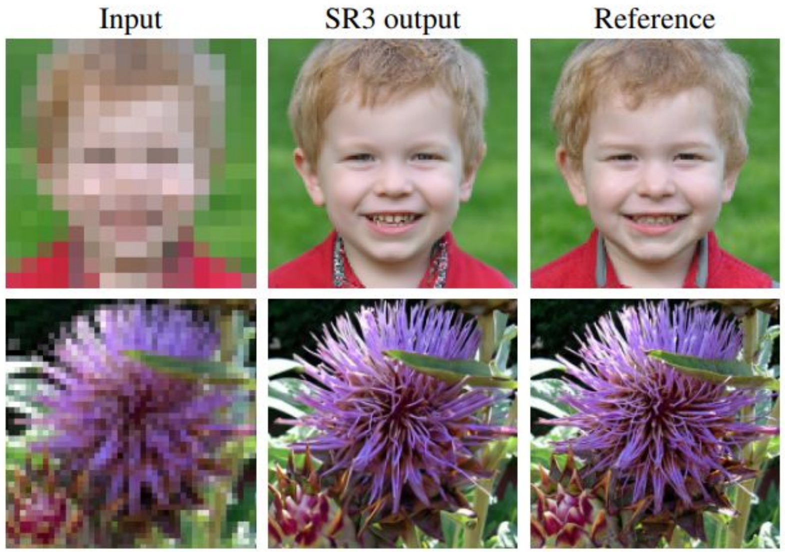
Image Super-Resolution via Iterative Refinement (SR3)

Chaeeun Ryu



Korea Institute of Science and Technology

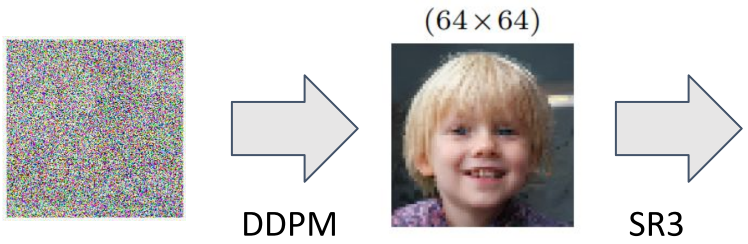
Super Resolution

64x64

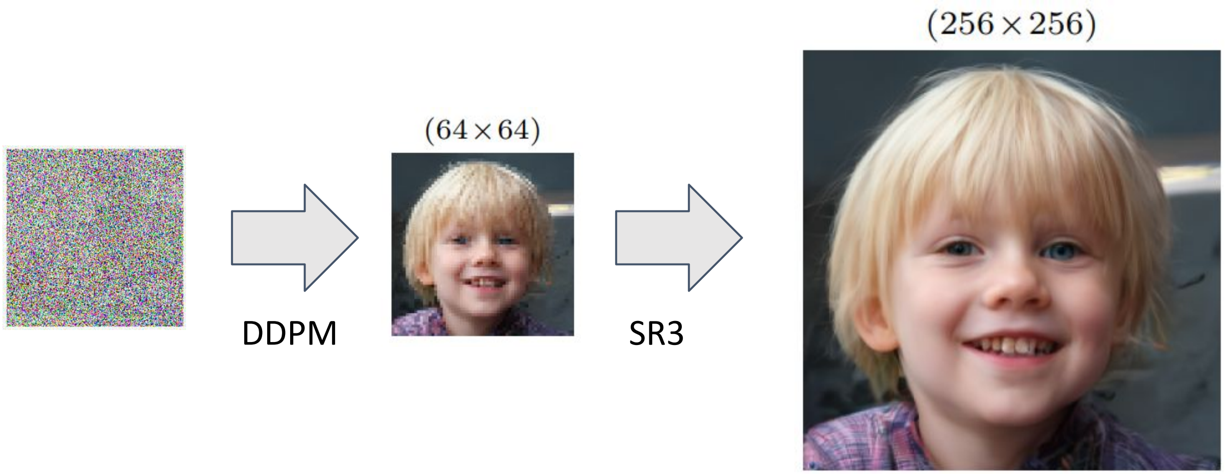
Generate more high quality image!

1024x1024

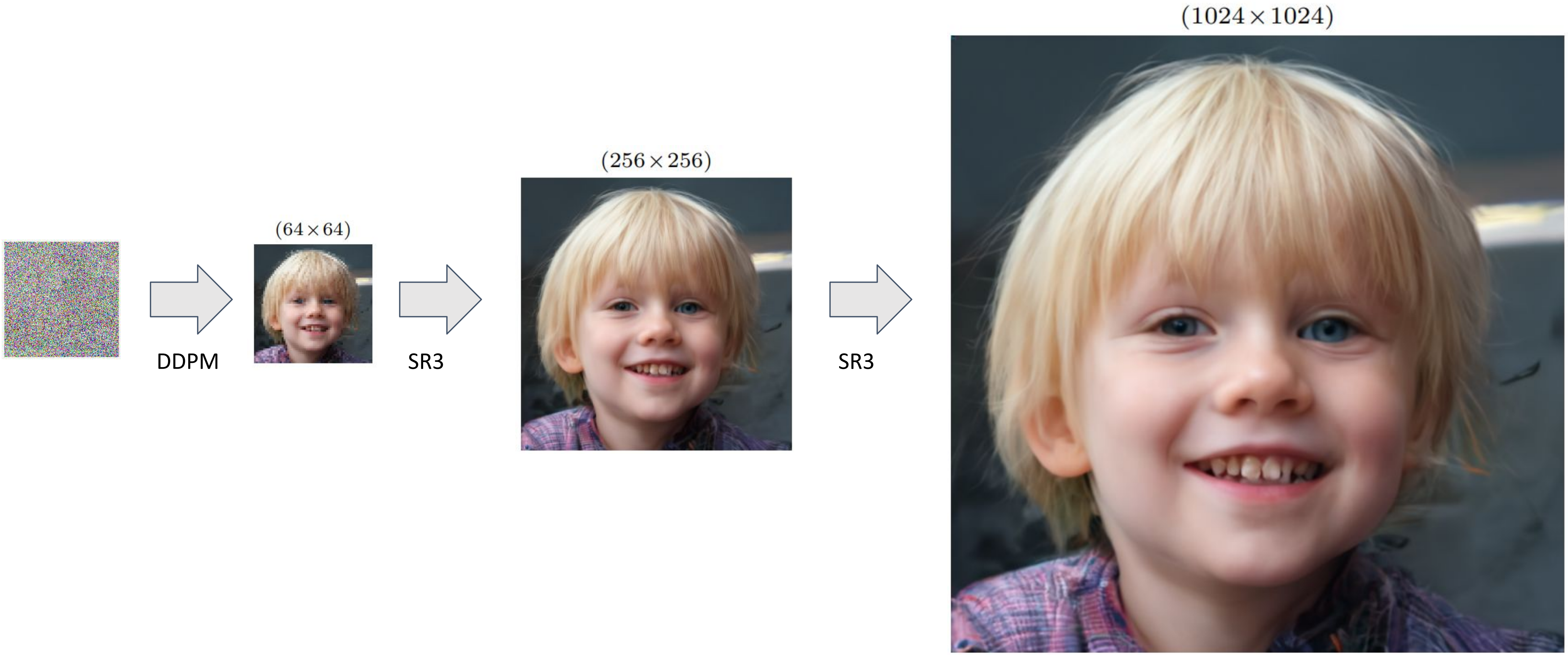
Cascaded High Resolution Synthesis



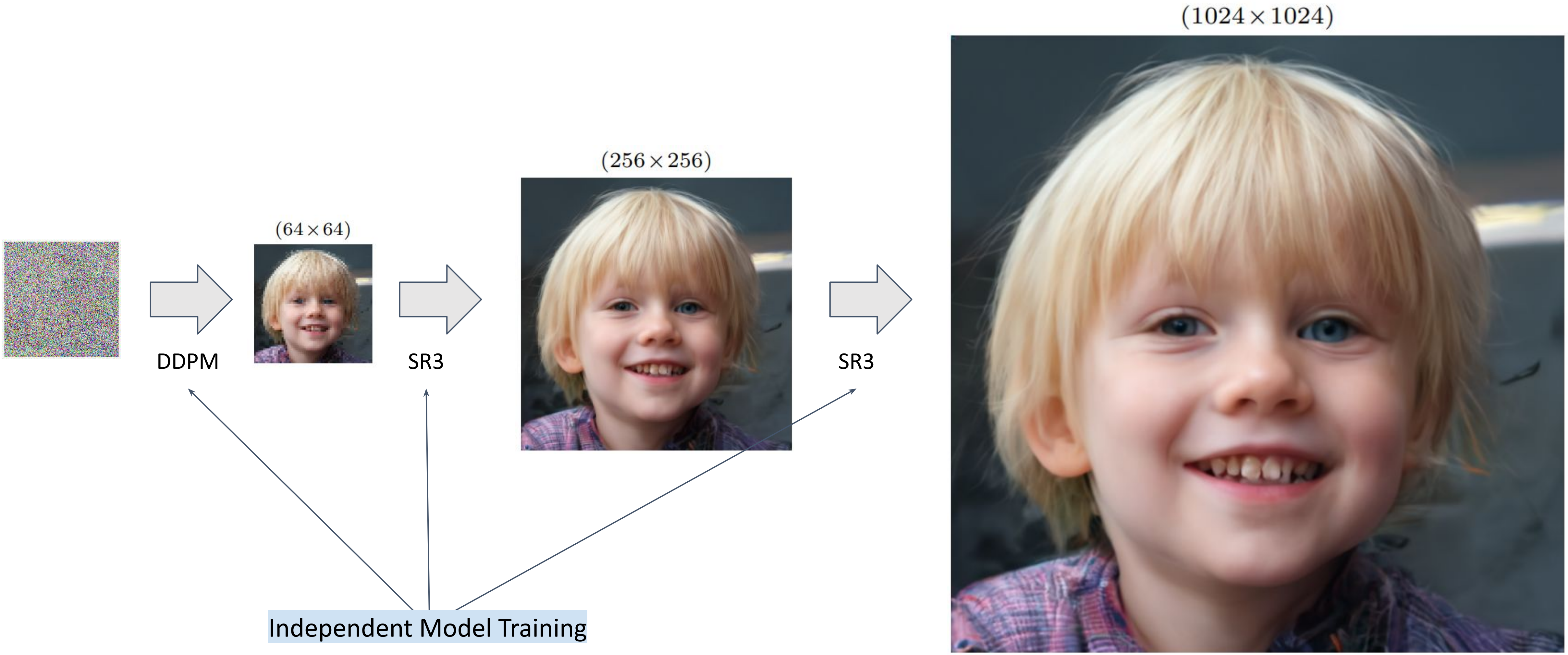
Cascaded High Resolution Synthesis



Cascaded High Resolution Synthesis



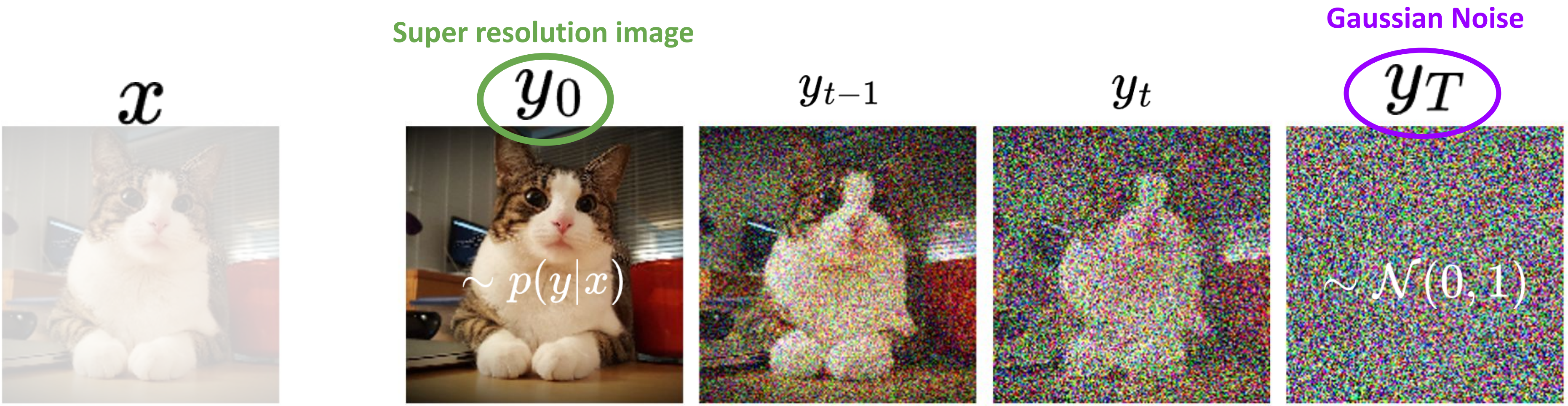
Cascaded High Resolution Synthesis



Methodology

x : source image, y: target image

- Dataset $D = \{x_i, y_i\}_{i=1}^{N}$ which represent samples drawn from an unknown conditional distribution $p(y|x)$ which represents one-to-many mapping
- Goal: Learn $\boxed{p(y|x)}$!! ← Adaptation of DDPM to conditional image generation

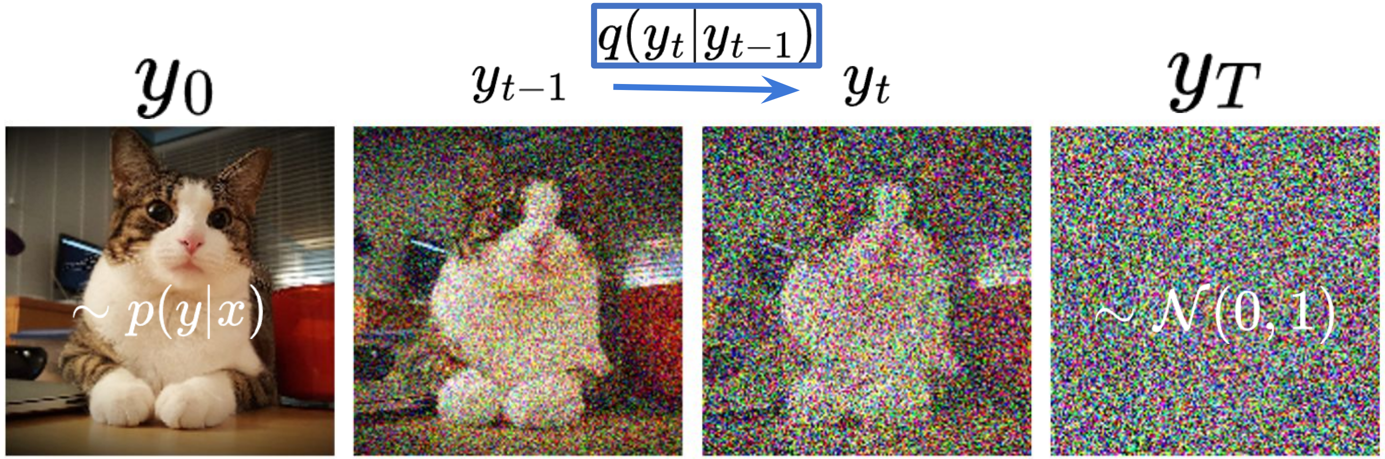


Korea Institute of Science and Technology

Methodology

x : source image, y: target image

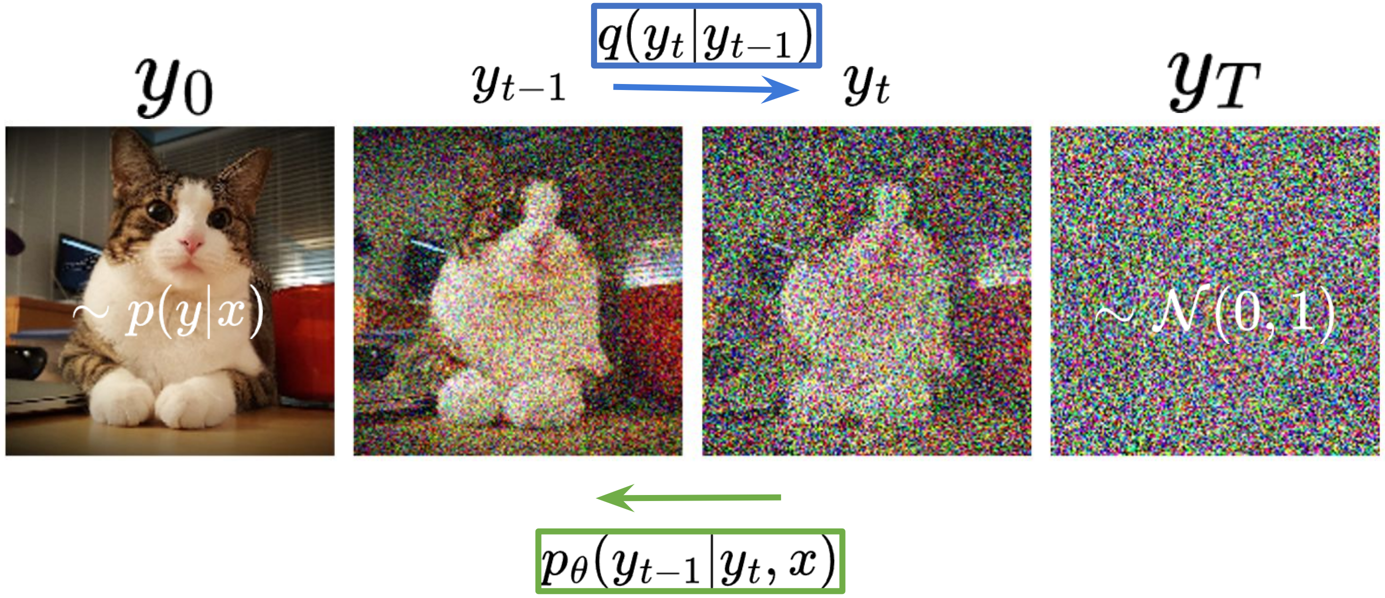
- Dataset $D = \{x_i, y_i\}_{i=1}^{N}$ which represent samples drawn from an unknown conditional distribution $p(y|x)$ which represents one-to-many mapping
- Goal: Learn $\boxed{p(y|x)}$!! ← Adaptation of DDPM to conditional image generation



Methodology

x : source image, y: target image

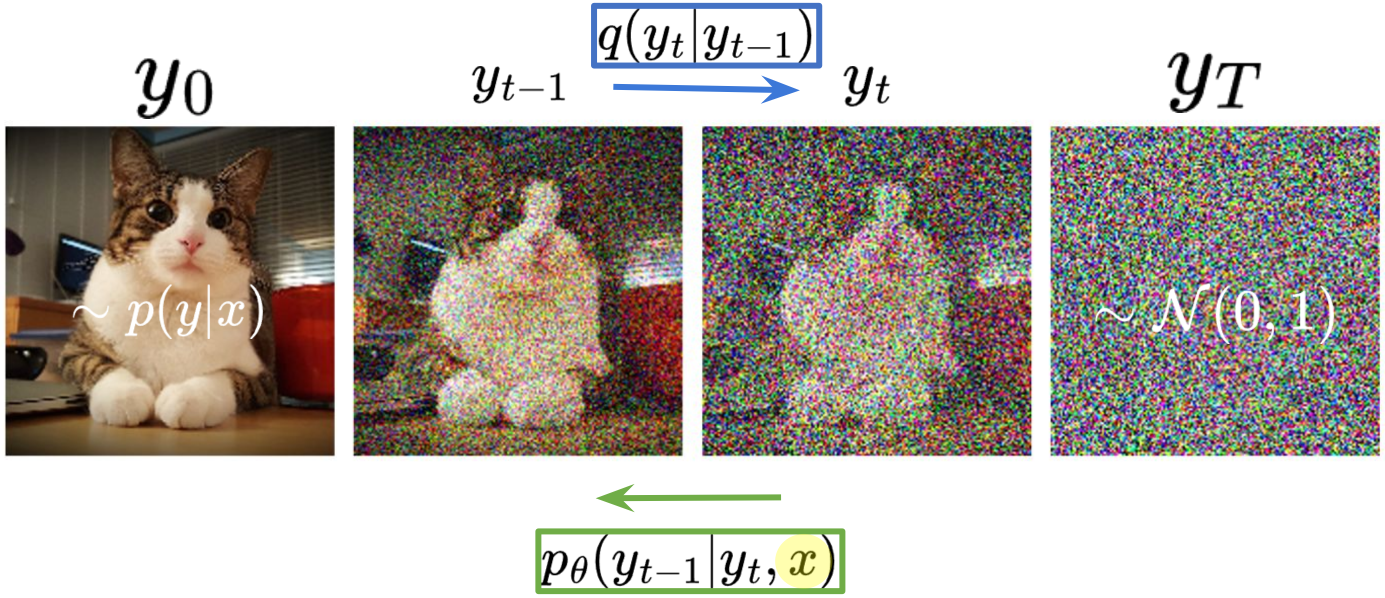
- Dataset $D = \{x_i, y_i\}_{i=1}^{N}$ which represent samples drawn from an unknown conditional distribution $p(y|x)$ which represents <u>one-to-many mapping</u>
- Goal: Learn $\boxed{p(y|x)}$!! ← Adaptation of DDPM to conditional image generation



Methodology

x : source image, y: target image

- Dataset $D = \{x_i, y_i\}_{i=1}^{N}$ which represent samples drawn from an unknown conditional distribution $p(y|x)$ which represents one-to-many mapping
- Goal: Learn $\boxed{p(y|x)}$!! ← Adaptation of DDPM to conditional image generation

