

Experiments

4.1. Qualitative Results

Face Images

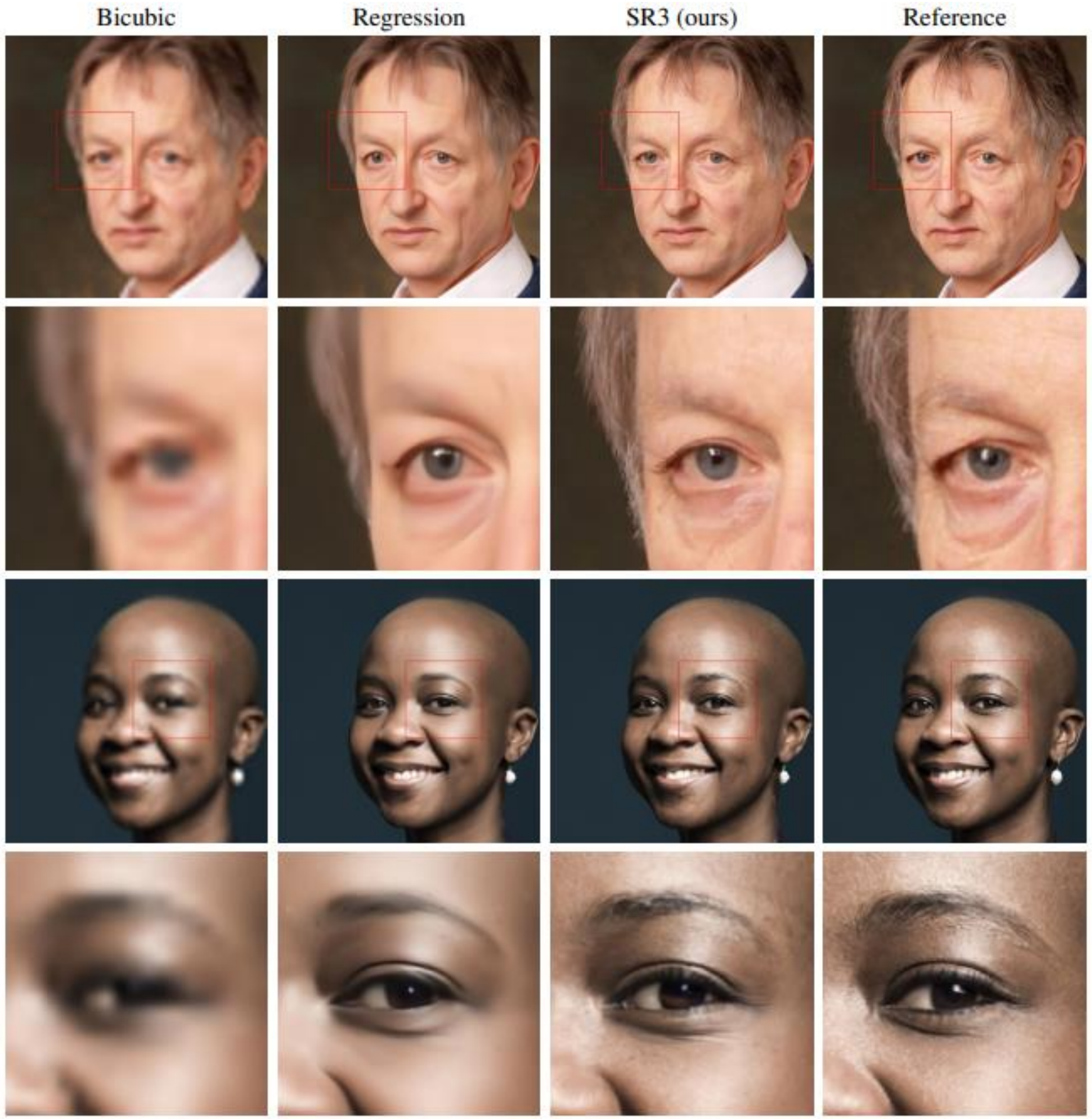


Figure 4: Results of a SR3 model (64×64 → 512×512), trained on FFHQ, and applied to images outside of the training set, along with enlarged patches to show finer details. Additional results are shown in Appendix C.1 and C.2.

Experiments

4.1. Qualitative Results

Natural Images

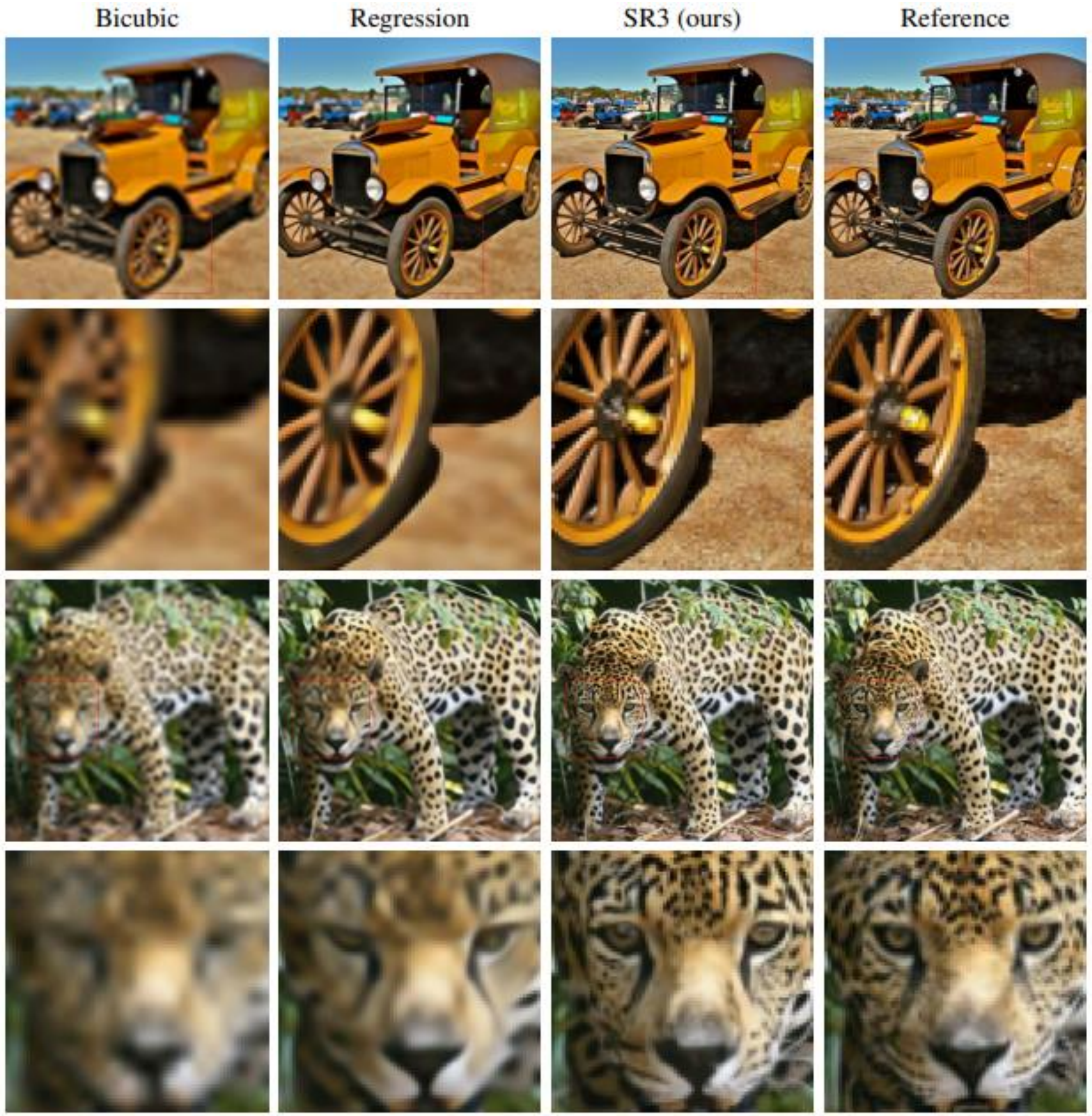


Figure 3: Results of a SR3 model (64×64 → 256×256), trained on ImageNet and evaluated on two ImageNet test images. For each we also show an enlarged patch in which finer details are more apparent. Additional samples are shown in Appendix C.3 and C.4.

