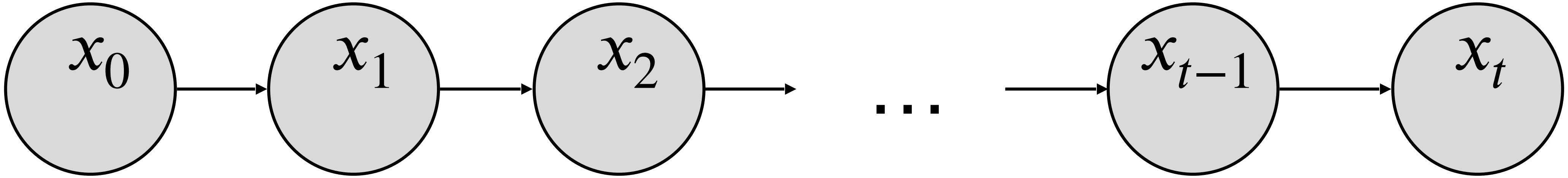
Diffusion Project
Problem Session

Task 1: Some Derivations (1.1)



$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\mathbf{z}_{t-1} \quad \text{where} \quad \mathbf{z}_{t-1} \sim \mathcal{N}(0, I)$$

Show that

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{z} \quad \text{where} \quad \mathbf{z} \sim \mathcal{N}(0, I)$$

Task 1: Some Derivations (1.2)

Score-based formulation $\Leftrightarrow$ $\hat{x}_0$-based formulation

---

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\boldsymbol{x}_t + (1 - \alpha_t)\boldsymbol{s}_\theta(x_t, t)) + \sqrt{1 - \alpha_t}\boldsymbol{z}$$

end for
return $\mathbf{x}_0$

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$

$$\hat{\boldsymbol{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t + (1 - \bar{\alpha}_t)\boldsymbol{s}_\theta(x_t, t))$$

$$\boldsymbol{x}_{t-1} = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}\boldsymbol{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)}{1 - \bar{\alpha}_t}\hat{\boldsymbol{x}}_0 + \sqrt{1 - \alpha_t}\boldsymbol{z}$$

end for
return $\mathbf{x}_0$

This is the formulation you will have to use

Task 1: Some Derivations (1.3)

Original DDPM publication $\Longleftrightarrow$ **Score-based formulation**

Original DDPM publication

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \dots, 1$ **do**
$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t + \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sqrt{1-\alpha_t}\boldsymbol{z}$$

end for
return $\mathbf{x}_0$

Score-based formulation

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \dots, 1$ **do**
$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\boldsymbol{x}_t + (1-\alpha_t)\boldsymbol{s}_\theta(x_t, t)) + \sqrt{1-\alpha_t}\boldsymbol{z}$$

end for
return $\mathbf{x}_0$

Hint: Use Tweedie's formula and the forward diffusion equation $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\hat{\mathbf{x}}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$

Task 2: Denoising

You will have to:

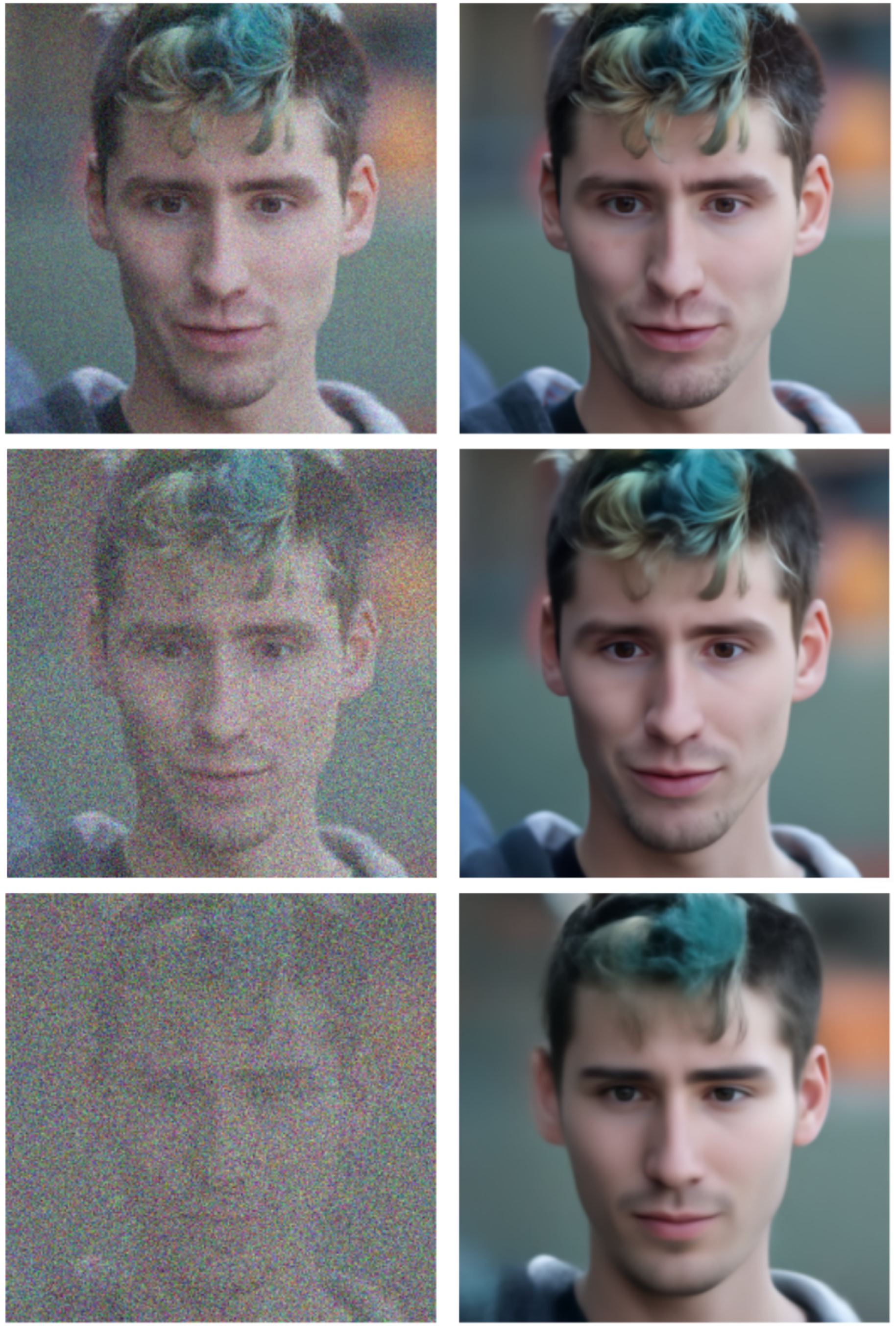
- Implement the forward (noising) process: $\boldsymbol{x}_0 \rightarrow \boldsymbol{x}_t$
- Implement the denoiser: $\boldsymbol{x}_t \rightarrow \hat{\boldsymbol{x}}_0$
- Visualize and analyze your results

➢ Hint: useful quantities $\left(\frac{1}{\sqrt{\bar{\alpha}_t}}, \frac{1-\bar{\alpha}_t}{\sqrt{\bar{\alpha}_t}}\right)$ are pre-computed

Task 2: Denoising

x_0

x_t | Denoised



	PSNR/ LPIPS
t=30	37.4 0.0378
t=100	32.7 0.0941
t=300	27.3 0.203

Task 3: Unconditional Generation

You will have to:

- Implement the denoiser using the score model: $\boldsymbol{x}_t \rightarrow \hat{\boldsymbol{x}}_0$
- Implement the backward step: $\hat{\boldsymbol{x}}_0,\ \boldsymbol{x}_t \rightarrow \boldsymbol{x}_{t-1}$
- Run unconditional generation: $\varnothing \rightarrow \boldsymbol{x}_0$
- Visualize and analyze your results

➢ Hint: useful quantities are pre-computed

Task 3: Unconditional generation

Example results

Task 4: SDEdit

$$\boldsymbol{y} \xrightarrow{\text{Add noise}} \boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\,\boldsymbol{y} + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{z} \xrightarrow{\text{Denoise}} \boldsymbol{x}_0$$