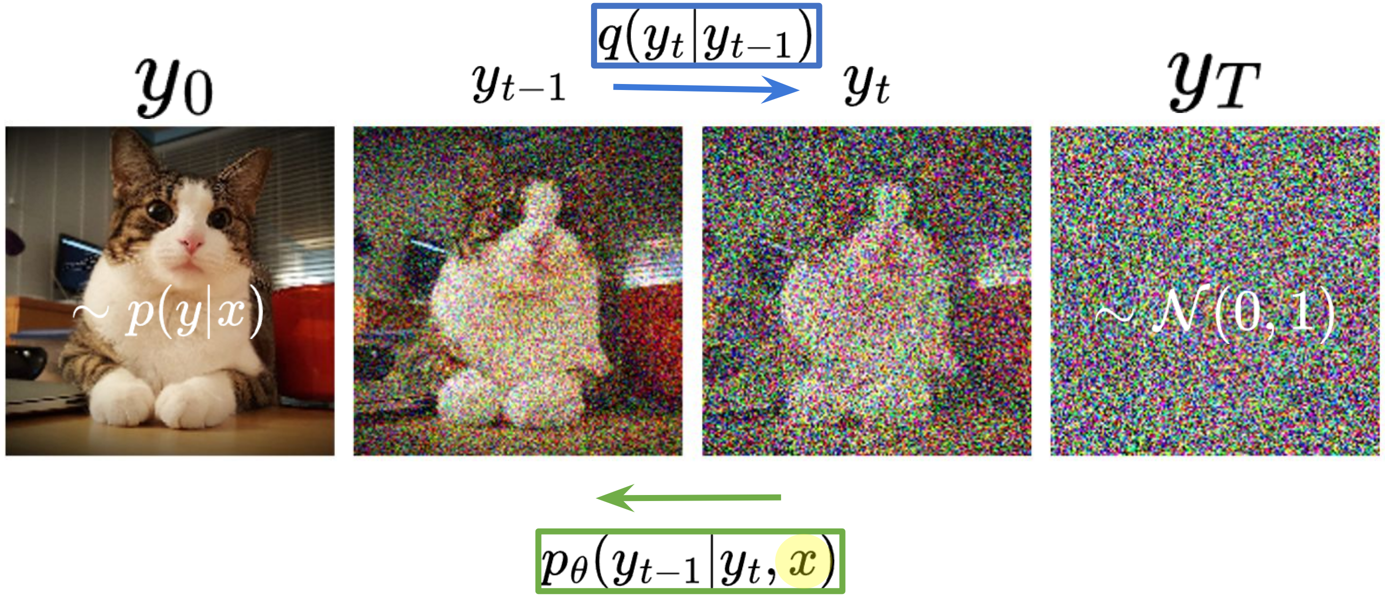


Korea Institute of Science and Technology

Methodology: Forward Process

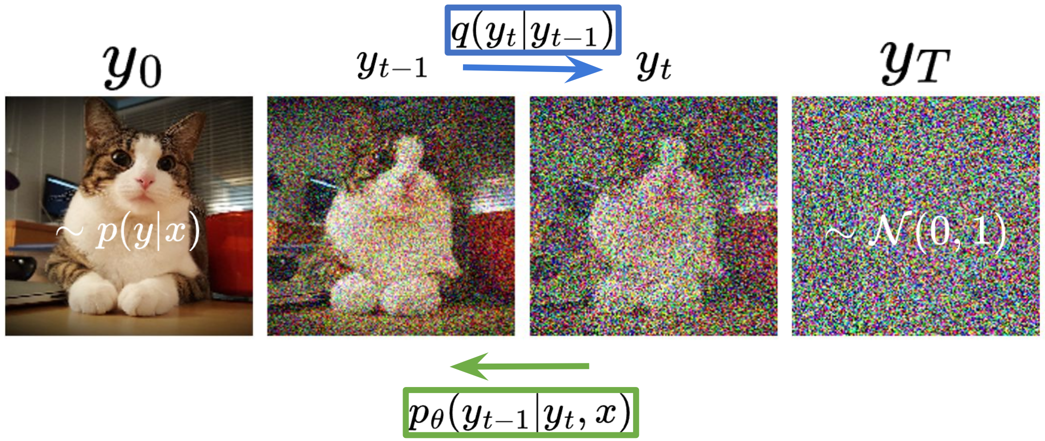
x : source image, y: target image

$$\boxed{\begin{aligned} q(\boldsymbol{y}_{1:T} \mid \boldsymbol{y}_0) &= \prod\nolimits_{t=1}^{T} q(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}) \,, \\ q(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}) &= \mathcal{N}(\boldsymbol{y}_t \mid \sqrt{\alpha_t}\, \boldsymbol{y}_{t-1}, (1-\alpha_t)\boldsymbol{I}) \end{aligned}}$$



Methodology: Forward Process

x : source image, y: target image



$$q(\boldsymbol{y}_{1:T} \mid \boldsymbol{y}_0) = \prod_{t=1}^{T} q(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}) ,$$
$$q(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}) = \mathcal{N}(\boldsymbol{y}_t \mid \sqrt{\alpha_t}\, \boldsymbol{y}_{t-1}, (1-\alpha_t)\boldsymbol{I})$$

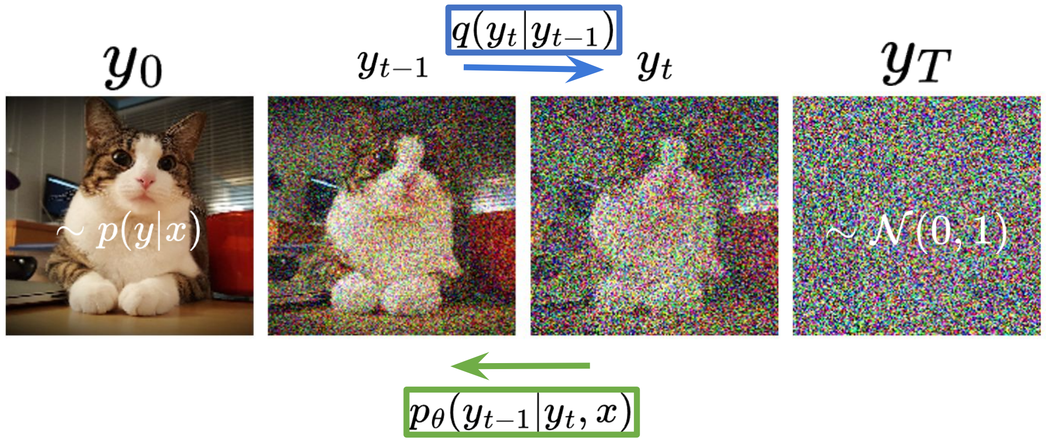
Original DDPM Notation

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

Korea Institute of Science and Technology

Methodology: Forward Process

x : source image, y: target image



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

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

Original DDPM Notation

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

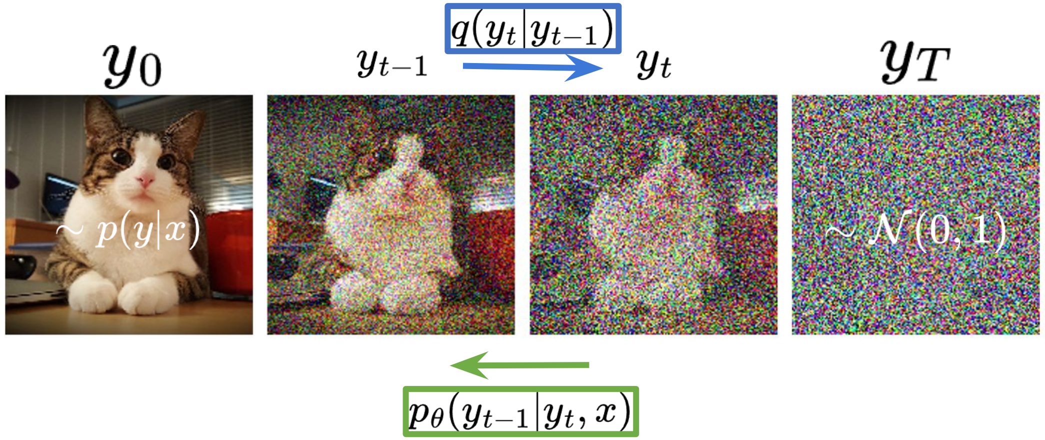
$$\gamma_t \;=\; \prod_{i=1}^{t} \alpha_i \qquad q(\boldsymbol{y}_t \mid \boldsymbol{y}_0) \;=\; \mathcal{N}(\boldsymbol{y}_t \mid \sqrt{\gamma_t}\, \boldsymbol{y}_0, (1-\gamma_t)\boldsymbol{I})$$

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$

Korea Institute of Science and Technology

Methodology: Forward Process

x : source image, y: target image



$$q(\boldsymbol{y}_{1:T} \mid \boldsymbol{y}_0) = \prod_{t=1}^{T} q(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}),$$
$$q(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}) = \mathcal{N}(\boldsymbol{y}_t \mid \sqrt{\alpha_t}\,\boldsymbol{y}_{t-1}, (1-\alpha_t)\boldsymbol{I})$$

Original DDPM Notation

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

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

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$

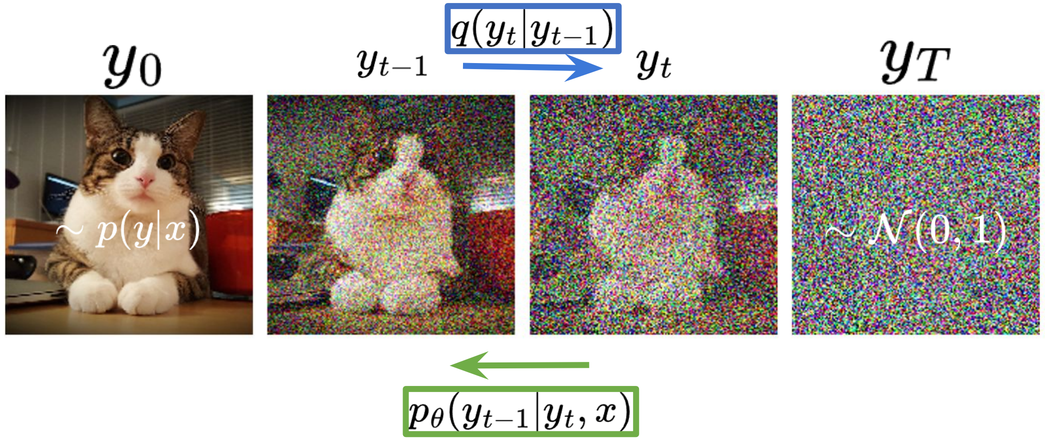
$$q(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_0, \boldsymbol{y}_t) = \mathcal{N}(\boldsymbol{y}_{t-1} \mid \boldsymbol{\mu}, \sigma^2\boldsymbol{I})$$
$$\boldsymbol{\mu} = \frac{\sqrt{\gamma_{t-1}}\,(1-\alpha_t)}{1-\gamma_t}\boldsymbol{y}_0 + \frac{\sqrt{\alpha_t}\,(1-\gamma_{t-1})}{1-\gamma_t}\boldsymbol{y}_t$$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

$$\tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \tag{10}$$
$$\tilde{\mu}_t(x_t, x_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t \tag{11}$$
$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t\mathbf{I}) \tag{12}$$

Korea Institute of Science and Technology

Methodology: Forward Process

x : source image, y: target image



$$q(\boldsymbol{y}_{1:T} \mid \boldsymbol{y}_0) = \prod_{t=1}^{T} q(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}),$$
$$q(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}) = \mathcal{N}(\boldsymbol{y}_t \mid \sqrt{\alpha_t}\, \boldsymbol{y}_{t-1}, (1-\alpha_t)\boldsymbol{I})$$

Original DDPM Notation

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I})$$

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

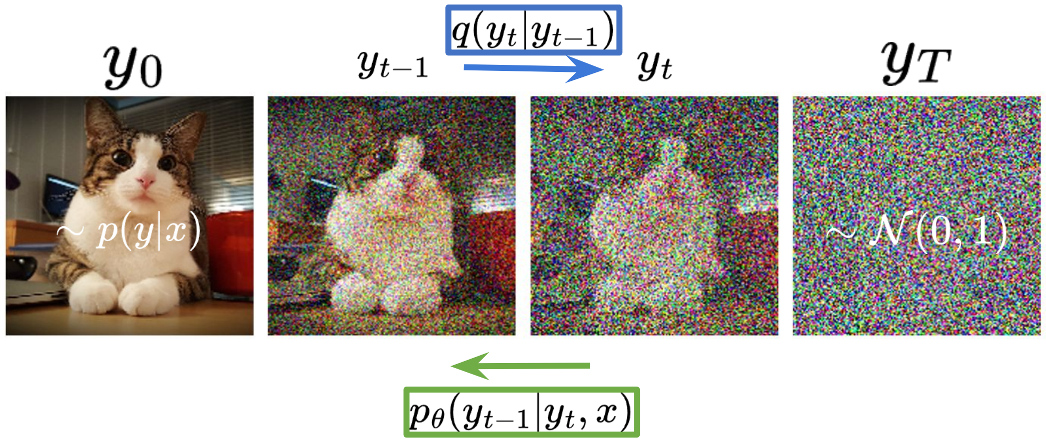
$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$

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

$$\tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \tag{10}$$
$$\tilde{\mu}_t(x_t, x_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t \tag{11}$$
$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t \mathbf{I}) \tag{12}$$

Methodology: Reverse Process

x : source image, y: target image



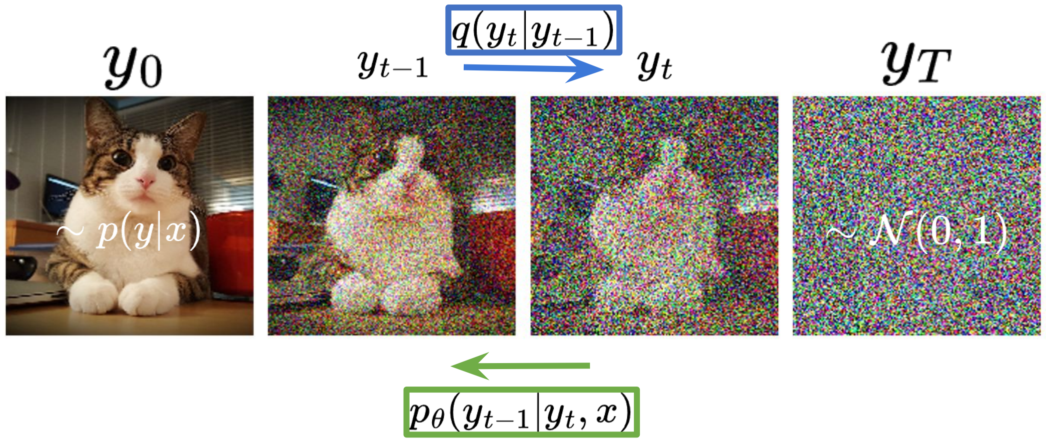
$$
\begin{aligned}
p_\theta(\boldsymbol{y}_{0:T}|\boldsymbol{x}) &= p(\boldsymbol{y}_T)\prod\nolimits_{t=1}^{T} p_\theta(\boldsymbol{y}_{t-1}|\boldsymbol{y}_t,\boldsymbol{x}) && (7)\\
p(\boldsymbol{y}_T) &= \mathcal{N}(\boldsymbol{y}_T \mid \boldsymbol{0}, \boldsymbol{I}) && (8)\\
p_\theta(\boldsymbol{y}_{t-1}|\boldsymbol{y}_t,\boldsymbol{x})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

$$
p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t,t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t,t))
$$

Korea Institute of Science and Technology

Methodology: Reverse Process

x : source image, y: target image



$$p_\theta(\boldsymbol{y}_{0:T}|\boldsymbol{x}) = p(\boldsymbol{y}_T)\prod_{t=1}^{T} p_\theta(\boldsymbol{y}_{t-1}|\boldsymbol{y}_t, \boldsymbol{x}) \quad (7)$$

$$p(\boldsymbol{y}_T) = \mathcal{N}(\boldsymbol{y}_T \mid \boldsymbol{0}, \boldsymbol{I}) \quad (8)$$

$$p_\theta(\boldsymbol{y}_{t-1}|\boldsymbol{y}_t, \boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}_{t-1} \mid \mu_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t), \sigma_t^2 \boldsymbol{I}) \,. \quad (9)$$

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

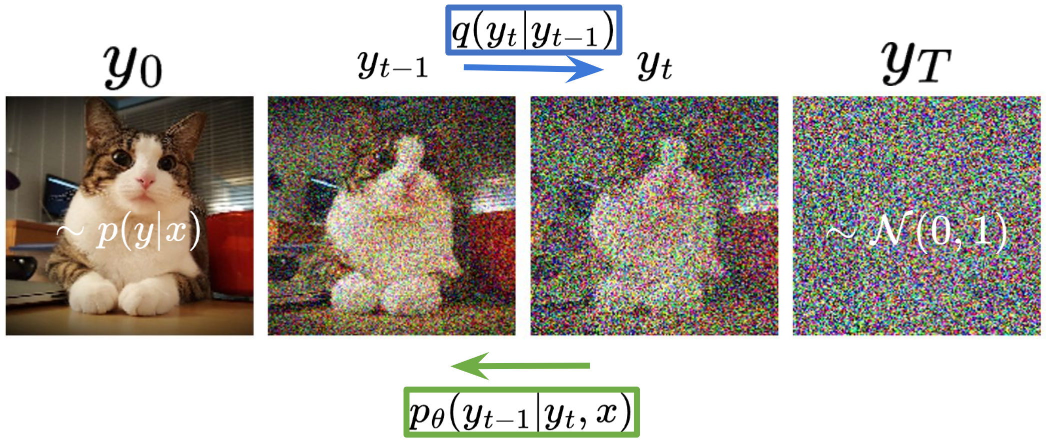
$$\mu_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{y}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t)\right)$$

$$\mu_\theta = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t)\right)$$

Korea Institute of Science and Technology

Methodology: Reverse Process

x : source image, y: target image



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

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

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

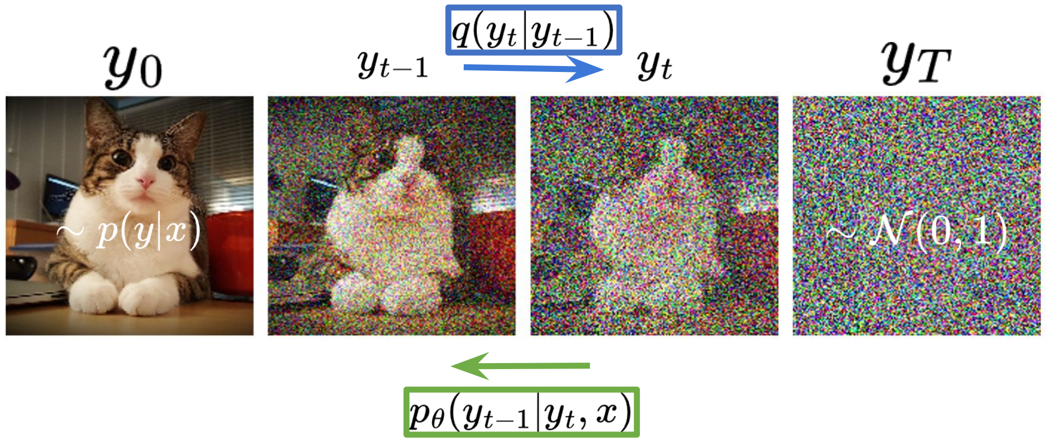
$$\mu_\theta = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t)\right)$$

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

predicted noise!

Methodology: Reverse Process

$\boldsymbol{x}$: source image, $\boldsymbol{y}$: target image



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

$$
p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t,t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t,t))
$$

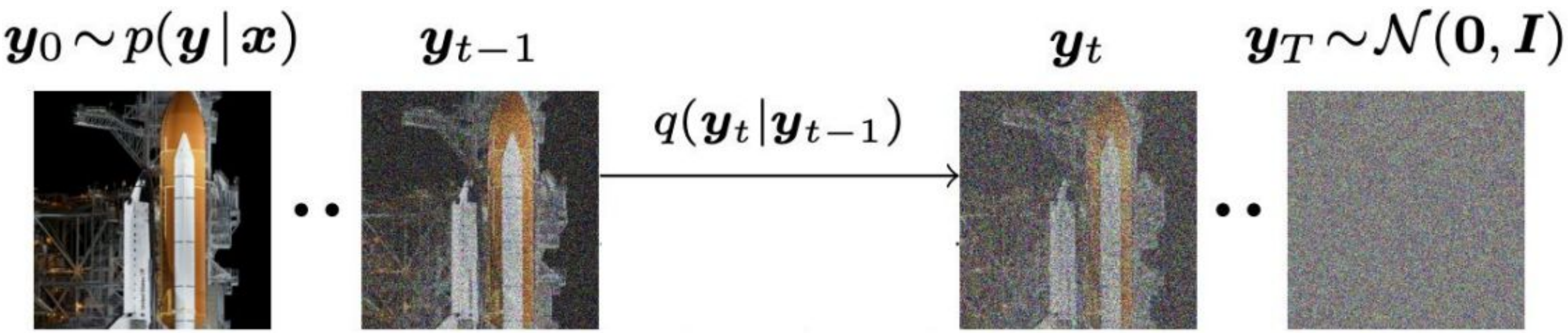
$$
\mu_\theta(\boldsymbol{x},\boldsymbol{y}_t,\gamma_t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{y}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta(\boldsymbol{x},\boldsymbol{y}_t,\gamma_t)\right)
$$

$$
\mu_\theta = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t)\right)
$$

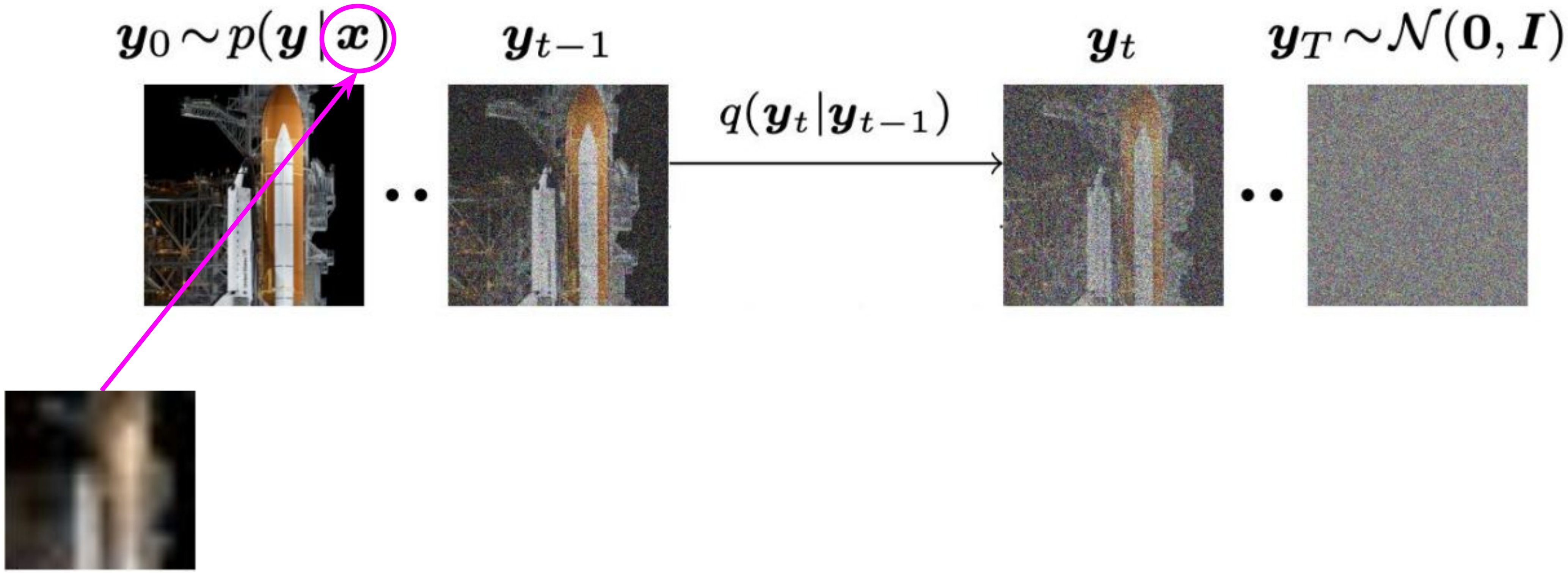
$$
\boldsymbol{y}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{y}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta(\boldsymbol{x},\boldsymbol{y}_t,\gamma_t)\right) + \sqrt{1-\alpha_t}\,\boldsymbol{\epsilon}_t
$$

