# Free-Lunch Color-Texture Disentanglement for Stylized Image Generation

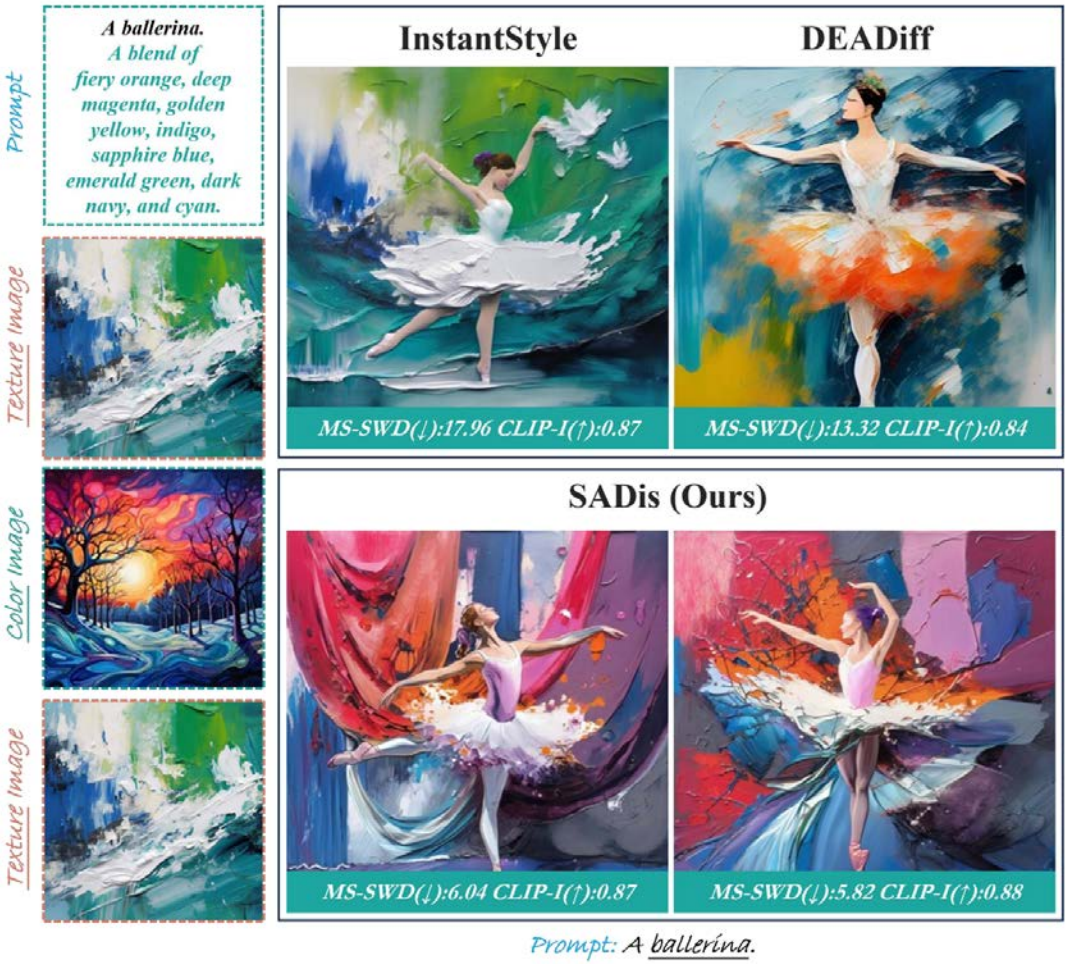
Jiang Qin[1,*], Alexandra Gomez-Villa[3,4,*], Senmao Li[2,*,‡],
Shiqi Yang[2,†], Yaxing Wang[2], Kai Wang[5,6,3,‡], Joost van de Weijer[3,4]

[1]Harbin Institute of Technology, China; [2]VCIP, CS, Nankai University, China; [3]Computer Vision Center, Spain;
[4]Universitat Autònoma de Barcelona, Spain; [5]Program of Computer Science, City University of Hong Kong (Dongguan), China;
[6]City University of Hong Kong, HK SAR, China

https://deepffff.github.io/sadis.github.io

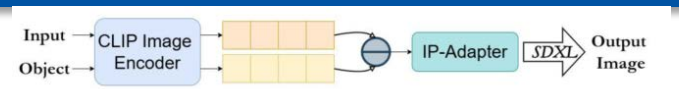
[*] Equal contribution
[‡] The corresponding author

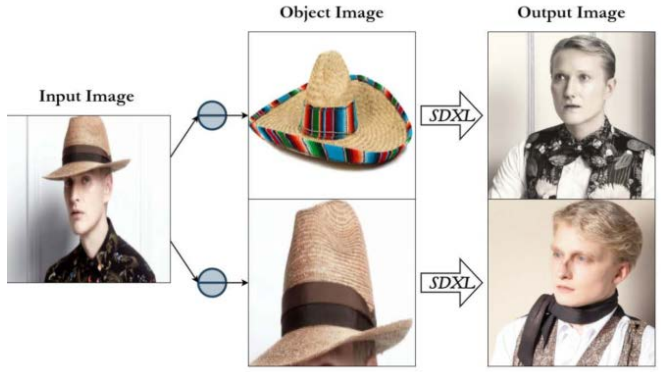
# Problem of Disentangled Stylized Image Generation



