# StyleGuide

## Crafting visual style prompting with negative visual query guidance

NAVER AI Lab, Generation research

김준호

https://github.com/taki0112

NAVER AI LAB

NAVER AI LAB

## Visual Style Prompting with Swapping Self-Attention

Jaeseok Jeong [*1,2] Junho Kim [*2] Yunjey Choi [2] Gayoung Lee [2] Youngjung Uh [1]

**Abstract**

In the evolving domain of text-to-image generation, diffusion models have emerged as powerful tools in content creation. Despite their remarkable capability, existing models still face challenges in achieving controlled generation with a consistent style, requiring costly fine-tuning or often inadequately transferring the visual elements due to content leakage. To address these challenges, we propose a novel approach, visual style prompting, to produce a diverse range of images while maintaining specific style elements and nuances. During the denoising process, we keep the query from original features while swapping the key and value with those from reference features in the late self-attention layers. This approach allows for the visual style prompting without any fine-tuning, en-

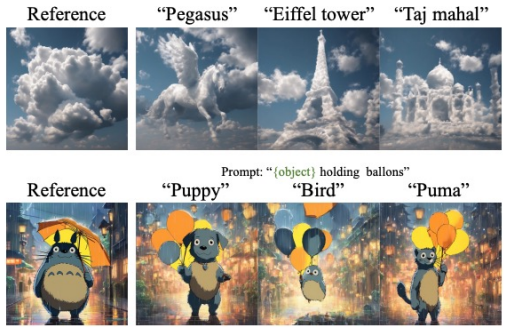



*Figure 1.* We tackle visual style prompting, reflecting style elements from reference images and contents from text prompts, in a training-free manner.

arxiv

## StyleGuide: Prevent Content Leakage Using Negative Query Guidance

**Anonymous authors**
Paper under double-blind review

**Abstract**

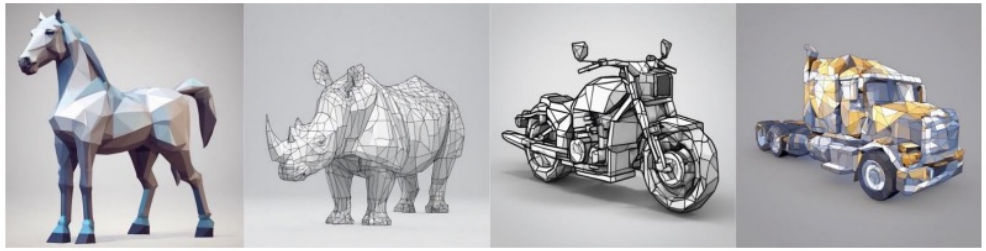
In the domain of text-to-image generation, diffusion models have emerged as powerful tools. Recently, studies on visual prompting, where images are used as prompts, have enabled more precise control over style and content. However, existing methods often suffer from content leakage, where undesired elements from the visual style prompt are transferred along with the intended style (content leakage). To address this issue, we 1) extend classifier-free guidance (CFG) to utilize swapping self-attention and propose 2) negative visual query guidance (NVQG) to reduce the transfer of unwanted contents. NVQG employs negative score by intentionally simulating content leakage scenarios which swaps queries instead of key and values of self-attention layers from visual style prompts. This simple yet effective method significantly reduces content leakage. Furthermore, we provide careful solutions for using a real image as a visual style prompts and for image-to-image (I2I) tasks. Through extensive evaluation across various styles and text prompts, our method demonstrates superiority over existing approaches, reflecting the style of the references and ensuring that resulting images match the text prompts.

ICLR 2025

NAVER AI LAB

"low-poly style horse"

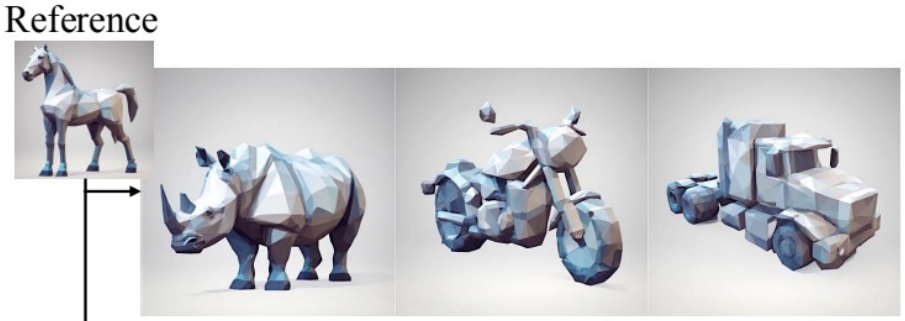
(a) Various style results within a text description



"Horse" "Rhino" "Motorcycle" "Truck"

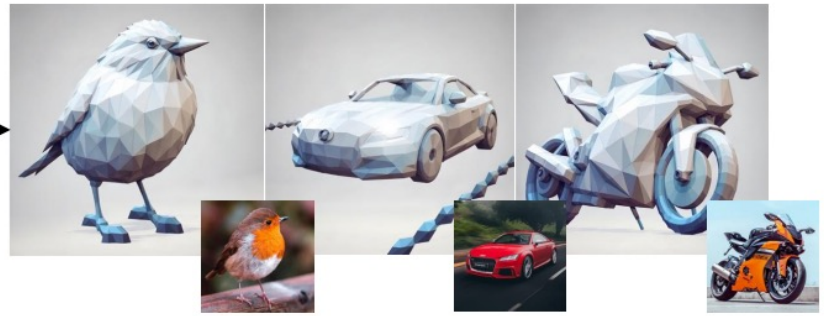
+ "low-poly style {object}. low-poly game art, polygon mesh, jagged, blocky, wireframe edges, centered composition, simple .."

(b) Various style results with a highly detailed text description




"Rhino" "Motorcycle" "Truck"

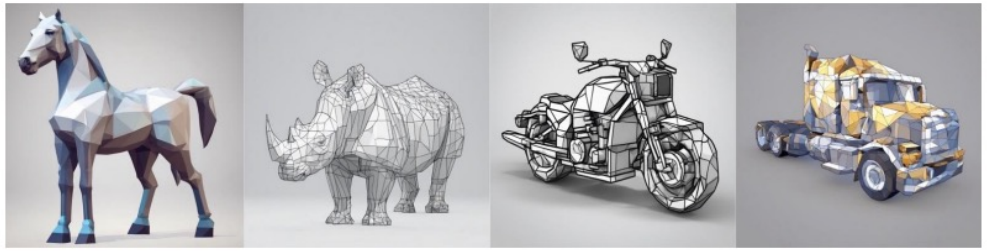
(c) T2I results specified by a style reference



(d) Style transfer results specified by a style reference

"low-poly style horse"

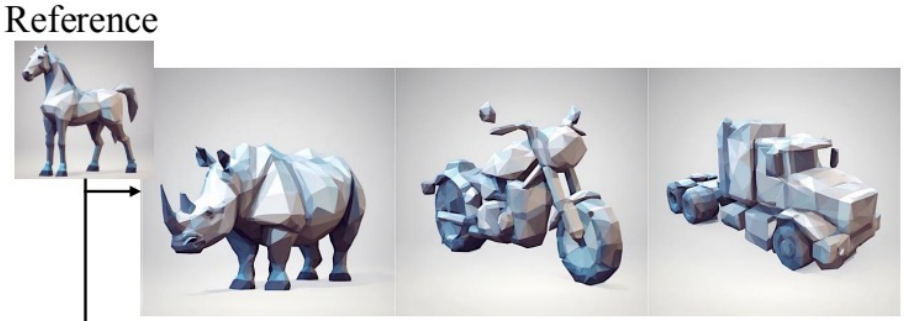
(a) Various style results within a text description



"Horse" "Rhino" "Motorcycle" "Truck"

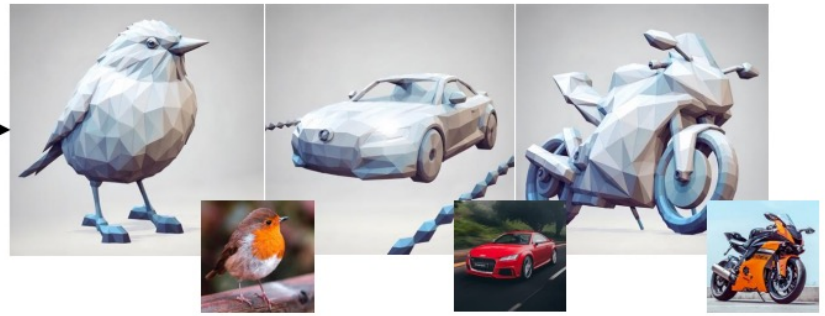
+ "low-poly style {object}. low-poly game art, polygon mesh, jagged, blocky, wireframe edges, centered composition, simple .."

(b) Various style results with a highly detailed text description




"Rhino" "Motorcycle" "Truck"

(c) T2I results specified by a style reference

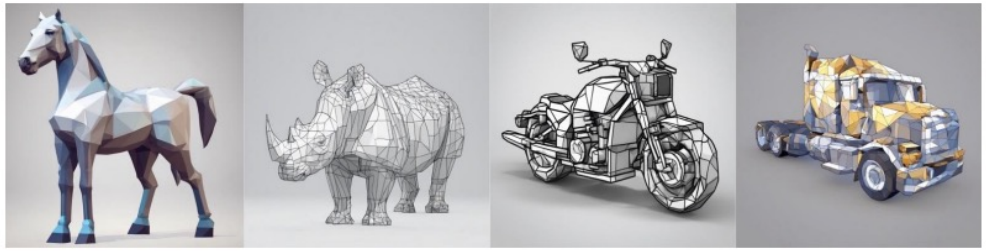


(d) Style transfer results specified by a style reference

NAVER AI LAB

"low-poly style horse"

(a) Various style results within a text description

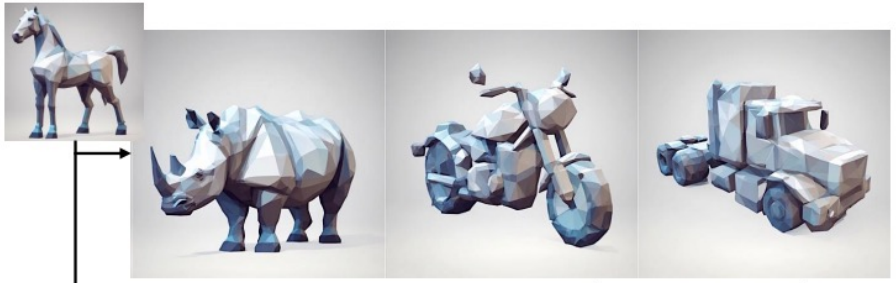


"Horse" "Rhino" "Motorcycle" "Truck"

+ "low-poly style {object}. low-poly game art, polygon mesh, jagged, blocky, wireframe edges, centered composition, simple .."