Korea Institute of Science and Technology

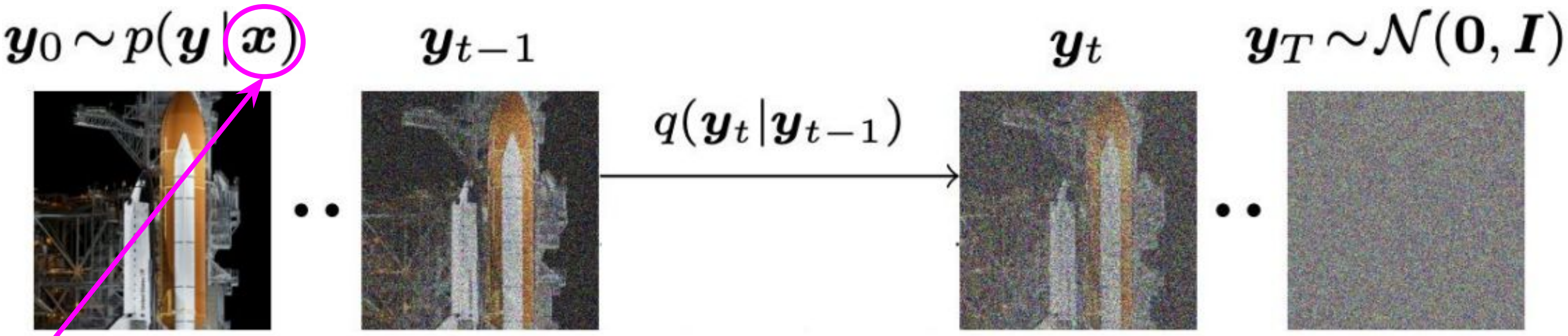
Algorithm: Forward Process

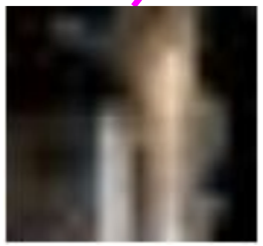


Algorithm: Forward Process



Algorithm: Forward Process





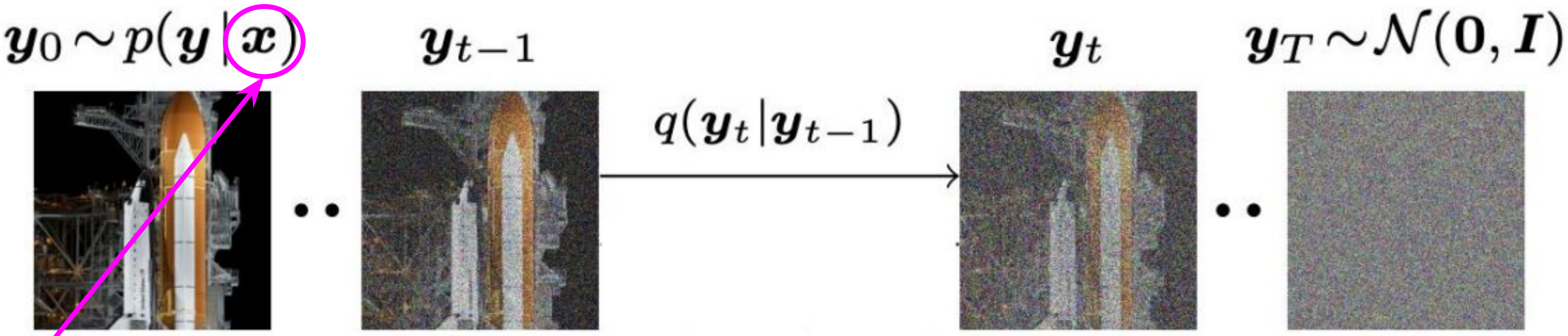
$$q(\boldsymbol{y}_{1:T} \mid \boldsymbol{y}_0) = \prod_{t=1}^{T} q(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1})$$

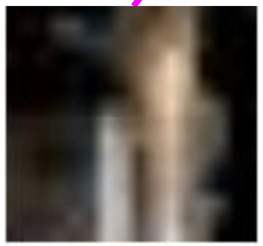
$$q(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}) = \mathcal{N}(\boldsymbol{y}_t \mid \sqrt{\alpha_t}\, \boldsymbol{y}_{t-1}, (1-\alpha_t)\boldsymbol{I})$$

noise variance

$0 < \alpha_t < 1$

Algorithm: Forward Process





$$q(y_{1:T} \mid y_0) = \prod_{t=1}^{T} q(y_t \mid y_{t-1}) \qquad q(y_t \mid y_{t-1}) = \mathcal{N}(y_t \mid \sqrt{\alpha_t}\, y_{t-1}, (1-\alpha_t)I)$$

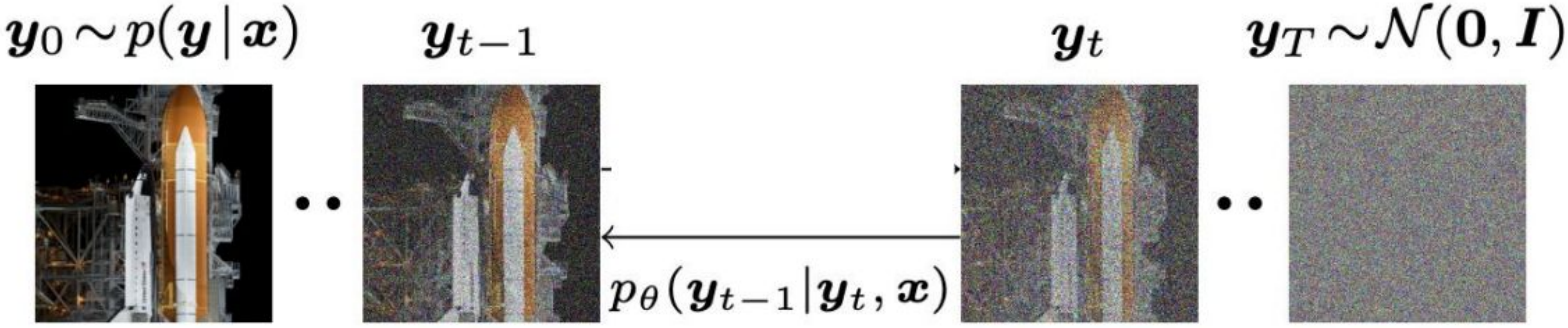
noise variance

$$0 < \alpha_t < 1$$

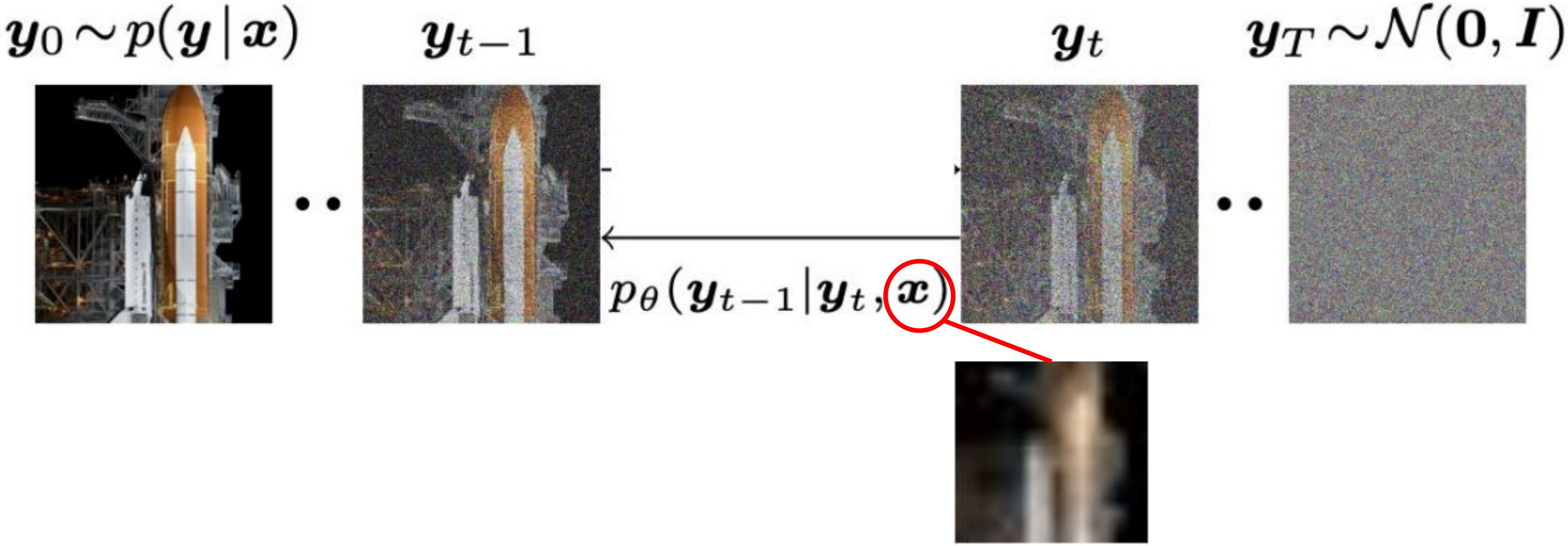
$$q(y_t \mid y_0) = \mathcal{N}(y_t \mid \sqrt{\gamma_t}\, y_0, (1-\gamma_t)I) \qquad \gamma_t = \prod_{i=1}^{t} \alpha_i$$

intermediate step

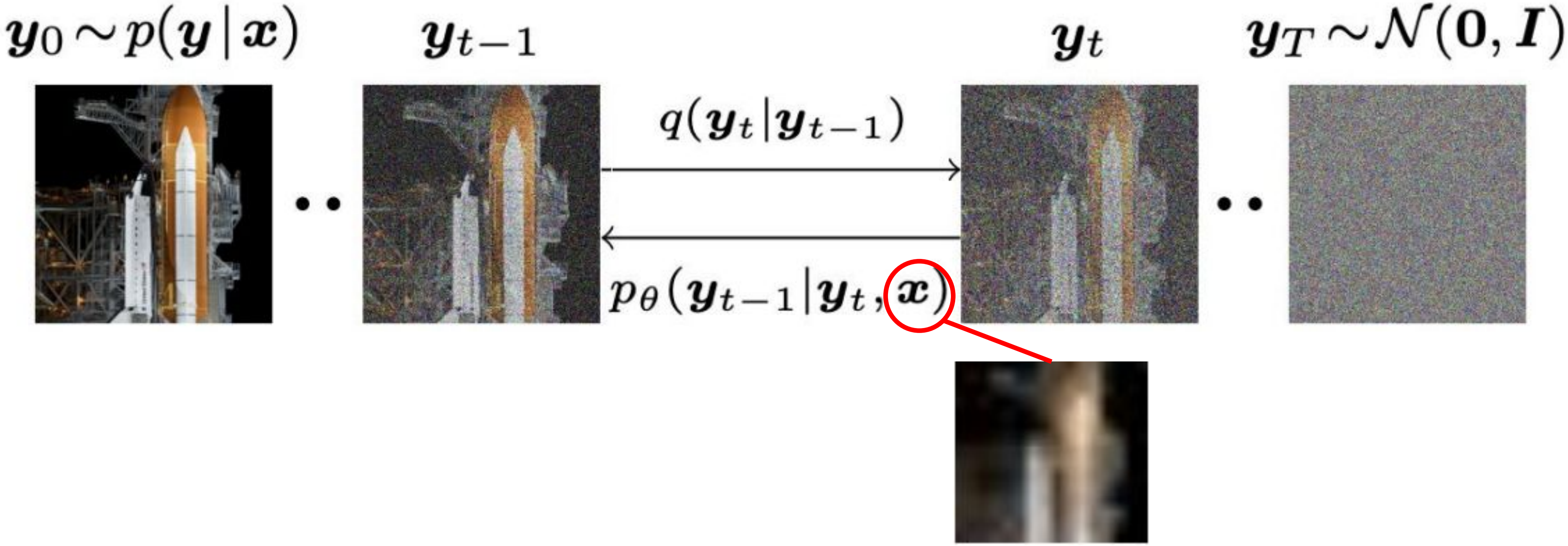
Algorithm: Reverse Process



Algorithm: Reverse Process

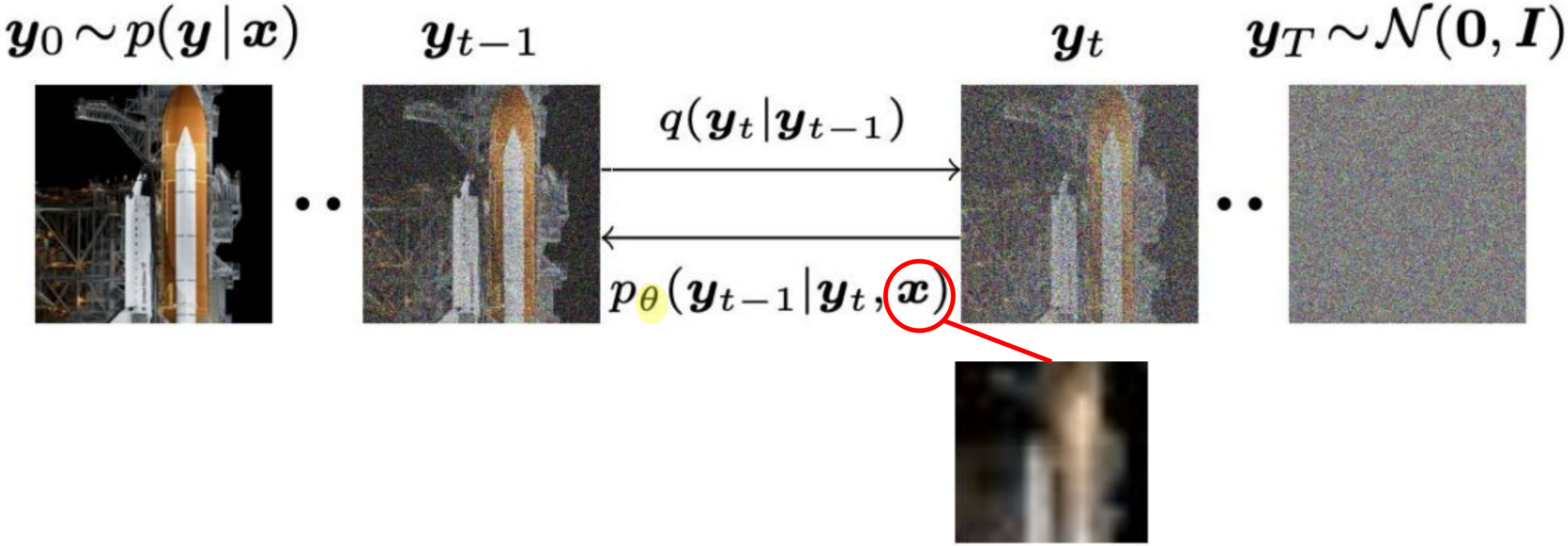


Algorithm: Reverse Process



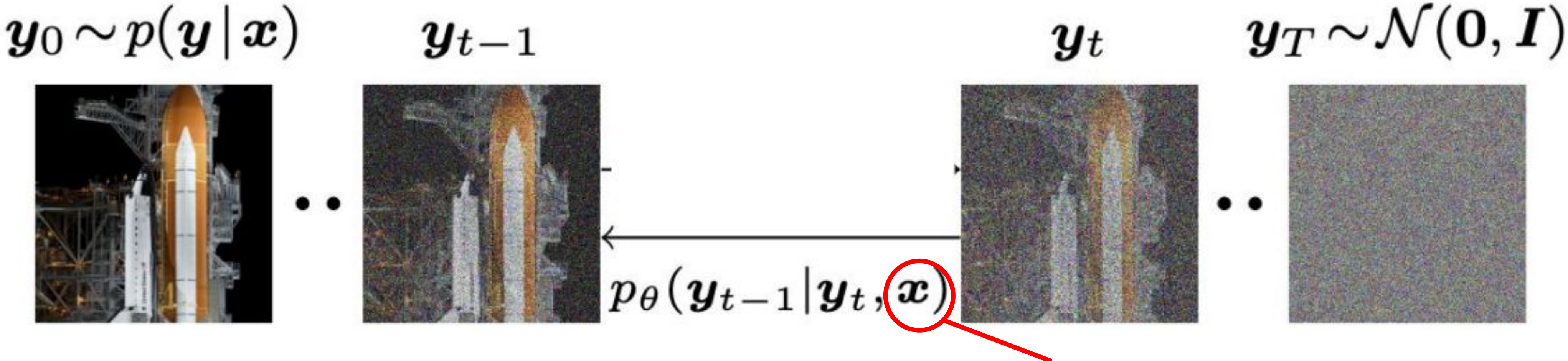
Korea Institute of Science and Technology

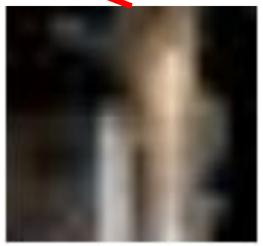
Algorithm: Reverse Process



Korea Institute of
Science and Technology

Algorithm: Reverse Process

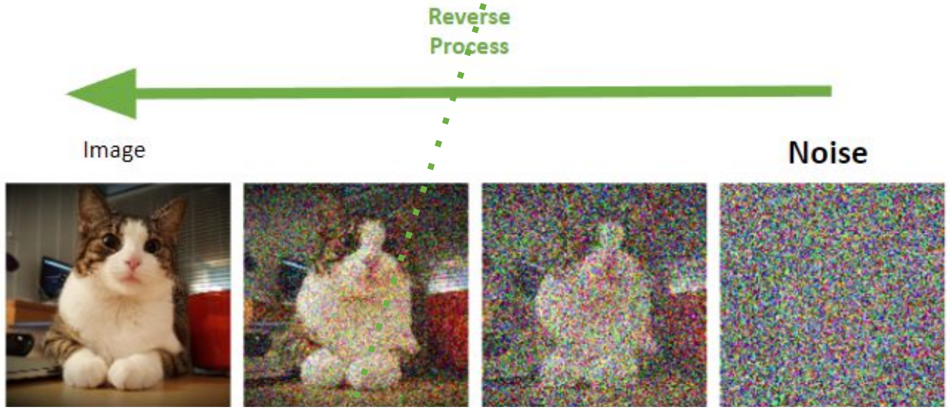




$$
\begin{aligned}
p_\theta(\boldsymbol{y}_{0:T}|\boldsymbol{x}) &= p(\boldsymbol{y}_T) \prod_{t=1}^{T} p_\theta(\boldsymbol{y}_{t-1}|\boldsymbol{y}_t, \boldsymbol{x}) && (7)\\
p(\boldsymbol{y}_T) &= \mathcal{N}(\boldsymbol{y}_T \mid \mathbf{0}, \boldsymbol{I}) && (8)\\
p_\theta(\boldsymbol{y}_{t-1}|\boldsymbol{y}_t,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

Training

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$



$$-log(p_\theta(x)) \leq log\frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}$$

$$-log(p_\theta(x_0)) \leq \boxed{log\frac{\prod_{t=1}^{T} q(x_t|x_{t-1})}{p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)}}$$

$$= \boxed{log(\frac{q(x_T|x_0)}{p(x_T)})} + \sum_{t=2}^{T} \boxed{log(\frac{q(x_{t-1}|x_t, x_0)}{p_\theta(x_{t-1}|x_t)})} - logp_\theta(x_0|x_1)$$

$$\boxed{D_{KL}(q(x_T|x_0)||p(x_T))}$$

$$\boxed{D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))}$$

$$\boxed{\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon)}$$

$$\tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \qquad (10)$$

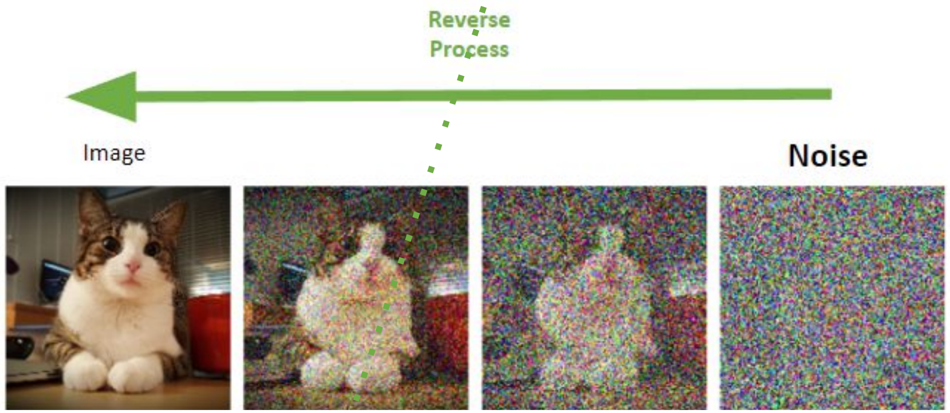
$$\tilde{\mu}_t(x_t, x_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t \qquad (11)$$

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t\mathbf{I}) \qquad (12)$$

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$

Training

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$



$$-log(p_\theta(x)) \le log\frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}$$

$$-log(p_\theta(x_0)) \le \boxed{log\frac{\prod_{t=1}^{T} q(x_t|x_{t-1})}{p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)}}$$

$$= \boxed{log(\frac{q(x_T|x_0)}{p(x_T)})} + \sum_{t=2}^{T} \boxed{log(\frac{q(x_{t-1}|x_t, x_0)}{p_\theta(x_{t-1}|x_t)})} - logp_\theta(x_0|x_1)$$

$$\boxed{D_{KL}(q(x_T|x_0)||p(x_T))}$$

$$\boxed{D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))}$$

$$\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon)$$

$$\tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \tag{10}$$

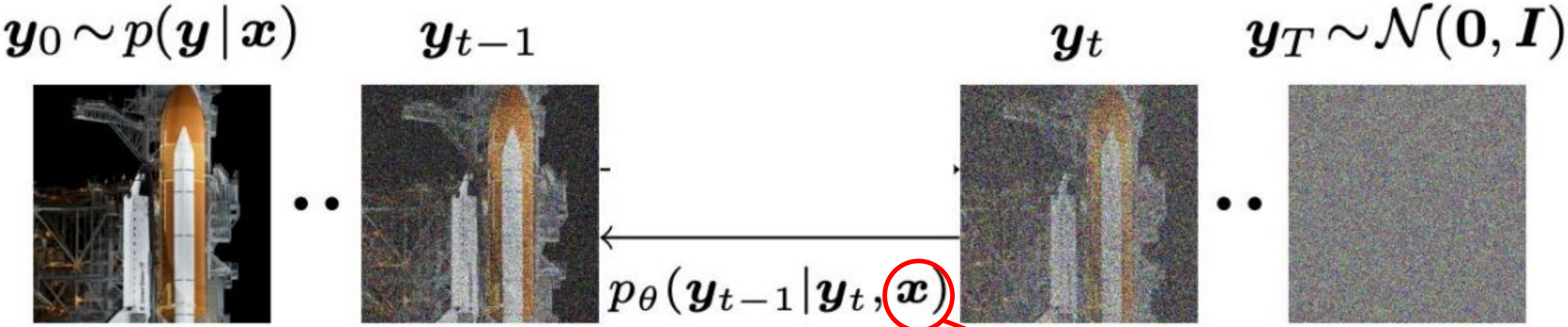
$$\tilde{\mu}_t(x_t, x_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t \tag{11}$$

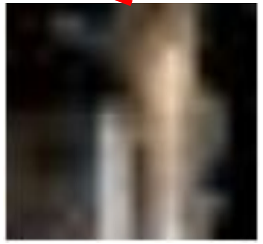
$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t\mathbf{I}) \tag{12}$$

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$

Korea Institute of Science and Technology

Algorithm: Reverse Process





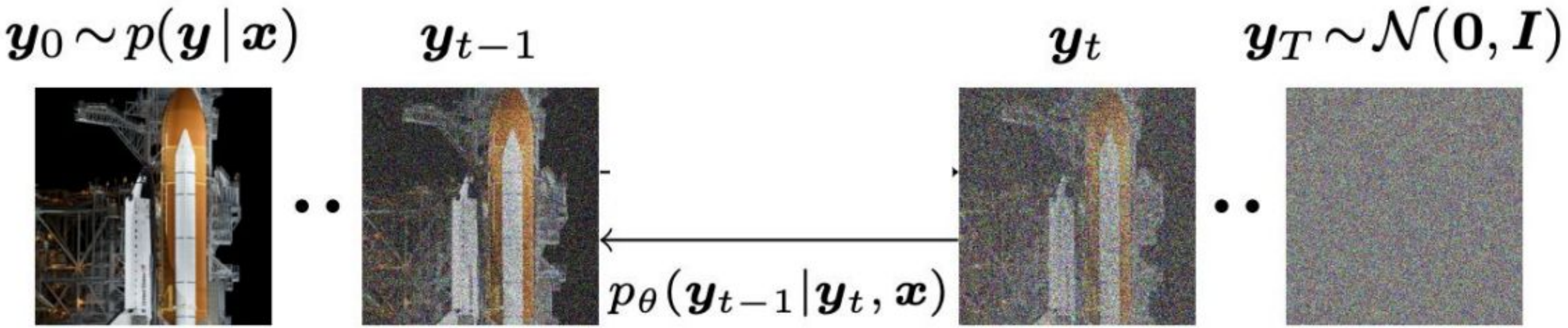
$$
\begin{aligned}
p_\theta(\boldsymbol{y}_{0:T} \mid \boldsymbol{x}) &= p(\boldsymbol{y}_T) \prod_{t=1}^{T} p_\theta(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_t, \boldsymbol{x}) && (7) \\
p(\boldsymbol{y}_T) &= \mathcal{N}(\boldsymbol{y}_T \mid \mathbf{0}, \boldsymbol{I}) && (8) \\
p_\theta(\boldsymbol{y}_{t-1}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

$$
\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right)
$$

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

Korea Institute of Science and Technology

Algorithm: Reverse Process



$$p_\theta(y_{0:T}|x) = p(y_T) \prod_{t=1}^{T} p_\theta(y_{t-1}|y_t, x) \quad (7)$$

$$p(y_T) = \mathcal{N}(y_T \mid 0, I) \quad (8)$$

$$p_\theta(y_{t-1}|y_t, x) = \mathcal{N}(y_{t-1} \mid \boxed{\mu_\theta(x, y_t, \gamma_t)}, \sigma_t^2 I)\,. \quad (9)$$