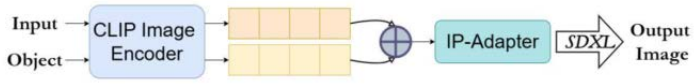

- **Lack of Fine-grained Style Control:** Existing methods struggle to accurately **control multiple stylization attributes**, such as color and texture, simultaneously.
- **Inadequate Color Palette Extraction:** Current methods fail to accurately extract and **manipulate color palettes** for precise control.
- **Image Prompts for Fine-Grained Control:** Image prompts provide more intuitive and detailed control over styles, unlike text prompts, which lack precision in **fine-grained attributes**.
- ***SADis achieves precise control over color and texture by leveraging style element images, enabling accurate specification of visual attributes.***

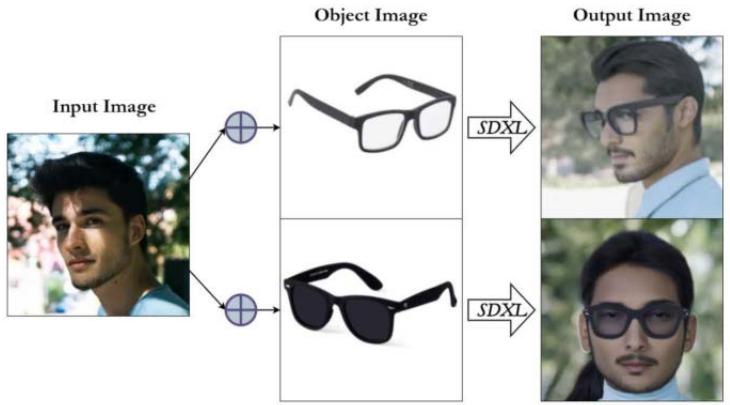
# Image-Prompt Additivity




**subtraction**




**addition**

**Image-prompt additivity**

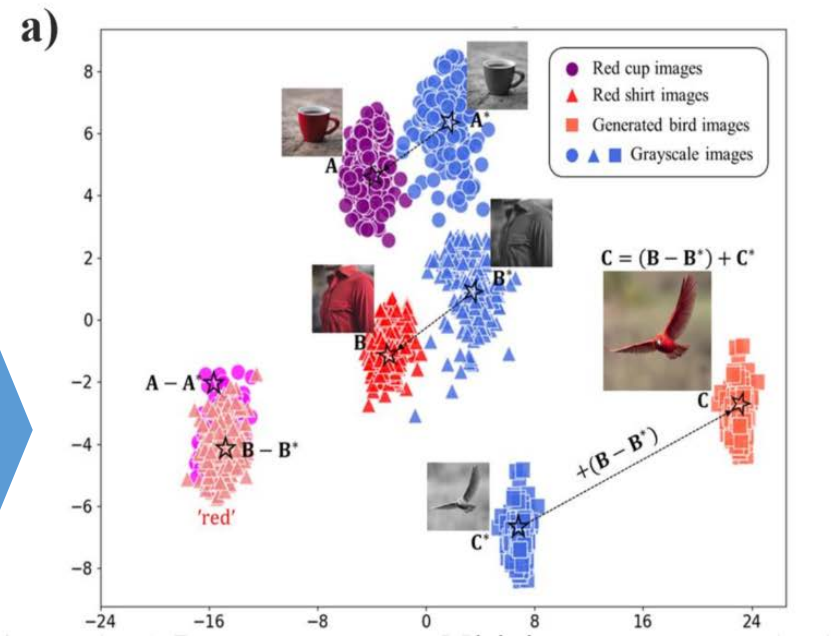
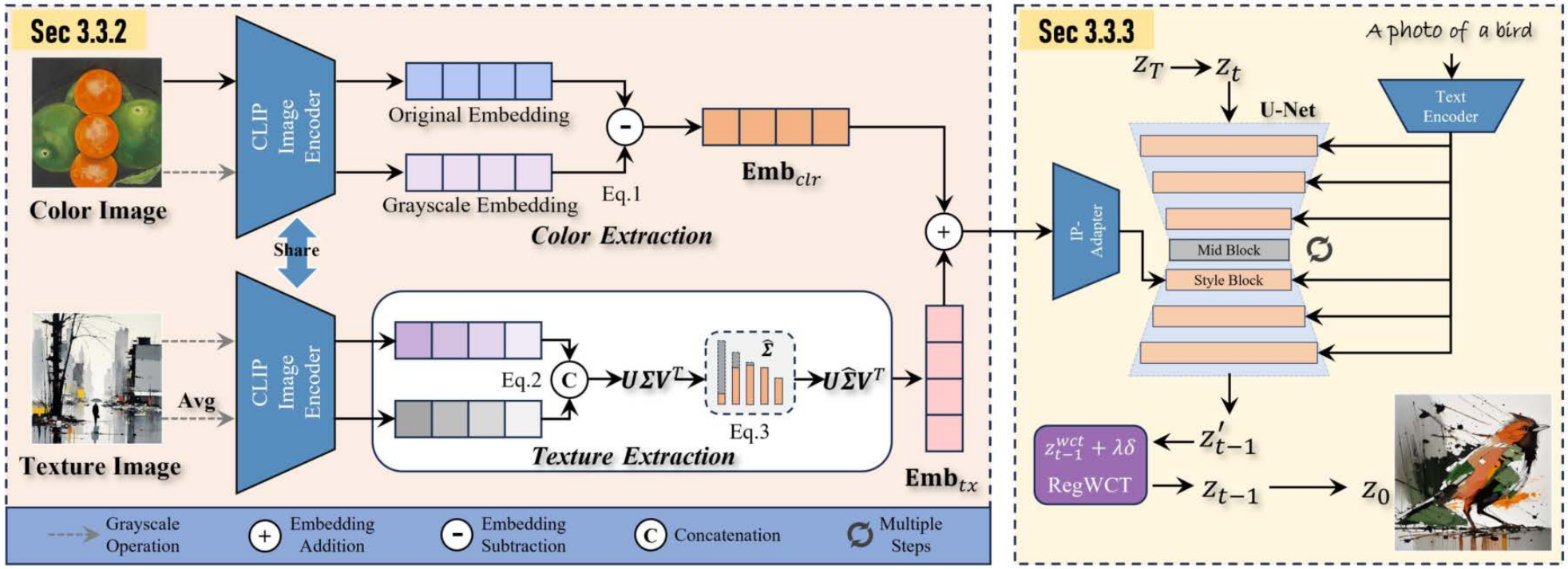