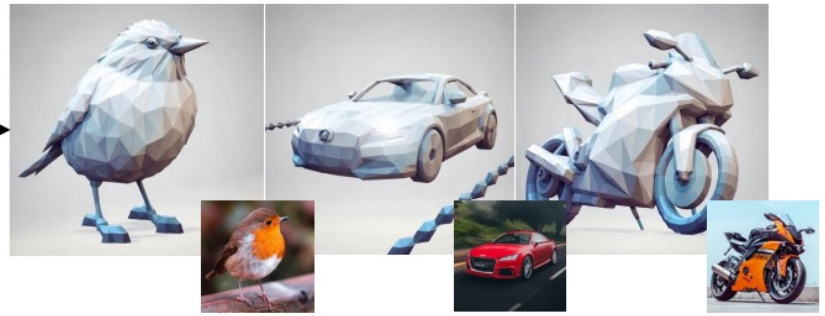
(b) Various style results with a highly detailed text description

Reference



"Rhino" "Motorcycle" "Truck"

(c) T2I results specified by a style reference

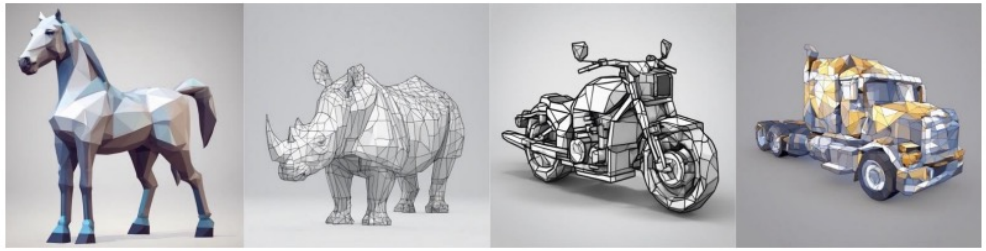


(d) Style transfer results specified by a style reference

NAVER AI LAB

"low-poly style horse"

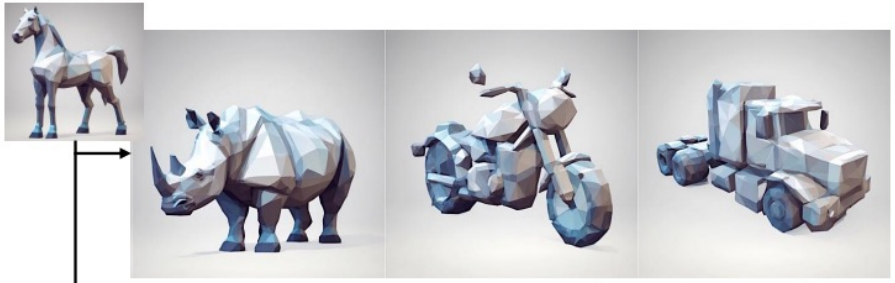
(a) Various style results within a text description



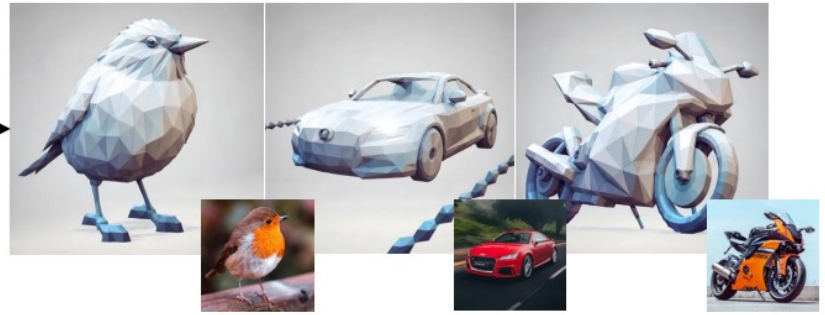
"Horse" "Rhino" "Motorcycle" "Truck"

+ "low-poly style {object}. low-poly game art, polygon mesh, jagged, blocky, wireframe edges, centered composition, simple .."

(b) Various style results with a highly detailed text description




(c) T2I results specified by a style reference

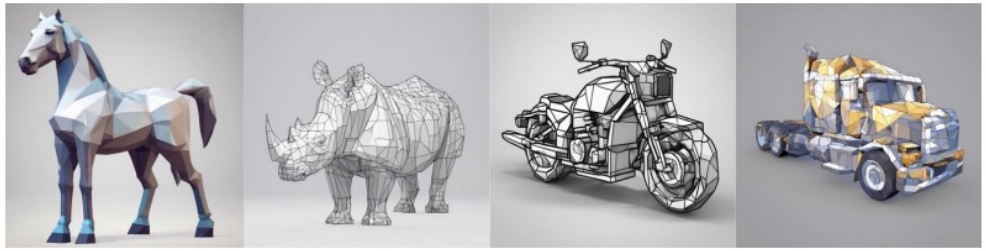


(d) Style transfer results specified by a style reference

NAVER AI LAB

"low-poly style horse"

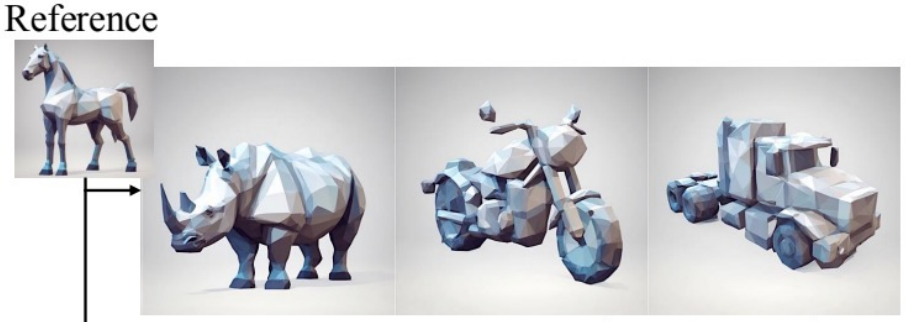
(a) Various style results within a text description



"Horse" "Rhino" "Motorcycle" "Truck"

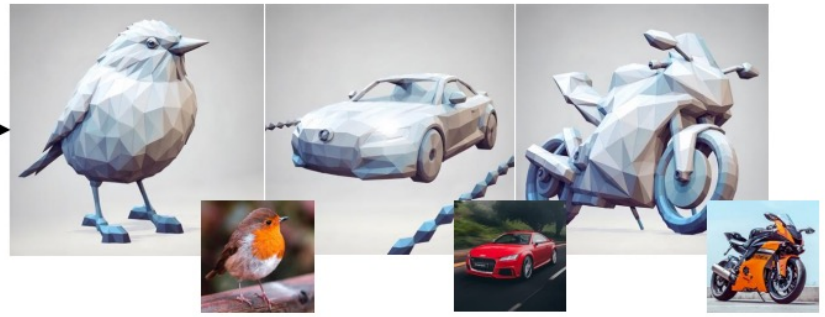
+ "low-poly style {object}. low-poly game art, polygon mesh, jagged, blocky, wireframe edges, centered composition, simple .."

(b) Various style results with a highly detailed text description




"Rhino" "Motorcycle" "Truck"

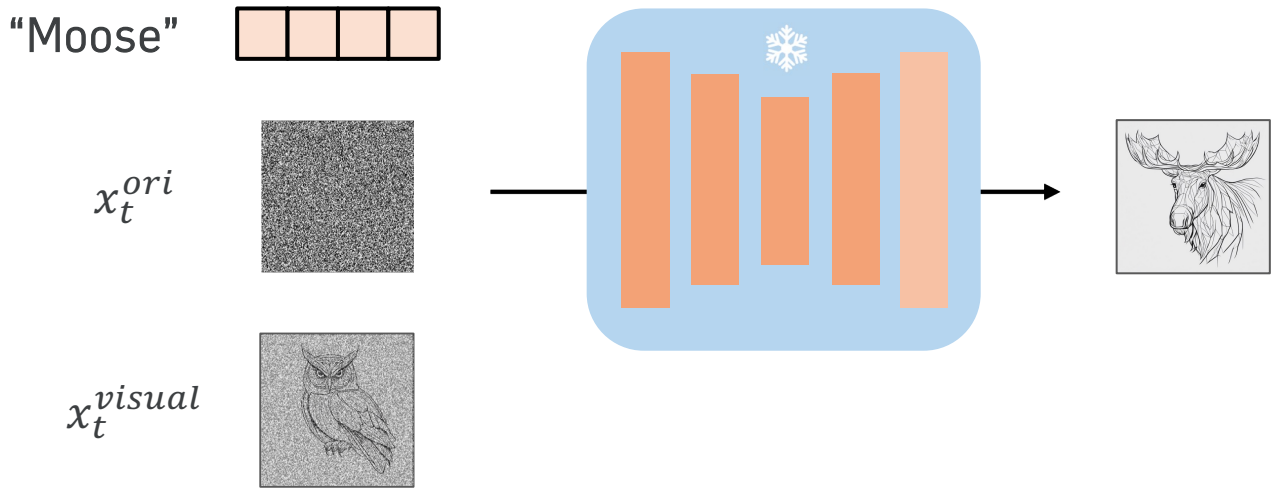
(c) T2I results specified by a style reference