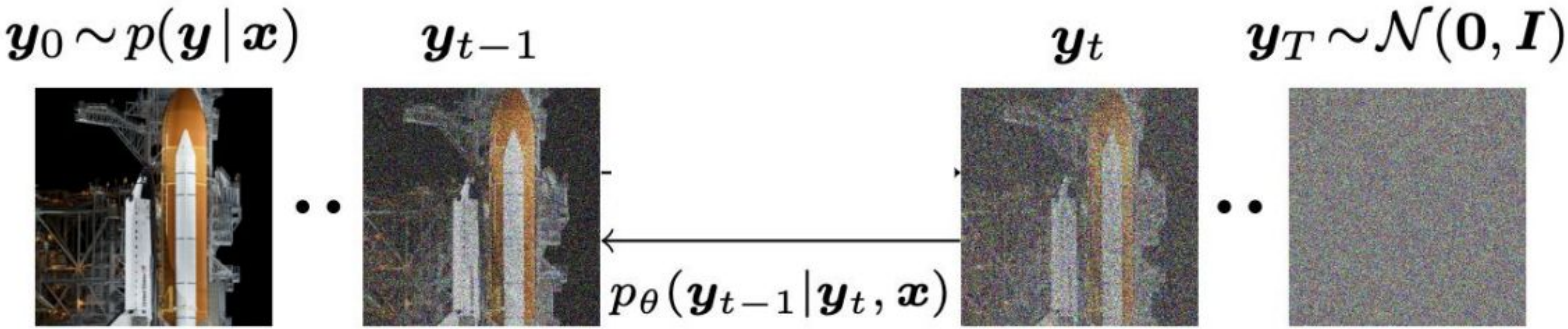
$$\boxed{\mu_\theta(x, y_t, \gamma_t) = \frac{1}{\sqrt{\alpha_t}}\left(y_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta(x, y_t, \gamma_t)\right)}$$

Algorithm 2 Inference in T iterative refinement steps

```
y_T ~ N(0, I)
for t = T, ..., 1 do
  z ~ N(0, I) if t > 1, else z = 0
  y_{t-1} = 1/sqrt(α_t) (y_t − (1−α_t)/sqrt(1−γ_t) f_θ(x, y_t, γ_t)) + sqrt(1 − α_t) z
end for
return y_0
```

Korea Institute of Science and Technology

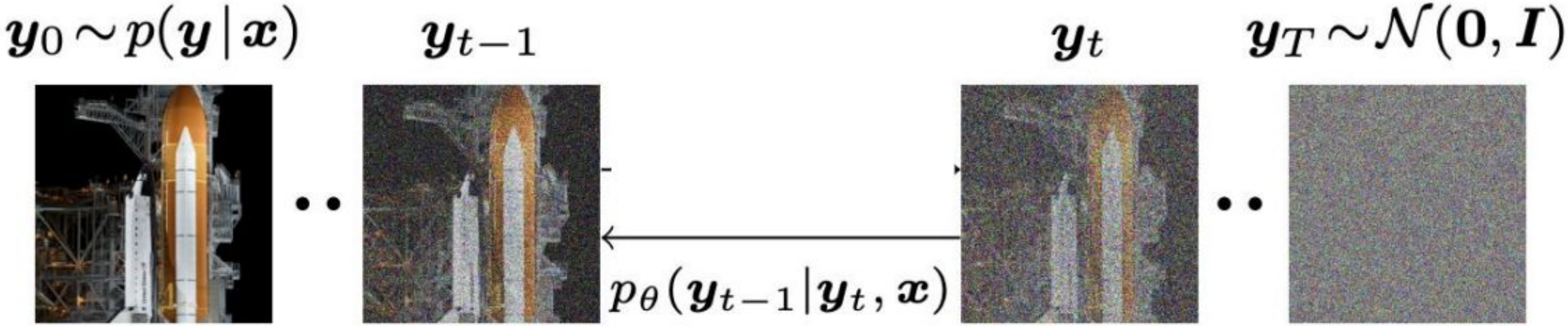
Algorithm: Reverse Process



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

objective

Algorithm: Reverse Process



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

objective

$$||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||^2$$

Pseudocode

Algorithm 1 Training

repeat
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $t \sim \mathrm{Uniform}(\{1, \dots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on
    $\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
until converged

Algorithm 2 Sampling

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

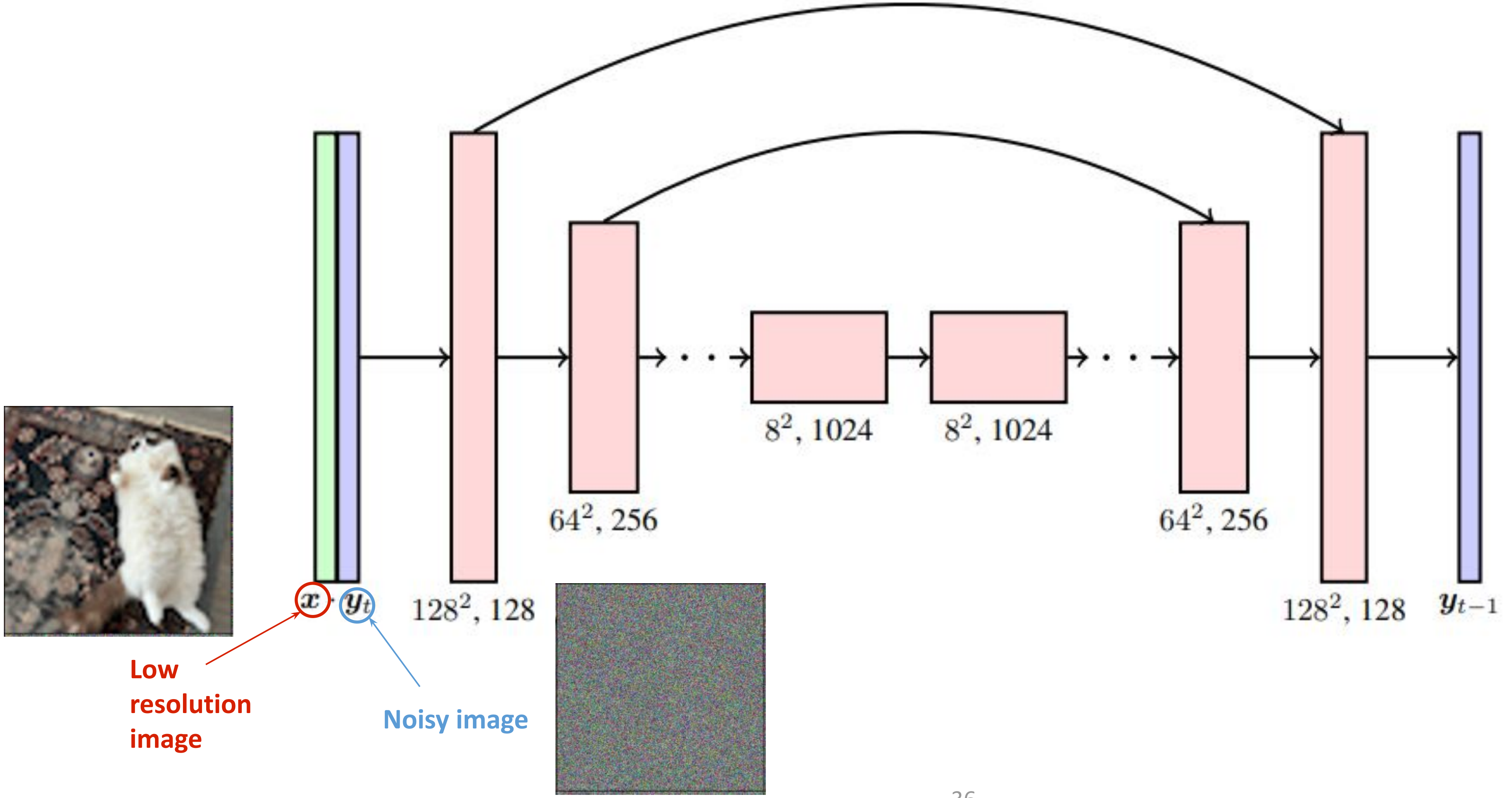
Algorithm 1 Training a denoising model f_θ

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

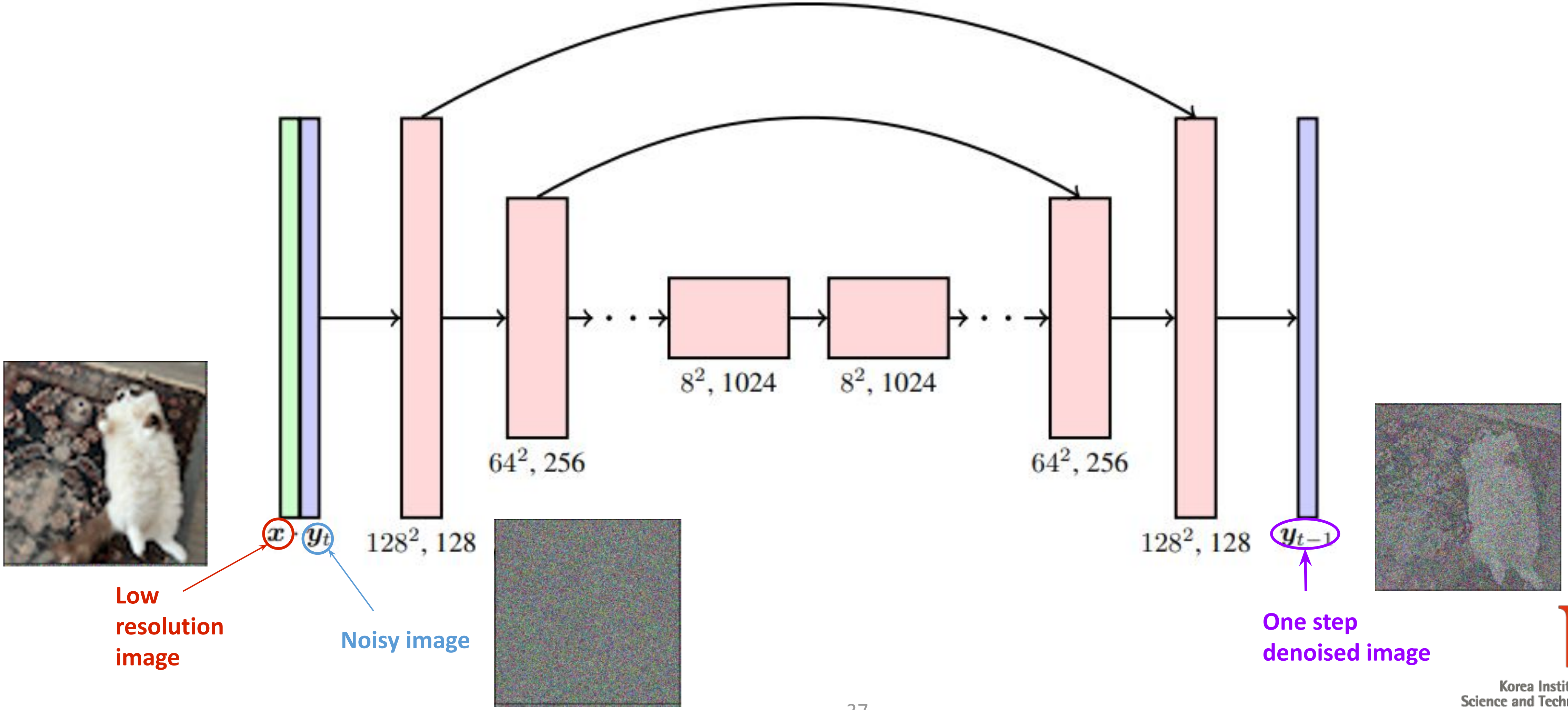
Algorithm 2 Inference in T iterative refinement steps

$\boldsymbol{y}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \dots, 1$ **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 $\boldsymbol{y}_0$

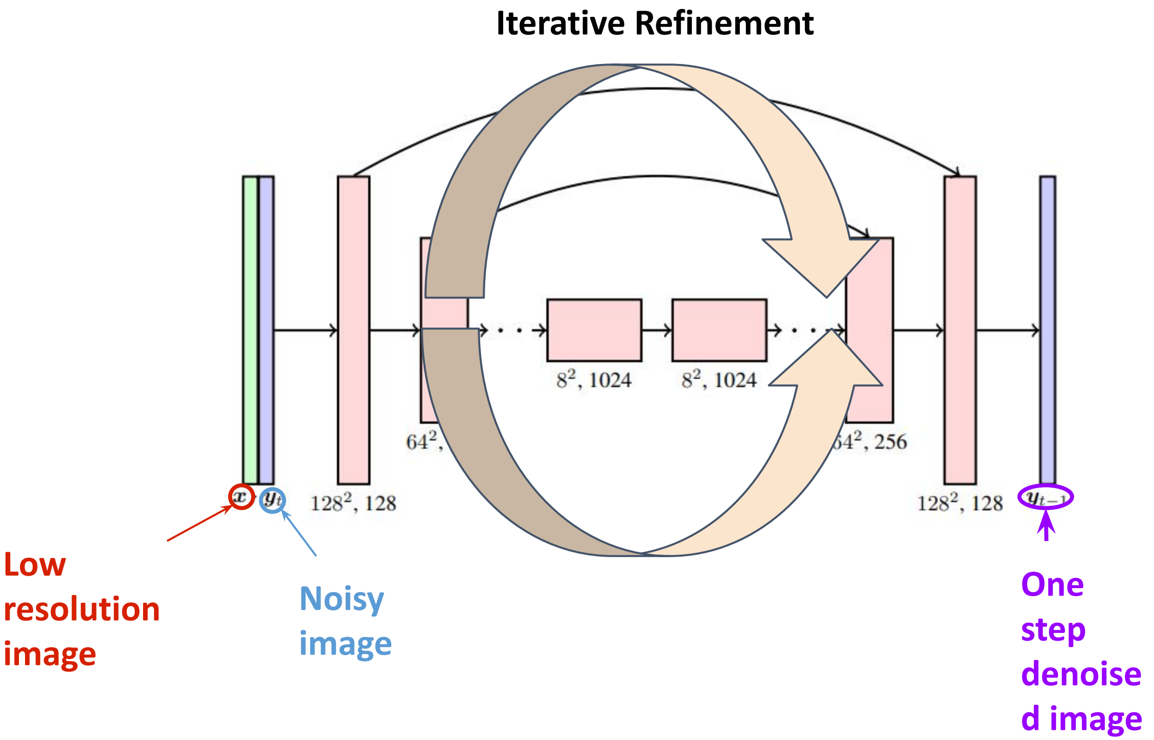
Architecture



Architecture

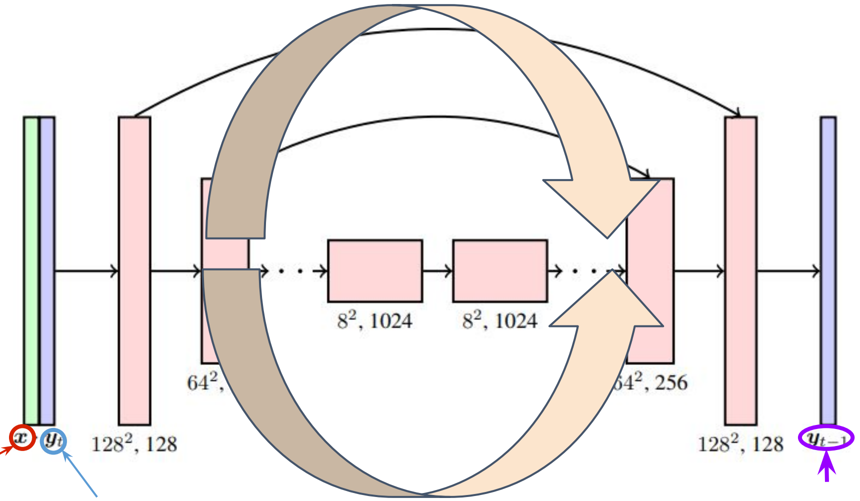


Architecture



Architecture

Iterative Refinement



Low resolution image

Noisy image

One step denoised image

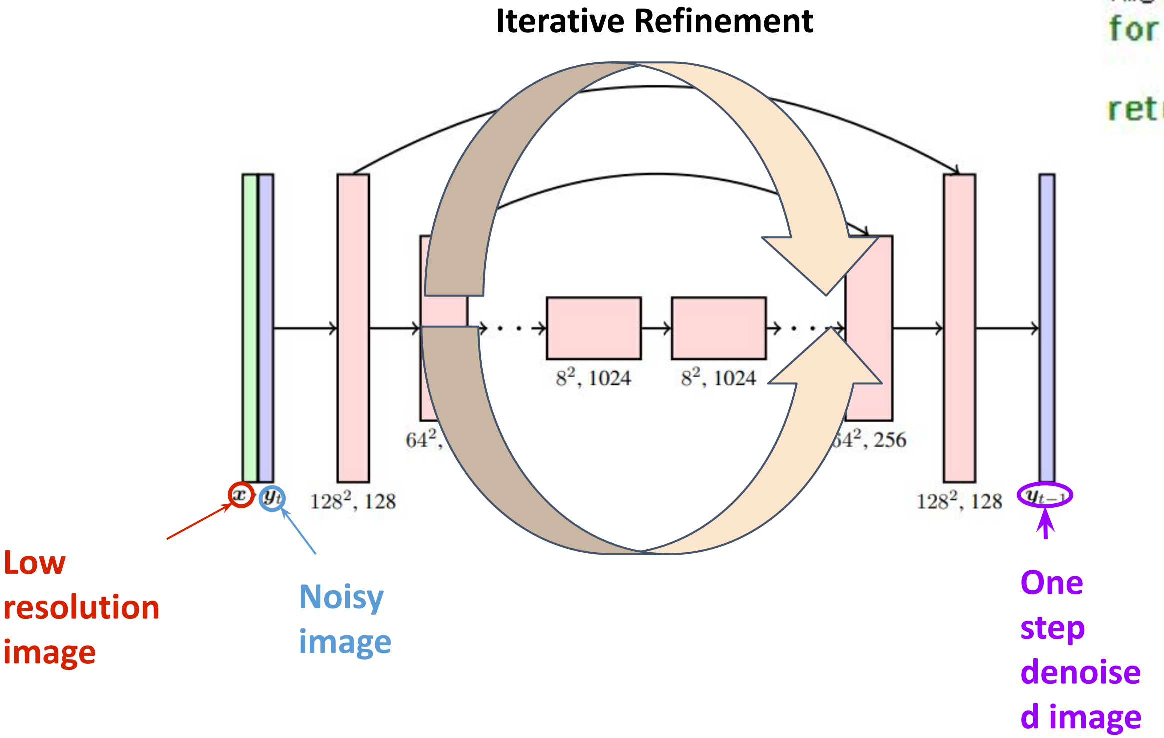
```
# The model predict actual noise added at time step t
pred_noise = self.model(torch.cat([lr_imgs, x_noisy], dim=1), noise_level=sqrt_alpha)
```

Concatenation of Low Resolution Image and Noisy Image

Architecture

```python
# Progress whole reverse diffusion process
@torch.no_grad()
def super_resolution(self, x_in):
    img = torch.rand_like(x_in, device=x_in.device)
    for i in reversed(range(0, self.num_timesteps)):
        img = self.p_sample(img, i, condition_x=x_in)
    return img
```