**Image-prompt additivity** enables color extraction via grayscale subtraction

# Method: *SADis for Disentangled Stylized Image Generation*



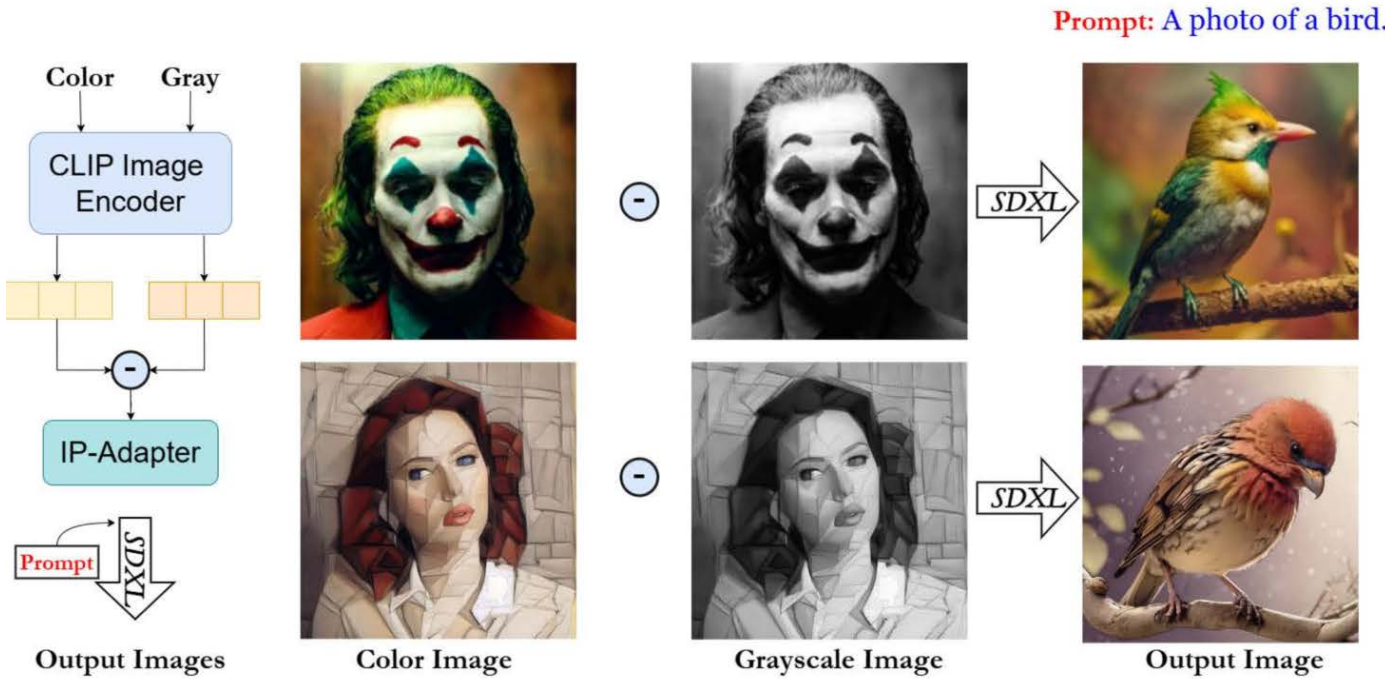

- **SADis**, begins with color-texture extraction, leveraging the ImagePrompt Additivity property;
- The color embedding $\mathbf{Emb}_{clr}$ is obtained by exploiting the Image-Prompt Additivity property;
- The texture embedding $\mathbf{Emb}_{tx}$ is extracted via a SVD operation;
- Color and texture embeddings are fed into the style cross-attention layer of the SDXL model;
- The latent $z_{t-1}$ is refined at each timestep with RegWCT, aligning color palettes precisely while retaining essential texture details

# Color extraction

For color extraction, we take the color image embedding and subtract the grayscale image embedding:

$$\mathbf{Emb}_{clr} = \tau_\phi(\mathcal{I}_{clr}) \ominus \tau_\phi(\mathbf{GS}(\mathcal{I}_{clr}))$$

