JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

Neural Texture Synthesis with Guided Correspondence


Yang Zhou


Kaijian Chen

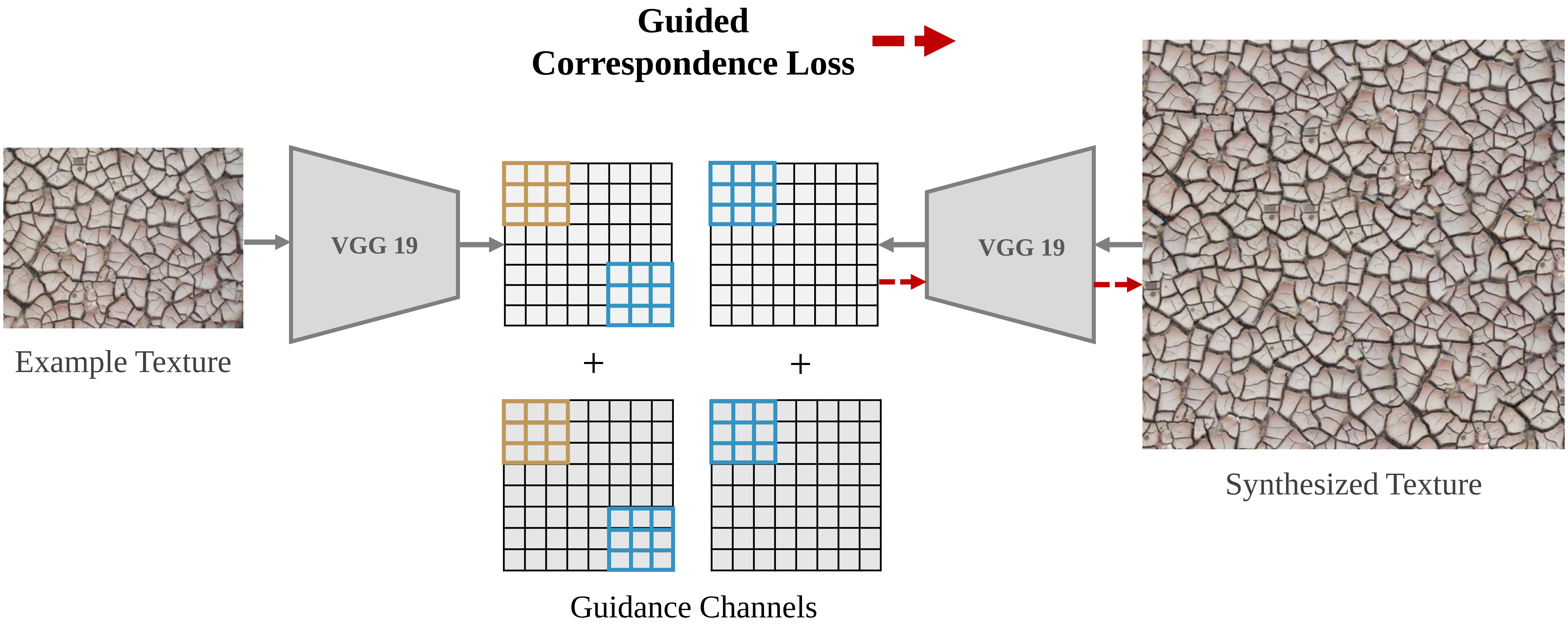

Rongjun Xiao


Hui Huang

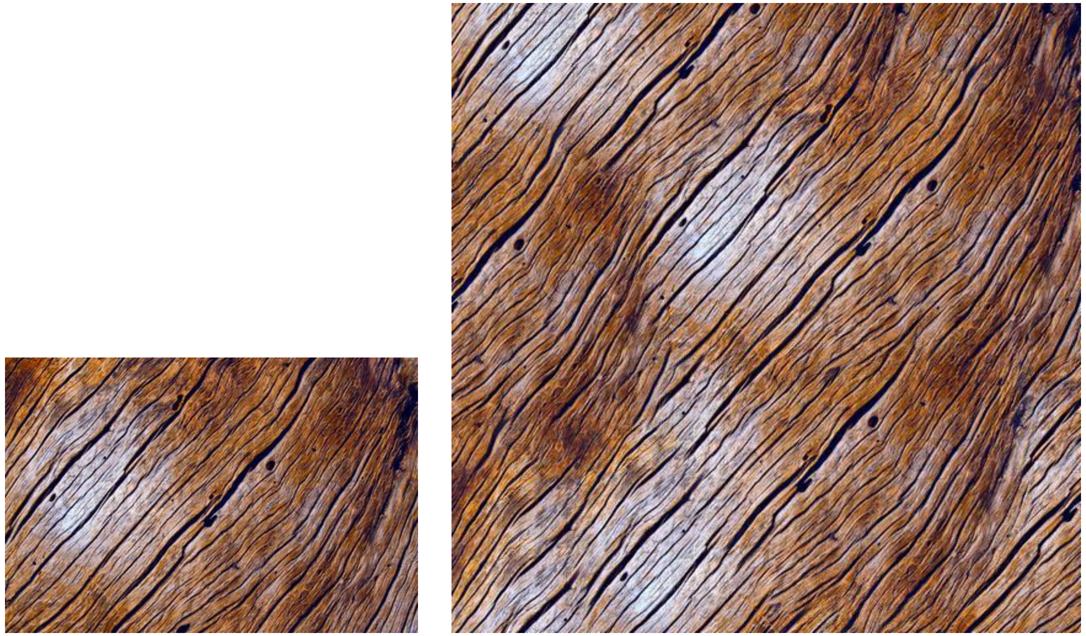
深圳大学
SHENZHEN UNIVERSITY

深圳大学可视计算研究中心
VISUAL COMPUTING RESEARCH CENTER

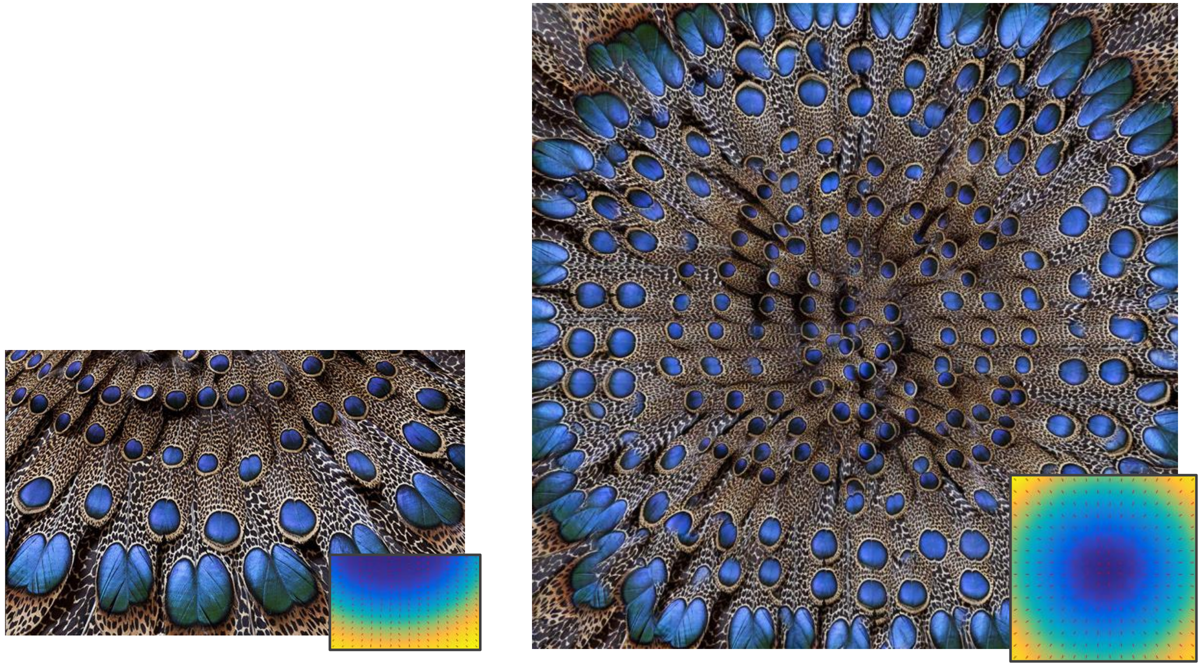
Example-based Texture Synthesis