(d) Style transfer results specified by a style reference

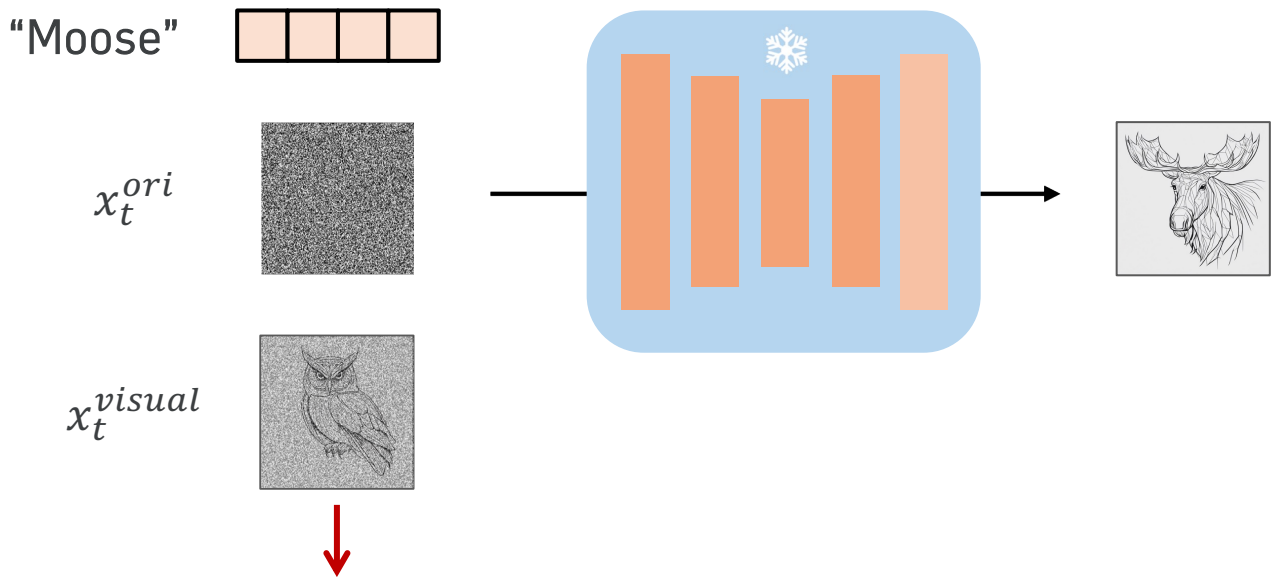
NAVER AI LAB

## Overview



NAVER AI LAB

## Overview

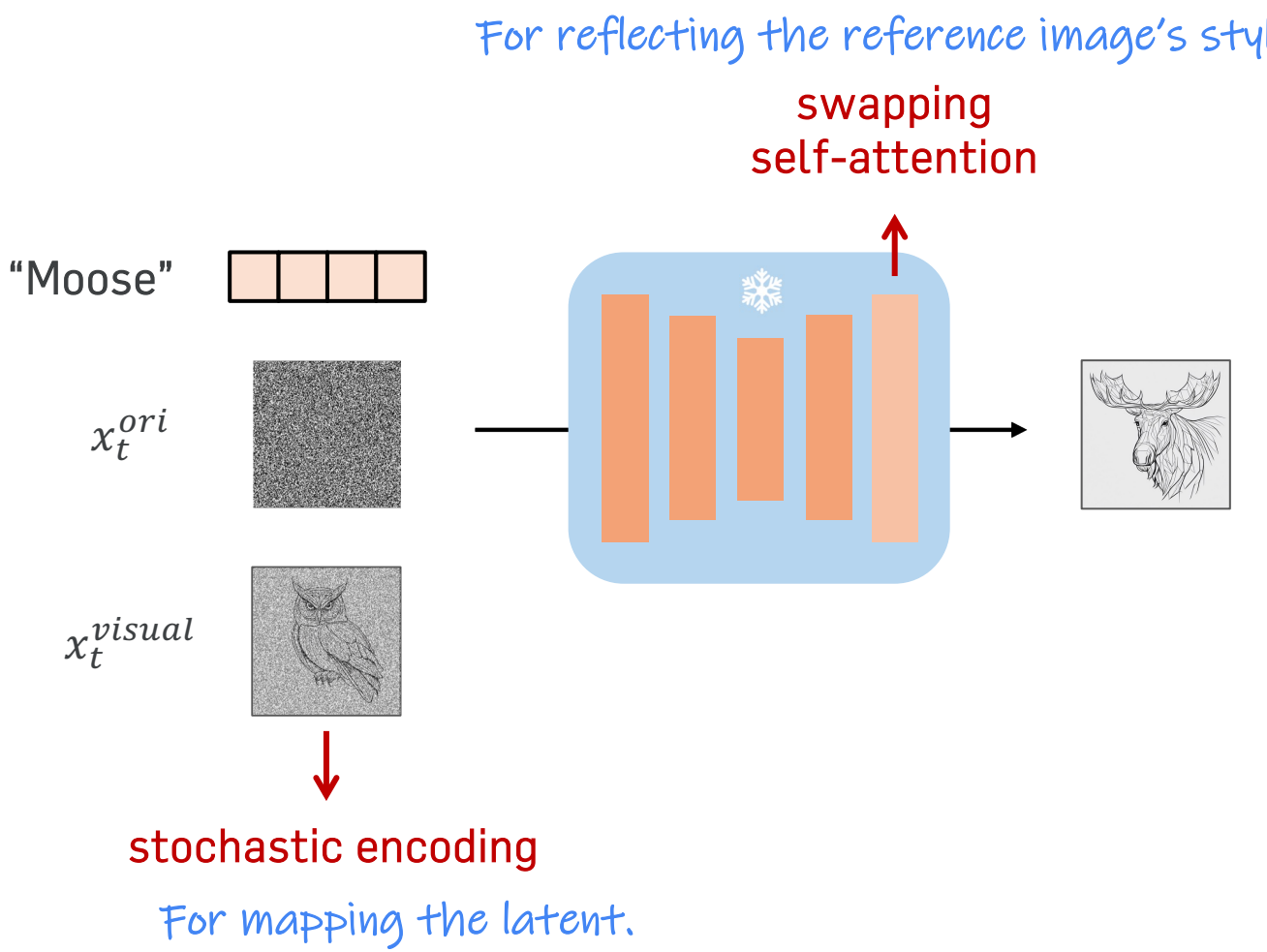



stochastic encoding

For mapping the latent.