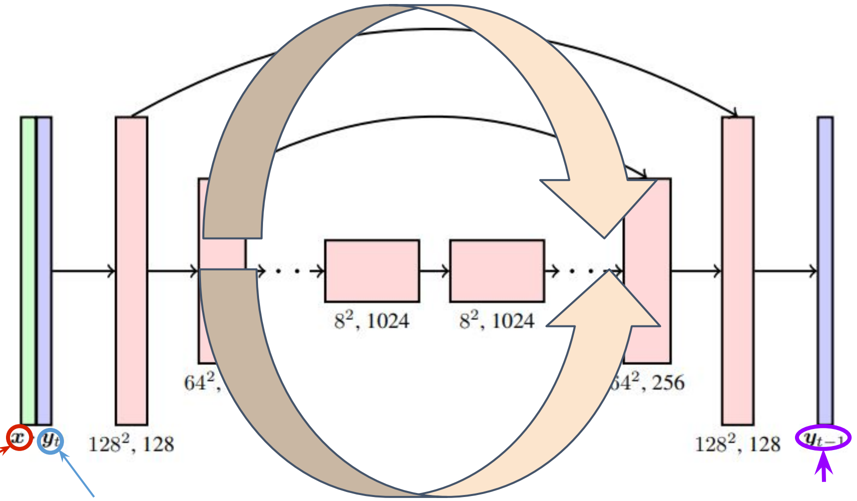
T ⟶ 0



Architecture

16x16

Iterative Refinement



Low resolution image

Noisy image

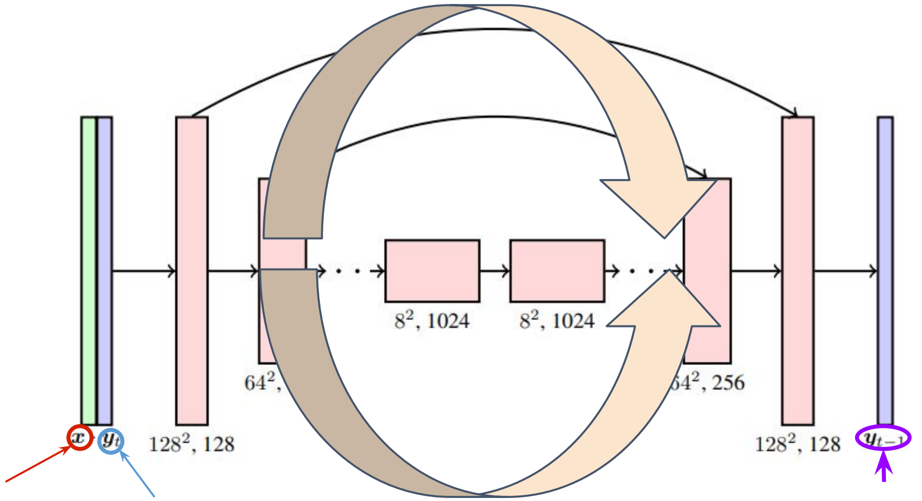
One step denoised image

Korea Institute of Science and Technology

Architecture

16x16

Iterative Refinement



Low resolution image

Noisy image

One step denoised image

64x64

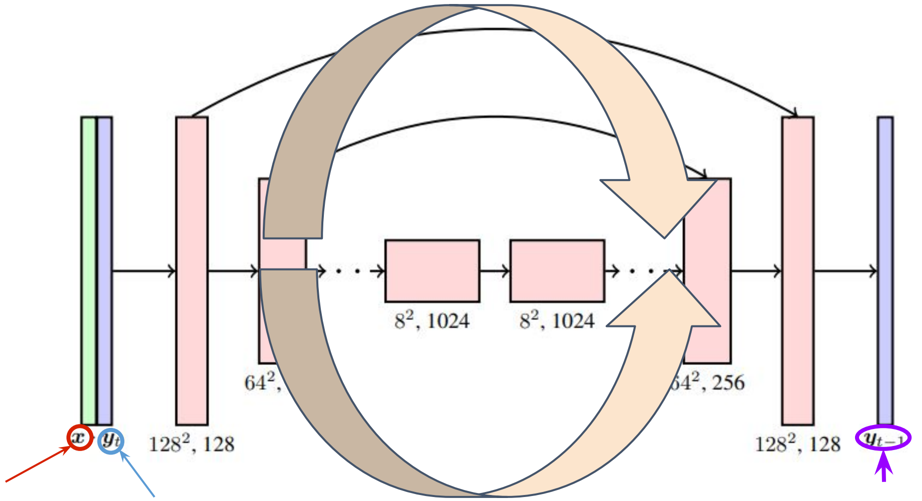
Architecture

16x16

64x64

resolution image

Iterative Refinement



Noisy image

One step denoise d image

64x64

Architecture

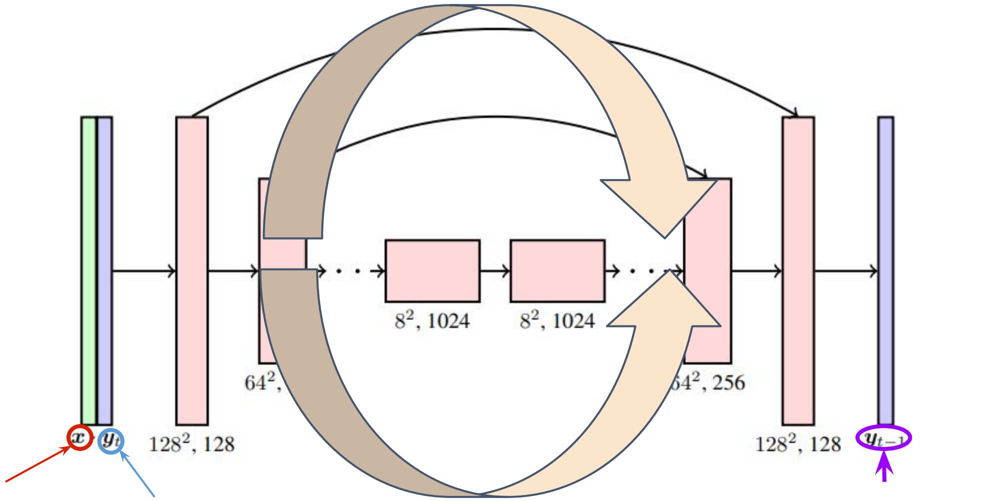
Bicubic Interpolation

16x16

64x64

resolution image

Iterative Refinement



Noisy image

One step denoised image

64x64

Experiments: Synthetic Face Generation

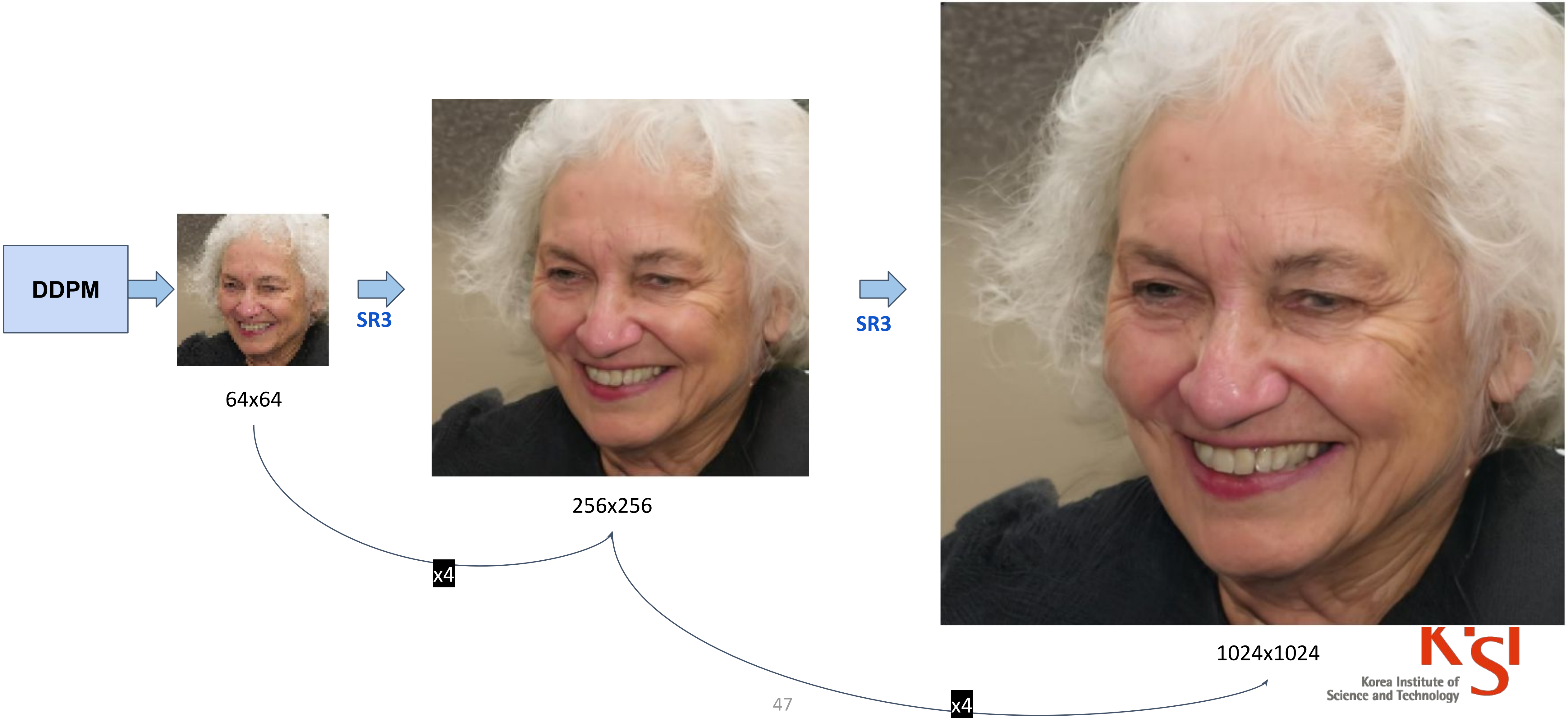
Korea Institute of Science and Technology

Experiments: Synthetic Face Generation

DDPM →

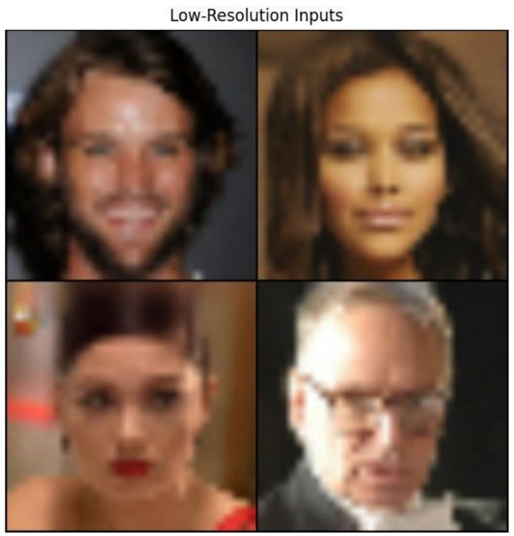
64x64

Experiments: Synthetic Face Generation



Results at epoch 120

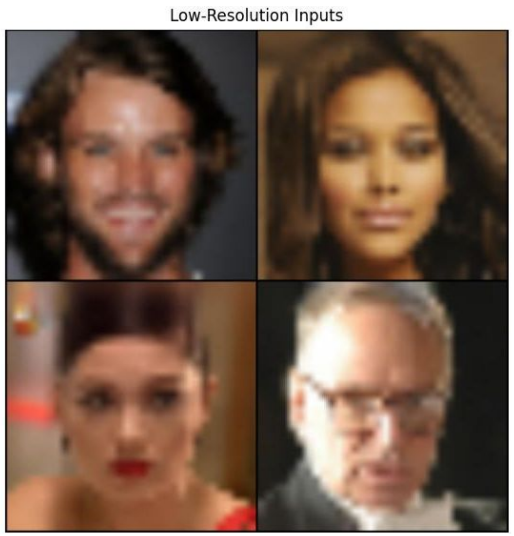
FFHQ dataset



32x32

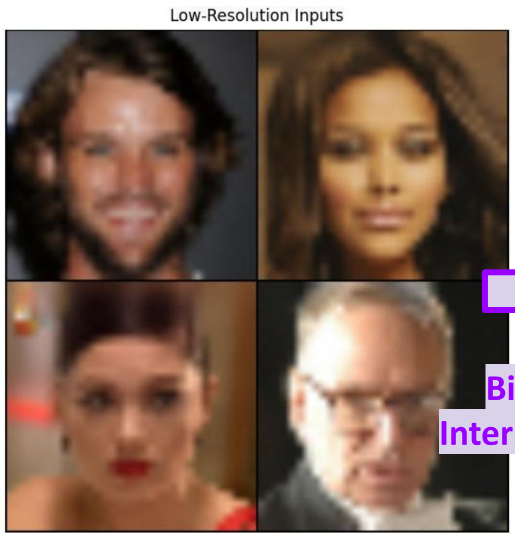
Artificially
Generate
Low-resolution
image by resizing

Results at epoch 120



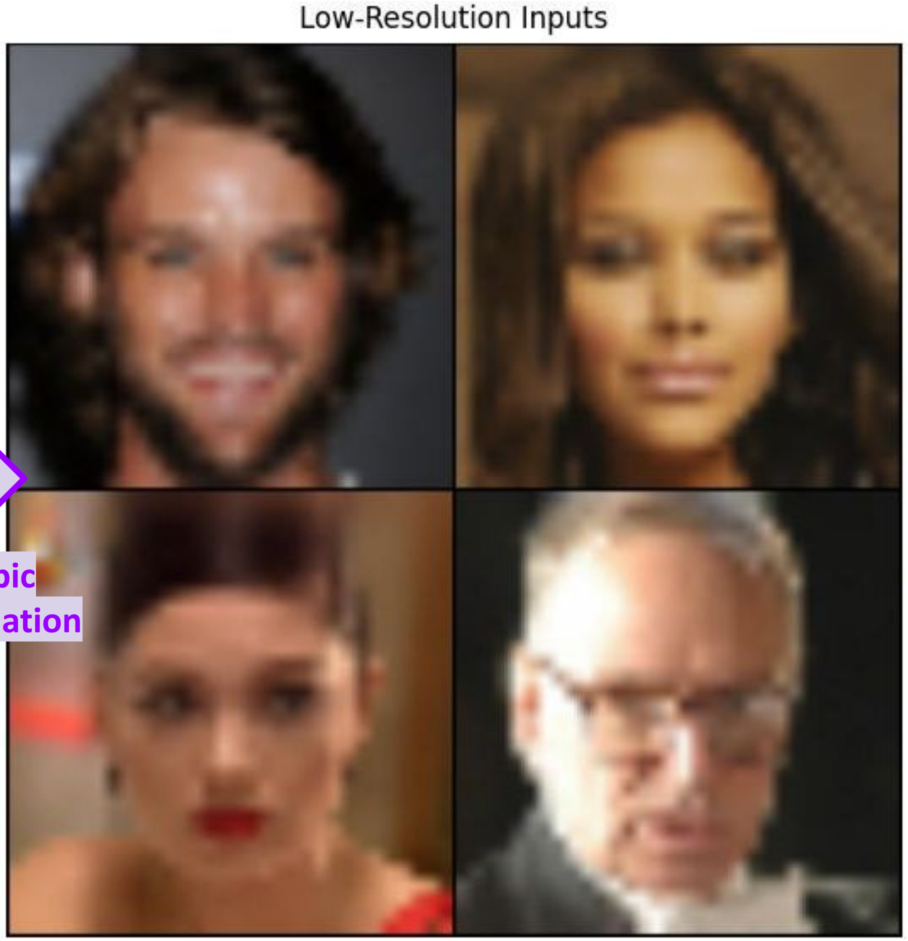
32x32

Results at epoch 120



32x32

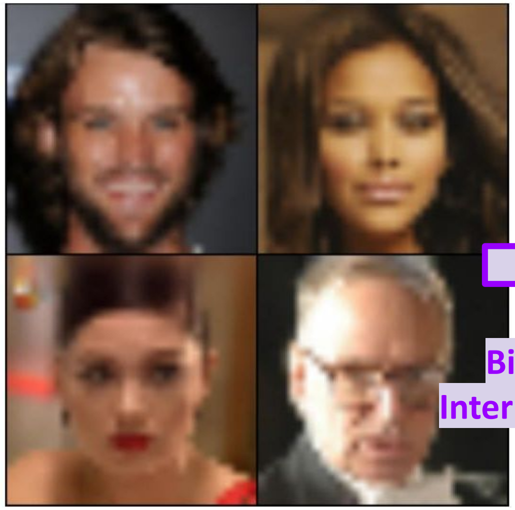
Bicubic Interpolation



Bicubic Interpolation 128x128

Results at epoch 120

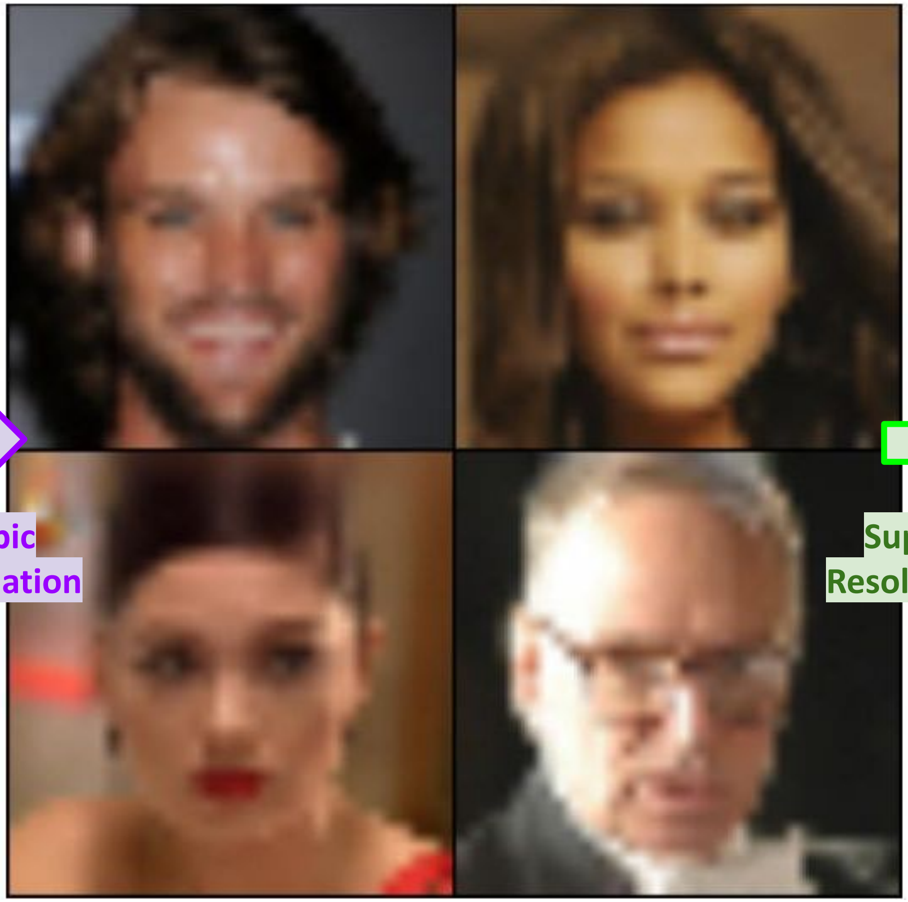
Low-Resolution Inputs



32x32

Bicubic
Interpolation

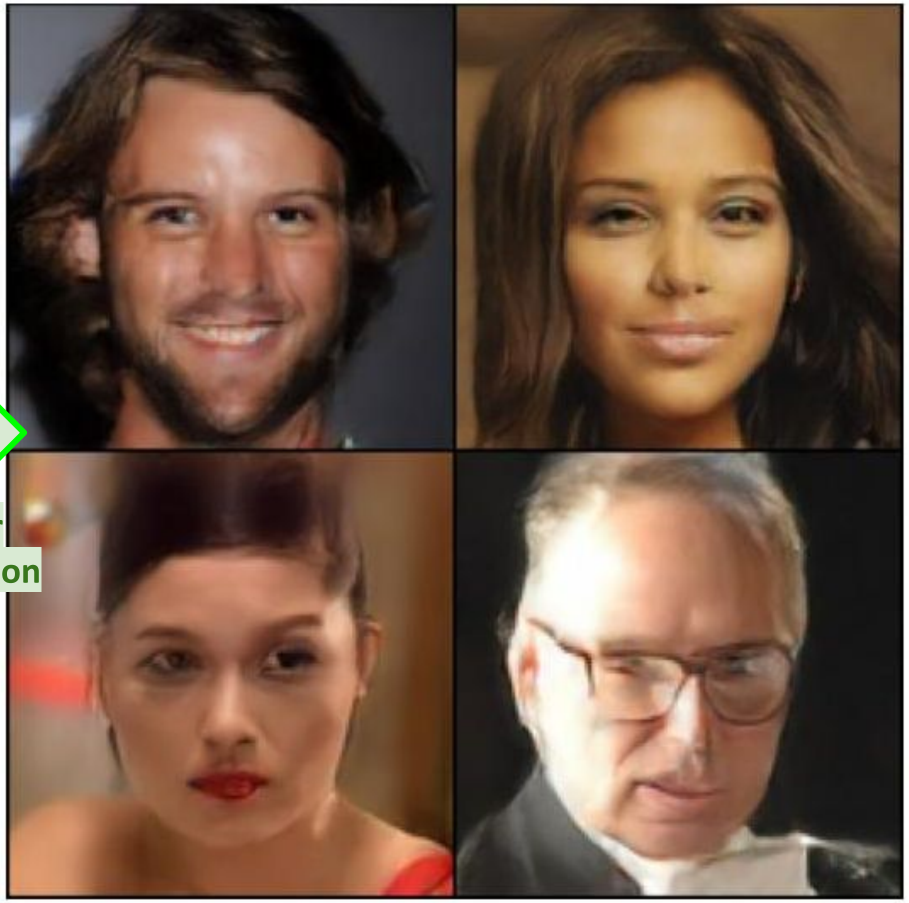
Low-Resolution Inputs



Bicubic Interpolation 128x128

Super
Resolution

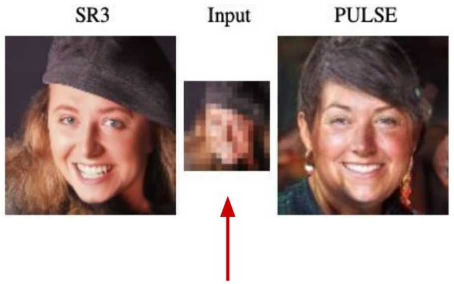
Super-Resolution Results 128x128



Korea Institute of Science and Technology

Experiments: Human Fool Rates

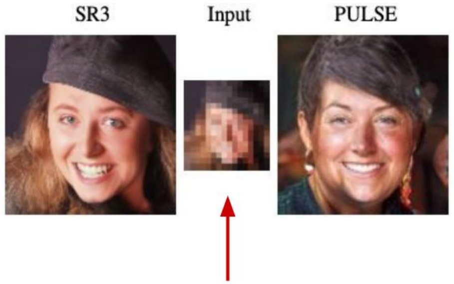
Task 1: “Which of the two images is a **better high quality version** of the low resolution image in the middle?”

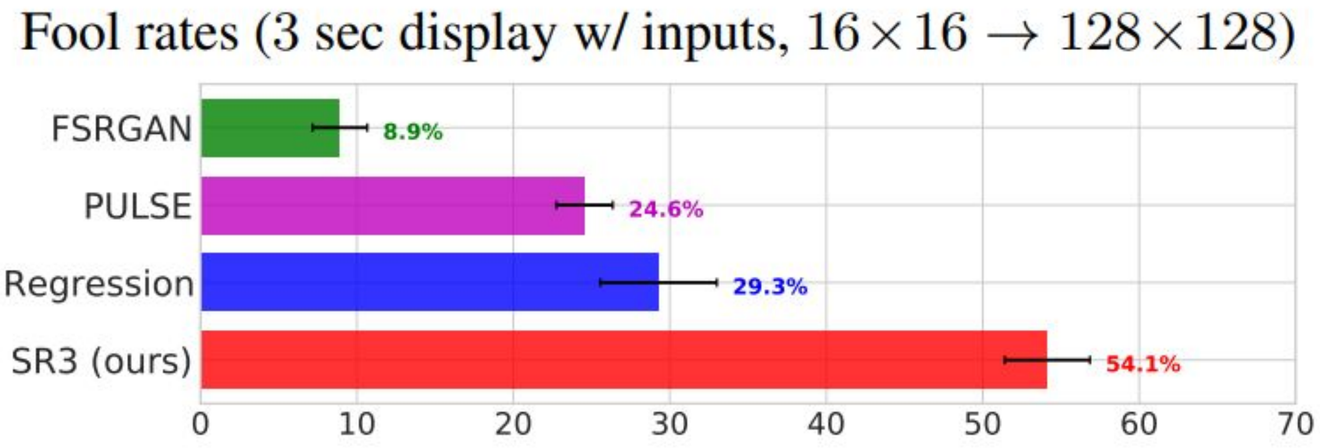


Task 2: “Which image would you guess is from a **camera**?”

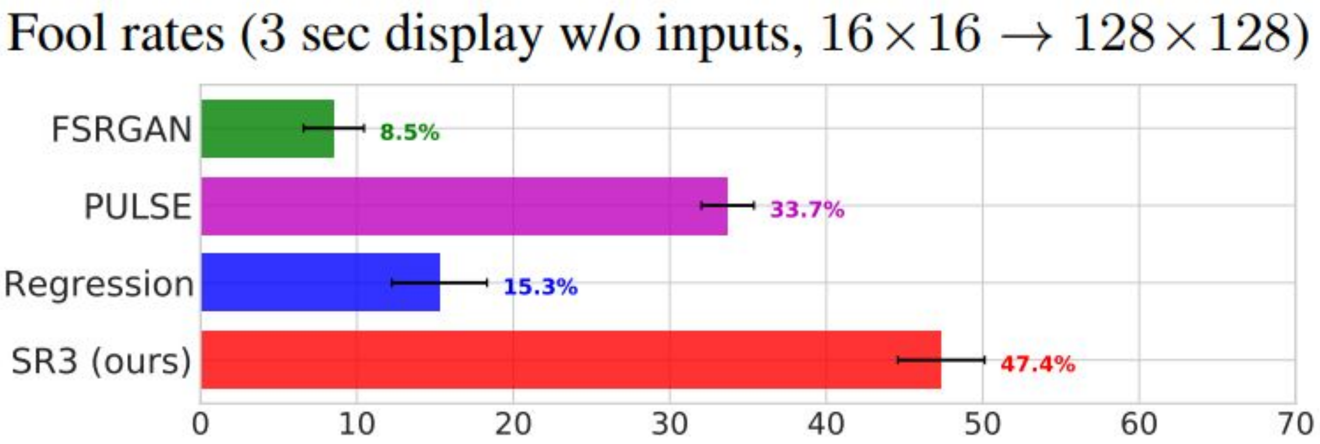
Experiments: Human Fool Rates

Task 1: "Which of the two images is a **better high quality version** of the low resolution image in the middle?"