Texture Optimization

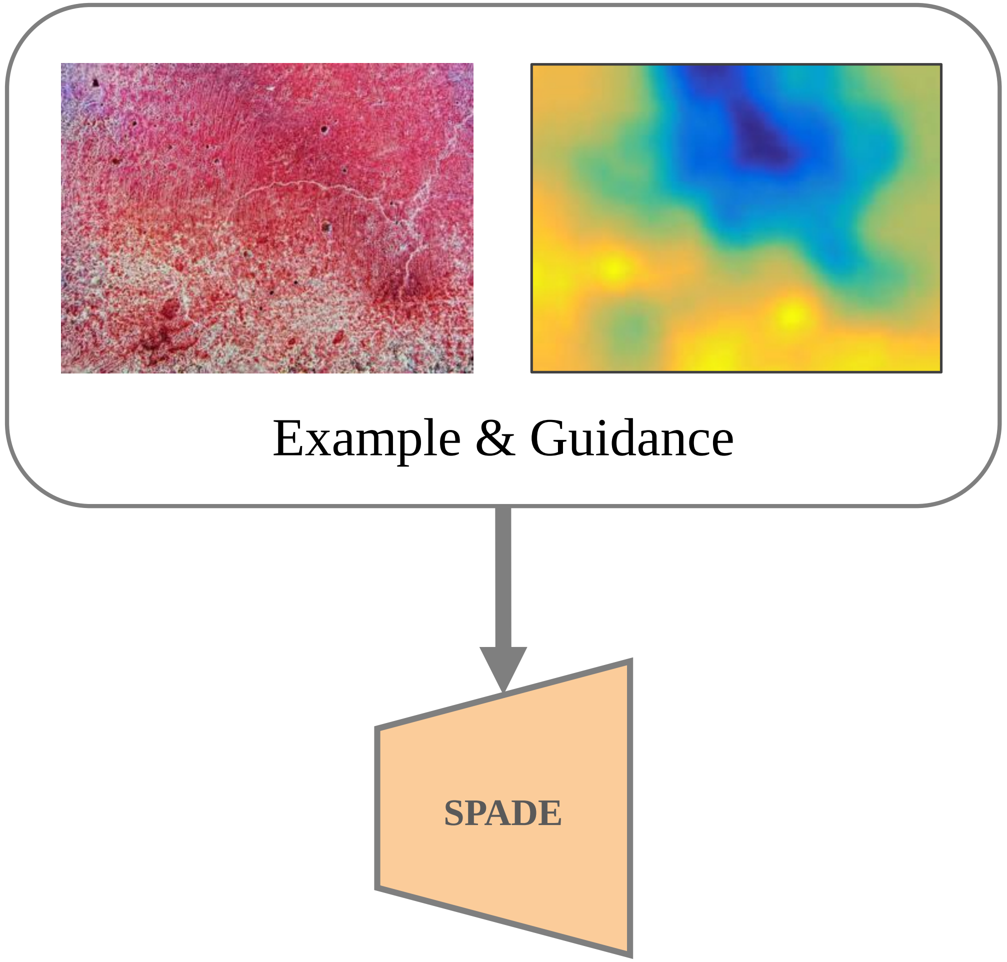


Uncontrolled Texture Optimization



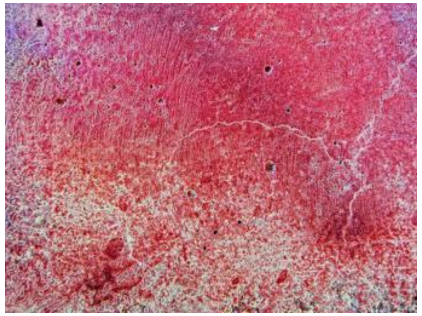
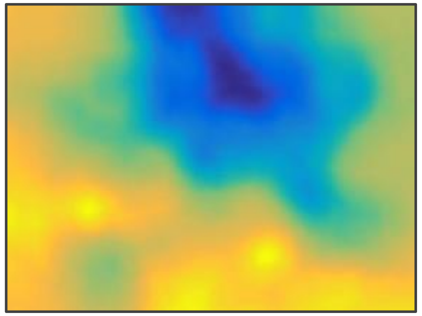
Controlled Texture Optimization with Guidance

Real-time Texture Synthesis

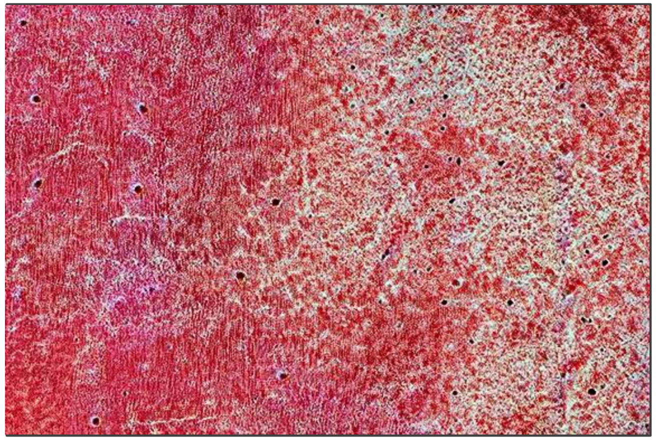
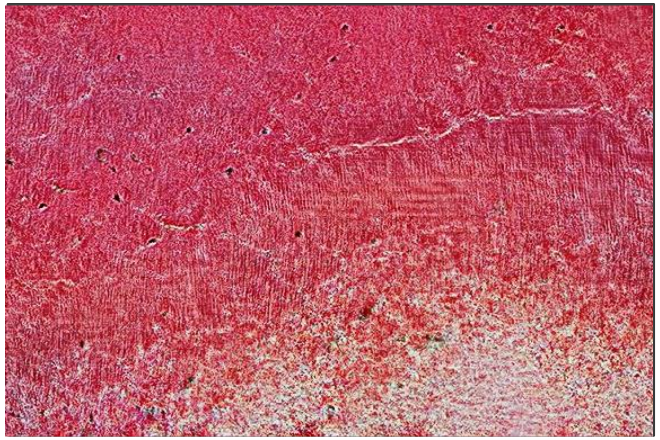
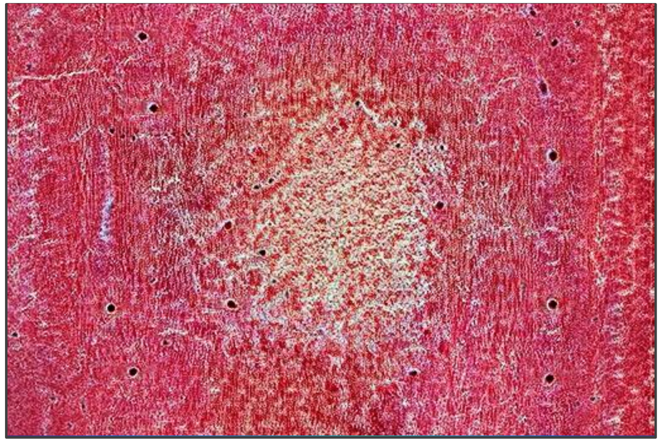