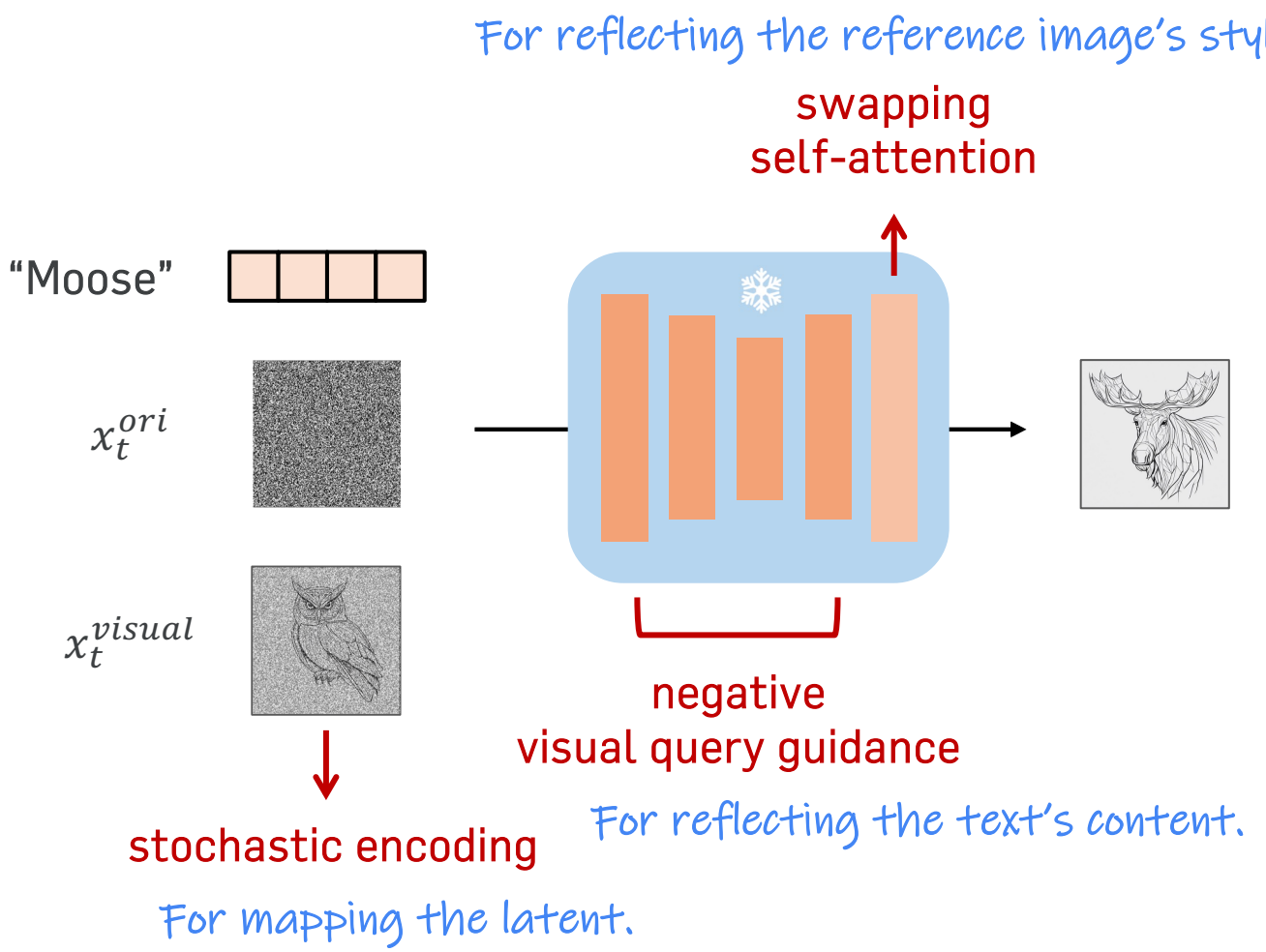
NAVER AI LAB

## Overview



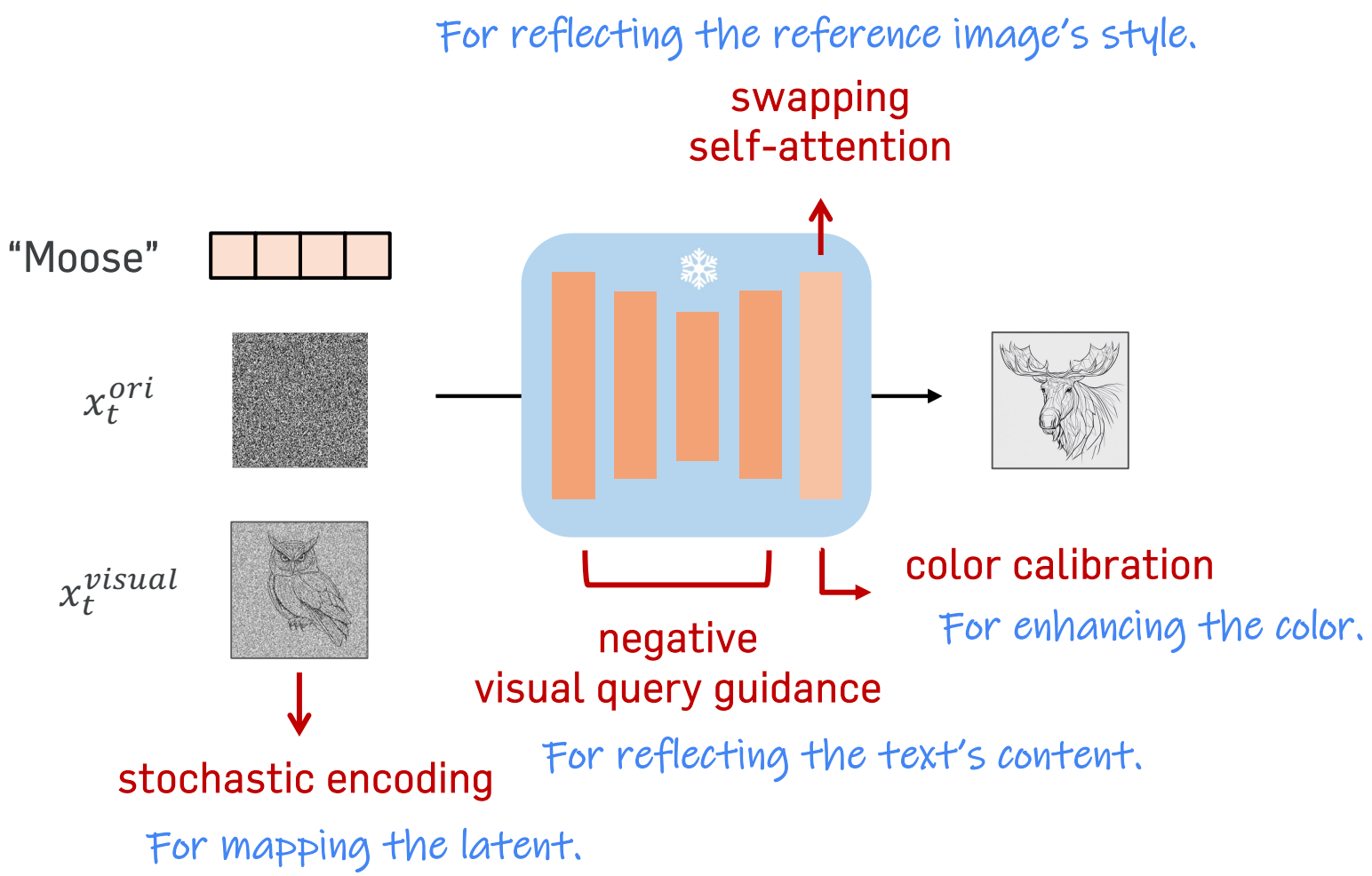
NAVER AI LAB

## Overview



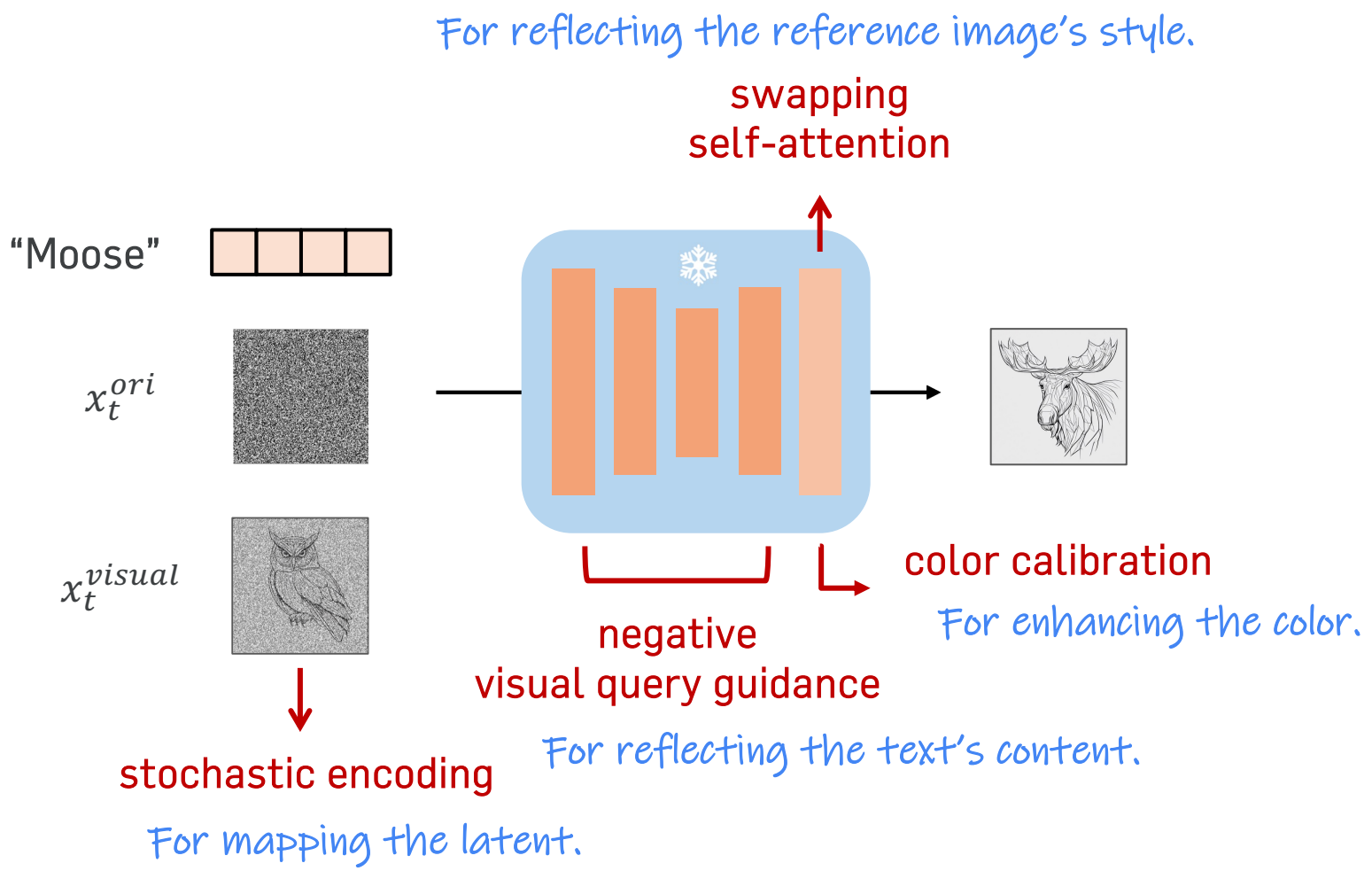
NAVER AI LAB

## Overview



NAVER AI LAB

## Overview

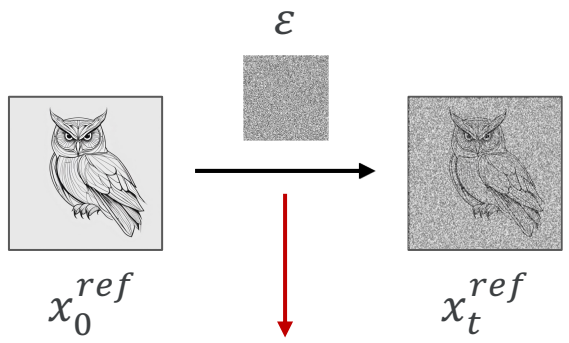



### Summary

- Training-free
- Versatile
- SoTA
- Efficient & Reliable

NAVER AI LAB

## Stochastic encoding




$$x_t^{ref} = \sqrt{\alpha_t} * x_0^{ref} + \sqrt{1-\alpha_t} * \varepsilon$$

For mapping the latent.

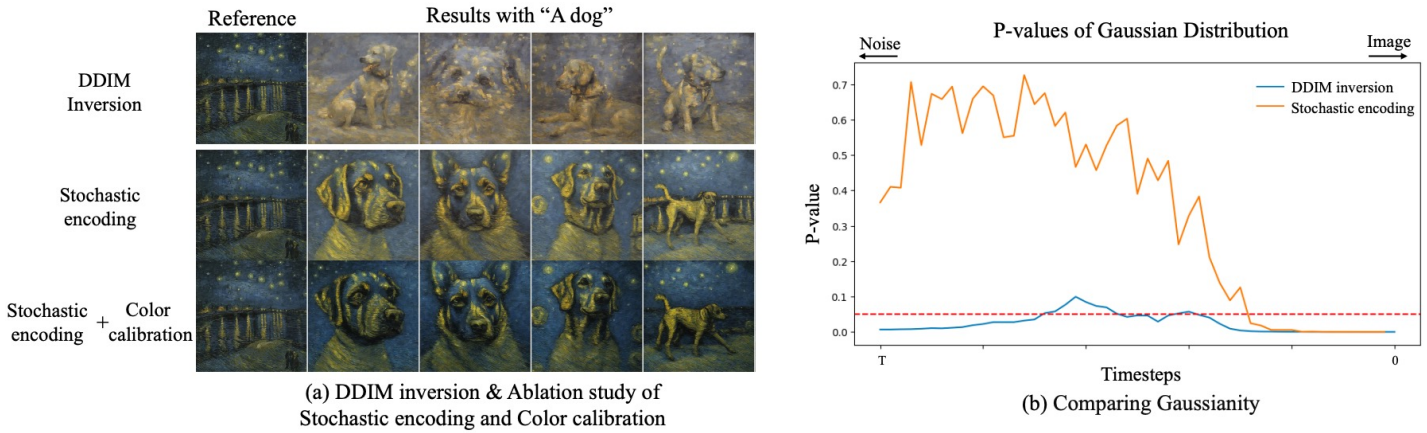
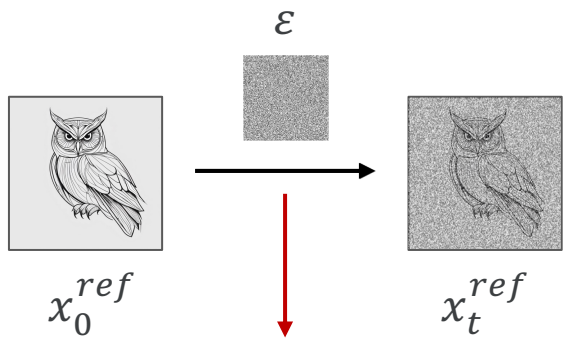



Figure 12: **(a) Comparison of DDIM inversion vs. stochastic encoding and the effect of color calibration.** Stochastic encoding reduces artifacts in the resulting images, while color calibration better reflects the colors of the reference image. **(b) Stochastic encoding produces the latents closer to the standard Gaussian distribution compared to DDIM inversion.** A P-value above 0.05 suggests that the data likely follows the standard Gaussian distribution.

NAVER AI LAB

## Stochastic encoding




$$x_t^{ref} = \sqrt{\alpha_t} * x_0^{ref} + \sqrt{1-\alpha_t} * \varepsilon$$

For mapping the latent.