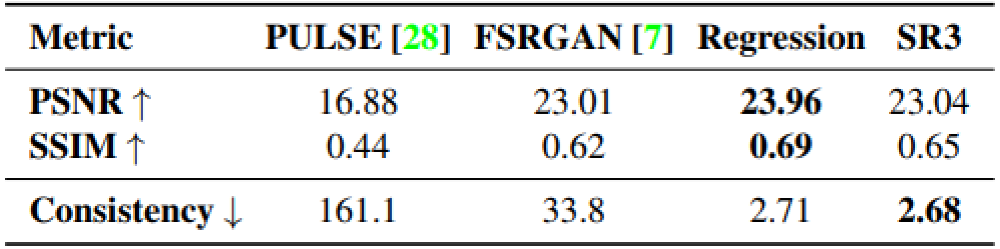
Experiments

4.2. Benchmark Comparison

Experiments

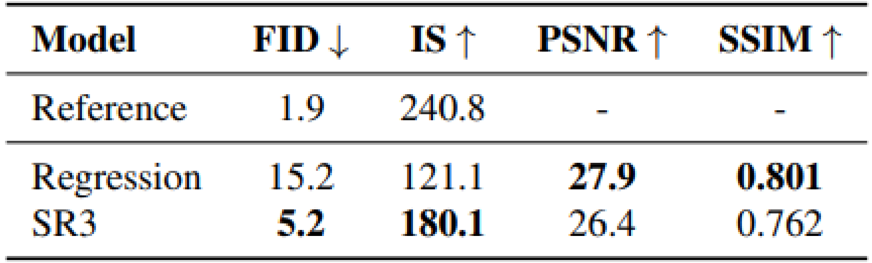
4.2. Benchmark Comparison

4.2.1 Automated metrics



Metric	PULSE [28]	FSRGAN [7]	Regression	SR3
PSNR ↑	16.88	23.01	**23.96**	23.04
SSIM ↑	0.44	0.62	**0.69**	0.65
Consistency ↓	161.1	33.8	2.71	**2.68**

Table 1: PSNR & SSIM on 16×16 → 128×128 face super-resolution. Consistency measures MSE ($\times 10^{-5}$) between the low-resolution inputs and the down-sampled super-resolution outputs.



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

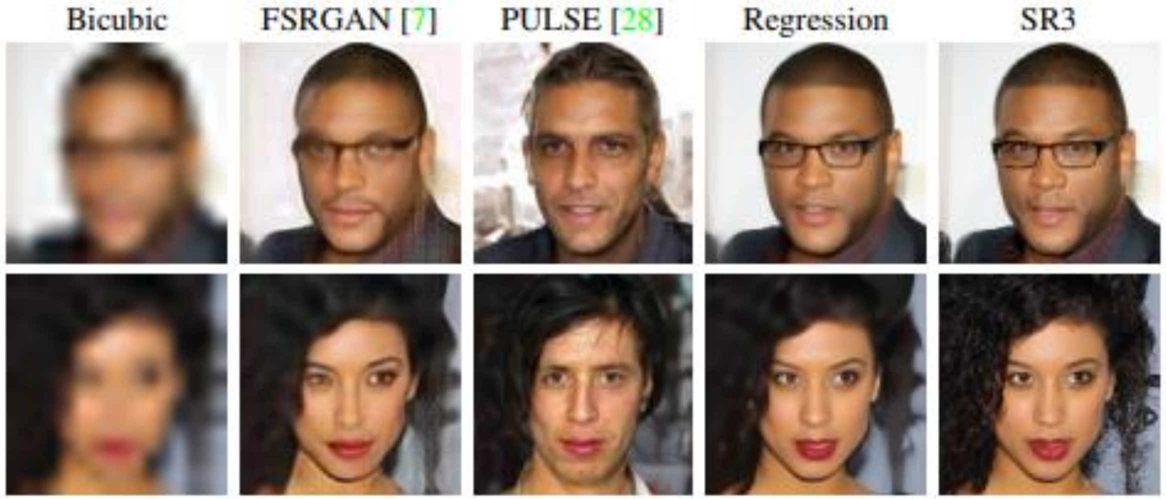
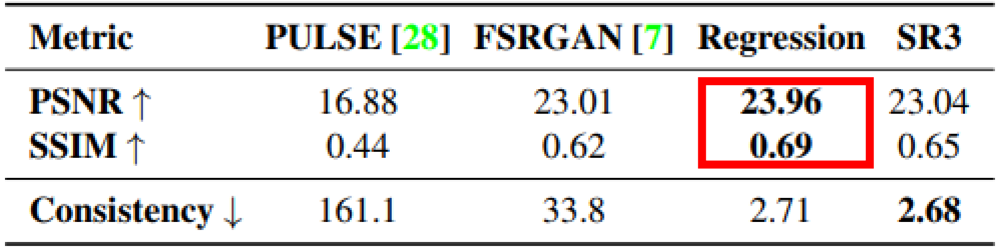


Figure 5: Comparison of different methods on the 16×16 → 128×128 face super-resolution task. Reference image has not been included because of privacy concerns. Additional results in Appendix C.5.

Experiments

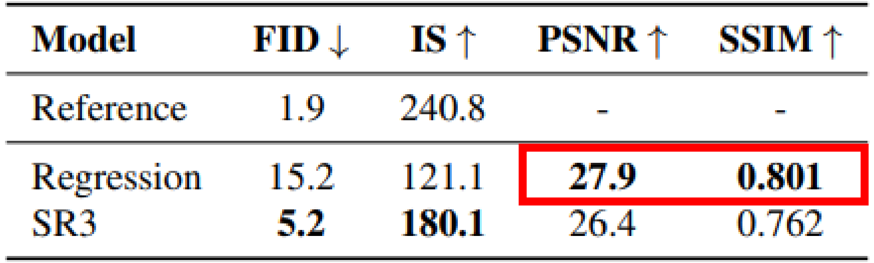
4.2. Benchmark Comparison

4.2.1 Automated metrics



Metric	PULSE [28]	FSRGAN [7]	Regression	SR3
PSNR ↑	16.88	23.01	**23.96**	23.04
SSIM ↑	0.44	0.62	**0.69**	0.65
Consistency ↓	161.1	33.8	2.71	**2.68**

Table 1: PSNR & SSIM on 16×16 → 128×128 face super-resolution. Consistency measures MSE ($\times 10^{-5}$) between the low-resolution inputs and the down-sampled super-resolution outputs.



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

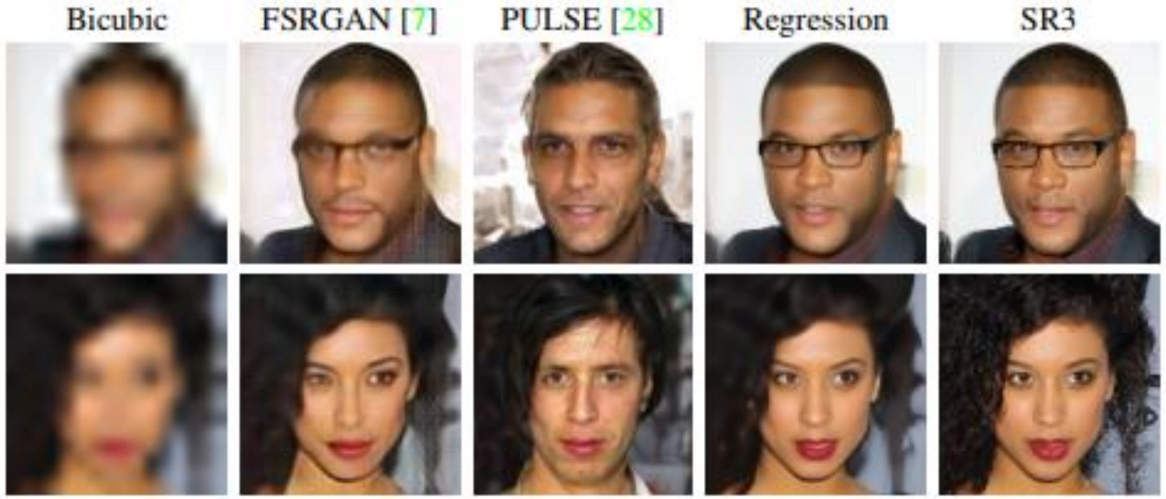


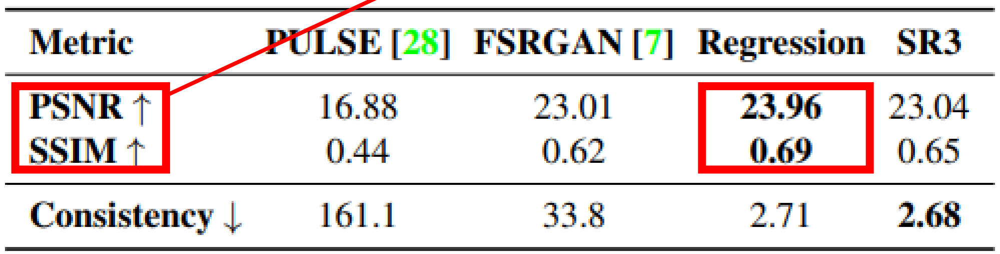
Figure 5: Comparison of different methods on the 16×16 → 128×128 face super-resolution task. Reference image has not been included because of privacy concerns. Additional results in Appendix C.5.