Feed-forward Generative Networks

Real-time Texture Synthesis

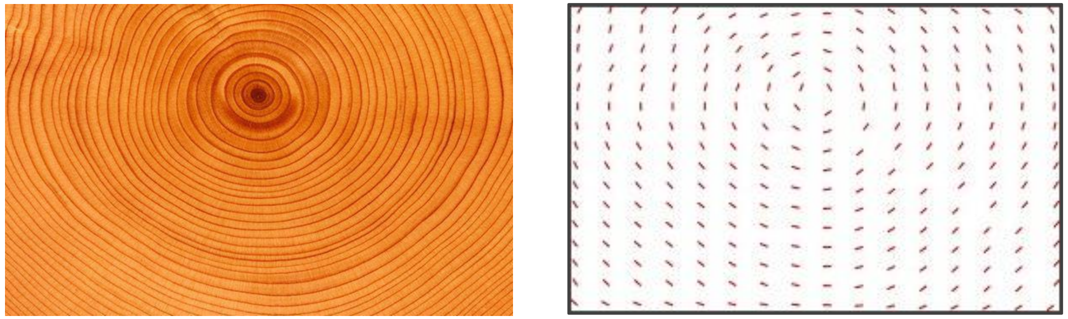
 

Example & Guidance

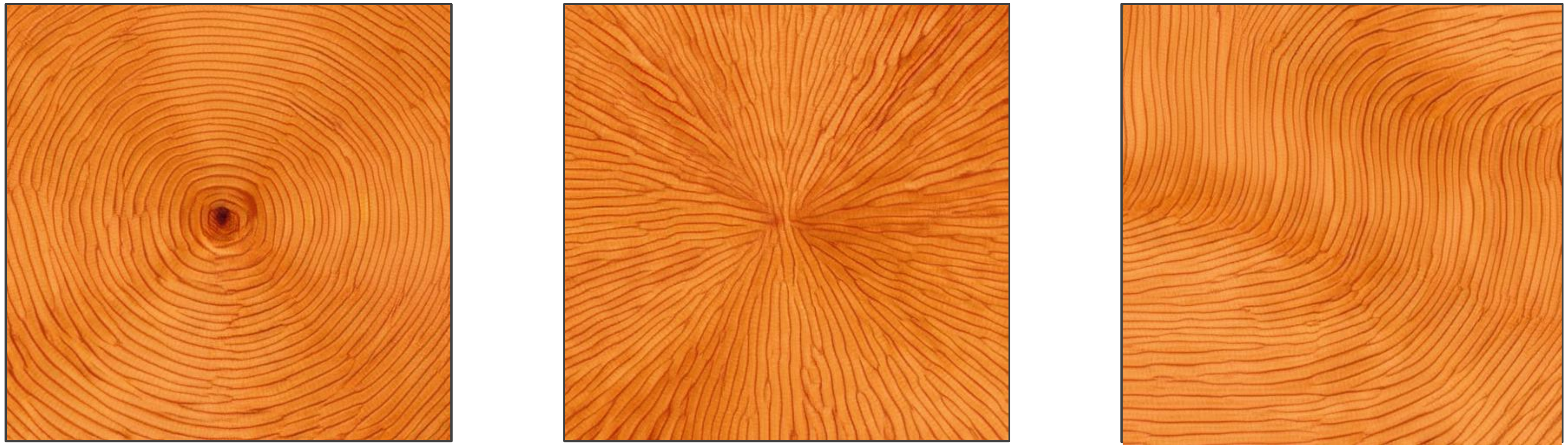
  

Feed-forward Generative Networks