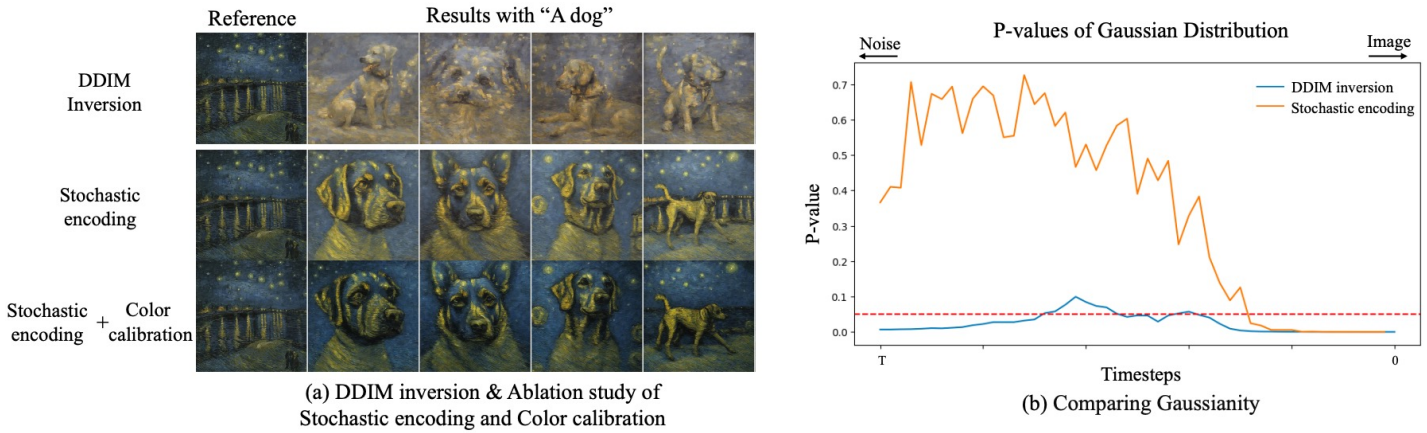
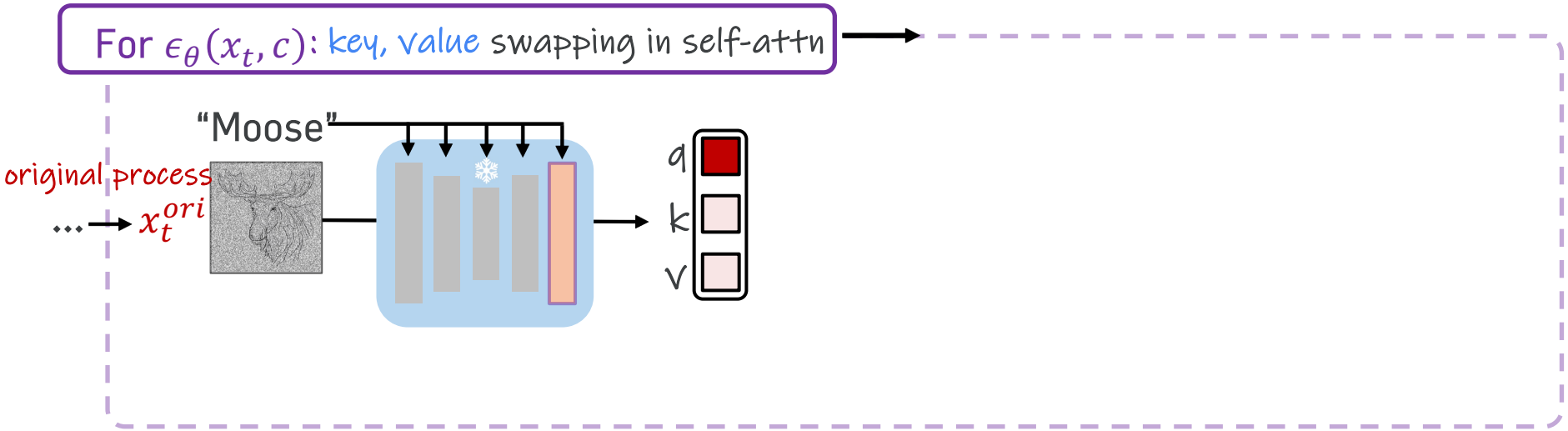



Figure 12: **(a) Comparison of DDIM inversion vs. stochastic encoding and the effect of color calibration.** Stochastic encoding reduces artifacts in the resulting images, while color calibration better reflects the colors of the reference image. **(b) Stochastic encoding produces the latents closer to the standard Gaussian distribution compared to DDIM inversion.** A P-value above 0.05 suggests that the data likely follows the standard Gaussian distribution.

# Method | Guide in self-attention

NAVER AI LAB

* q, k, v = query, key, value

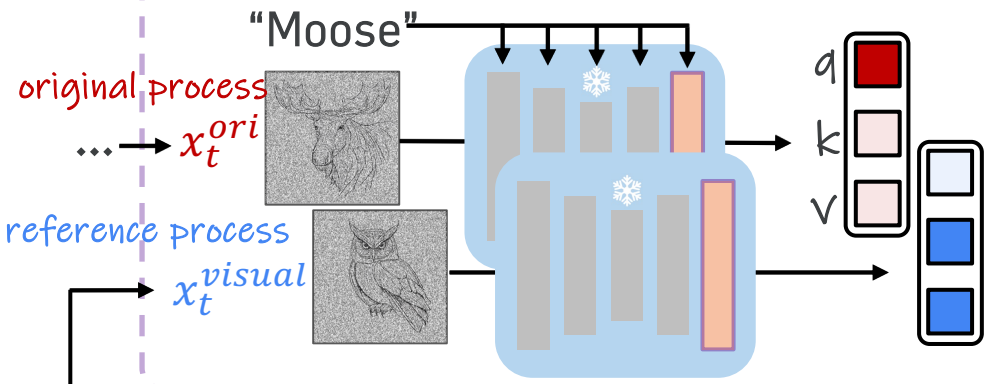


# Method | Guide in self-attention

NAVER AI LAB

* q, k, v = query, key, value

For $\epsilon_\theta(x_t, c)$: key, value swapping in self-attn




Sec 3
stochastic encoding

$x_0^{ref}$

For mapping the latent

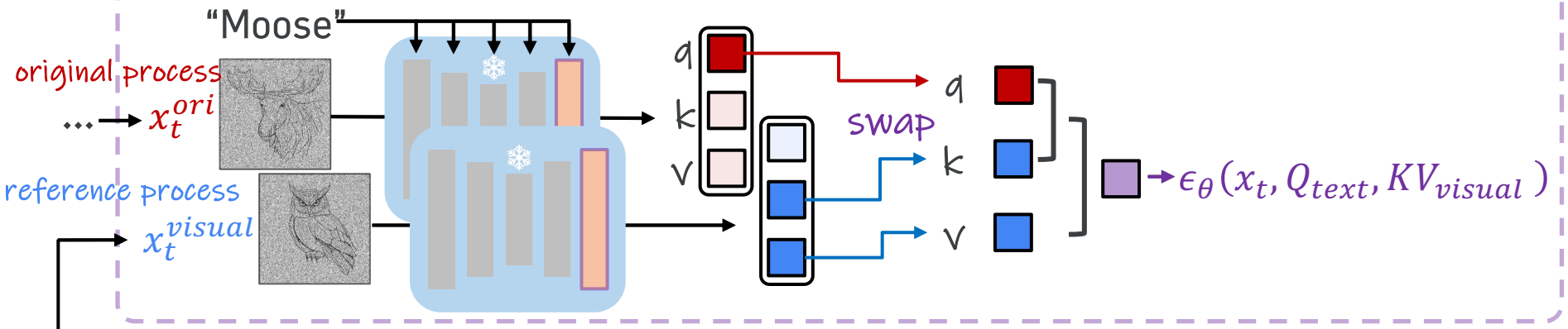
# Method | Guide in self-attention

NAVER AI LAB

For reflecting the reference image's style

* q, k, v = query, key, value

For $\epsilon_\theta(x_t, c)$: key, value swapping in self-attn → Sec 2 Swapping self-attention




Sec 3 stochastic encoding

$x_0^{ref}$

For mapping the latent

**Motivation:** $\tilde{\epsilon}_\theta(x_t, c) = \epsilon_\theta(x_t, c) + \omega * (\epsilon_\theta(x_t, c) - \epsilon_\theta(x_t, \phi))$

# Method | Guide in self-attention

NAVER AI LAB

* q, k, v = query, key, value

For reflecting the reference image's style

For $\epsilon_\theta(x_t, c)$: key, value swapping in self-attn

Sec 2
Swapping self-attention

"Moose"

original process

$\cdots \rightarrow x_t^{ori}$

reference process

$x_t^{visual}$

q k v

swap

$\epsilon_\theta(x_t, Q_{text}, KV_{visual})$

For $\epsilon_\theta(x_t)$: query swapping in self-attn

Sec 2
Negative visual query guidance

$\cdots \rightarrow x_t^{ori}$

$x_t^{visual}$

swap

For reflecting
the text's content

$\epsilon_\theta(x_t, Q_{visual}, KV_\phi)$

Sec 3
stochastic encoding

$x_0^{ref}$

For mapping the latent

**Motivation:** $\tilde{\epsilon}_\theta(x_t, c) = \epsilon_\theta(x_t, c) + \omega * (\epsilon_\theta(x_t, c) - \epsilon_\theta(x_t, \phi))$

# Method | Guide in self-attention

NAVER AI LAB

* q, k, v = query, key, value

For reflecting the reference image's style

For $\epsilon_\theta(x_t, c)$: key, value swapping in self-attn

Sec 2
Swapping self-attention

"Moose"

original process

$\cdots \rightarrow x_t^{ori}$

reference process

$x_t^{visual}$

q
k
v

swap

$\epsilon_\theta(x_t, Q_{text}, KV_{visual})$

For enhancing the color

Sec 3
color calibration

$* (w + 1)$

$\tilde{x}_t^{ori} \rightarrow \cdots$ denoising

"Moose"