Experiments

4.2. Benchmark Comparison

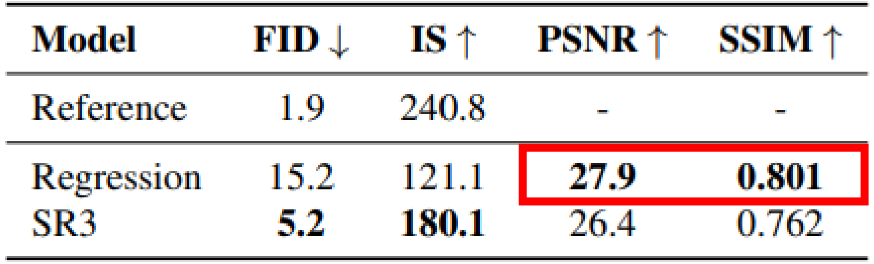
4.2.1 Automated metrics

PSNR and SSIM tend to prefer MSE regression-based techniques that are extremely conservative with high-frequency details



Metric	PULSE [28]	FSRGAN [7]	Regression	SR3
PSNR ↑	16.88	23.01	**23.96**	23.04
SSIM ↑	0.44	0.62	**0.69**	0.65
Consistency ↓	161.1	33.8	2.71	**2.68**

Table 1: PSNR & SSIM on 16×16 → 128×128 face super-resolution. Consistency measures MSE ($\times 10^{-5}$) between the low-resolution inputs and the down-sampled super-resolution outputs.



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

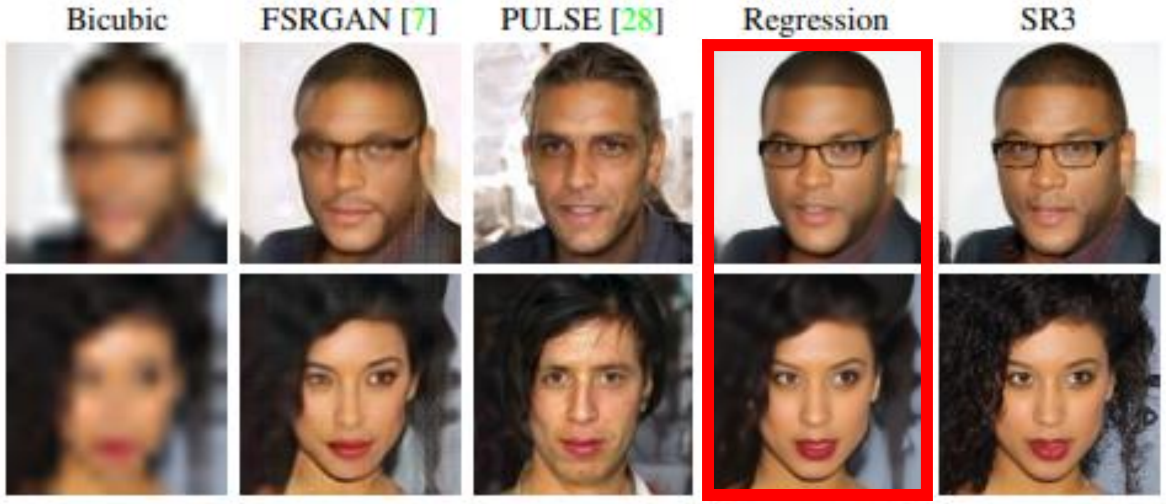


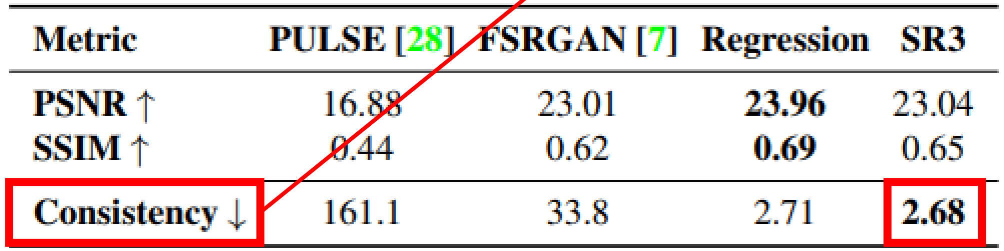
Figure 5: Comparison of different methods on the 16×16 → 128×128 face super-resolution task. Reference image has not been included because of privacy concerns. Additional results in Appendix C.5.

Experiments

4.2. Benchmark Comparison

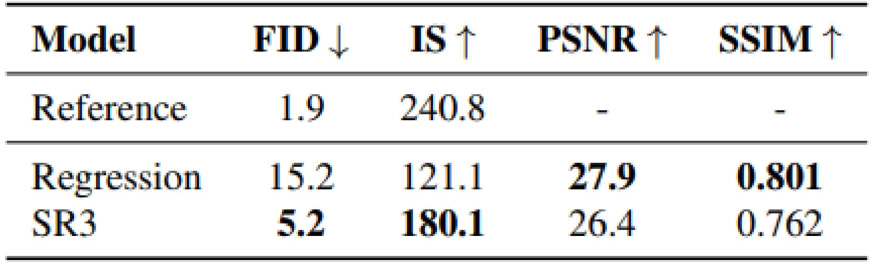
4.2.1 Automated metrics

As a measure of the consistency, we compute MSE between the downsampled outputs and the low resolution inputs



Metric	PULSE [28]	FSRGAN [7]	Regression	SR3
PSNR ↑	16.88	23.01	**23.96**	23.04
SSIM ↑	0.44	0.62	**0.69**	0.65
Consistency ↓	161.1	33.8	2.71	**2.68**

Table 1: PSNR & SSIM on $16\times16 \rightarrow 128\times128$ face super-resolution. Consistency measures MSE ($\times10^{-5}$) between the low-resolution inputs and the down-sampled super-resolution outputs.



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

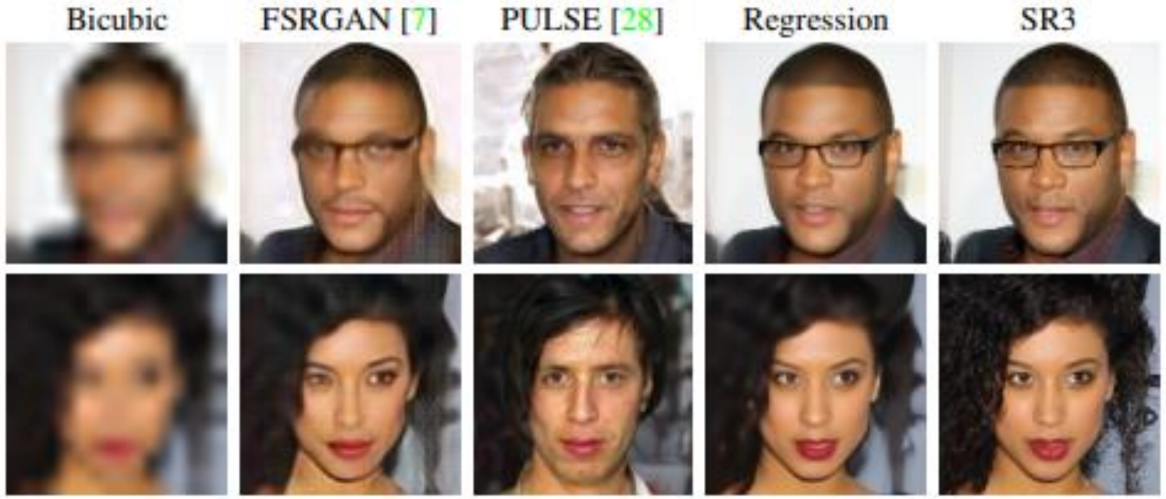
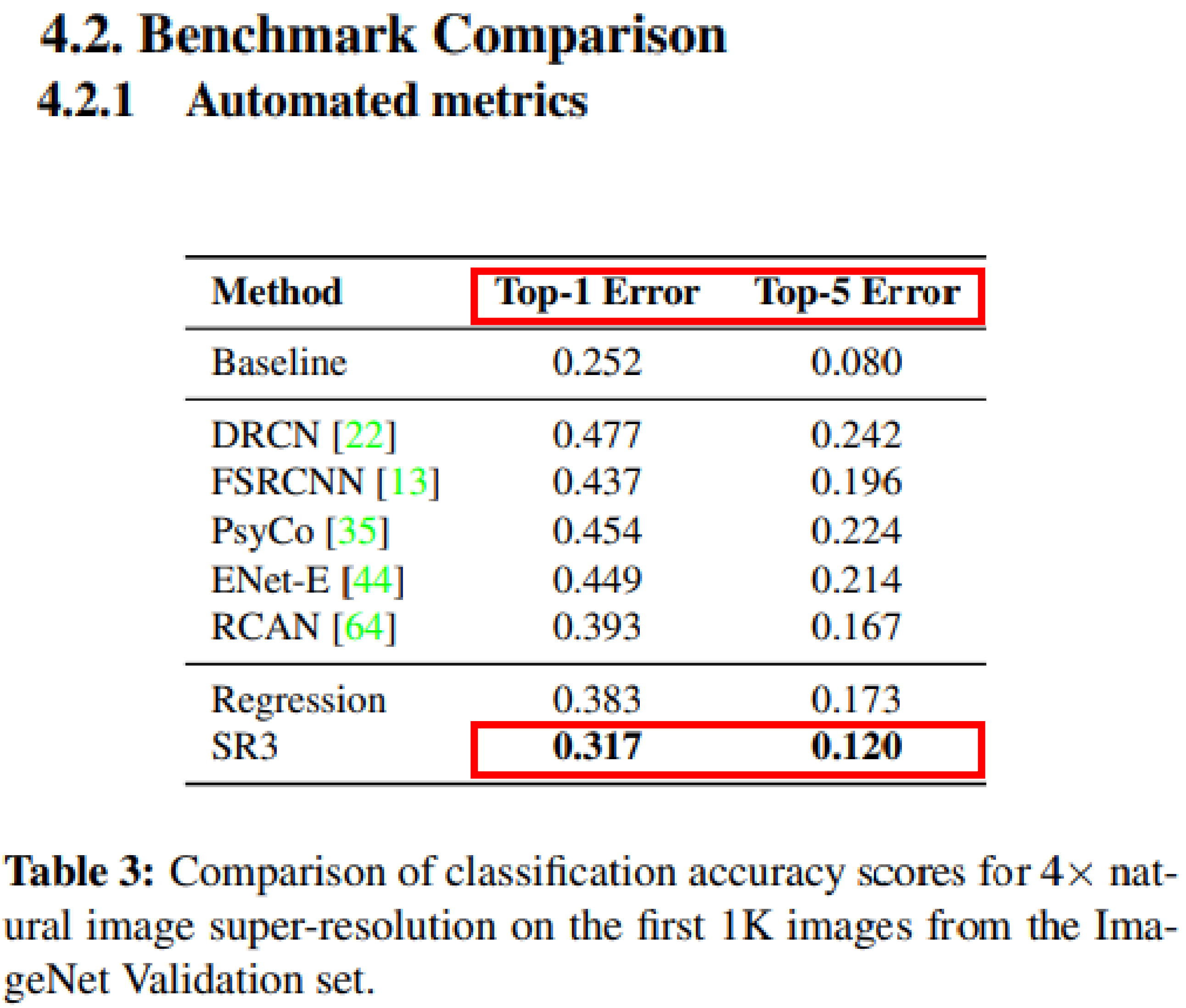