Real-time Texture Synthesis



Example & Guidance



Feed-forward Generative Networks

Texture Transfer

Content Image

Reference Texture

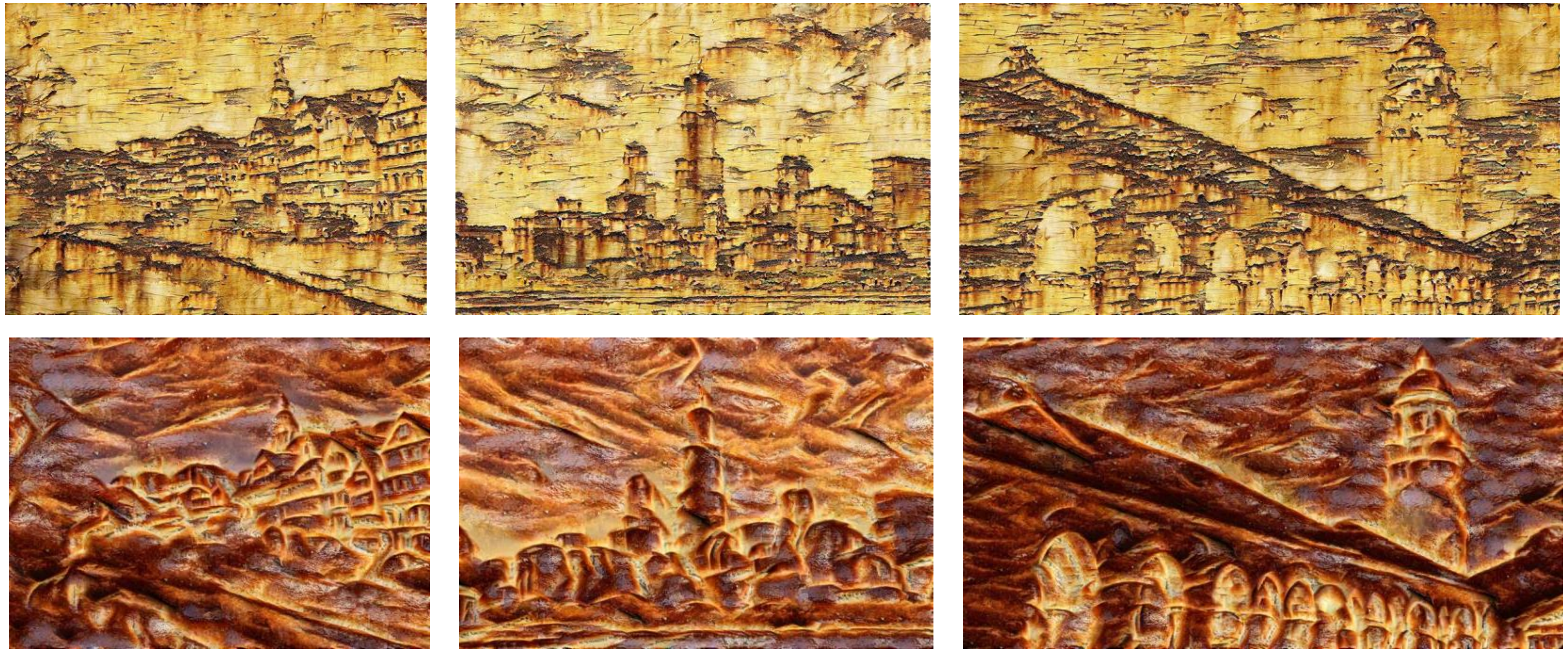
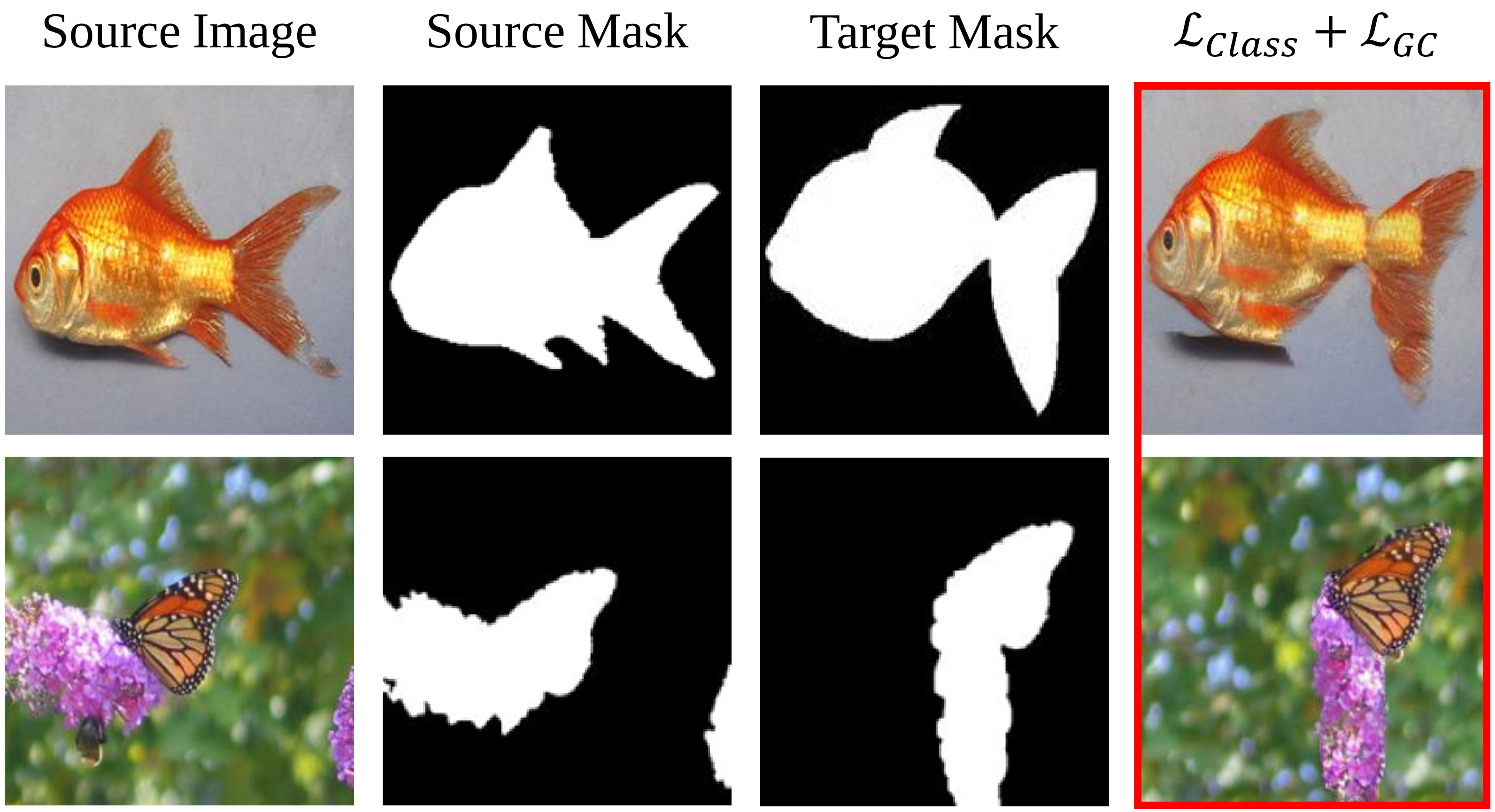