**GS** is the grayscale operation. This strips away semantic information to retain only the color attributes.



# Texture extraction using SVD

Extracting texture embeddings using CLIP encoder:

$$\mathbf{Emb}_{tx}^{*} = \tau_{\phi}(\mathbf{GS}(\mathcal{I}_{tx}))$$

Concatenating pure gray image vectors via token dimension and applying SVD transformation:

$$\mathbf{Emb}_{tx}^{'} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{T} \qquad \mathbf{Emb}_{tx}^{'} = \mathbf{Emb}_{tx}^{*} \copyright \tau_{\phi}(\mathbf{Avg}(\mathbf{GS}(\mathcal{I}_{tx})))$$

$$\boldsymbol{\Sigma} = diag(\sigma_0, \sigma_1, \cdots, \sigma_{n_t}, \cdots, \sigma_{2n_t-1})$$

Suppressing principal eigenvalues for grayscale suppression and inverse transformation to extract texture:

$$\hat{\sigma} = \beta e^{-\gamma\sigma} * \sigma. \qquad \hat{\boldsymbol{\Sigma}} = diag(\hat{\sigma_0}, \hat{\sigma_1}, \cdots, \hat{\sigma}_{2n_t-1})$$

$$\mathbf{Emb}_{re} = \boldsymbol{U}\hat{\boldsymbol{\Sigma}}\boldsymbol{V}^{T}$$

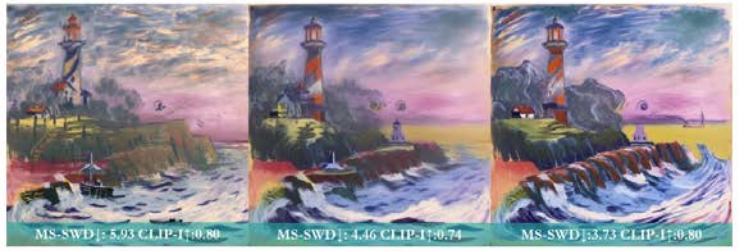
$$\mathbf{Emb}_{tx} = \mathbf{Emb}_{re}[:n_t, :]$$

# RegWCT: Regularized Whitening-Coloring Transforms

The WCT method is applied in the latent space to enhance color consistency.

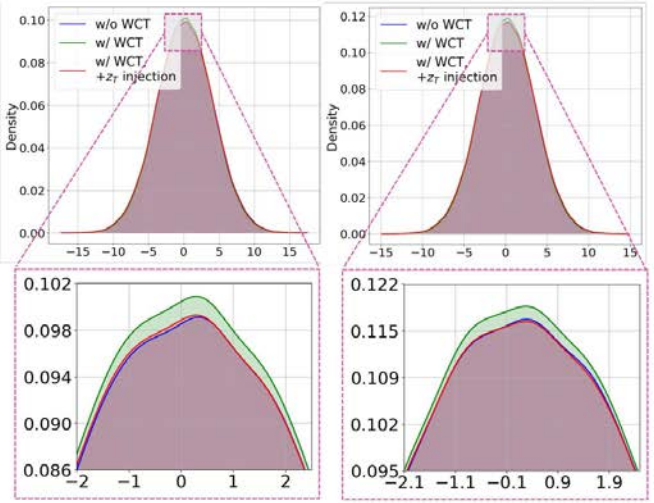
$$z_t^{wct} = \mathbf{WCT}(z_t^{'})$$

**Problems with Directly Using WCT**




SADis w/o WCT | SADis w/ WCT | SADis w/ WCT + $z_T$ injection

*(b) WCT compress texture details in noised latent*




*(c) ReWCT rectify the distribution*

- **WCT improves color fidelity but causes the loss of high-frequency information, leading to a decline in texture fidelity.**
- **Analysis reveals that the latent representation after WCT lacks a certain amount of noise, which contributes to the loss of texture details.**

Analyzing the WCT latent distribution reveals a bias that can be corrected by adding noise.

$$z_t = z_t^{wct} + \lambda \cdot \delta, \delta \sim \mathcal{N}(0,1)$$

This alleviates texture loss while ensuring color consistency.

# Comparison Experiments: color-texture disentangled stylized image generation


Table 1: Quantitative Comparison with existing image stylization methods. The best and second-best numbers are marked with **bold** and underlined respectively.

| Method | CLIP | Color | | | Texture | | Time Cost (s) | User study (%) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | MS-SWD↓ | C-Hist↓ | GPT4o↑ | CLIP-I↑ | KID↓ | | Color↑ | Texture↑ | Both↑ |
| SDXL [53] | 0.272 | 9.51 | 1.20 | 3.01 | 0.69 | 0.08 | 9.26 | 16.99 | 5.84 | 11.65 |
| IPAdapter [76] | 0.233 | 11.54 | 1.23 | 2.84 | **0.84** | **0.043** | 9.52 | 6.08 | 22.54 | 14.55 |
| InstantStyle [67] | 0.261 | 12.53 | 1.32 | 2.80 | 0.74 | 0.056 | 9.31 | 4.81 | 24.94 | 13.10 |
| Artist [39] | 0.269 | 10.48 | 1.39 | 2.82 | 0.69 | 0.089 | 12.32 | 9.38 | 1.79 | 3.40 |
| DEADiff [54] | 0.267 | 11.20 | 1.24 | 2.73 | 0.69 | 0.087 | **1.86** | 4.57 | 2.59 | 2.67 |
| StyleDrop [62] | 0.275 | 13.52 | 1.43 | 2.67 | 0.70 | 0.054 | 6.91 | 5.07 | 3.57 | 5.34 |
| DreamStyler [3] | 0.277 | 12.17 | 1.26 | 2.39 | 0.71 | 0.060 | 5.23 | 4.67 | 3.57 | 5.39 |
| CSGO [74] | 0.280 | 14.25 | 1.36 | 2.63 | 0.69 | 0.071 | 15.99 | 6.59 | 9.73 | 6.06 |
| *SADis* (Ours) | **0.281** | **5.57** | **0.96** | **3.34** | 0.74 | 0.049 | 10.30 | **41.83** | **25.42** | **37.84** |