Figure 5: Comparison of different methods on the $16\times16 \rightarrow 128\times128$ face super-resolution task. Reference image has not been included because of privacy concerns. Additional results in Appendix C.5.

Experiments



4.2. Benchmark Comparison

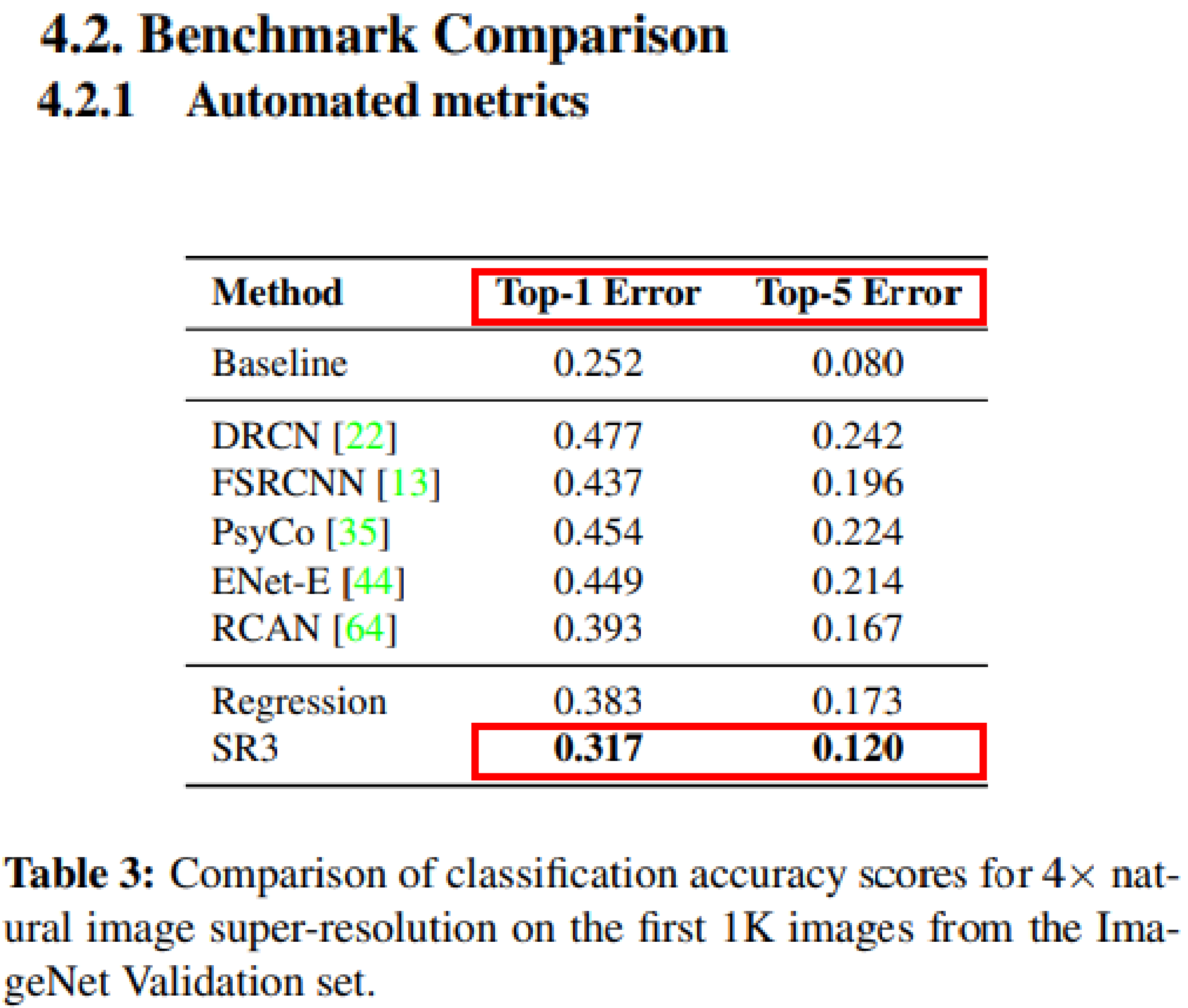
4.2.1 Automated metrics

Method	Top-1 Error	Top-5 Error
Baseline	0.252	0.080
DRCN [22]	0.477	0.242
FSRCNN [13]	0.437	0.196
PsyCo [35]	0.454	0.224
ENet-E [44]	0.449	0.214
RCAN [64]	0.393	0.167
Regression	0.383	0.173
SR3	**0.317**	**0.120**

Table 3: Comparison of classification accuracy scores for 4× natural image super-resolution on the first 1K images from the ImageNet Validation set.

224x224 ImageNet dataset image

Experiments

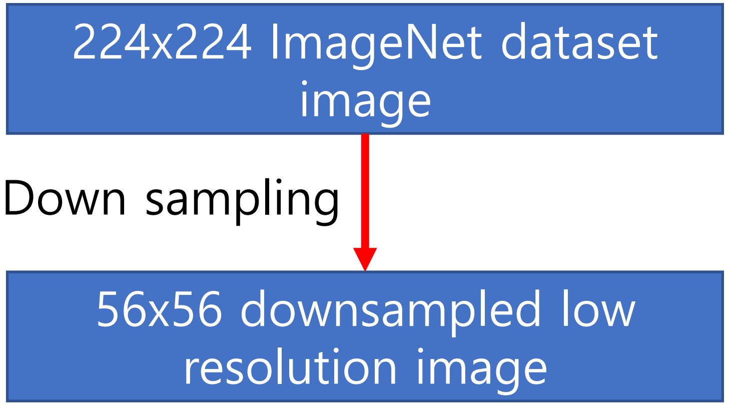


4.2. Benchmark Comparison

4.2.1 Automated metrics

Method	Top-1 Error	Top-5 Error
Baseline	0.252	0.080
DRCN [22]	0.477	0.242
FSRCNN [13]	0.437	0.196
PsyCo [35]	0.454	0.224
ENet-E [44]	0.449	0.214
RCAN [64]	0.393	0.167
Regression	0.383	0.173
SR3	**0.317**	**0.120**

Table 3: Comparison of classification accuracy scores for 4× natural image super-resolution on the first 1K images from the ImageNet Validation set.