Image Inpainting



Single-image Editing



Background

- Example-based texture synthesis

Source Texture

Texture Synthesis

Target Texture

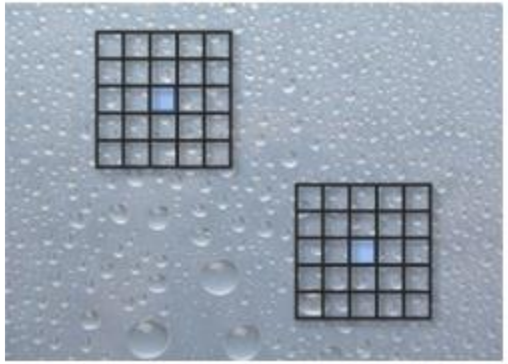
Classical approaches

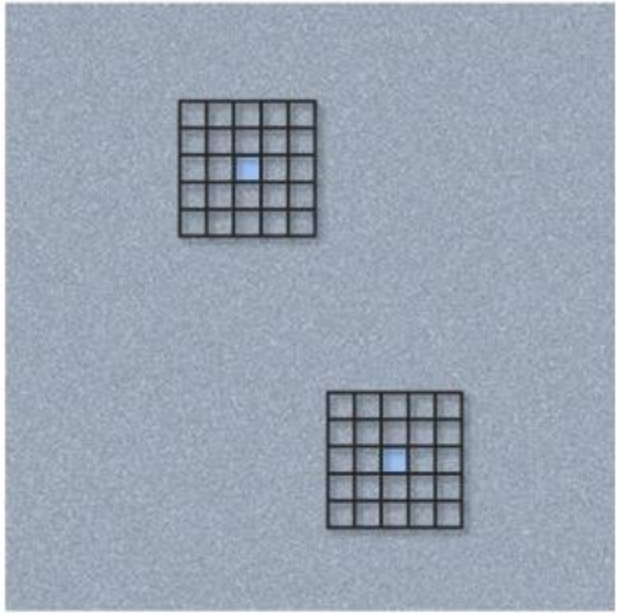
➢Markov Random Field (MRF)-based Texture Optimization:

- **Goal:** optimize all overlapping output patches to be similar to their nearest neighbor in the input