Result

For $\epsilon_\theta(x_t)$: query swapping in self-attn

Sec 2
Negative visual query guidance

$\cdots \rightarrow x_t^{ori}$

$x_t^{visual}$

swap

For reflecting
the text's content

$\epsilon_\theta(x_t, Q_{visual}, KV_\phi)$

$* (-w)$

Sec 3
stochastic encoding

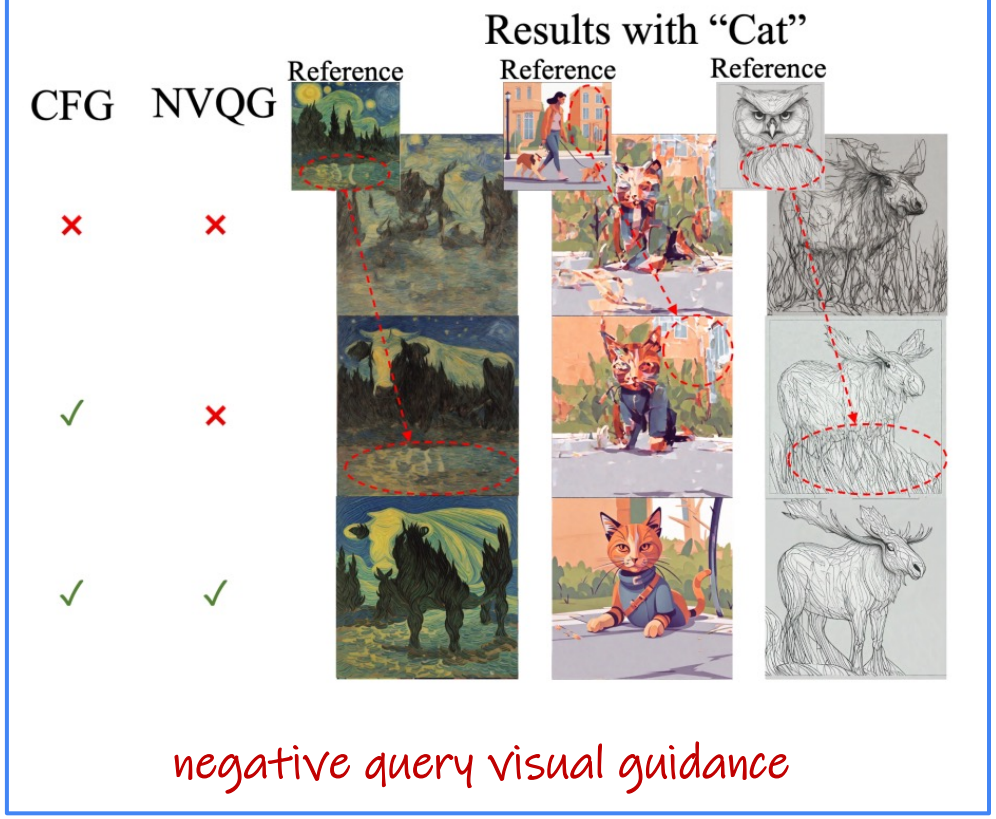
$x_0^{visual}$

For mapping the latent

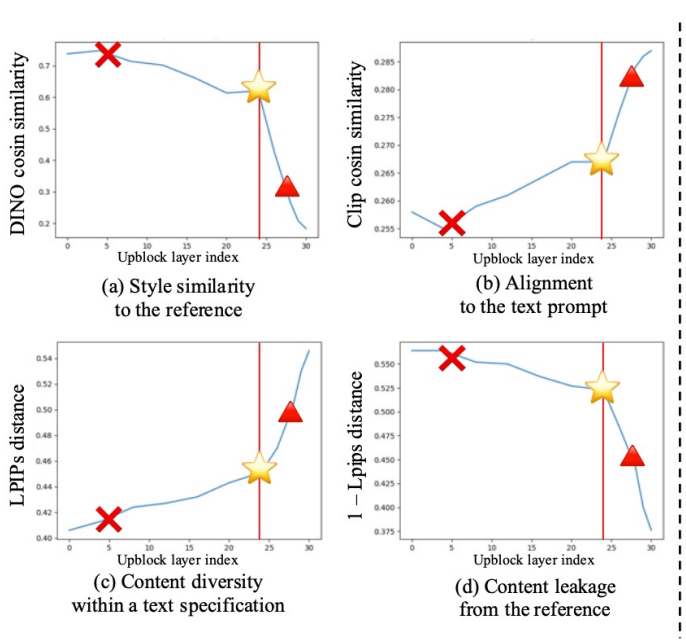
**Motivation:** $\tilde{\epsilon}_\theta(x_t, c) = \epsilon_\theta(x_t, c) + \omega * (\epsilon_\theta(x_t, c) - \epsilon_\theta(x_t, \phi))$

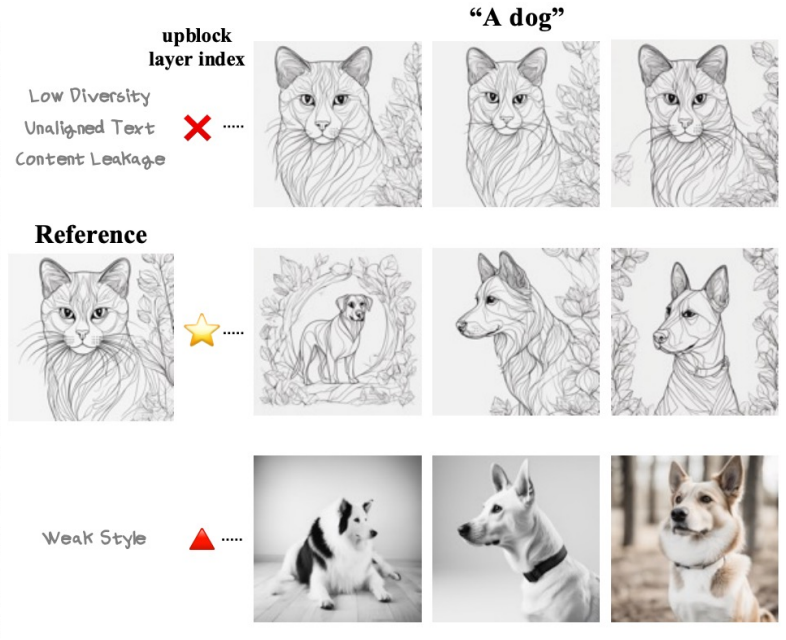
# Method | Guide in self-attention

NAVER AI LAB




negative query visual guidance

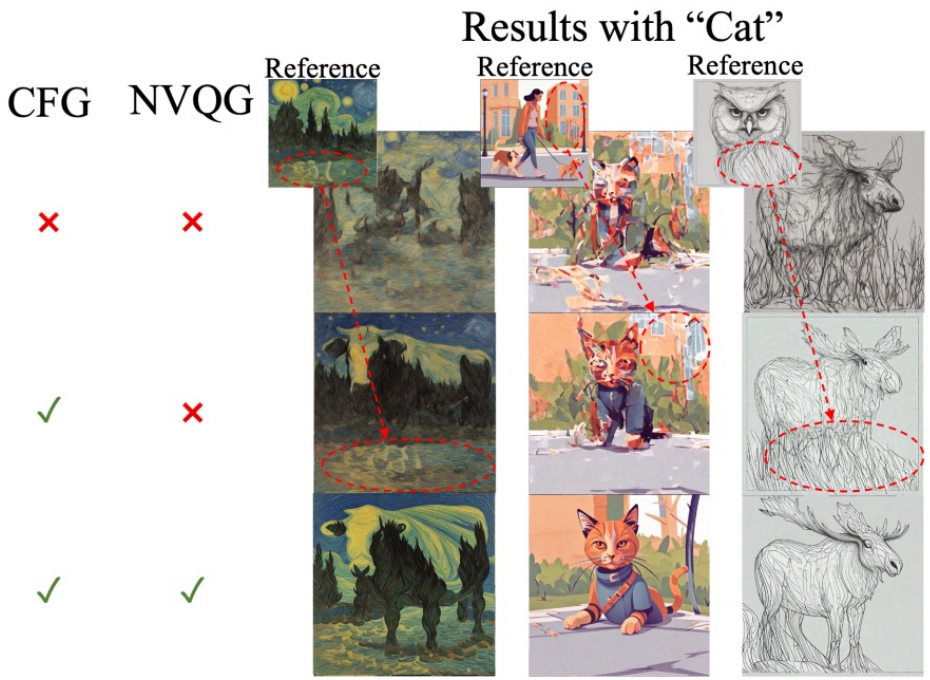


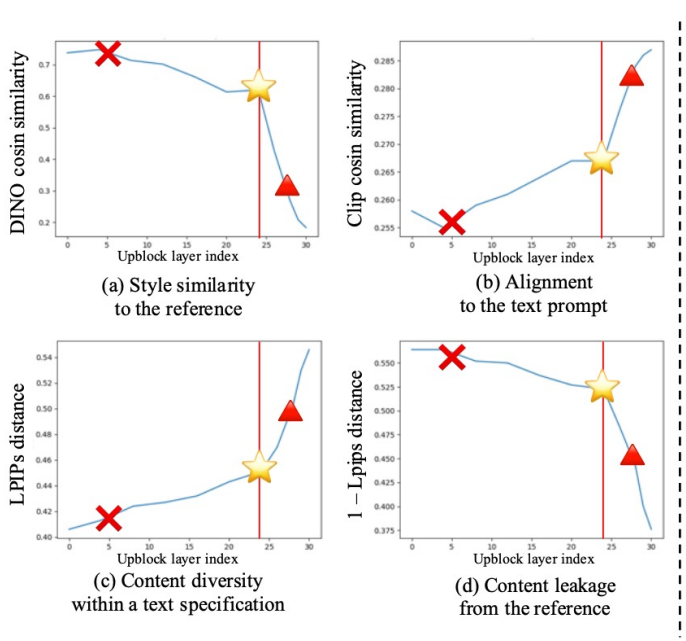

layer analysis

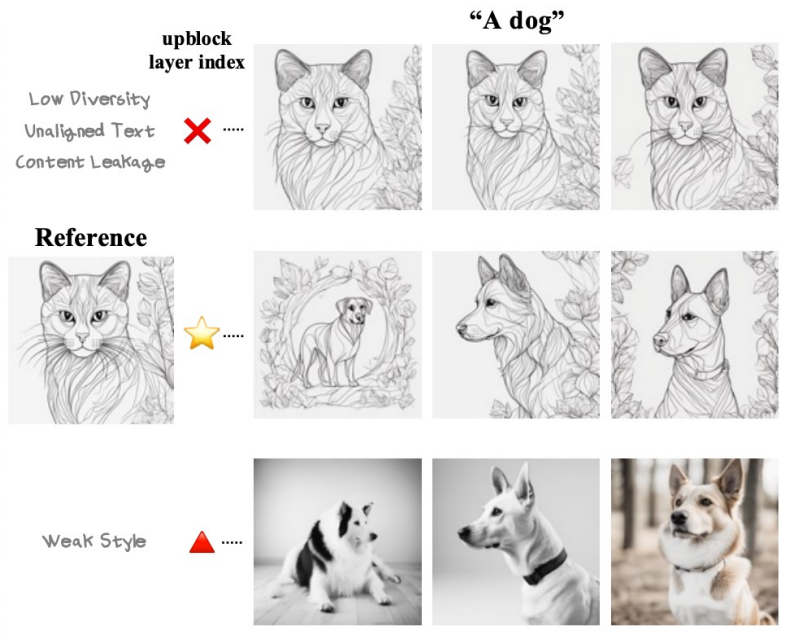
# Method | Guide in self-attention

NAVER AI LAB




negative query visual guidance




layer analysis

NAVER AI LAB

**Function** `color_calibration` $(x_t, \hat{x}_t, x_0^{visual})$ **:**

$x_{pred} \leftarrow \frac{x_t - \sqrt{1-\alpha_t} \cdot \epsilon_\theta(\hat{x}_t)}{\sqrt{\alpha_t}}$ ;

$x_{dir} \leftarrow \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \epsilon_\theta(\hat{x}_t)$ ;

$\epsilon \sim \mathcal{N}(0, I)$ ;

$x_{noise} \leftarrow \sigma_t \cdot \epsilon$ ;

$\hat{x}_{pred} \leftarrow \text{adain}(x_{pred}, x_0^{visual})$ ;