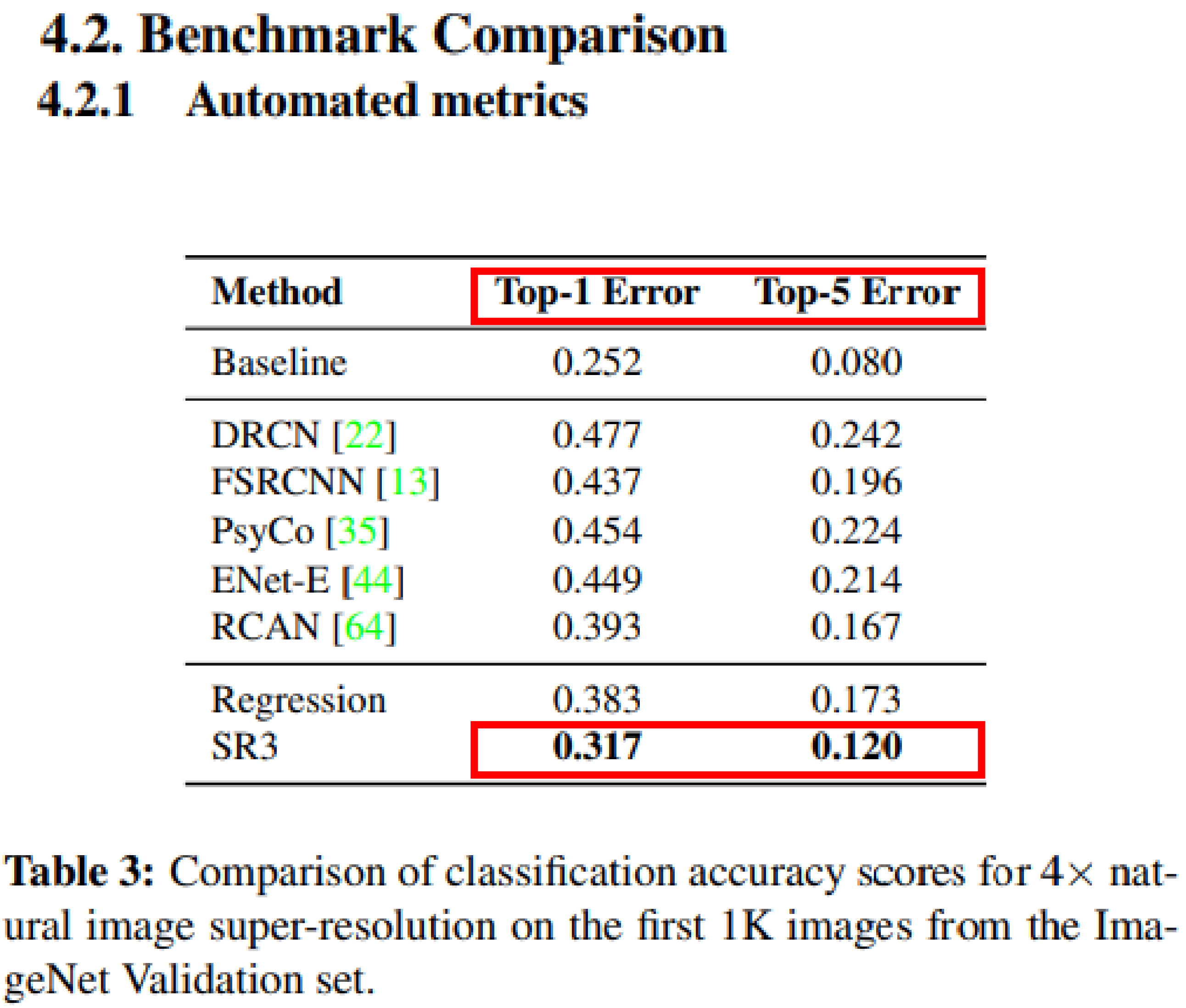


224x224 ImageNet dataset image

Down sampling

56x56 downsampled low resolution image

Experiments

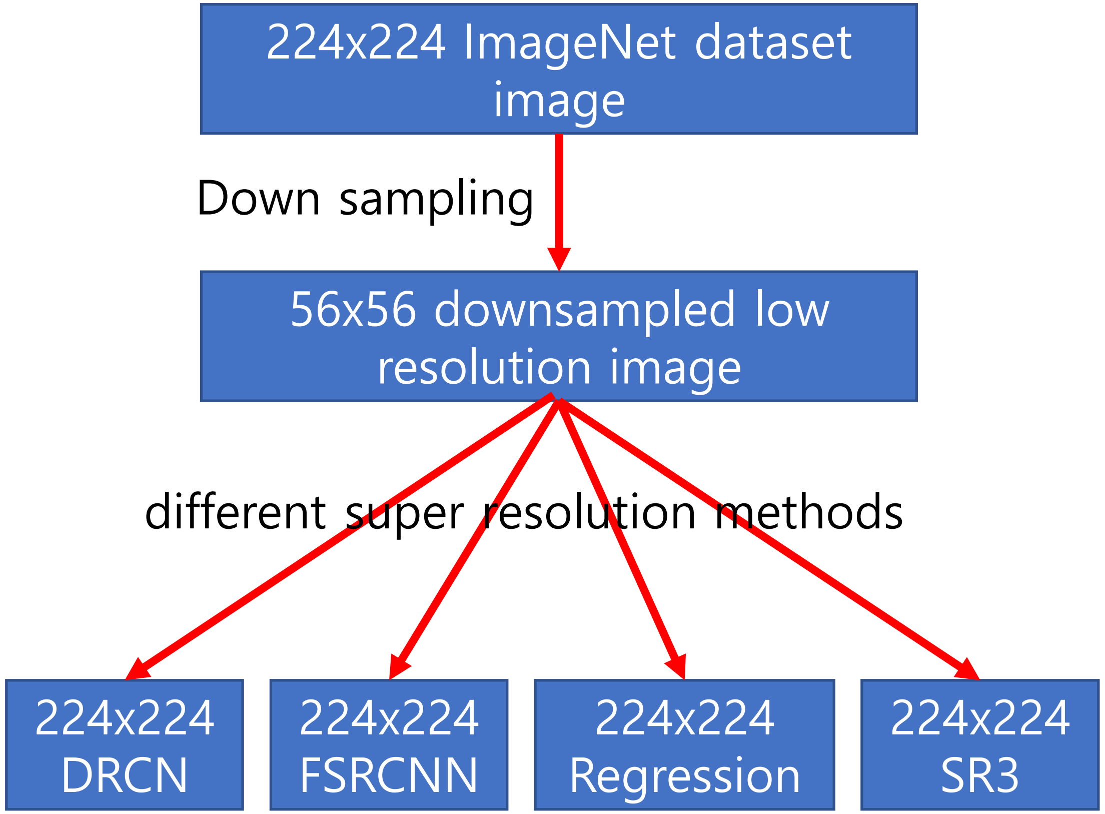


4.2. Benchmark Comparison

4.2.1 Automated metrics

Method	Top-1 Error	Top-5 Error
Baseline	0.252	0.080
DRCN [22]	0.477	0.242
FSRCNN [13]	0.437	0.196
PsyCo [35]	0.454	0.224
ENet-E [44]	0.449	0.214
RCAN [64]	0.393	0.167
Regression	0.383	0.173
SR3	**0.317**	**0.120**

Table 3: Comparison of classification accuracy scores for 4× natural image super-resolution on the first 1K images from the ImageNet Validation set.