Target Texture

Source Texture



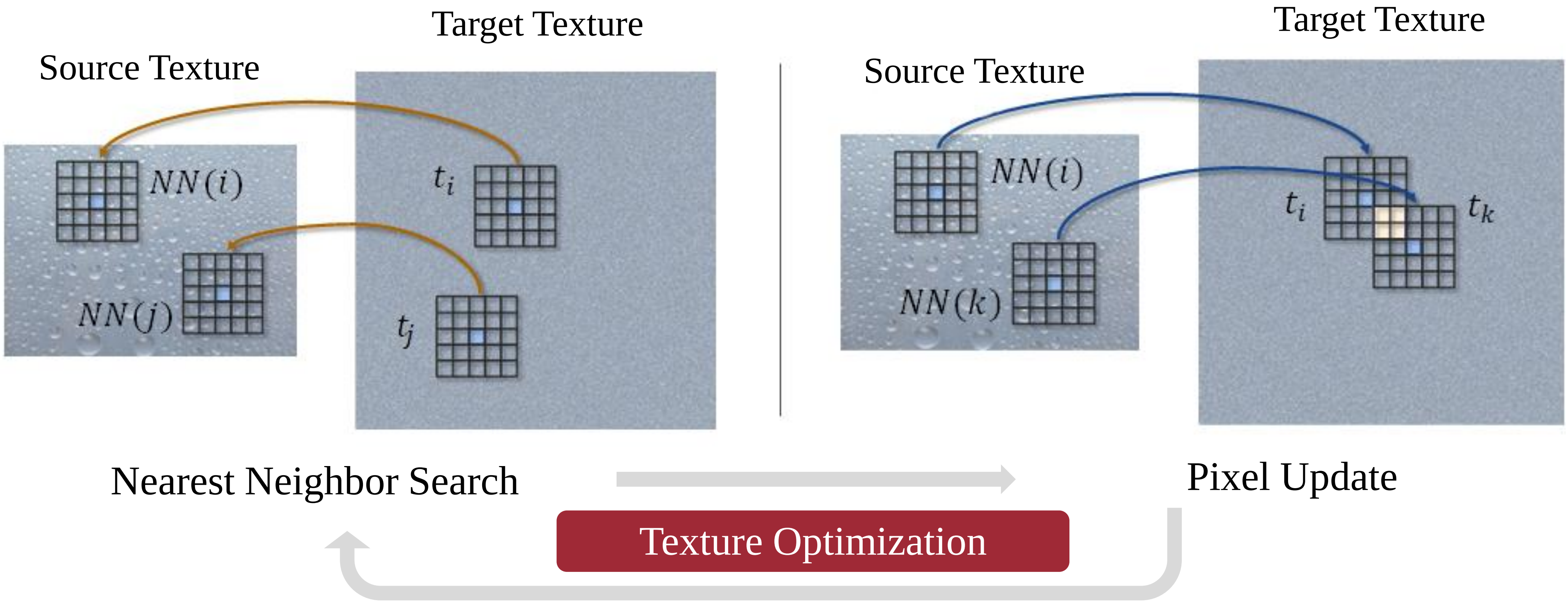


[Kwatra et al. Texture Optimization for Example-based Synthesis. 2005]

Classical approaches

➢Markov Random Field (MRF)-based Texture Optimization:

- **Goal:** optimize all overlapping output patches to be similar to their nearest neighbor in the input.

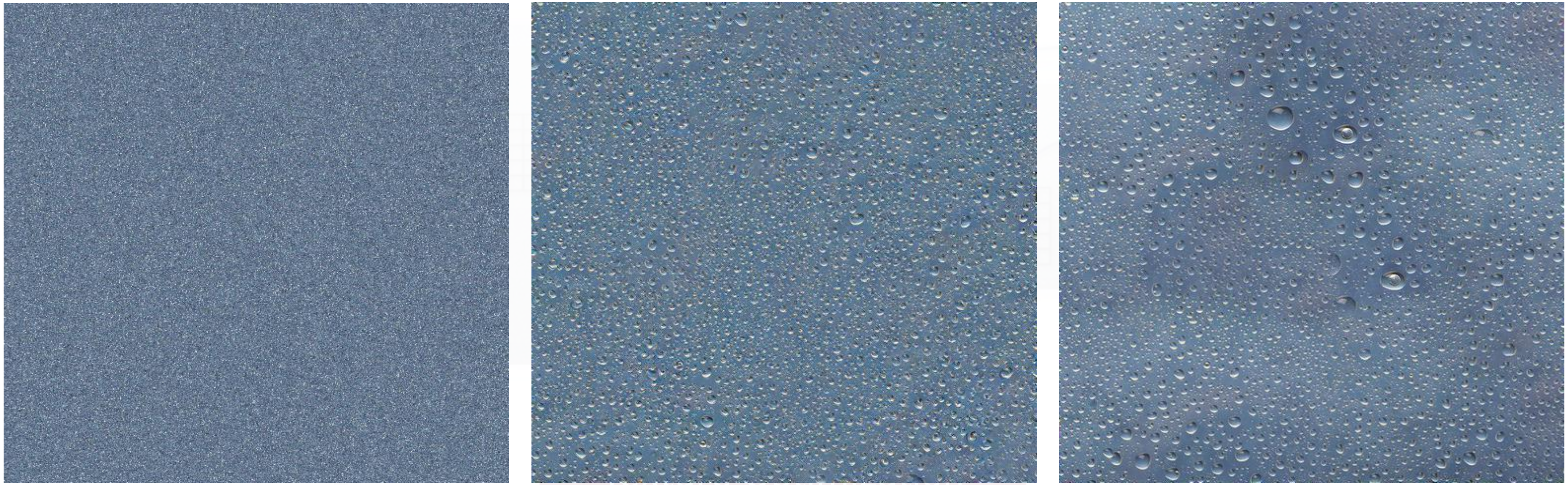


[Kwatra et al. Texture Optimization for example-based Synthesis. 2005]

Classical approaches

➢Markov Random Field (MRF)-based Texture Optimization:

- **Goal:** optimize all overlapping output patches to be similar to their nearest neighbor in the input.