- SADis demonstrates superior alignment with textual prompts compared to other methods.
- It significantly outperforms others in color alignment.
- While IP-Adapter achieves high texture scores, it suffers from semantic content leakage (e.g., identical stars in the Van Gogh bear).
- User studies show SADis exhibits a clear superiority, with results highly aligned with human preferences.

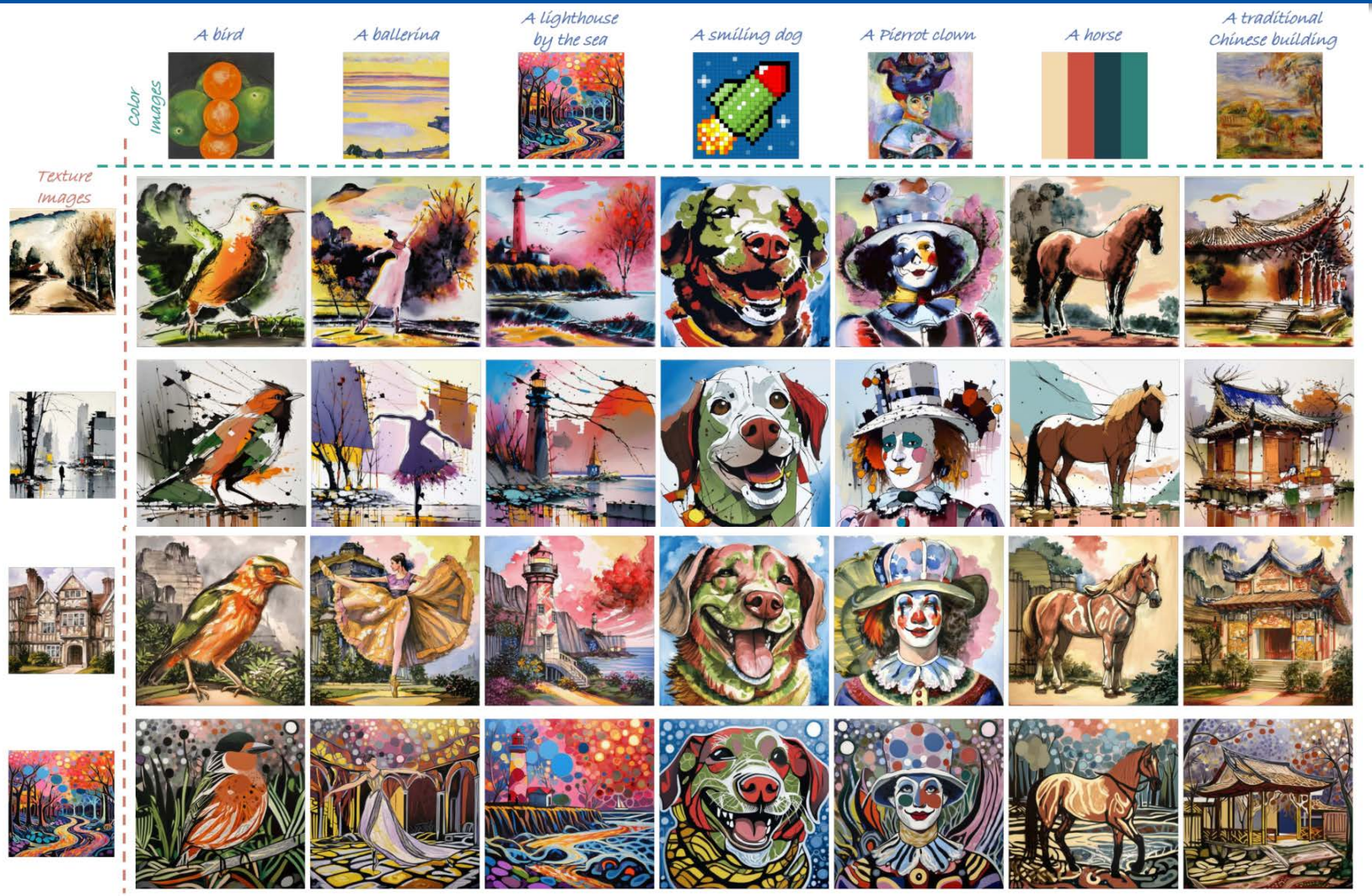
# Comparison Experiments: when both color and texture are from the same image

Table 4: Quantitative comparison where both color and texture are derived from the same image. The best and second-best numbers are marked with **bold** and underlined, respectively.