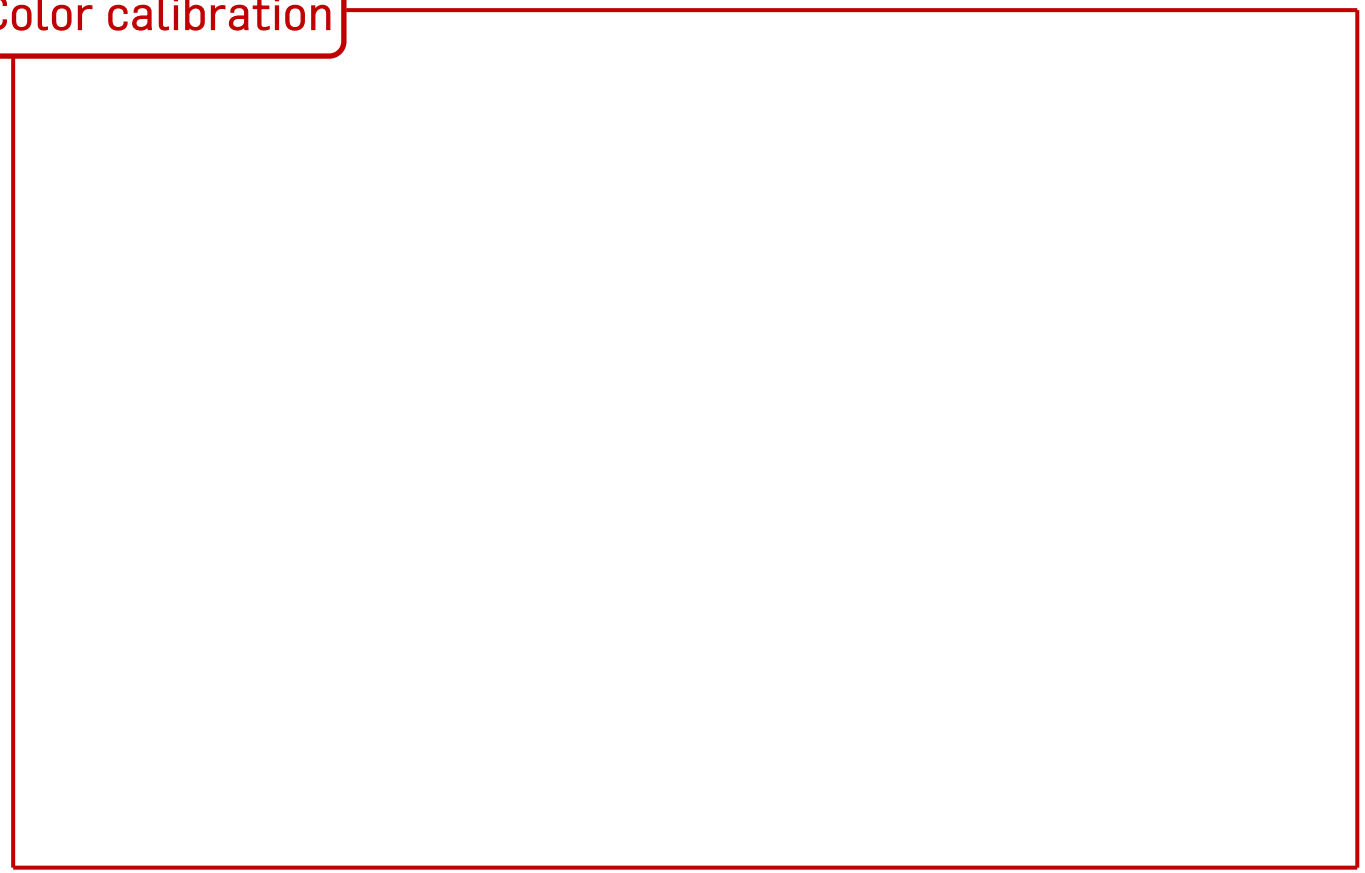
$x_{t-1} \leftarrow \sqrt{\alpha_{t-1}} \cdot \hat{x}_{pred} + x_{dir} + x_{noise}$ ;

**return** $x_{t-1}$

# Method | Color calibration

NAVER AI LAB

Color calibration



For enhancing the color

**Function** `color_calibration` $(x_t, \hat{x}_t, x_0^{visual})$ **:**

$x_{pred} \leftarrow \frac{x_t - \sqrt{1-\alpha_t} \cdot \epsilon_\theta(\hat{x}_t)}{\sqrt{\alpha_t}}$ ;

$x_{dir} \leftarrow \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \epsilon_\theta(\hat{x}_t)$ ;

$\epsilon \sim \mathcal{N}(0, I)$ ;

$x_{noise} \leftarrow \sigma_t \cdot \epsilon$ ;

$\hat{x}_{pred} \leftarrow \text{adain}(x_{pred}, x_0^{visual})$ ;

$x_{t-1} \leftarrow \sqrt{\alpha_{t-1}} \cdot \hat{x}_{pred} + x_{dir} + x_{noise}$ ;

**return** $x_{t-1}$

# Method | Color calibration

NAVER AI LAB

Color calibration

$x_0^{visual}$ 

For enhancing the color

**Function** `color_calibration` $(x_t, \hat{x}_t, x_0^{visual})$ **:**

$x_{pred} \leftarrow \frac{x_t - \sqrt{1-\alpha_t} \cdot \epsilon_\theta(\hat{x}_t)}{\sqrt{\alpha_t}}$ ;

$x_{dir} \leftarrow \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \epsilon_\theta(\hat{x}_t)$ ;

$\epsilon \sim \mathcal{N}(0, I)$ ;

$x_{noise} \leftarrow \sigma_t \cdot \epsilon$ ;

$\hat{x}_{pred} \leftarrow \text{adain}(x_{pred}, x_0^{visual})$ ;

$x_{t-1} \leftarrow \sqrt{\alpha_{t-1}} \cdot \hat{x}_{pred} + x_{dir} + x_{noise}$ ;

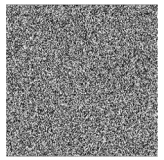
**return** $x_{t-1}$

# Method | Color calibration

NAVER AI LAB

Color calibration

$x_0^{visual}$ 

$x_t^{ori}$ 

For enhancing the color

**Function** `color_calibration` $(x_t, \hat{x}_t, x_0^{visual})$ **:**

$x_{pred} \leftarrow \frac{x_t - \sqrt{1-\alpha_t} \cdot \epsilon_\theta(\hat{x}_t)}{\sqrt{\alpha_t}}$ ;

$x_{dir} \leftarrow \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \epsilon_\theta(\hat{x}_t)$ ;

$\epsilon \sim \mathcal{N}(0, I)$ ;

$x_{noise} \leftarrow \sigma_t \cdot \epsilon$ ;

$\hat{x}_{pred} \leftarrow \text{adain}(x_{pred}, x_0^{visual})$ ;

$x_{t-1} \leftarrow \sqrt{\alpha_{t-1}} \cdot \hat{x}_{pred} + x_{dir} + x_{noise}$ ;

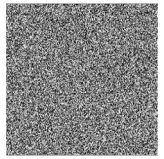
**return** $x_{t-1}$

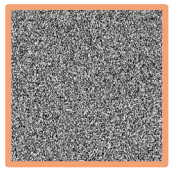
# Method | Color calibration

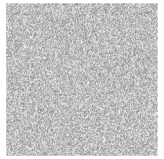
NAVER AI LAB

Color calibration

$x_0^{visual}$ 

$x_t^{ori}$ 

$\tilde{x}_t^{ori}$ 

$\varepsilon$ 

For enhancing the color

**Function** `color_calibration` $(x_t, \hat{x}_t, x_0^{visual})$ **:**

$x_{pred} \leftarrow \frac{x_t - \sqrt{1-\alpha_t} \cdot \epsilon_\theta(\hat{x}_t)}{\sqrt{\alpha_t}}$ ;

$x_{dir} \leftarrow \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \epsilon_\theta(\hat{x}_t)$ ;

$\epsilon \sim \mathcal{N}(0, I)$ ;

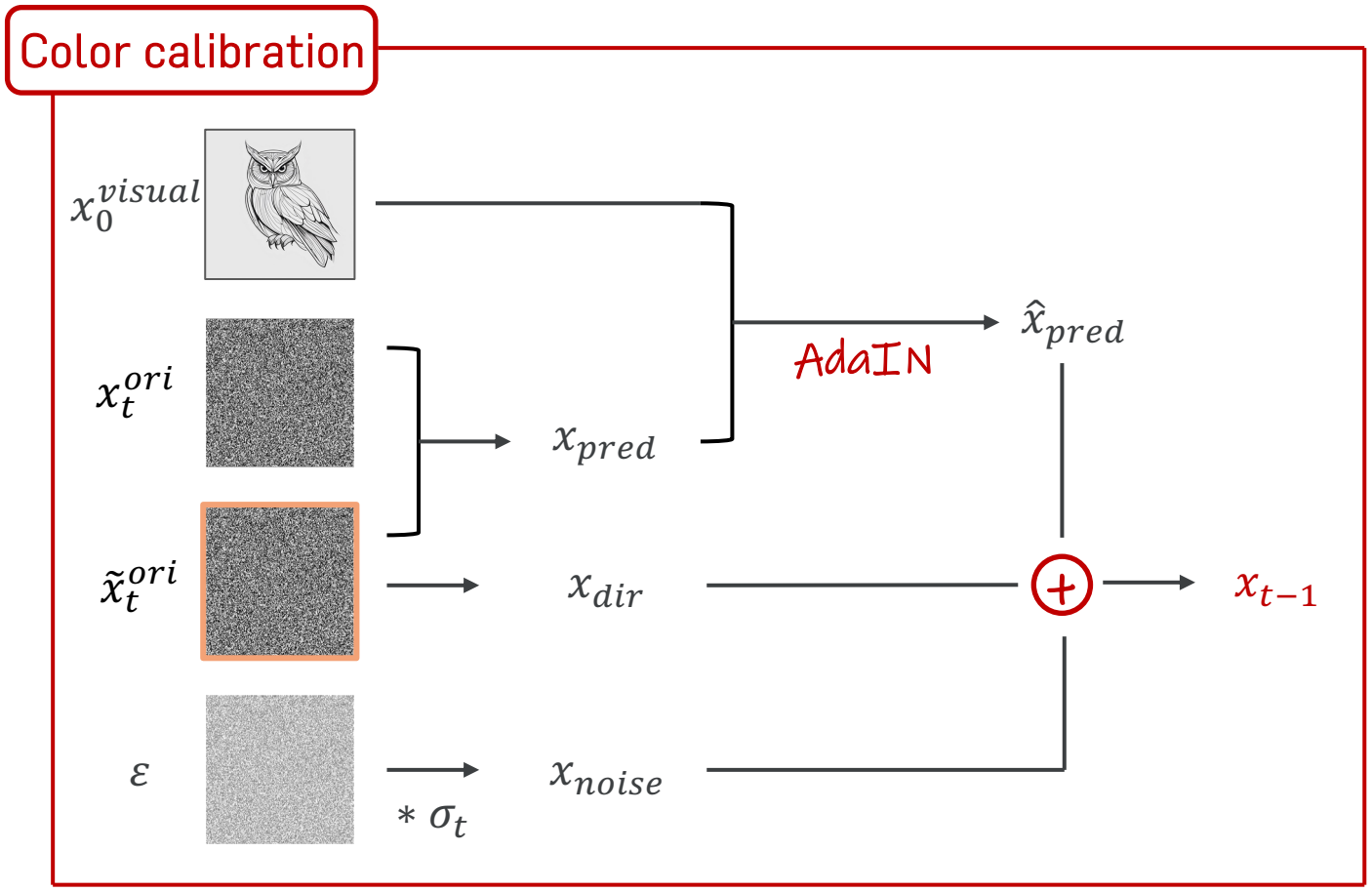
$x_{noise} \leftarrow \sigma_t \cdot \epsilon$ ;