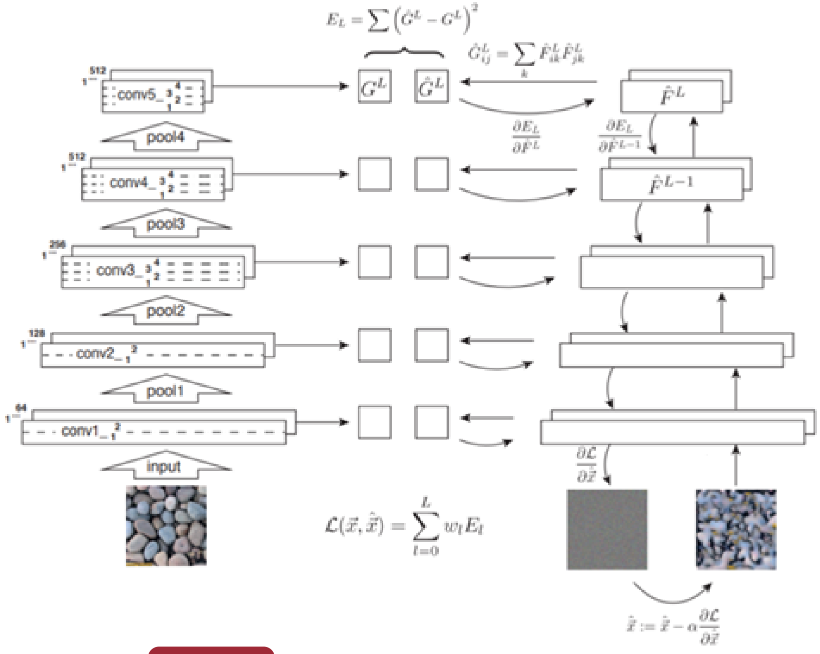


Textures are optimized gradually along with the iteration.

[Kwatra et al. Texture Optimization for example-based Synthesis. 2005]

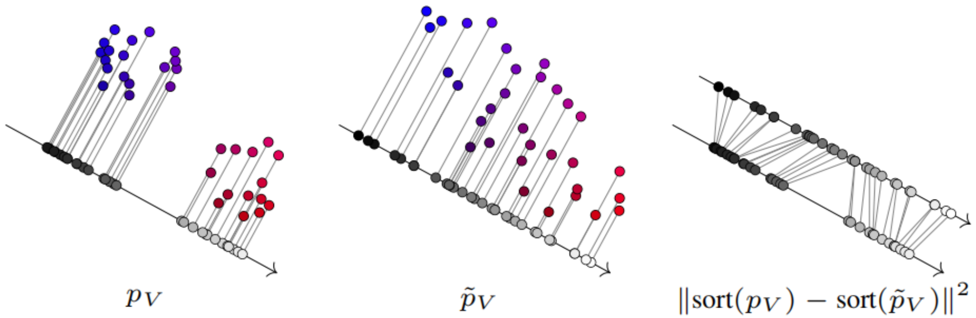
Deep learning based approaches

➢ Align the statistics of deep features



$\mathcal{L}_{Gram}$ Gram Loss

[Gatys et al. Image Style Transfer Using Convolutional Neural Networks. 2016]



$\mathcal{L}_{SW}$ Sliced Wasserstein Loss

[Heitz et al. A Sliced Wasserstein Loss for Neural Texture Synthesis. 2021]

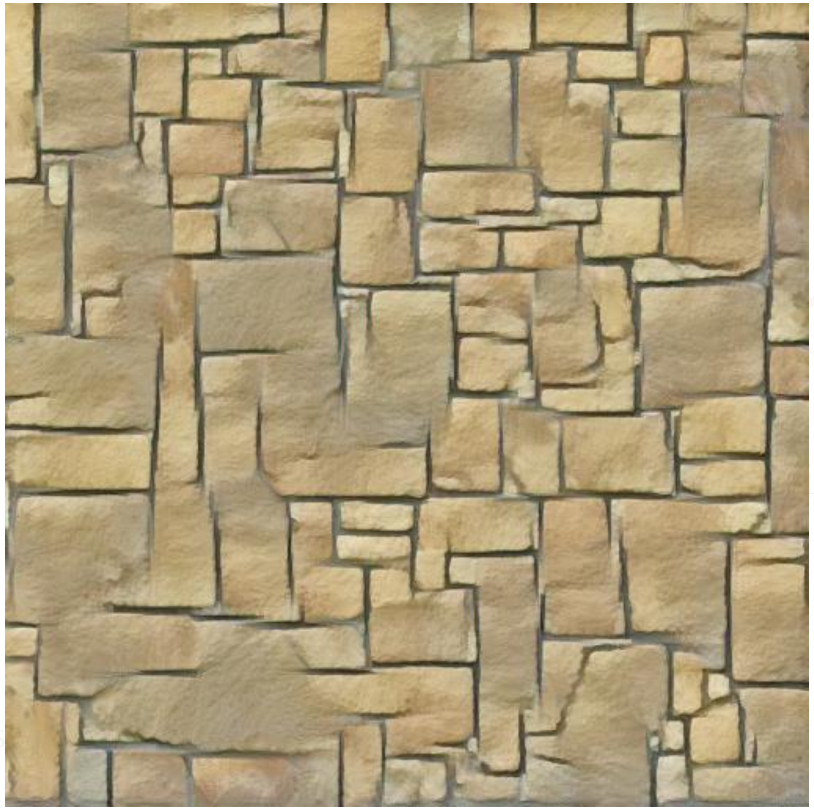
Deep learning based approaches

- Align the statistics of deep features

Only suitable for Homogeneous Textures!!!