224x224 ImageNet dataset image

Down sampling

56x56 downsampled low resolution image

different super resolution methods

224x224 DRCN

224x224 FSRCNN

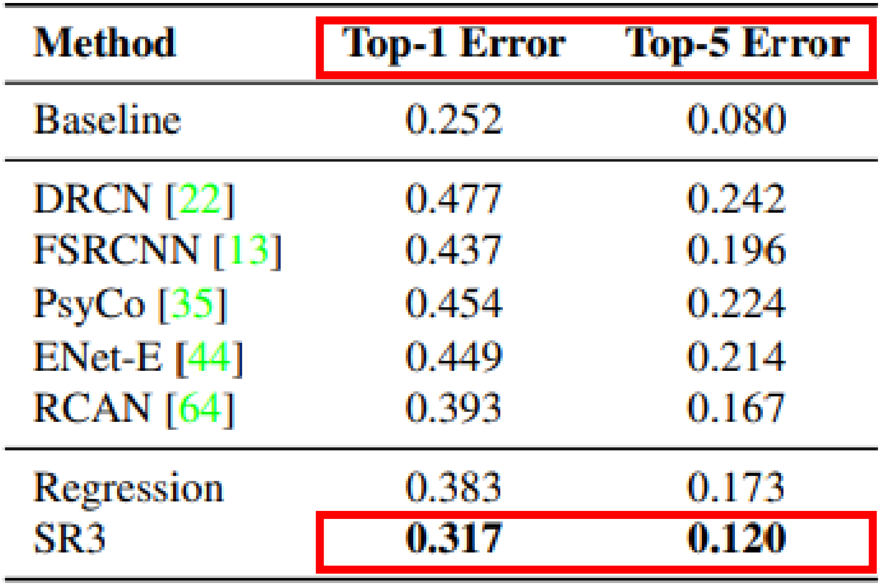
224x224 Regression

224x224 SR3

Experiments

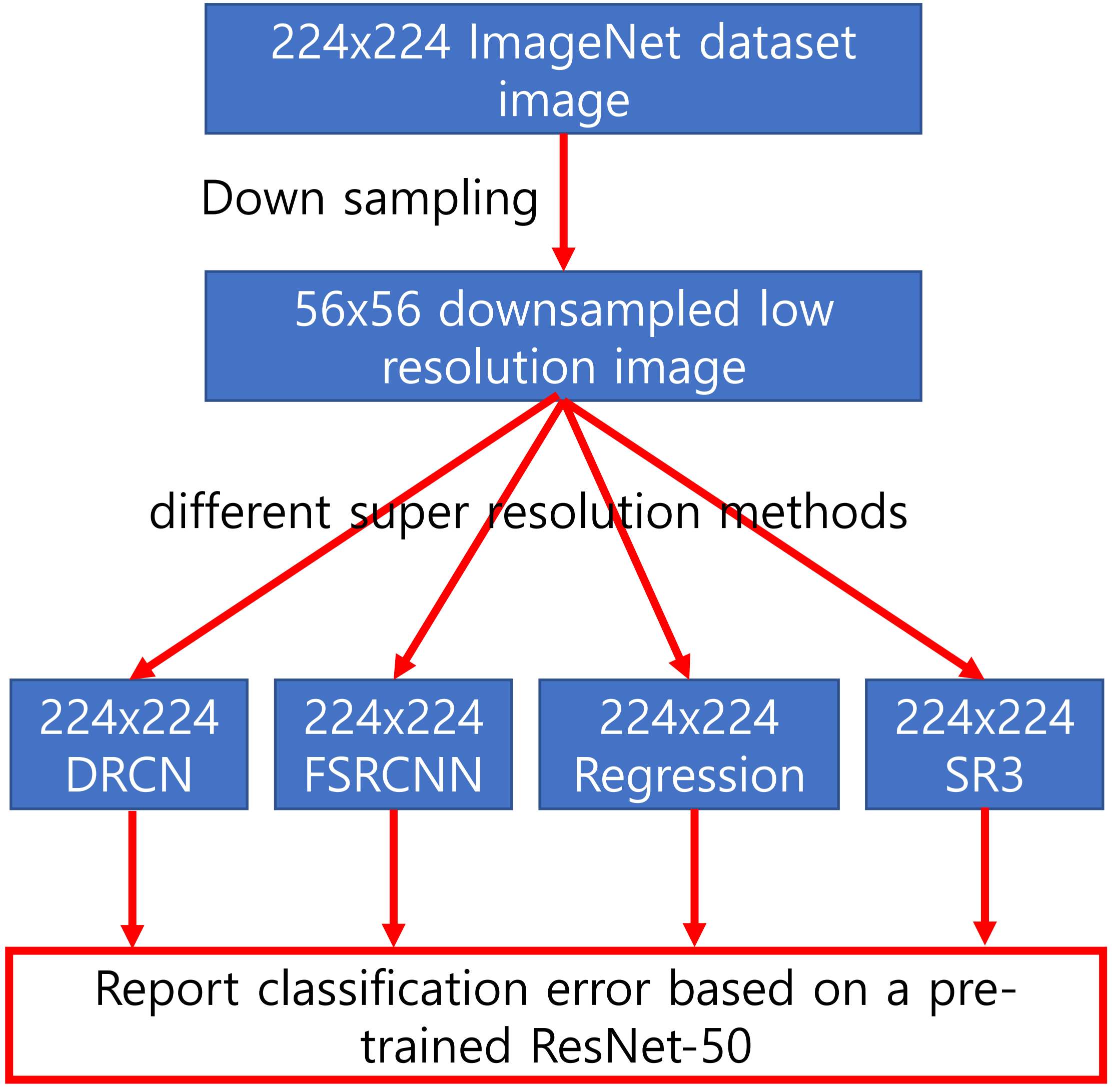
4.2. Benchmark Comparison

4.2.1 Automated metrics



Method	Top-1 Error	Top-5 Error
Baseline	0.252	0.080
DRCN [22]	0.477	0.242
FSRCNN [13]	0.437	0.196
PsyCo [35]	0.454	0.224
ENet-E [44]	0.449	0.214
RCAN [64]	0.393	0.167
Regression	0.383	0.173
SR3	**0.317**	**0.120**

Table 3: Comparison of classification accuracy scores for 4× natural image super-resolution on the first 1K images from the ImageNet Validation set.


224x224 ImageNet dataset image
Down sampling
56x56 downsampled low resolution image
different super resolution methods
224x224 DRCN
224x224 FSRCNN
224x224 Regression
224x224 SR3
Report classification error based on a pre-trained ResNet-50

Experiments

4.2. Benchmark Comparison

4.2.2 Human Evaluation (2AFC)

Human subjects are shown a low-resolution input and are required to select between a model output and a ground truth image.

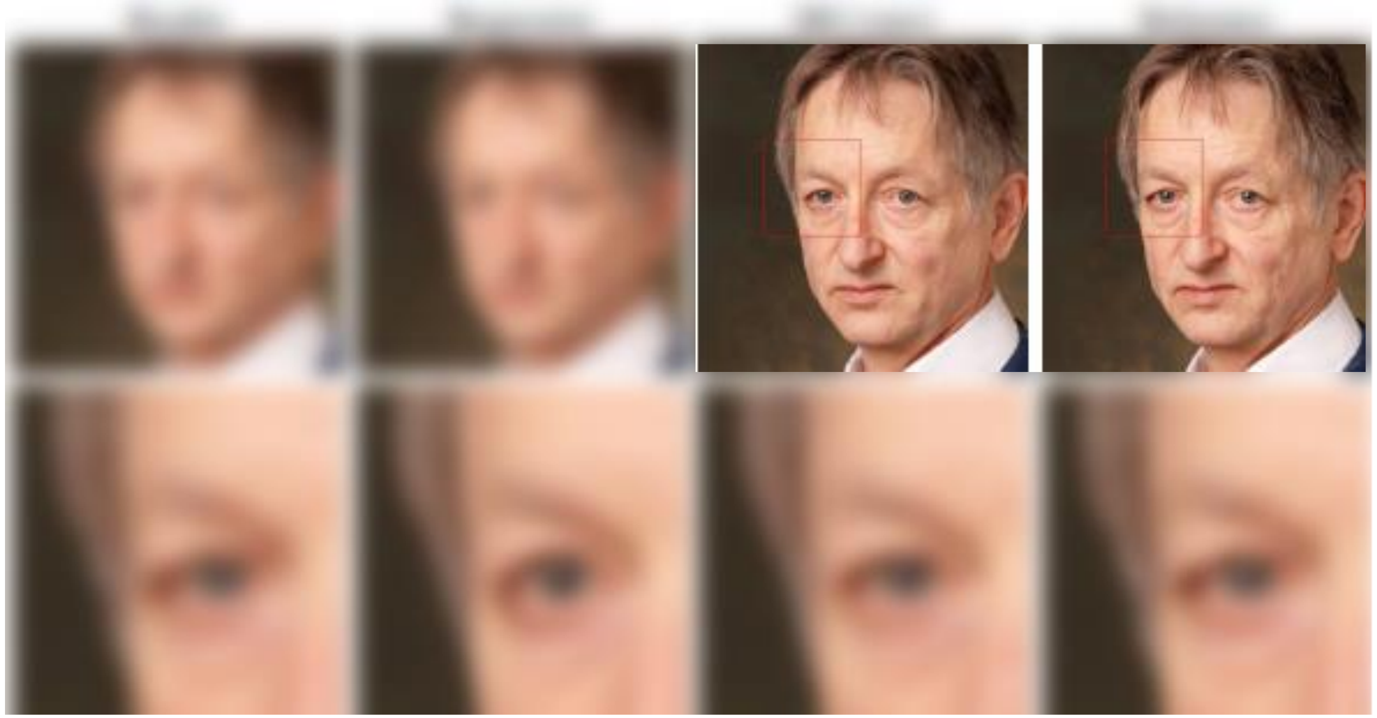
Low-resolution input

Experiments

4.2. Benchmark Comparison

4.2.2 Human Evaluation (2AFC)

Human subjects are shown a low-resolution input and are required to select between a model output and a ground truth image.