| Method | CLIP | Color | | | Texture | |
|---|---|---|---|---|---|---|
| | | MS-SWD↓ | C-Hist↓ | GPT4o↑ | CLIP-I↑ | KID↓ |
| SDXL [53] | 0.292 | 9.12 | 1.15 | 3.04 | 0.696 | 0.090 |
| IPAdapter [76] | 0.260 | 4.18 | 0.63 | 3.44 | **0.815** | **0.057** |
| InstantStyle [67] | 0.277 | 4.21 | 0.64 | 3.41 | 0.753 | 0.066 |
| Artist [39] | 0.272 | 10.98 | 1.18 | 2.83 | 0.744 | 0.077 |
| DEADiff [54] | 0.295 | 10.81 | 1.14 | 2.66 | 0.721 | 0.089 |
| StyleDrop [62] | 0.292 | 12.16 | 1.28 | 2.51 | 0.717 | 0.090 |
| DreamStyler [3] | 0.301 | 11.90 | 1.09 | 2.53 | 0.691 | 0.095 |
| CSGO [74] | 0.298 | 6.43 | 0.83 | 3.09 | 0.716 | 0.089 |
| *SADis* (Ours) | **0.302** | **3.19** | **0.53** | **3.51** | 0.754 | 0.066 |

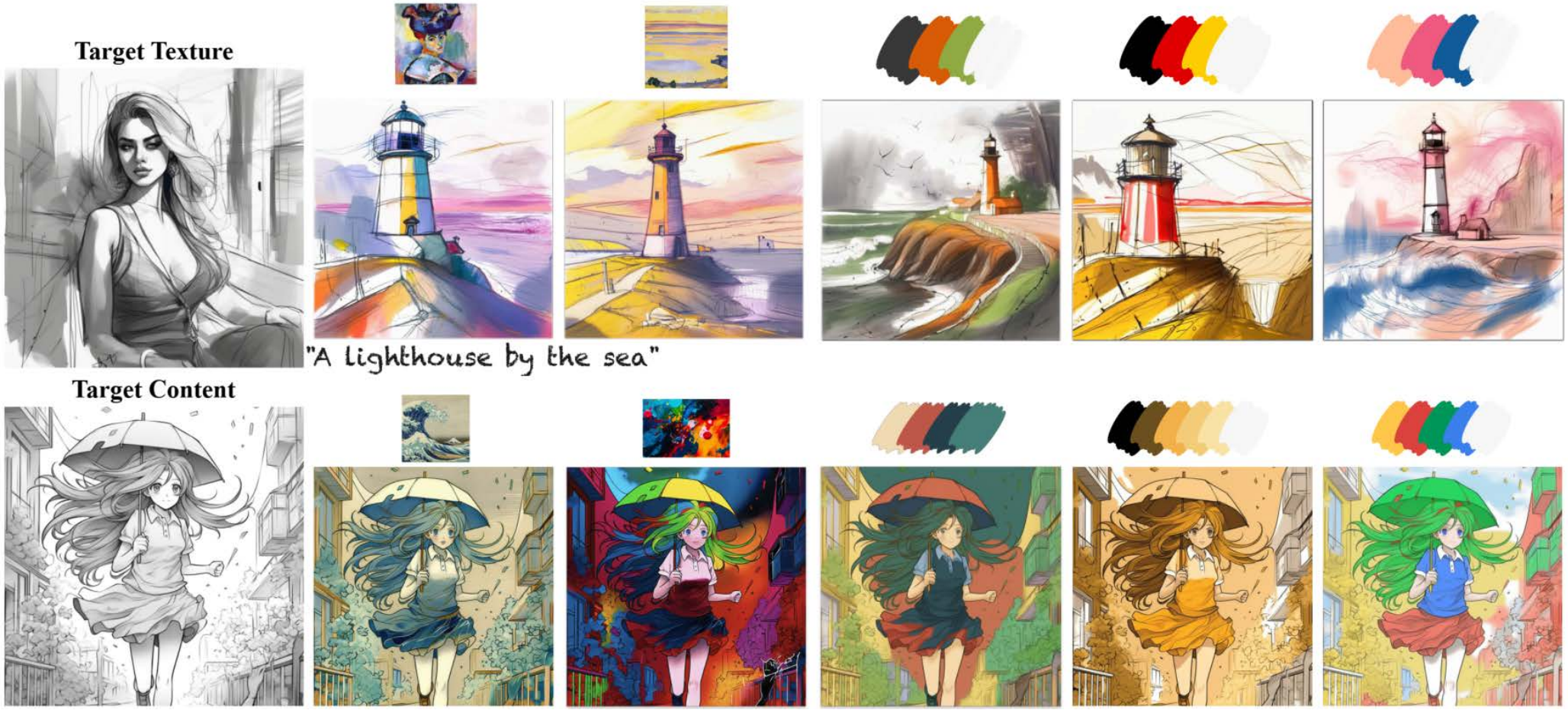

**SADis** performs well using the same image as both the color and texture reference, demonstrating its flexible capability.