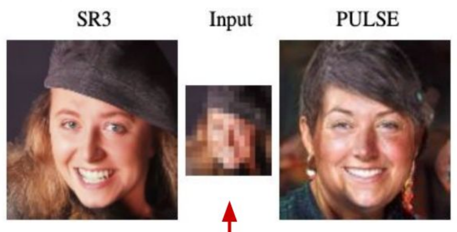


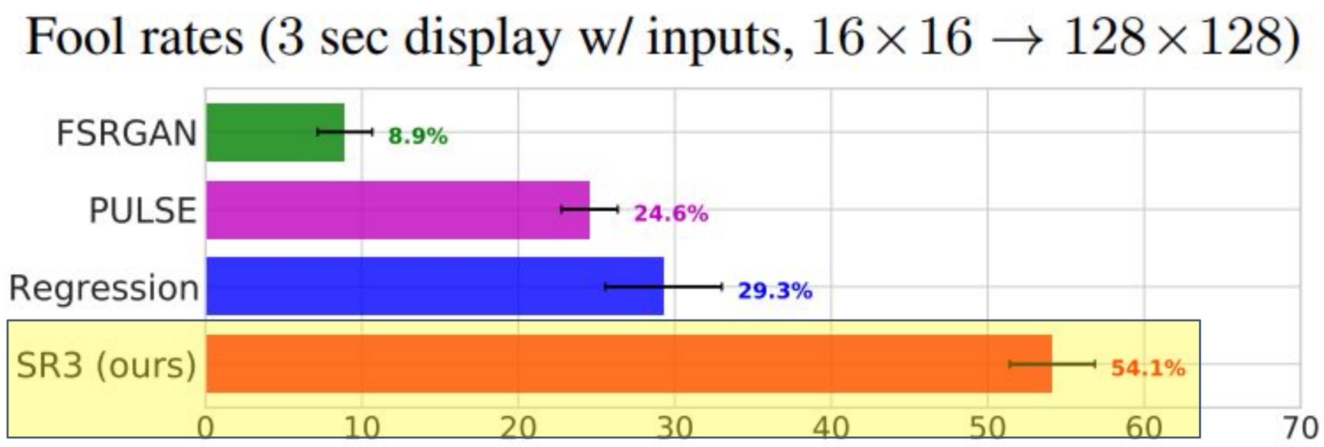
Task 2: "Which image would you guess is from a **camera**?"



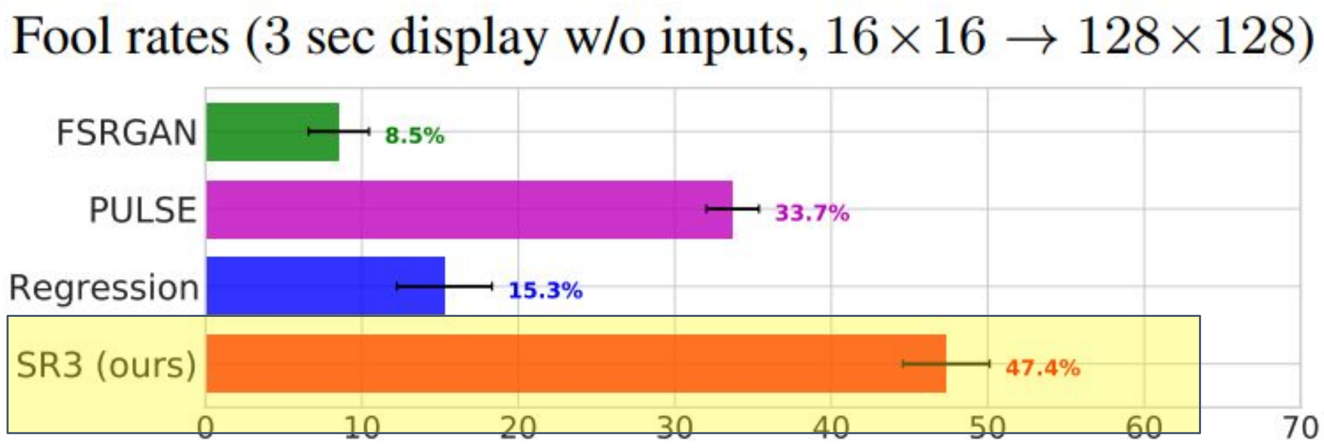
Experiments: Human Fool Rates

Task 1: "Which of the two images is a **better high quality version** of the low resolution image in the middle?"



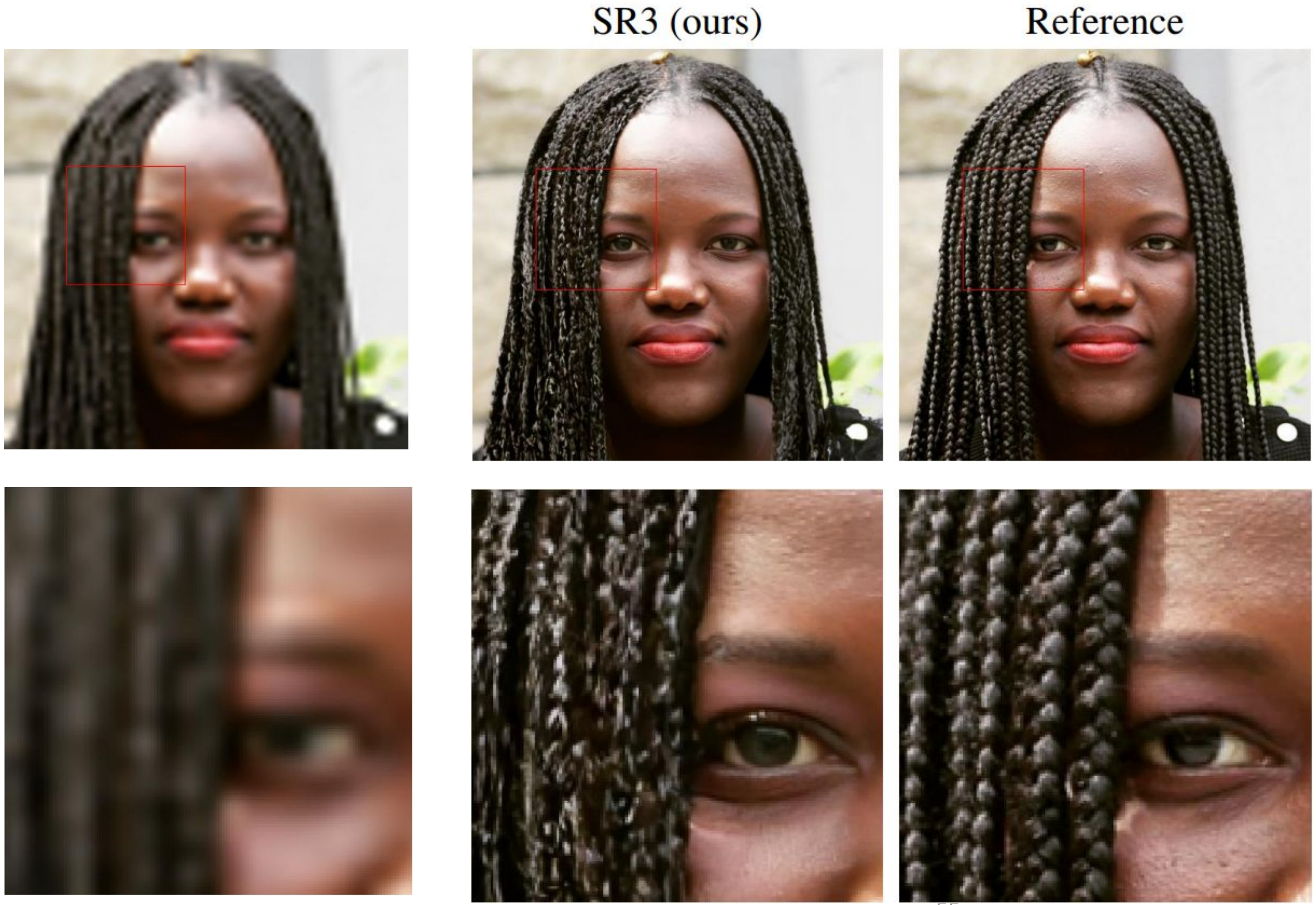


Task 2: "Which image would you guess is from a **camera**?"

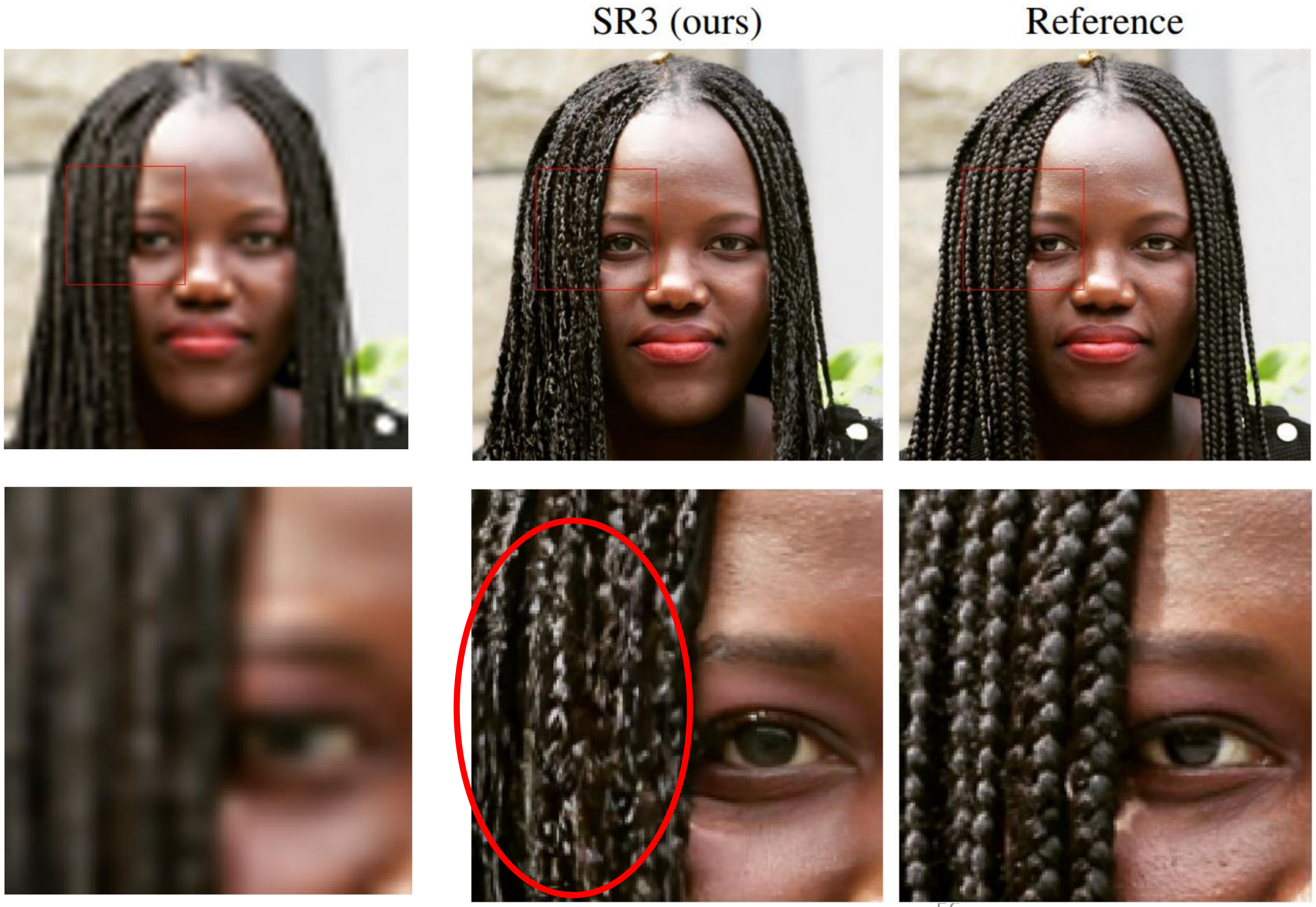


Korea Institute of Science and Technology

Limitations #1: Complex Hair Designs



Limitations #1: Complex Hair Designs



Limitations #2: Eye Glasses

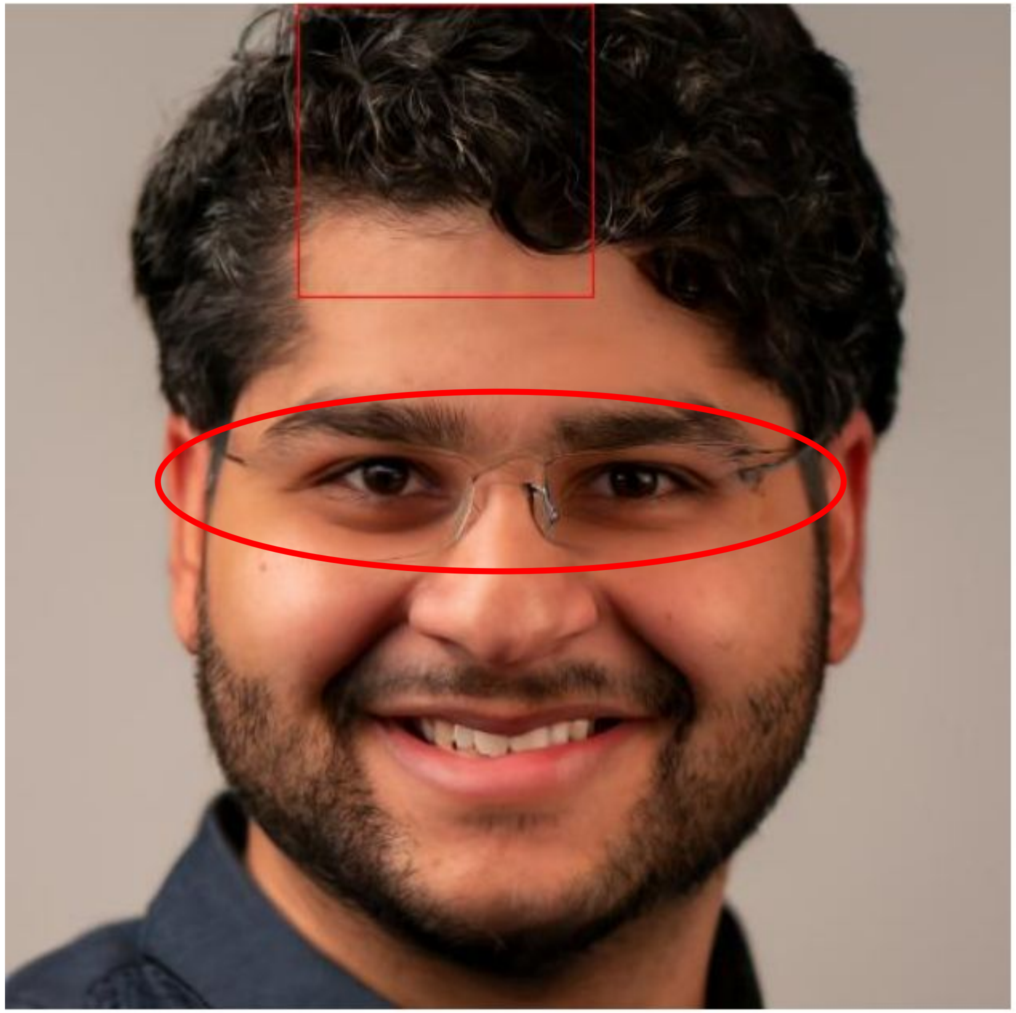
SR3

Reference

Korea Institute of Science and Technology

Limitations #2: Eye Glasses

SR3



Reference

Korea Institute of Science and Technology

Limitations #3: Meaningful Text Generation

SR3 **Reference**

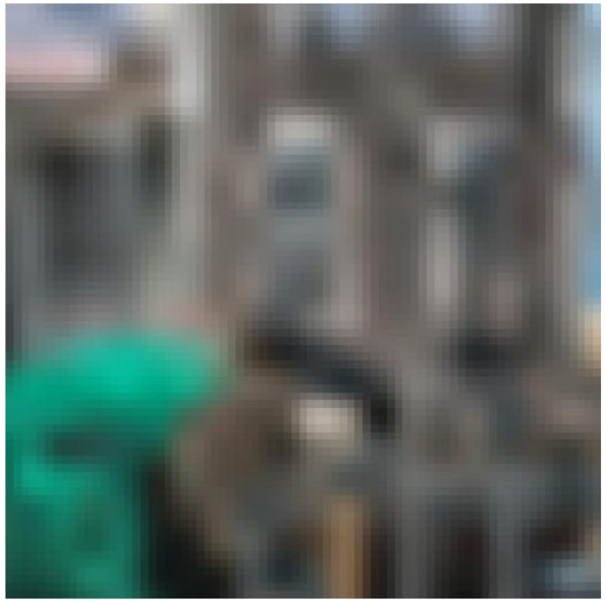
Korea Institute of Science and Technology

Limitations #3: Meaningful Text Generation

SR3 | **Reference**

Limitations #3: Certain Texture in Buildings

SR3 | Reference



Korea Institute of Science and Technology

Limitations #3: Certain Texture in Buildings

SR3 Reference

Related Works - SOTA

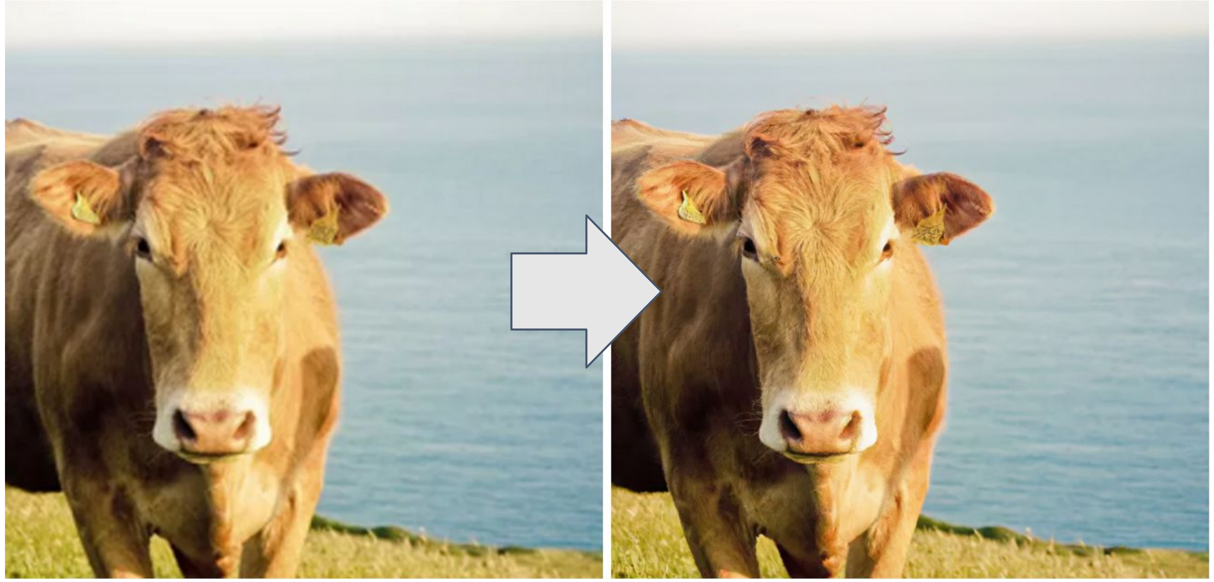
High-Resolution Image Synthesis with Latent Diffusion Models

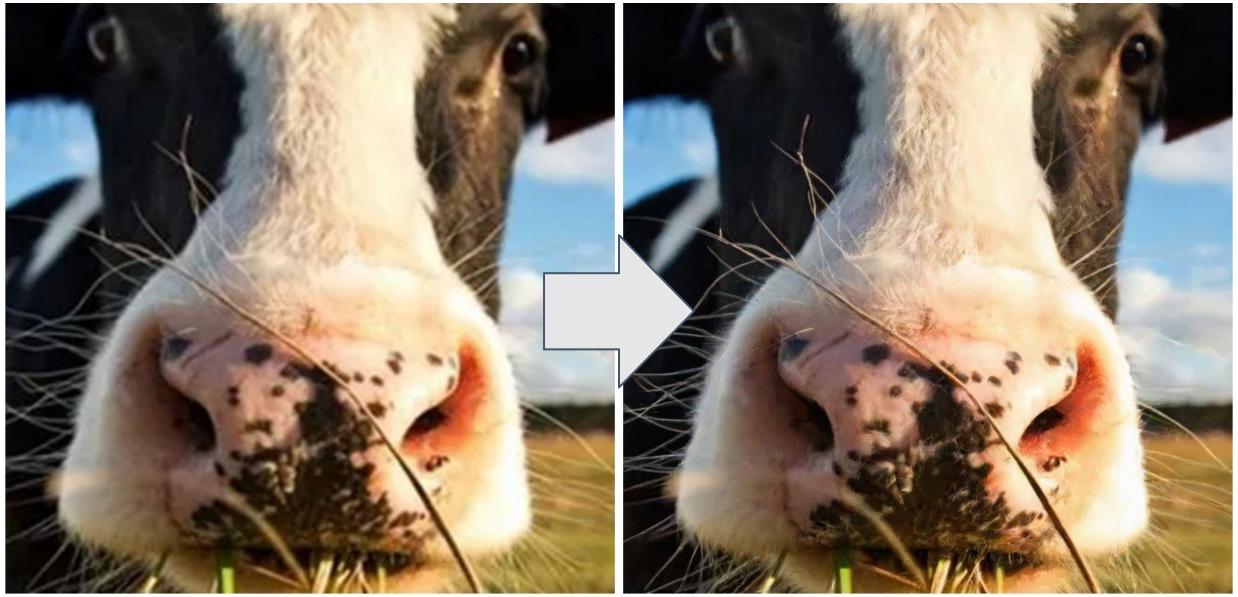
Robin Rombach[1] * Andreas Blattmann[1] * Dominik Lorenz[1] Patrick Esser[R] Björn Ommer[1]
[1]Ludwig Maximilian University of Munich & IWR, Heidelberg University, Germany [R]Runway ML
https://github.com/CompVis/latent-diffusion