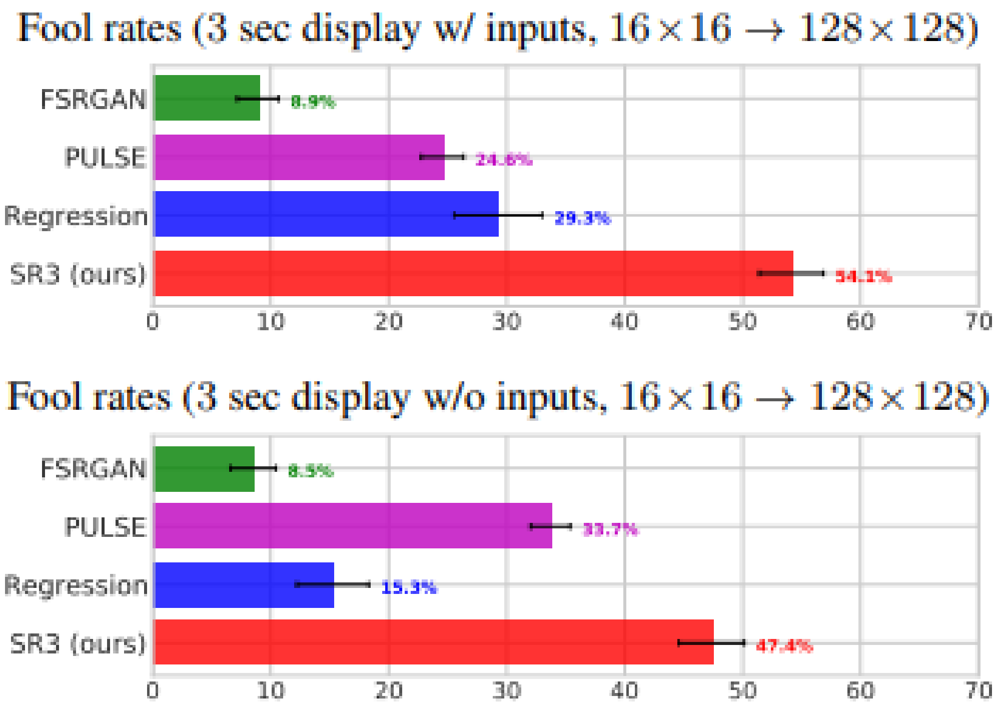


Select between Model output and ground truth

Experiments

4.2. Benchmark Comparison

4.2.2 Human Evaluation (2AFC)



Fool rates (3 sec display w/ inputs, 16 × 16 → 128 × 128)

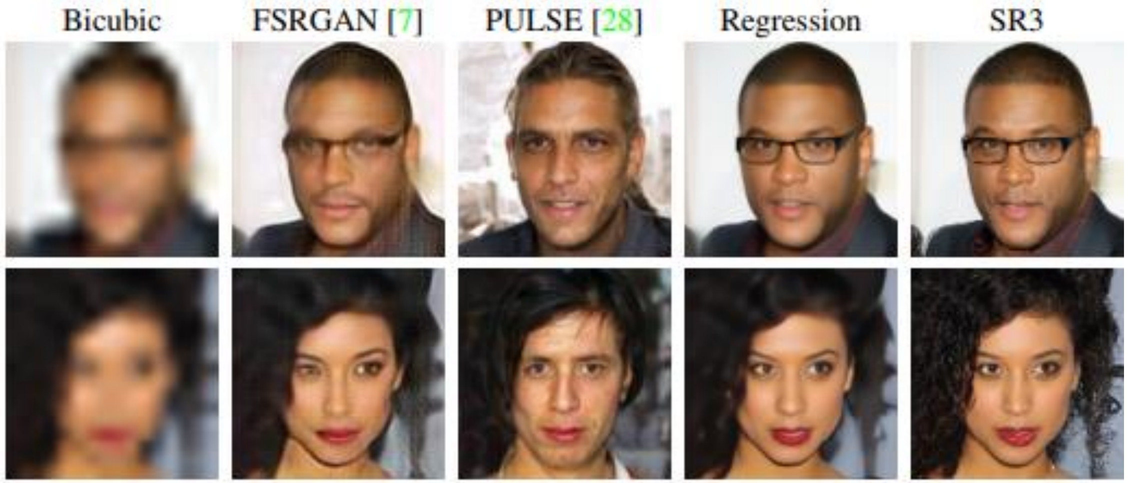
Fool rates (3 sec display w/o inputs, 16 × 16 → 128 × 128)

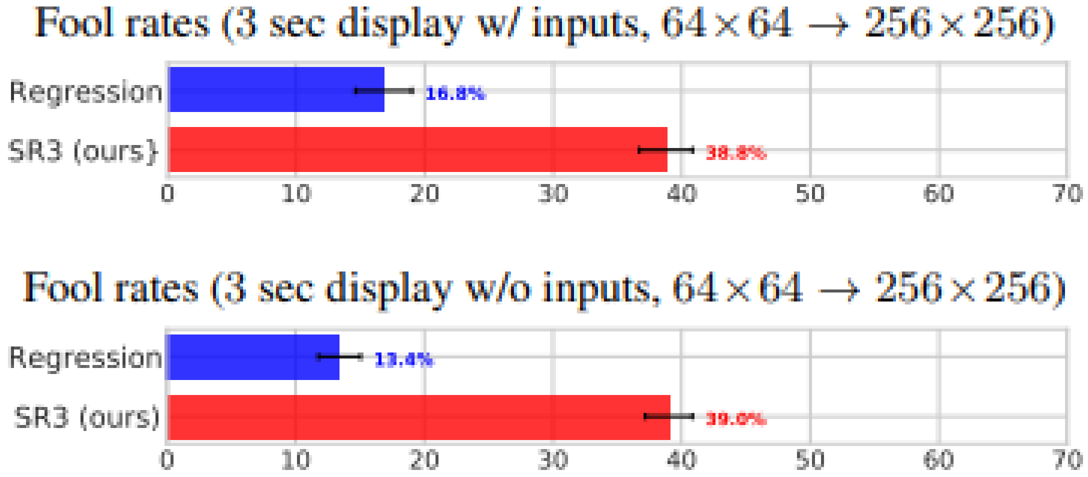
Figure 6: Face super-resolution human fool rates (higher is better, photo-realistic samples yield a fool rate of 50%). Outputs of 4 models are compared against ground truth. (top) Subjects are shown low-resolution inputs. (bottom) Inputs are not shown.

Not consistent

Consistent, but blurry

Consistent, and detail





Fool rates (3 sec display w/ inputs, 64 × 64 → 256 × 256)

Fool rates (3 sec display w/o inputs, 64 × 64 → 256 × 256)

Figure 7: 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.