$\hat{x}_{pred} \leftarrow \text{adain}(x_{pred}, x_0^{visual})$ ;

$x_{t-1} \leftarrow \sqrt{\alpha_{t-1}} \cdot \hat{x}_{pred} + x_{dir} + x_{noise}$ ;

**return** $x_{t-1}$

NAVER AI LAB




For enhancing the color

**Function** `color_calibration` $(x_t, \hat{x}_t, x_0^{visual})$ **:**

$x_{pred} \leftarrow \frac{x_t - \sqrt{1-\alpha_t} \cdot \epsilon_\theta(\hat{x}_t)}{\sqrt{\alpha_t}}$ ;

$x_{dir} \leftarrow \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \epsilon_\theta(\hat{x}_t)$ ;

$\epsilon \sim \mathcal{N}(0, I)$ ;

$x_{noise} \leftarrow \sigma_t \cdot \epsilon$ ;

$\hat{x}_{pred} \leftarrow \text{adain}(x_{pred}, x_0^{visual})$ ;

$x_{t-1} \leftarrow \sqrt{\alpha_{t-1}} \cdot \hat{x}_{pred} + x_{dir} + x_{noise}$ ;

**return** $x_{t-1}$

# Results | Comparison (text-to-image)

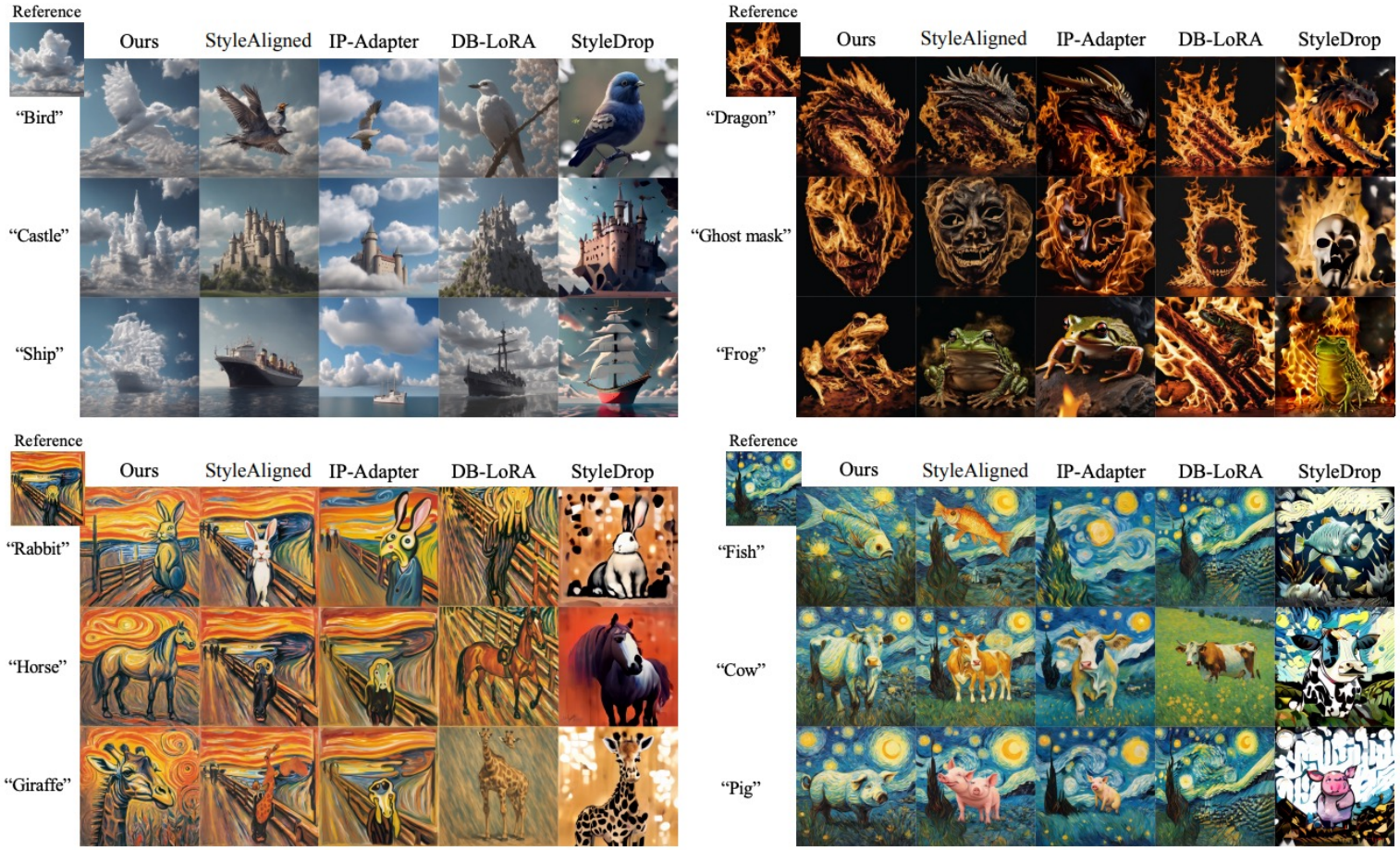
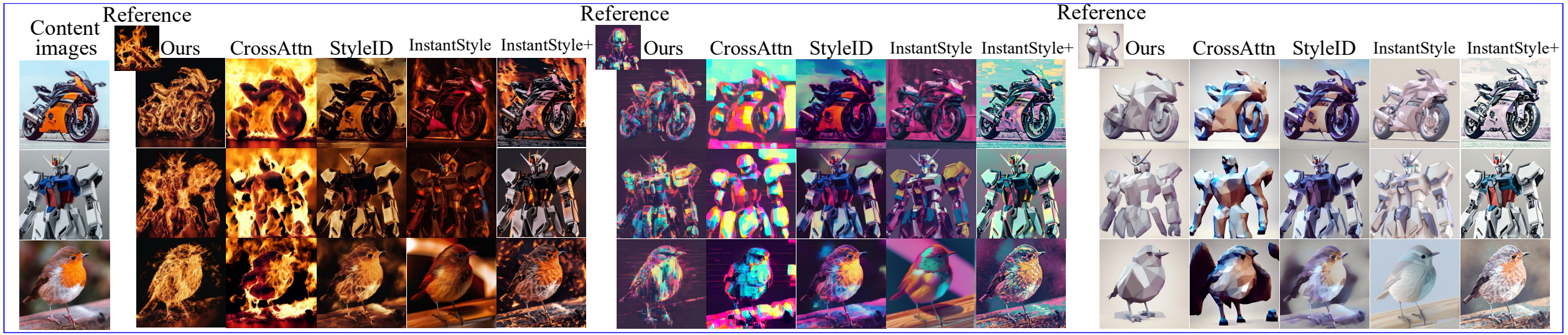


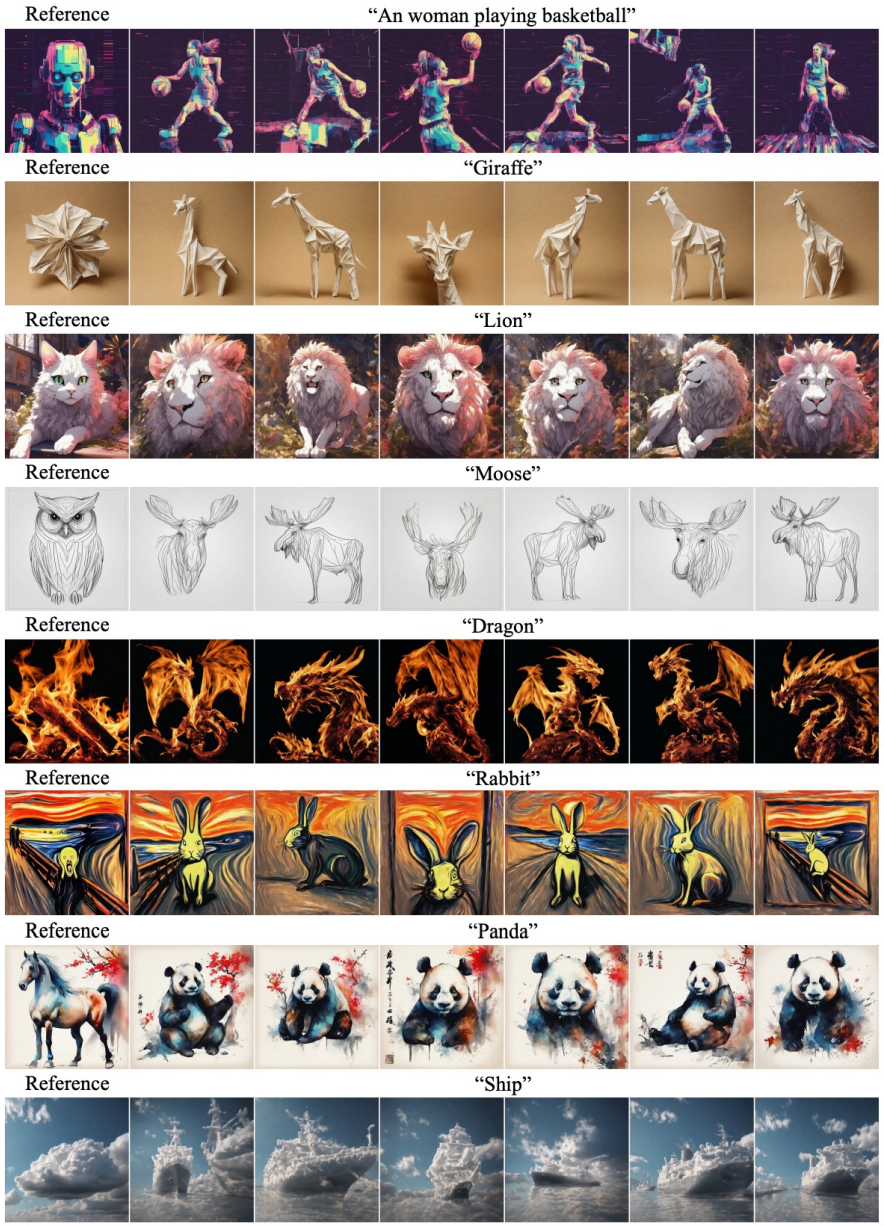

Figure 7: **Qualitative comparison across various styles and text prompts.** StyleGuide faithfully reflects style elements in reference images without causing content leakage from the reference images.

# Results | Comparison (image-to-image)

NAVER AI LAB



# Results | simple prompt & complex prompt





Junho Kim
(NAVER AI Lab)
Thank you for your attention !