Cited by 1077 CVPR 2022

High-Resolution Image Synthesis with Latent Diffusion Models

High Quality Samples



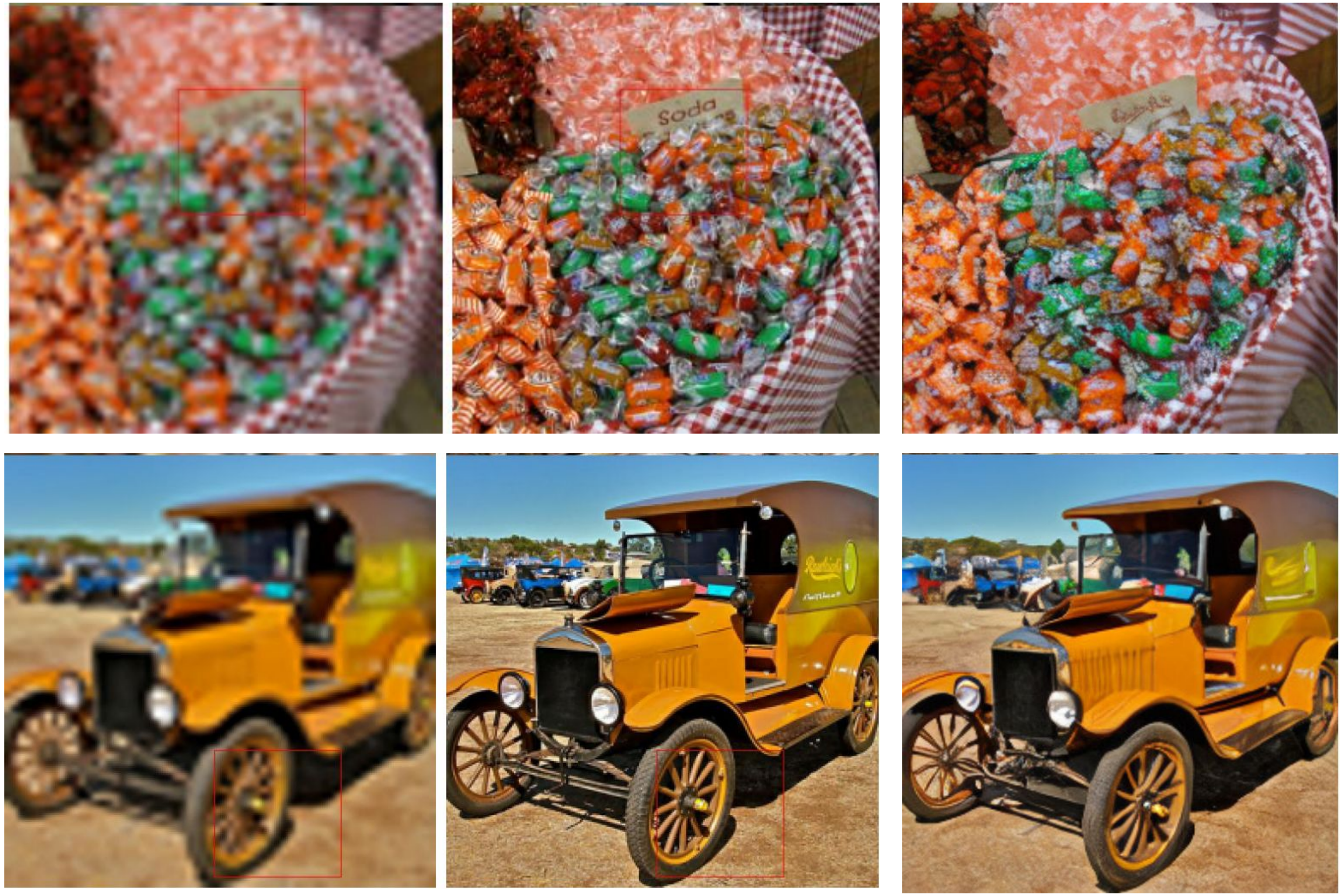


Korea Institute of Science and Technology

High-Resolution Image Synthesis with Latent Diffusion Models

Problematic Samples

input GT



Korea Institute of Science and Technology

High-Resolution Image Synthesis with Latent Diffusion Models

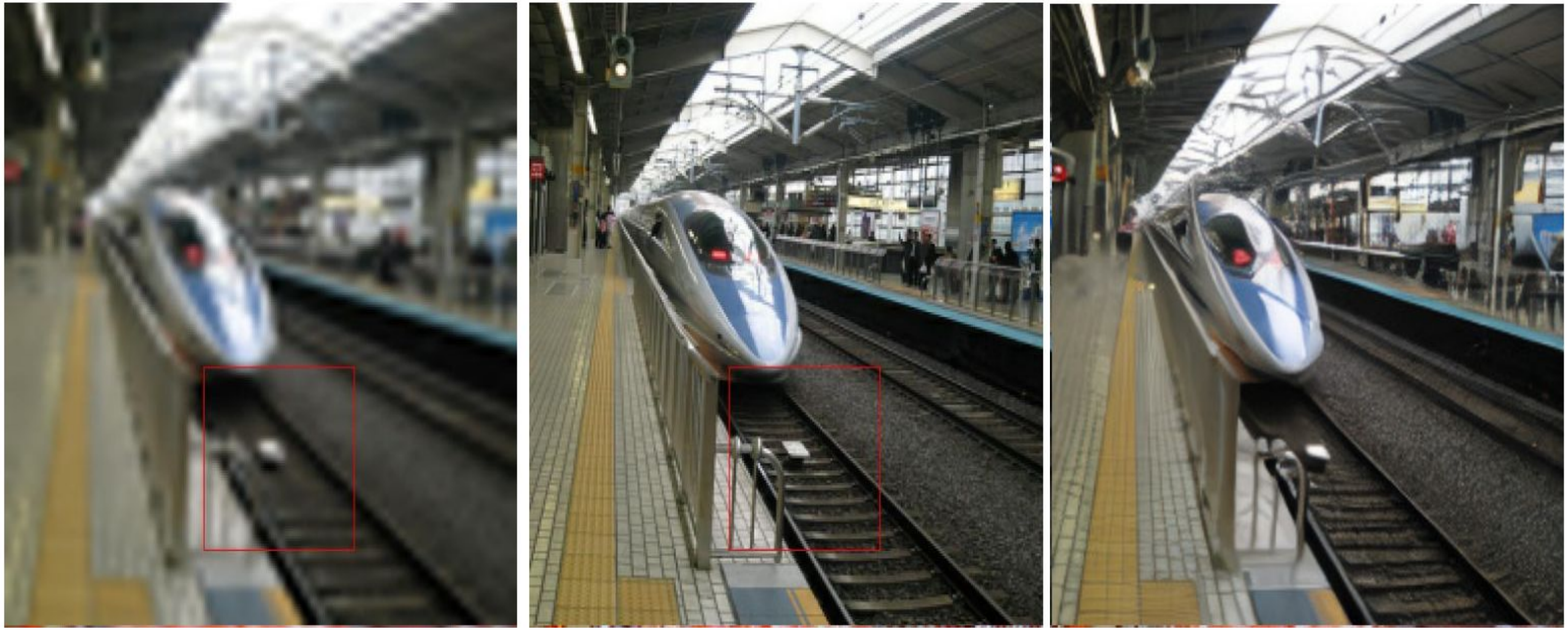
Problematic Samples

Korea Institute of
Science and Technology

High-Resolution Image Synthesis with Latent Diffusion Models

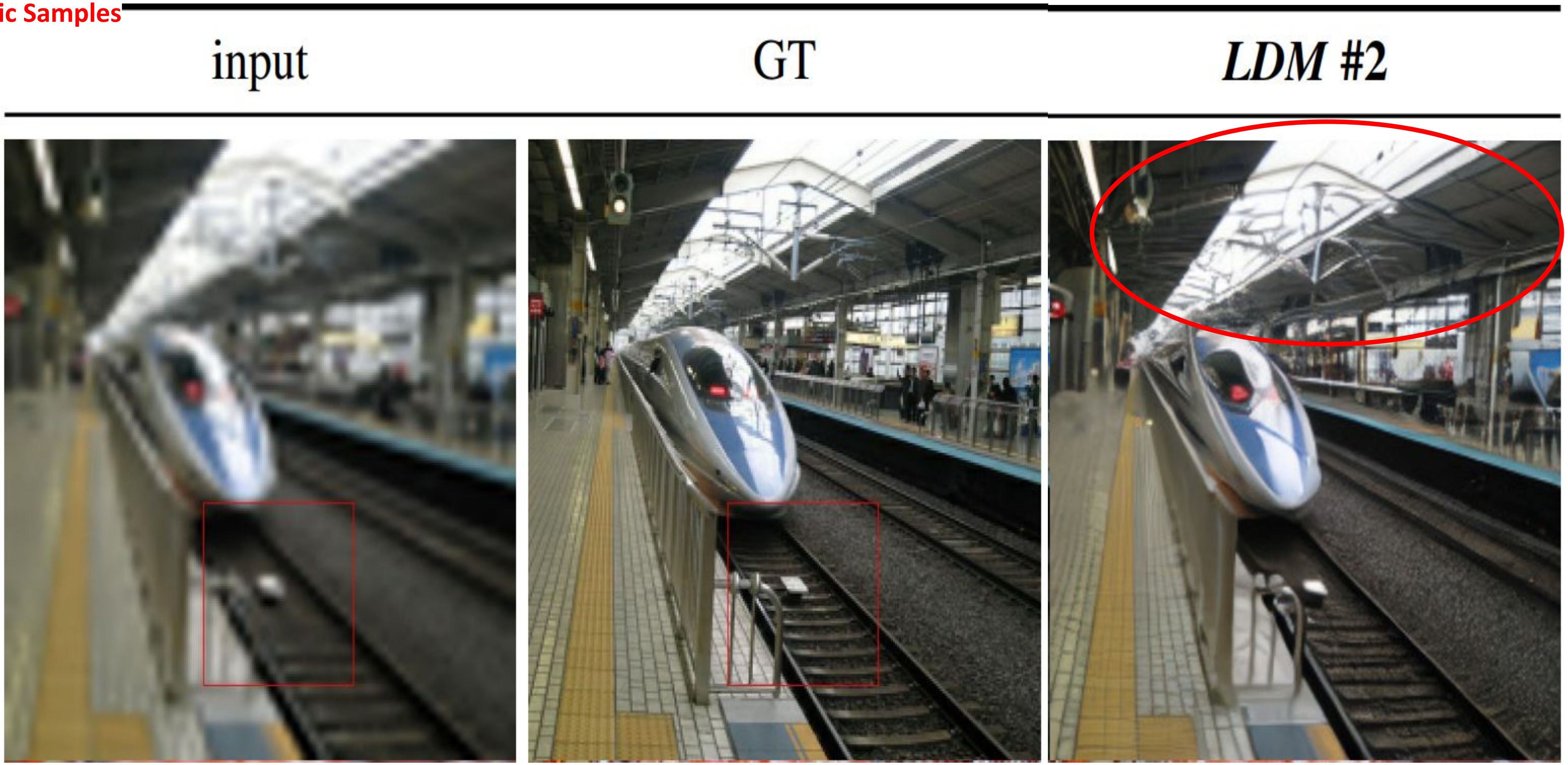
Problematic Samples

input	GT	*LDM #2*



High-Resolution Image Synthesis with Latent Diffusion Models

Problematic Samples



High-Resolution Image Synthesis with Latent Diffusion Models

Problematic Samples

Input | GT | LDM

Korea Institute of Science and Technology

High-Resolution Image Synthesis with Latent Diffusion Models

Problematic Samples

Input GT LDM

Common Shortcomings in Performance of Super resolution approaches

#1. Fails to generate meaningful text

GT

SR3

LDM

Korea Institute of
Science and Technology

Common Shortcomings in Performance of Super resolution approaches

#2. Fails to generate complex edges

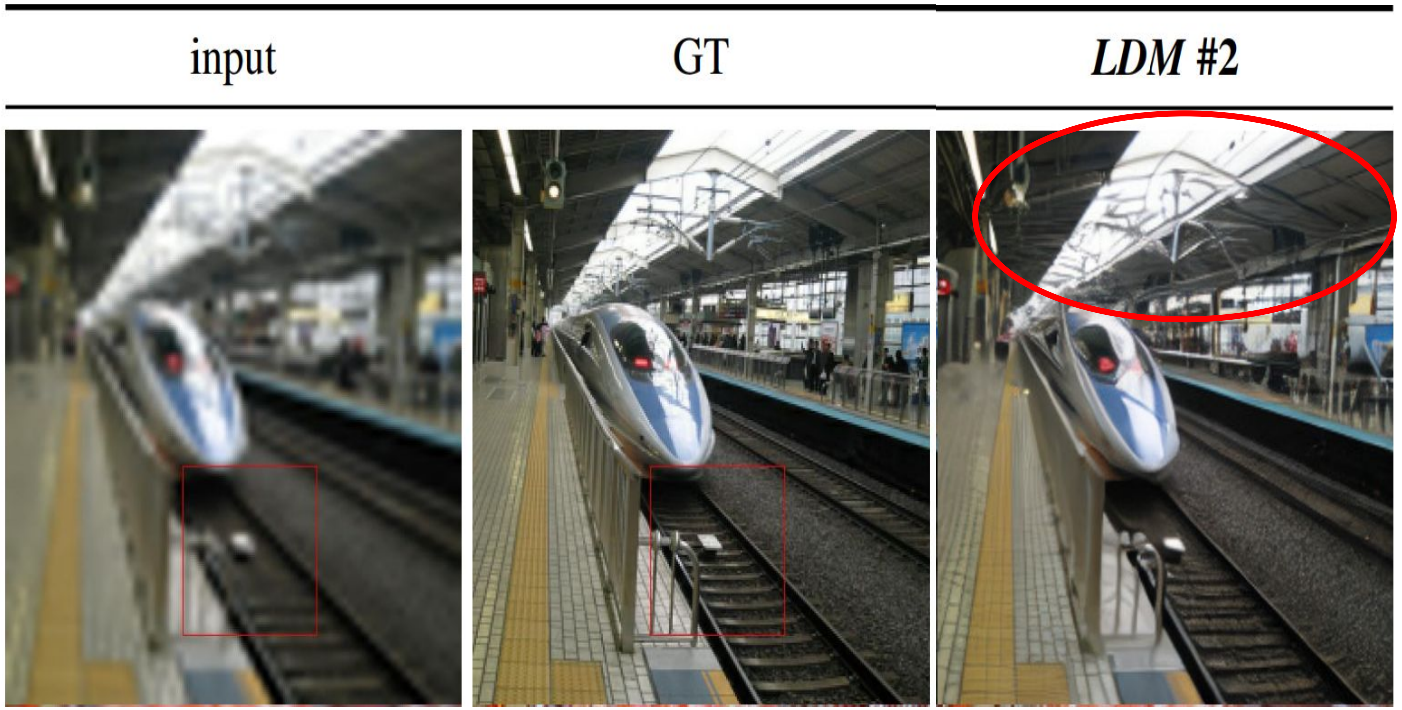
GT

SR3

LDM

Common Shortcomings in Performance of Super resolution approaches

#2. Fails to generate complex edges



Korea Institute of Science and Technology

Common Shortcomings in Performance of Super resolution approaches

#1. Fails to generate meaningful text

#2. Fails to generate complex edges

Korea Institute of Science and Technology

Common Shortcomings in Performance of Super resolution approaches

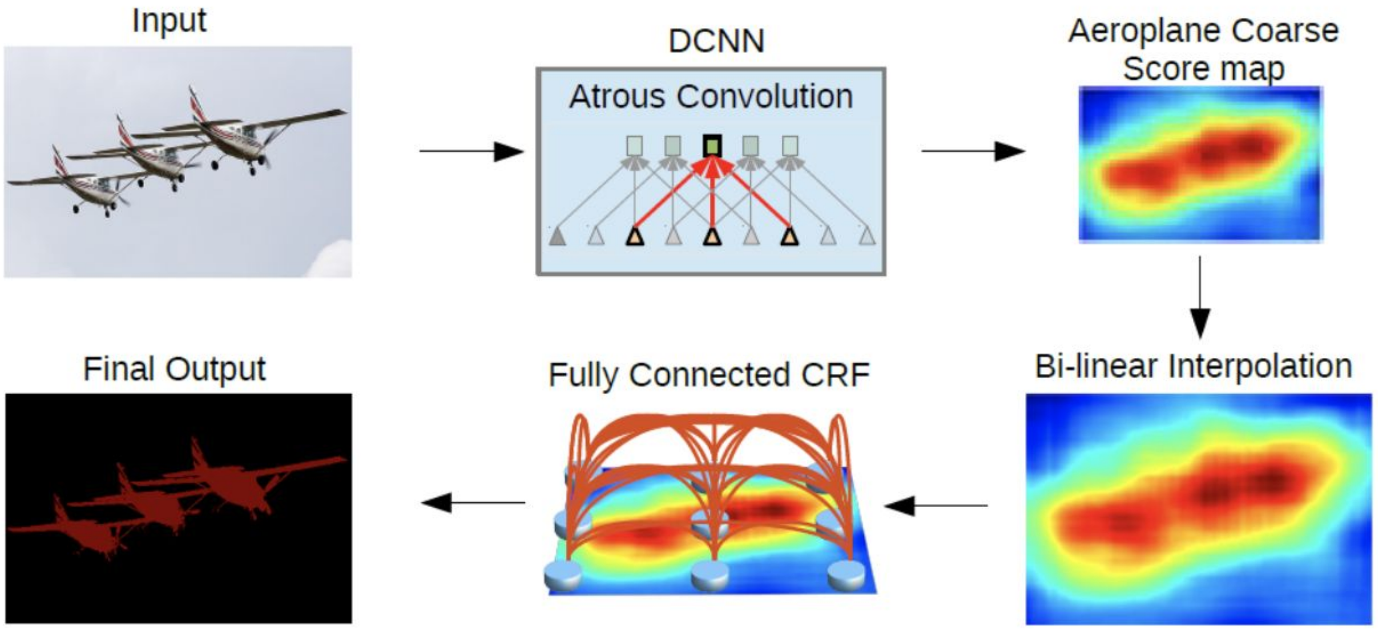
#1. Fails to generate meaningful text

#2. Fails to generate complex edges ✓

Possible Improvements to Diffusion-based Super Resolution Approaches

Rank	Model	Mean IoU	FLOPS	Params	Extra Training Data	Paper	Code	Result	Year	Tags
1	**DeepLabv3+** (Xception-65-JFT)	89.0%			✓	Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation			2018	
2	**DeepLabv3+** (Xception-JFT)	89.0%			✓	Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation			2018	
3	**DeepLabv3-JFT**	86.9%			✓	Rethinking Atrous Convolution for Semantic Image Segmentation			2017	

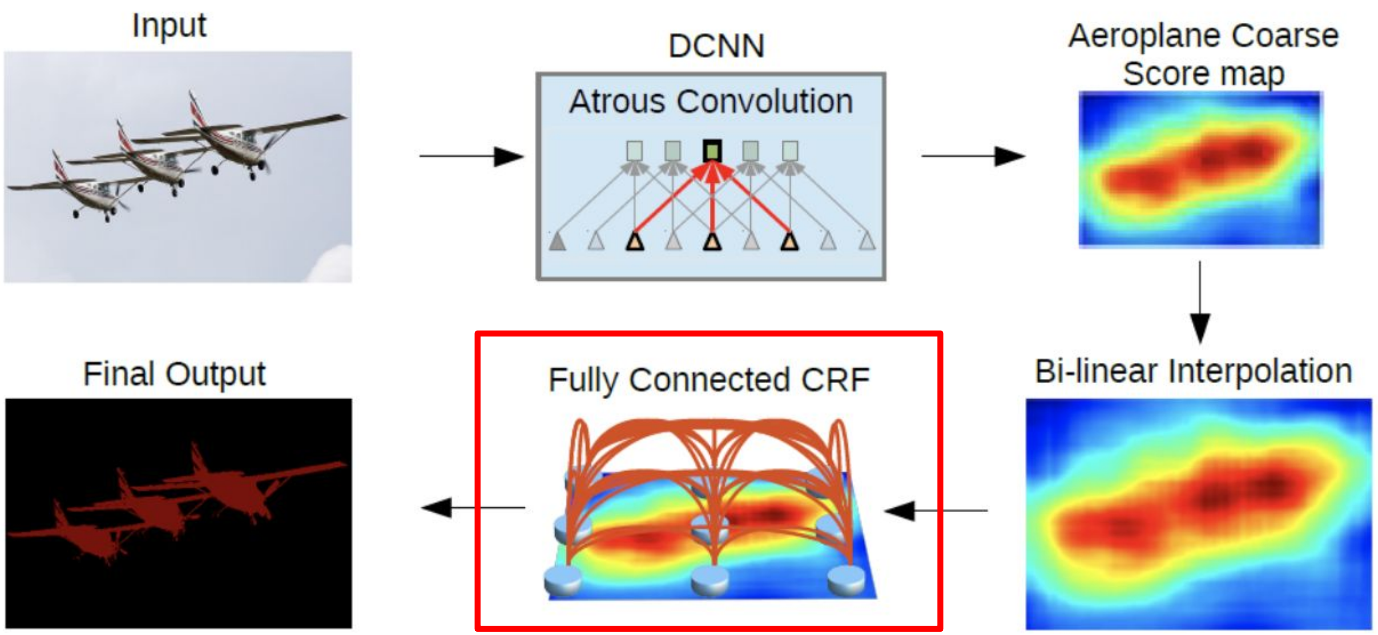
PASCAL VOC 2012 test



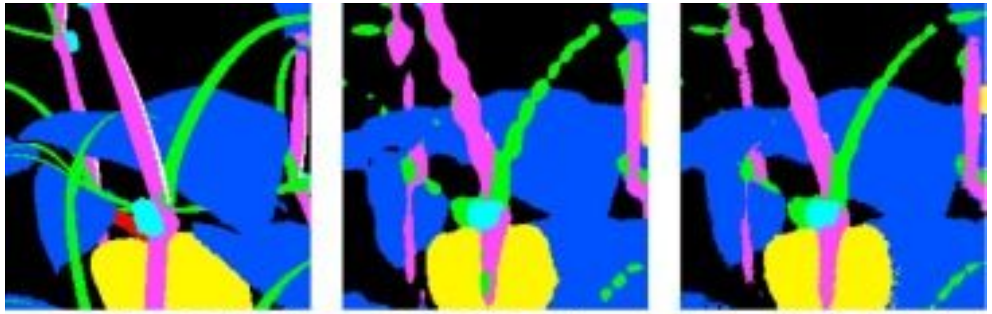
Possible Improvements to Diffusion-based Super Resolution Approaches

Rank	Model	Mean IoU	FLOPS	Params	Extra Training Data	Paper	Code	Result	Year	Tags
1	**DeepLabv3+** (Xception-65-JFT)	89.0%			✓	Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation			2018	
2	**DeepLabv3+** (Xception-JFT)	89.0%			✓	Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation			2018	
3	**DeepLabv3-JFT**	86.9%			✓	Rethinking Atrous Convolution for Semantic Image Segmentation			2017	

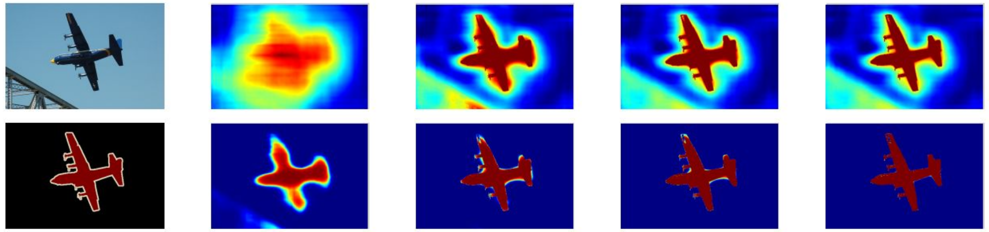
PASCAL VOC 2012 test



Possible Improvements to Diffusion-based Super Resolution Approaches

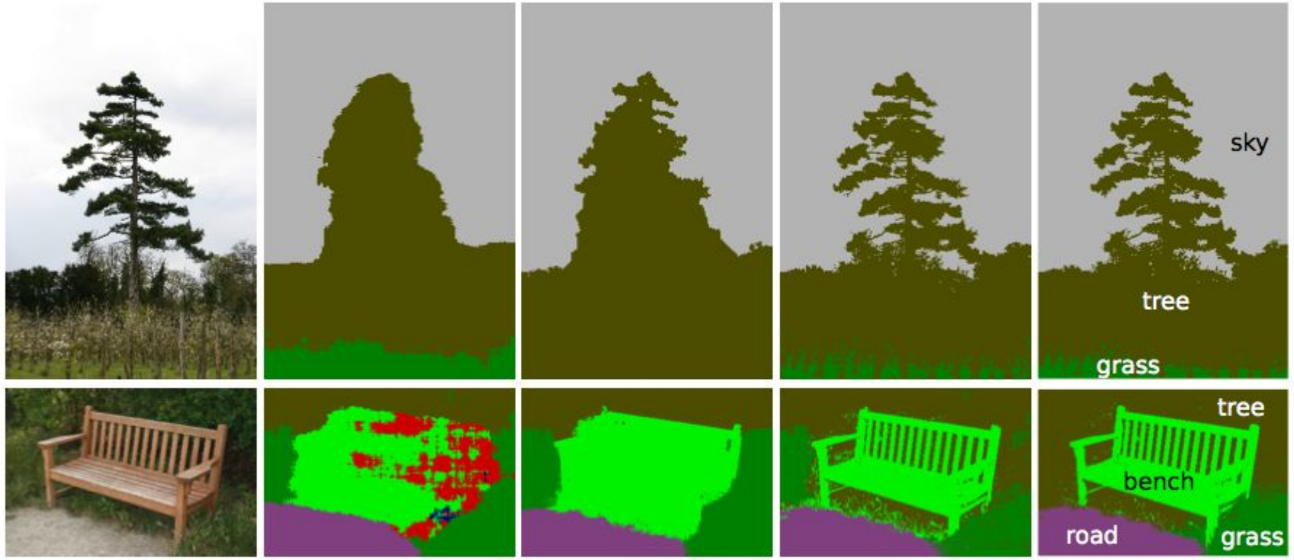


ground truth label | DeepLab-v2 | DeepLab-v2-crf

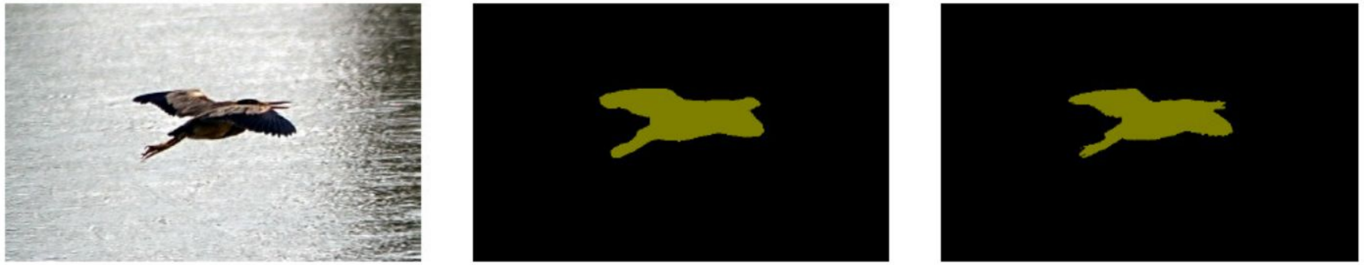


Image/G.T. | DCNN output | CRF Iteration 1 | CRF Iteration 2 | CRF Iteration 10

Fig. 5: Score map (input before softmax function) and belief map (output of softmax function) for Aeroplane. We show the score (1st row) and belief (2nd row) maps after each mean field iteration. The output of last DCNN layer is used as input to the mean field inference.