# Additional experimental results of SADis



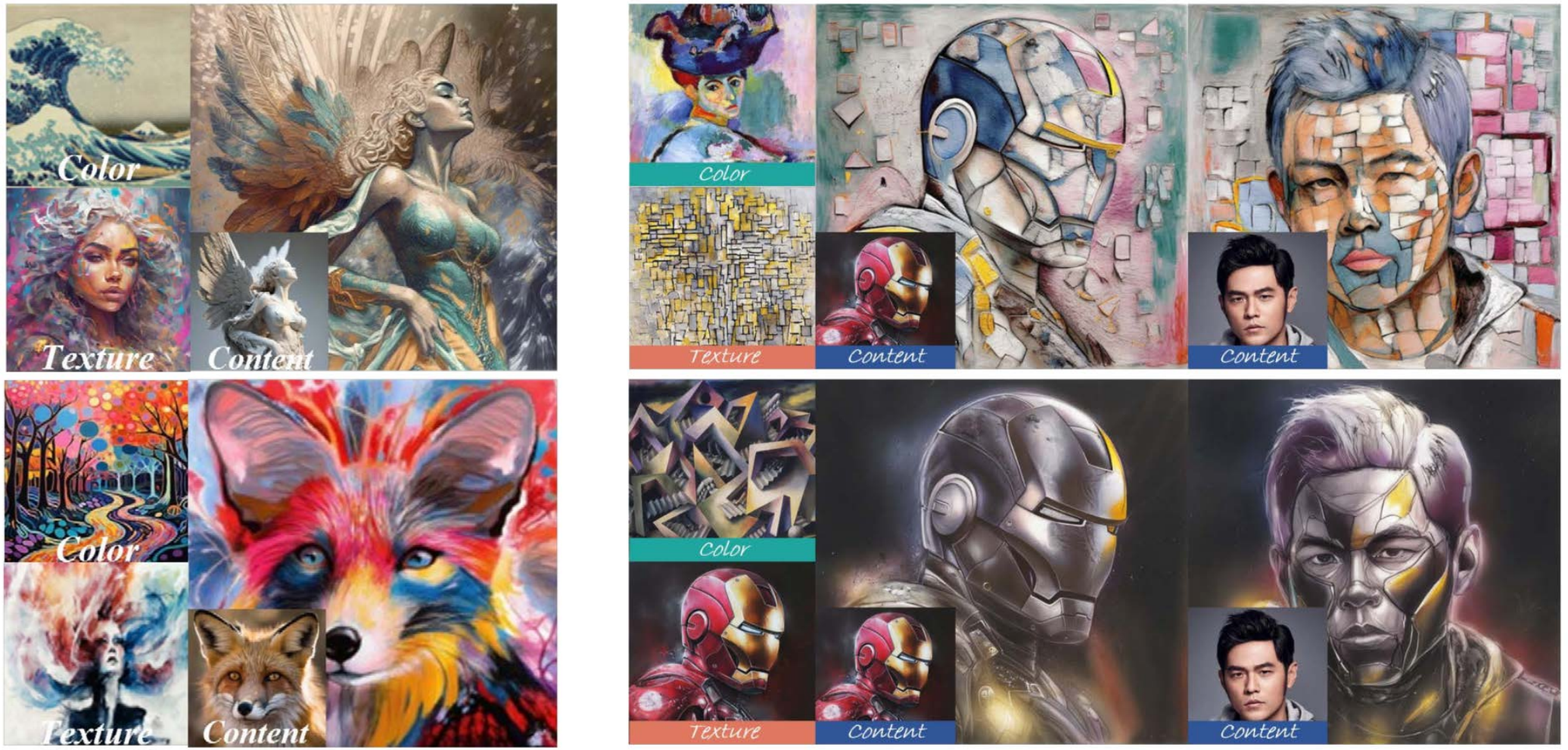

Given a content prompt, a color reference image, and a texture reference image, **SADis** can achieve color-texture controlled stylized image generation.

# Color-texture conditioned or color-content conditioned generation results




**SADis** can not only perform stylized image generation with color-texture control (first row) but also enable content-preserving colorization (second row).

# Experiments: image-based stylization with ControlNet




SADis can be combined with ControlNet to achieve content preservation while controlling color and texture for stylized generation.

# Experiments: color and material transfer



SADis can simultaneously control both color and material, independently control color, and support the editing of both color and material for objects.

# Experiments: compatible with MMDiT-based models


Figure 8: SADis is compatible with MMDiT-based models. (a) demonstrates color-texture disentangled stylization results on FLUX.1-dev, while (b) shows the corresponding results on SD3.5.

# Experiments: ablation study


Table 2: Ablation by removing each components in our method *SADis*.

| Method | Color MS-SWD↓ | Color C-Hist↓ | Texture CLIP-I↑ | Time cost (s) |
|---|---|---|---|---|
| *SADis* (Ours) | **7.27** | **0.96** | 0.74 | ≈ 10.30 |
| — SVD | <u>5.70</u> | <u>1.01</u> | **0.76** | ≈ 10.29 |
| — *RegWCT* | 8.05 | 1.06 | <u>0.75</u> | ≈ 9.42 |
| — SVD — *RegWCT* | 8.93 | 1.10 | **0.76** | ≈ 9.41 |

# Experiments: ablation study on the color scale


Figure 24: Results of different scales of color embedding $\mathbf{Emb}_{clr}$. Here, the scale of $\mathbf{Emb}_{tx}$ is fixed as 1.