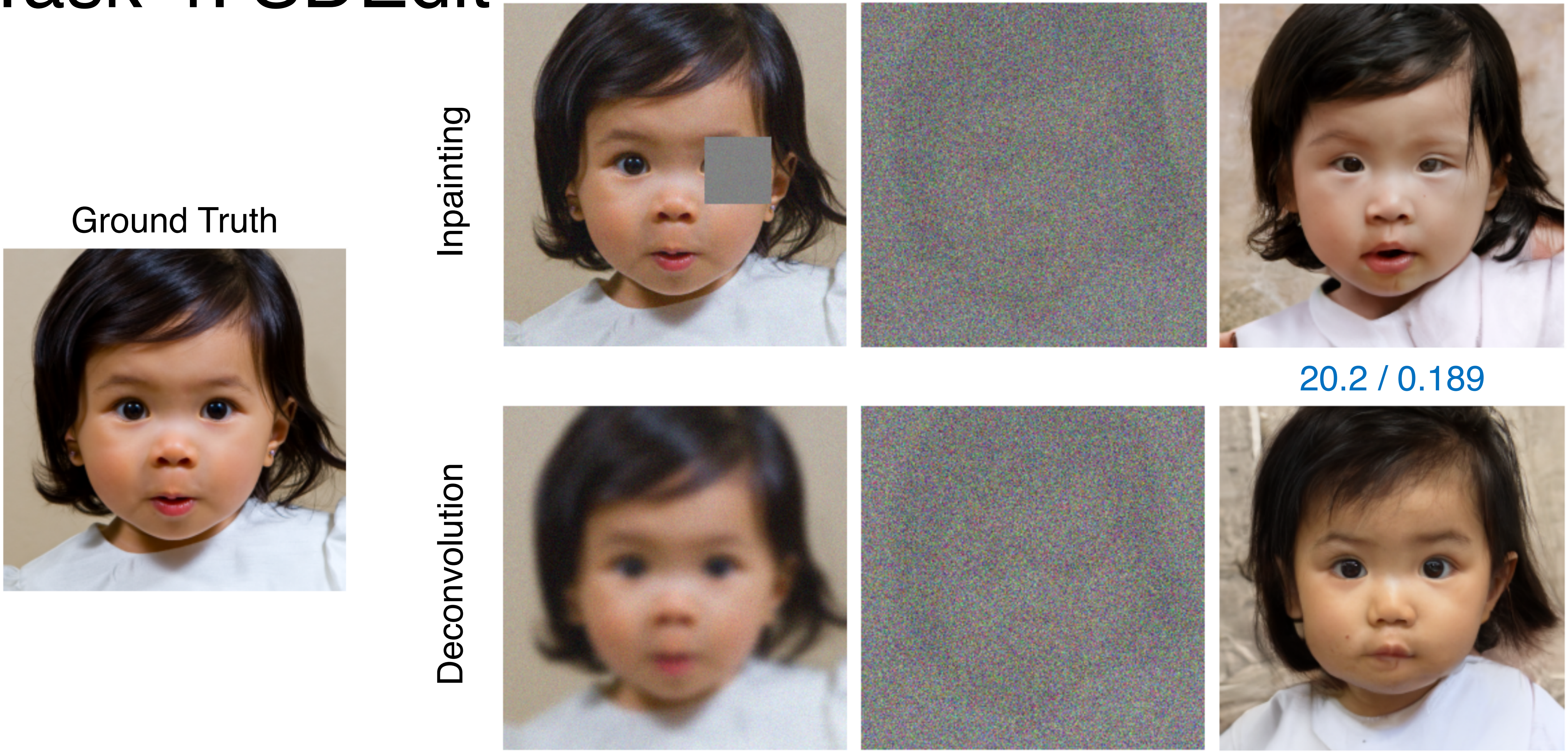
Measurement

t is a hyperparameter

You will have to:

- Implement the denoiser, the backward step
- Implement and run SDEdit: $\boldsymbol{y},\ t \rightarrow \boldsymbol{x}_0$
- Visualize and analyze your results

Task 4: SDEdit



Task 5: Score ALD

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

for $t = T, \ldots, 1$ **do**

$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$

$$\hat{\boldsymbol{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t + (1 - \bar{\alpha}_t)\boldsymbol{s}_\theta(x_t, t))$$

$$\boldsymbol{x}_{t-1} = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}\boldsymbol{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)}{1 - \bar{\alpha}_t}\hat{\boldsymbol{x}}_0 + \sqrt{1 - \alpha_t}\boldsymbol{z}$$

$$\boldsymbol{x}_{t-1} = \boldsymbol{x}_{t-1} - \frac{1}{2(\sigma^2 + \gamma_t^2)} \nabla_{\boldsymbol{x}_t} \left\| \mathscr{A}(\boldsymbol{x}_t) - \boldsymbol{y} \right\|^2$$

end for

return $\mathbf{x}_0$

Task 5: Score ALD

You will have to:

- Implement the denoiser, the backward step
- Implement the gradient computation (with `torch.autograd`)
- Implement and run ScoreALD: $\boldsymbol{y},\ \sigma, \gamma_t, \mathscr{A} \rightarrow \boldsymbol{x}_0$
- Visualize and analyze your results

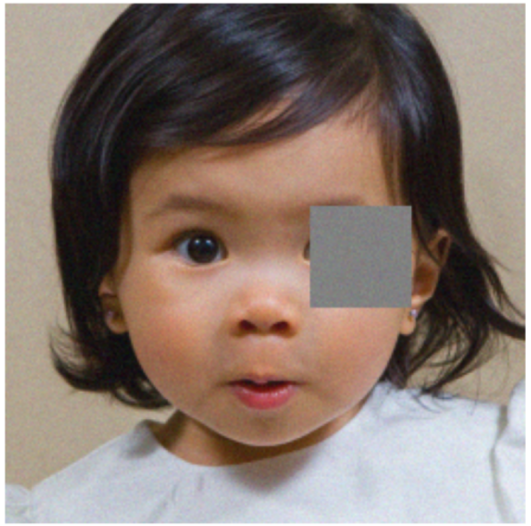
Task 5: Score ALD

PSNR/LPIPS

Input

Reconstruction

Inpainting



28.5 / 0.0605

Ground Truth

Deconvolution

22.2 / 0.144

Task 6: DPS

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

for $t = T, \ldots, 1$ **do**

$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$

$$\hat{\boldsymbol{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t + (1 - \bar{\alpha}_t)\boldsymbol{s}_\theta(x_t, t))$$

$$\boldsymbol{x}'_{t-1} = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}\boldsymbol{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)}{1 - \bar{\alpha}_t}\hat{\boldsymbol{x}}_0 + \sqrt{1 - \alpha_t}\boldsymbol{z}$$

$$\boldsymbol{x}_{t-1} = \boldsymbol{x}'_{t-1} - \zeta_t \nabla_{\boldsymbol{x}_t} \left\| \mathscr{A}\left(\hat{\boldsymbol{x}}_0\right) - \boldsymbol{y} \right\|^2$$

end for

return $\mathbf{x}_0$

In practice, $\zeta_t = \dfrac{\zeta}{\left\| \nabla_{\boldsymbol{x}_t} \left\| \mathscr{A}\left(\hat{\boldsymbol{x}}_0\right) - \boldsymbol{y} \right\|^2 \right\|}$ works well (see DPS paper p16)

Task 6: DPS

You will have to:

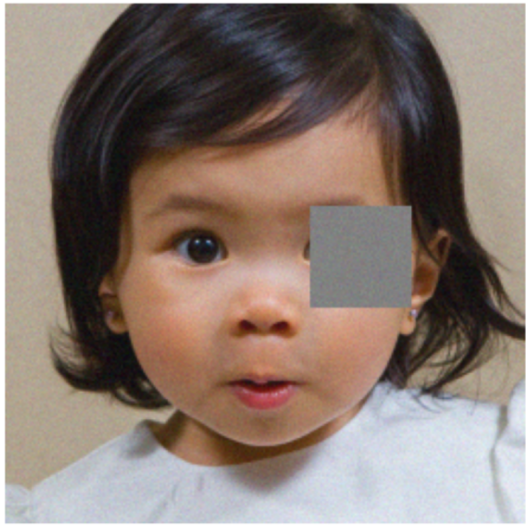
- Implement the denoiser, the backward step
- Implement the gradient computation (with `torch.autograd`)
- Implement and run DPS: $\boldsymbol{y},\ \sigma, \zeta, \mathscr{A} \rightarrow \boldsymbol{x}_0$
- Visualize and analyze your results

Task 6: DPS

Ground Truth

Input | PSNR/LPIPS Reconstruction

Inpainting



34.0 / 0.0239

Deconvolution

28.4 / 0.0518