(a) Image | (b) Unary classifiers | (c) Robust P^n CRF | (d) Fully connected CRF, MCMC inference, 36 hrs | (e) Fully connected CRF, our approach, 0.2 seconds

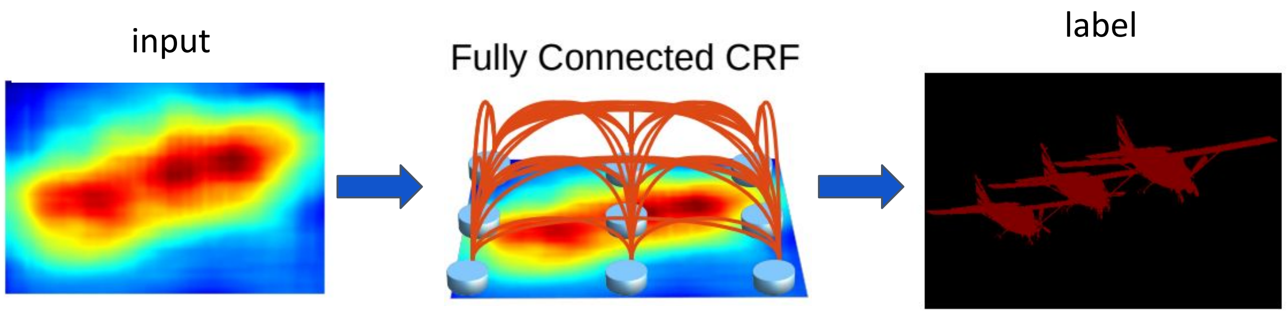


(a) Image | (b) Before CRF | (c) After CRF

Common Shortcomings in Performance of Super resolution approaches

#1. Fails to generate meaningful text

#2. Fails to generate complex edges



The pairwise potentials in our model have the form

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \underbrace{\sum_{m=1}^{K} w^{(m)} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j)}_{k(\mathbf{f}_i, \mathbf{f}_j)},$$

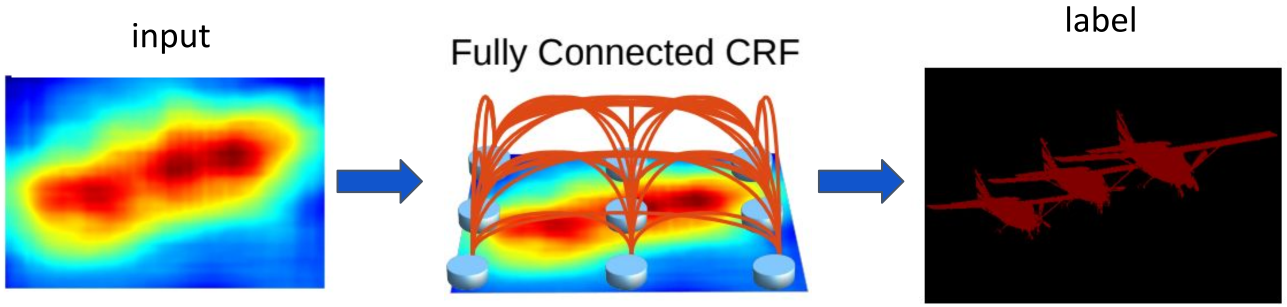
x_i, x_j : Labels

$$\mu(x_i, x_j) = \mathbb{1}\{x_i \neq x_j\}$$

Common Shortcomings in Performance of Super resolution approaches

#1. Fails to generate meaningful text

#2. Fails to generate complex edges



The pairwise potentials in our model have the form

$$\psi_p(x_i, x_j) = \boxed{\mu(x_i, x_j)} \underbrace{\sum_{m=1}^{K} w^{(m)} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j)}_{k(\mathbf{f}_i, \mathbf{f}_j)},$$

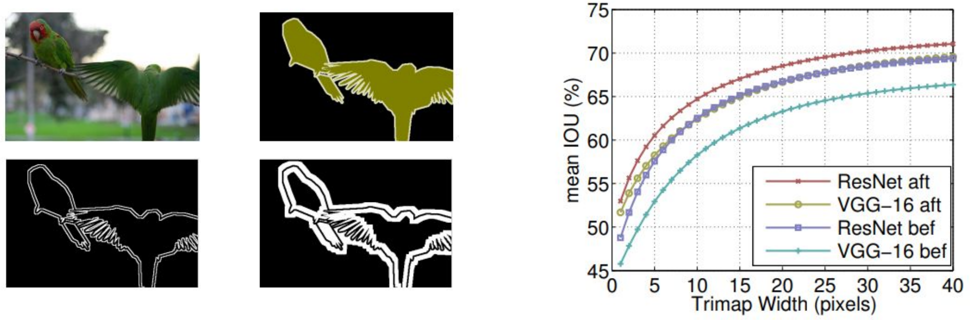
x_i, x_j : Labels

$$\boxed{\mu(x_i, x_j) = \mathbb{1}\{x_i \neq x_j\}}$$

introduces a penalty for nearby similar pixels that are assigned different labels.

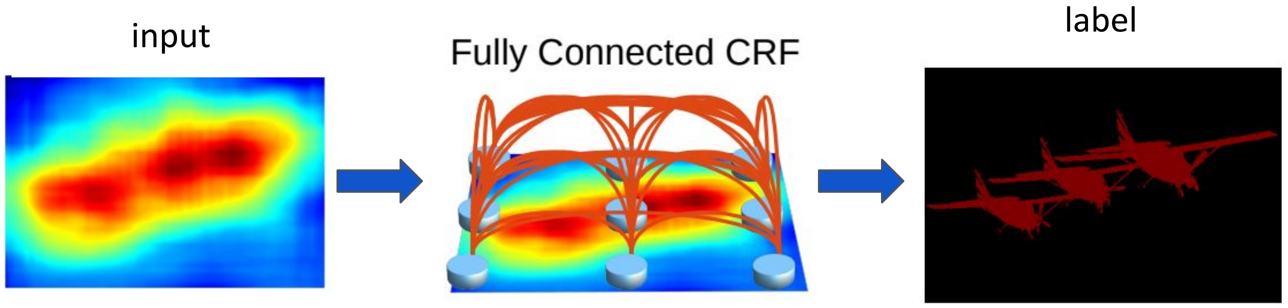
Korea Institute of Science and Technology

ce of Super resolution approaches



(a) (b)

Fig. 10: (a) Trimap examples (top-left: image. top-right: ground-truth. bottom-left: trimap of 2 pixels. bottom-right: trimap of 10 pixels). (b) Pixel mean IOU as a function of the band width around the object boundaries when employing VGG-16 or ResNet-101 before and after CRF.



The pairwise potentials in our model have the form

$$\psi_p(x_i, x_j) = \boxed{\mu(x_i, x_j)} \underbrace{\sum_{m=1}^{K} w^{(m)} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j)}_{k(\mathbf{f}_i, \mathbf{f}_j)},$$

x_i, x_j : Labels

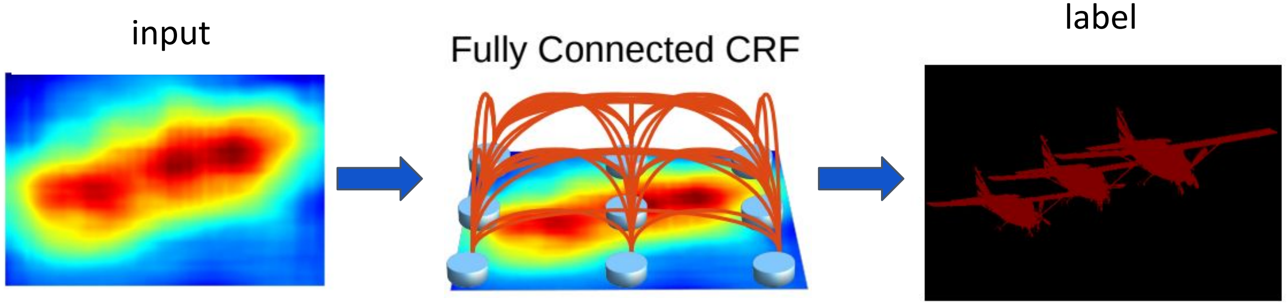
$$\boxed{\mu(x_i, x_j) = \mathbb{1}\{x_i \neq x_j\}}$$

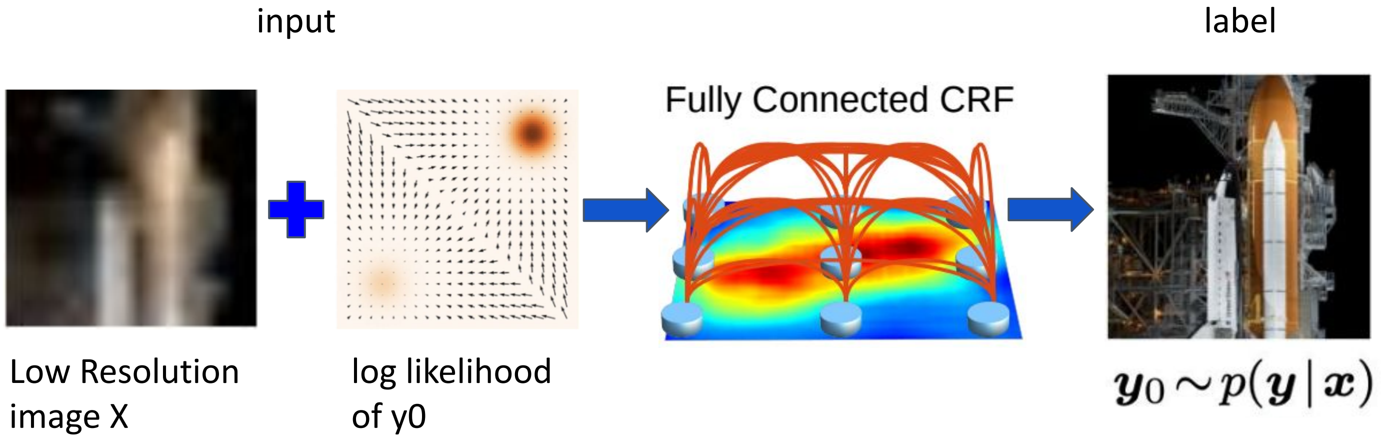
introduces a penalty for nearby similar pixels that are assigned different labels.

Common Shortcomings in Performance of Super resolution approaches

#1. Fails to generate meaningful text

#2. Fails to generate complex edges ✓





Korea Institute of Science and Technology

Common Shortcomings in Performance of Super resolution approaches

#1. Fails to generate meaningful text

#2. Fails to generate complex edges ✓

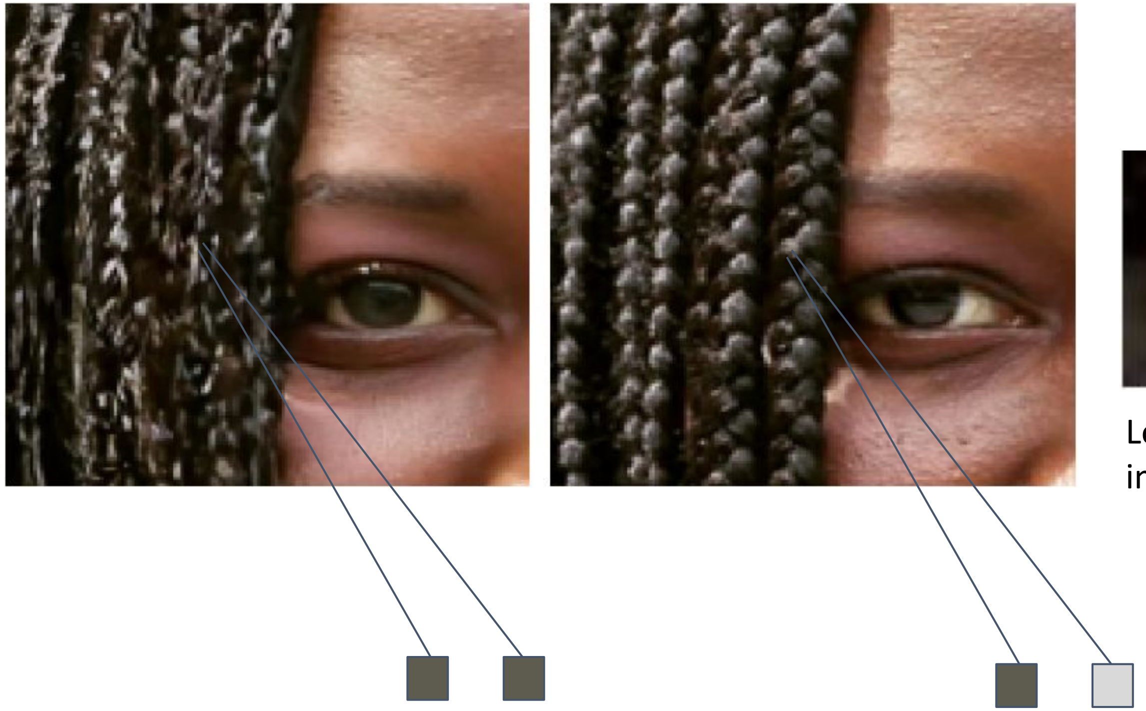
The pairwise potentials in our model have the form

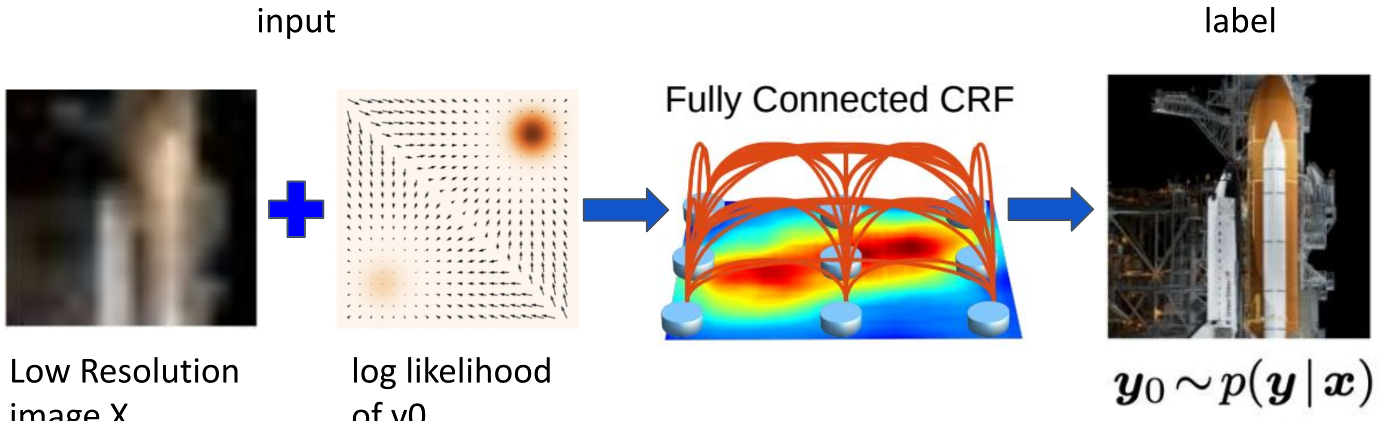
$$\psi_p(x_i, x_j) = \boxed{\mu(x_i, x_j)} \underbrace{\sum_{m=1}^{K} w^{(m)} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j)}_{k(\mathbf{f}_i, \mathbf{f}_j)},$$

x_i, x_j : Labels

$$\boxed{\mu(x_i, x_j) = \mathbb{1}\{x_i \neq x_j\}}$$

introduces a penalty for nearby similar pixels that are assigned different labels.





Korea Institute of Science and Technology

Common Shortcomings in Performance of Super resolution approaches

#1. Fails to generate meaningful text

#2. Fails to generate complex edges ✓

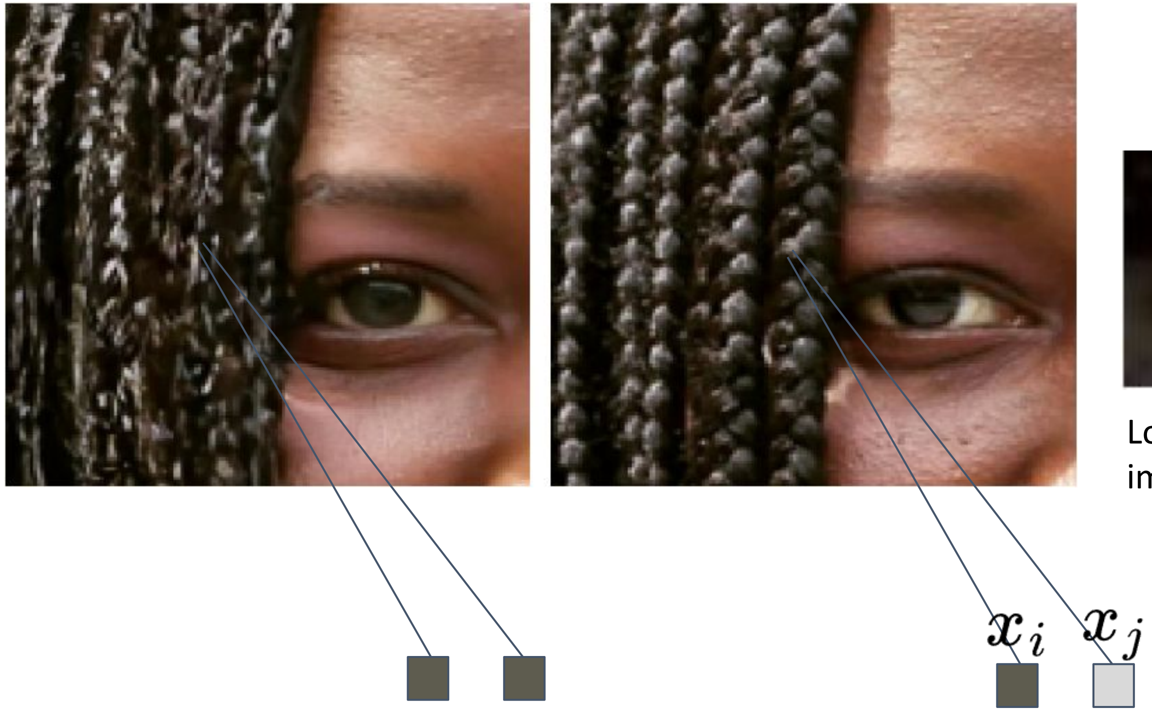
The pairwise potentials in our model have the form

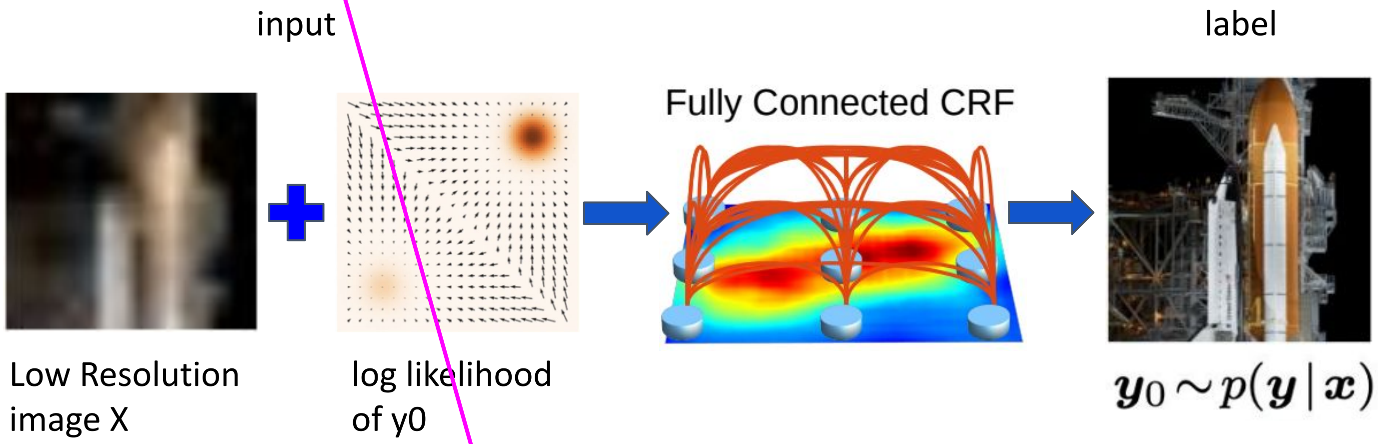
$$\psi_p(x_i, x_j) = \boxed{\mu(x_i, x_j)} \underbrace{\sum_{m=1}^{K} w^{(m)} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j)}_{k(\mathbf{f}_i, \mathbf{f}_j)},$$

$x_i, x_j :$ Labels

$\boxed{\mu(x_i, x_j) = \mathbb{1}\{x_i \neq x_j\}}$

introduces a penalty for nearby similar pixels that are assigned different labels.





introduces a penalty for nearby similar pixels that are assigned different colours.

$\boxed{\mu(x_i, x_j) = ||x_i - x_j||^2}$

Korea Institute of Science and Technology

Thank you!!!

- 영어로 모두 작성
- 요약 3-4 줄
- Major comment 조언 (Critical, 개괄식 가능 & 5번째까지 가능, 궁금한 점, 놓친 것)
- Minor comment (철자, Figure)
- 반 페이지 ~ 1페이지
- 내일까지