# Experiments: ablation study on scale γ and β of SVD

Table 7: Ablation studies of scaling factor $\gamma$. $\gamma$ is set to 0.003 according to the ablation result

| $\gamma$ ($\beta = 1$) | | 0 | 0.001 | 0.003 | 0.005 | 0.007 | 0.009 | 0.011 |
|---|---|---|---|---|---|---|---|---|
| Color | MS-SWD (↓) | 5.70 | 5.71 | 5.57 | 5.36 | 5.22 | 5.21 | 5.20 |
| | C-Hist (↓) | 1.01 | 1.017 | 0.962 | 0.939 | 0.897 | 0.887 | 0.861 |
| Texture | CLIP-I (↑) | 0.759 | 0.755 | 0.743 | 0.736 | 0.730 | 0.723 | 0.713 |

Table 8: Ablation studies of scaling factor $\beta$. $\beta$ is set to 1 according to the ablation results.

| $\beta$ ($\gamma = 0.003$) | | 0.5 | 0.7 | 0.9 | 1 | 1.1 | 1.3 | 1.5 |
|---|---|---|---|---|---|---|---|---|
| Color | MS-SWD (↓) | 5.20 | 5.28 | 5.45 | 5.57 | 5.70 | 5.92 | 6.17 |
| | C-Hist (↓) | 0.890 | 0.923 | 0.949 | 0.962 | 0.973 | 0.992 | 1.011 |
| Texture | CLIP-I (↑) | 0.711 | 0.729 | 0.740 | 0.743 | 0.747 | 0.749 | 0.752 |

# Experiments: ablation study on RegWCT

$$z_t = (1-\omega)\, z_t^{'} + \omega \cdot \mathbf{RegWCT}(z_t^{'}). \qquad \omega \text{ is the RegWCT scale.}$$

Table 6: Ablation studies of *RegWCT* scales. considering the trade-off of texture and color alignment, the *RegWCT* scale is set as 0.5 by default.

| *RegWCT* Scale | | 0 | 0.3 | 0.5 | 0.7 | 1.0 |
|---|---|---|---|---|---|---|
| Color | MS-SWD (↓) | 8.05 | 5.65 | 5.57 | 5.23 | 5.18 |
| | C-Hist (↓) | 1.06 | 0.98 | 0.96 | 0.89 | 0.889 |
| Texture | CLIP-I (↑) | 0.747 | 0.744 | 0.743 | 0.741 | 0.740 |

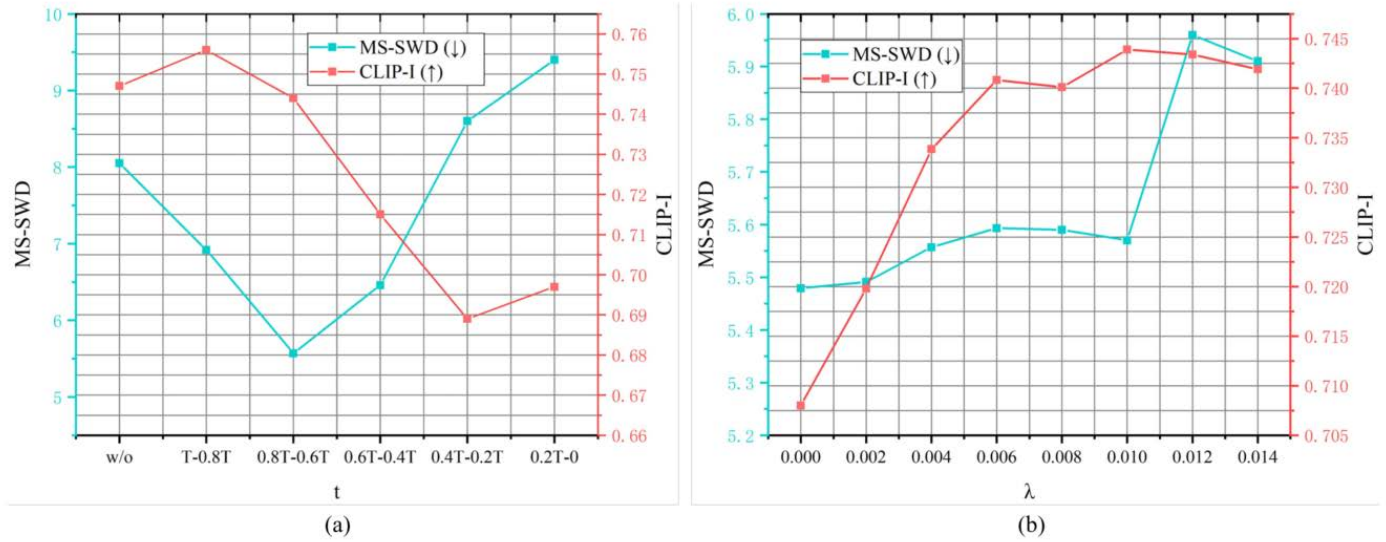



Figure 23: Ablation studies on applying *RegWCT* to different timestep intervals (left). Ablation studies on the scale $\lambda$ of noise injection (right) during applying *RegWCT*. By default, $\lambda$ is set to 0.01.






# Thanks for your attention

**Jiang Qin[1,*], Alexandra Gomez-Villa[3,4,*], Senmao Li[2,*,‡], Shiqi Yang[2,†], Yaxing Wang[2], Kai Wang[5,6,3,‡], Joost van de Weijer[3,4]**

[1]Harbin Institute of Technology, China; [2]VCIP, CS, Nankai University, China; [3]Computer Vision Center, Spain; [4]Universitat Autònoma de Barcelona, Spain; [5]Program of Computer Science, City University of Hong Kong (Dongguan), China; [6]City University of Hong Kong, HK SAR, China

https://deepffff.github.io/sadis.github.io

[*] Equal contribution

[‡] The corresponding author