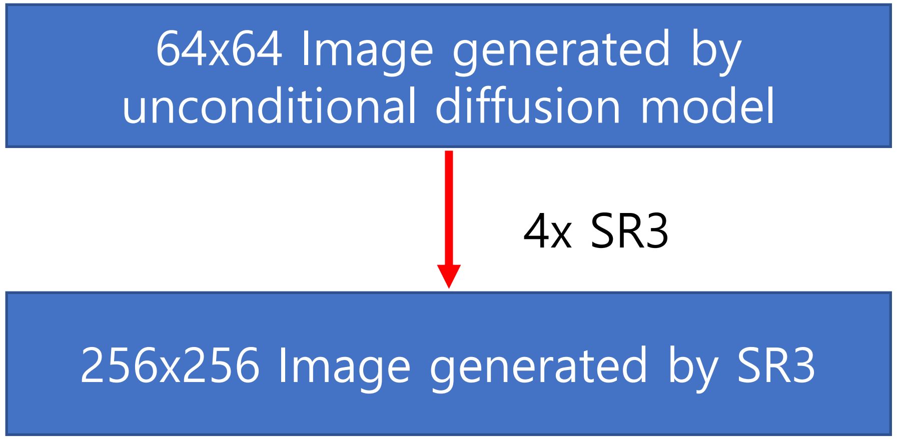
Experiments

4.3. Cascaded High-Resolution Image Synthesis

64x64 Image generated by
unconditional diffusion model

Experiments

4.3. Cascaded High-Resolution Image Synthesis



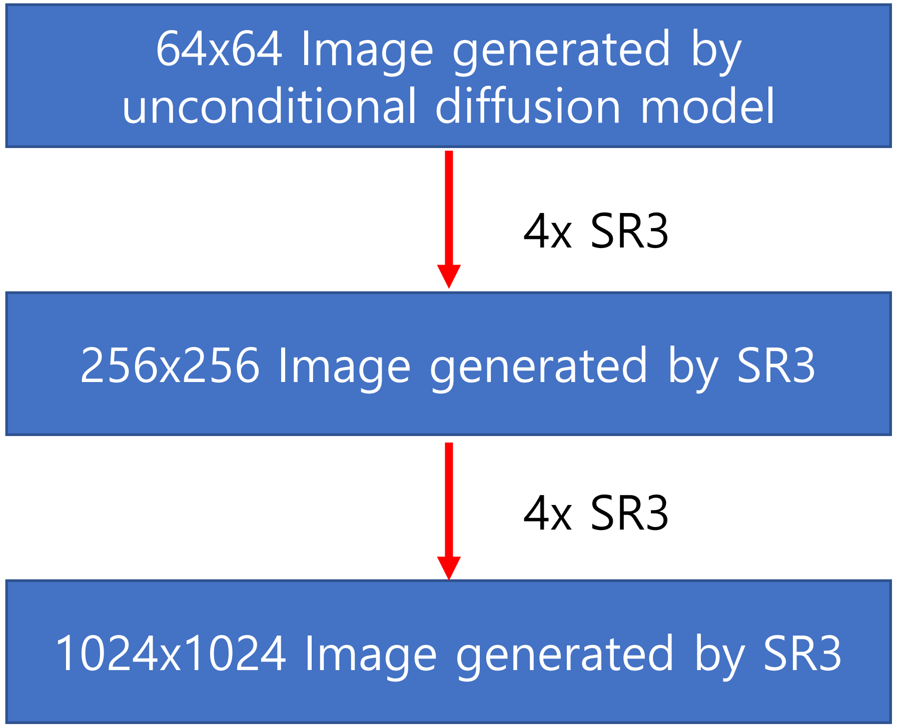
64x64 Image generated by unconditional diffusion model

4x SR3

256x256 Image generated by SR3

Experiments

4.3. Cascaded High-Resolution Image Synthesis



64x64 Image generated by unconditional diffusion model

4x SR3

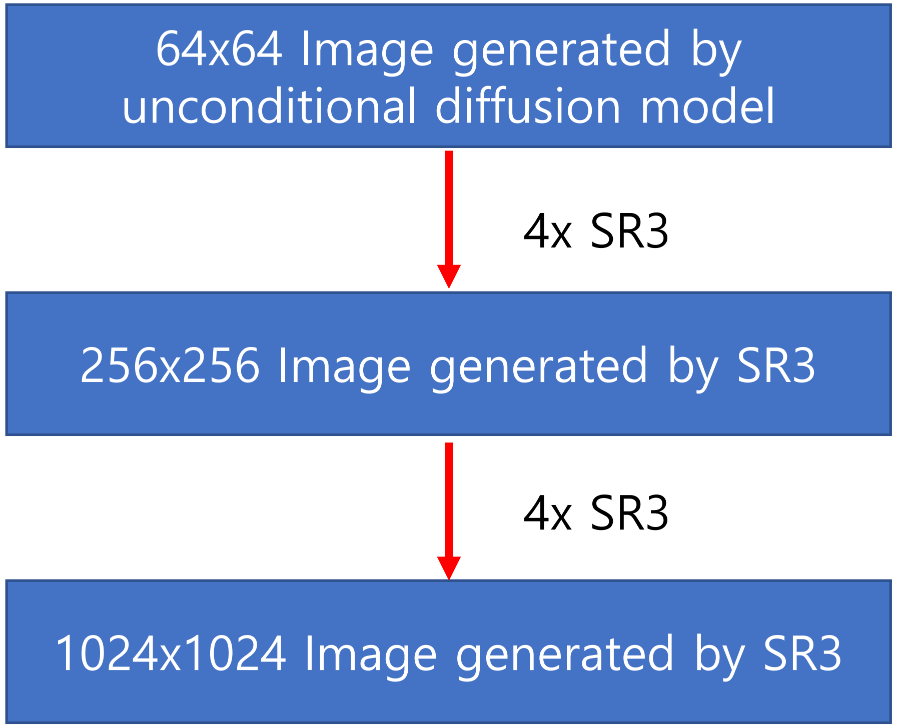
256x256 Image generated by SR3

4x SR3

1024x1024 Image generated by SR3

Experiments

4.3. Cascaded High-Resolution Image Synthesis



64x64 Image generated by unconditional diffusion model

4x SR3

256x256 Image generated by SR3

4x SR3

1024x1024 Image generated by SR3

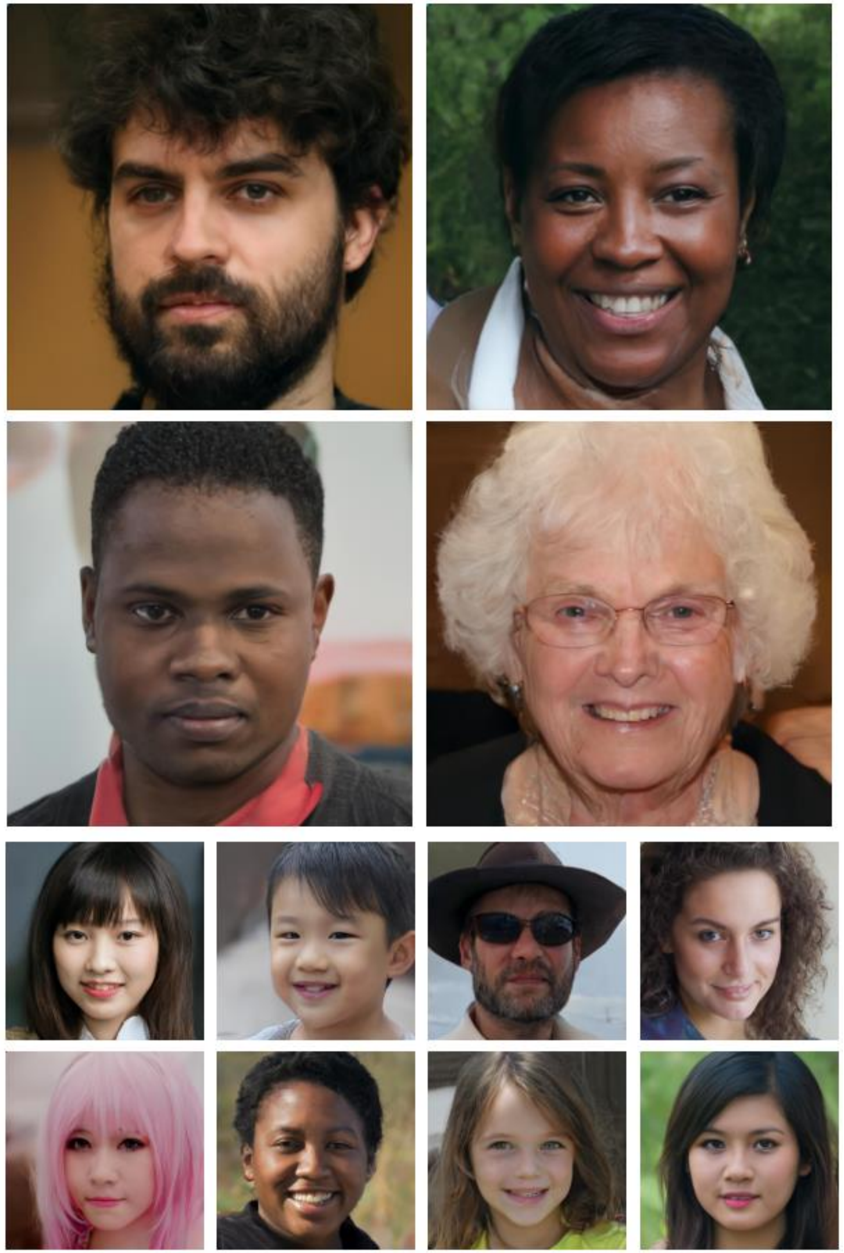


Figure 8: Synthetic 1024×1024 face images. We first sample from an unconditional 64×64 diffusion model, then pass the samples through two 4× SR3 models, *i.e.*, 64×64 → 256×256 → 1024×1024. Additional samples in Appendix C.7, C.8 and C.9.

Discussion and Conclusion

Bias is an important problem in all generative models.

SR3 also suffers from bias issues.

We believe it is likely our diffusion-based models drop modes.

We also observed the model to generate very continuous skin texture in face super-resolution, dropping moles, pimples, and pierceings found in the reference.

SR3 should not be used for any real world super-resolution tasks, until these biases are thoroughly understood and mitigated.

Discussion and Conclusion

In conclusion,

SR3 is an approach to image super-resolution via diffusion model.

SR3 can be used in a cascaded high-resolution image synthesis.

SR3 achieves a human fool rate close to 50%, suggesting photo-realistic outputs.

Q&A

Thank you