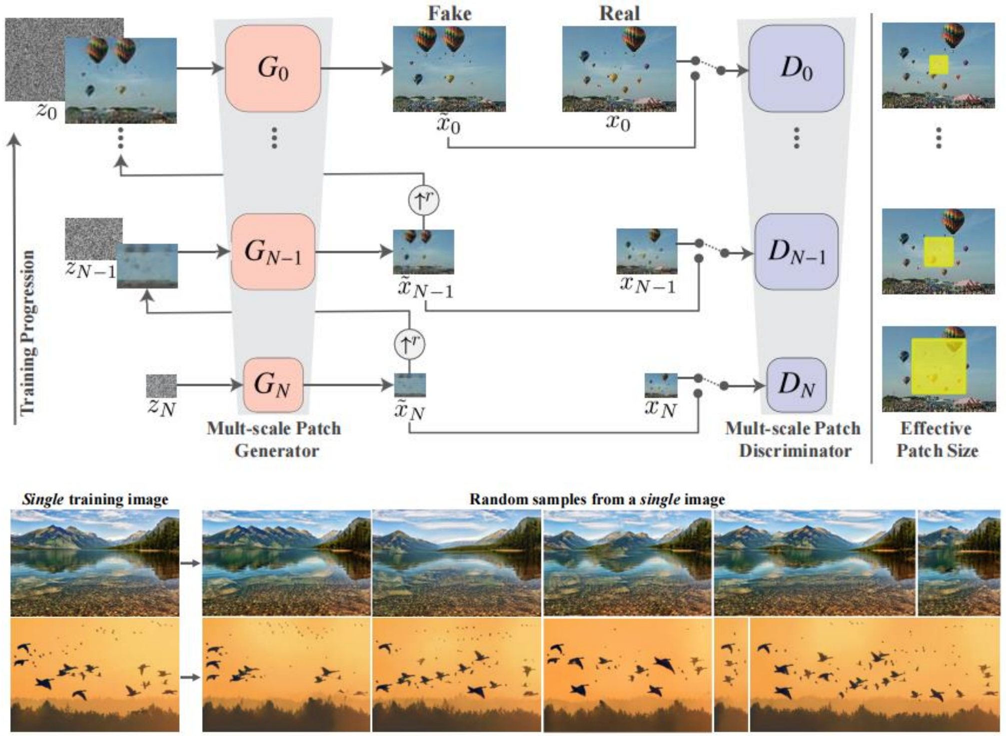
Source Texture

Sliced Wasserstein loss



Deep learning based approaches

➢GAN-based Methods



TexExp Texture Expansion Networks

[Zhou et al. Non-Stationary Texture Synthesis by Adversarial Expansion. 2018]

SinGAN Single Image Generative Adversarial Networks

[Shaham et al. Learning a Generative Model from a Single Natural Image. 2019]

Deep learning based approaches

Suffer from visual artifacts!!!

- GAN-based Methods

Source Texture

TexExp

SinGAN

[Zhou et al. Non-Stationary Texture Synthesis by Adversarial Expansion. 2018]

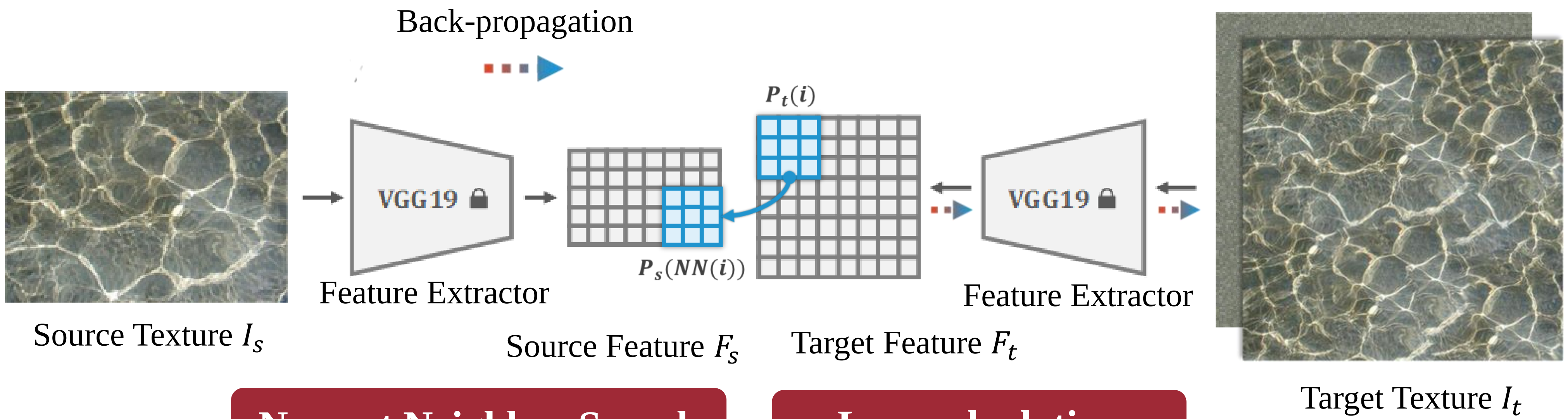
[Shaham et al. Learning a Generative Model from a Single Natural Image. 2019]

CNNMRF

CNNMRF CNNMRF (Convolution Neural Network + MRF)

- Nearest Neighbor Search: Color → Deep Feature
- Pixel Update: Color Averaging → Loss Back-propagation

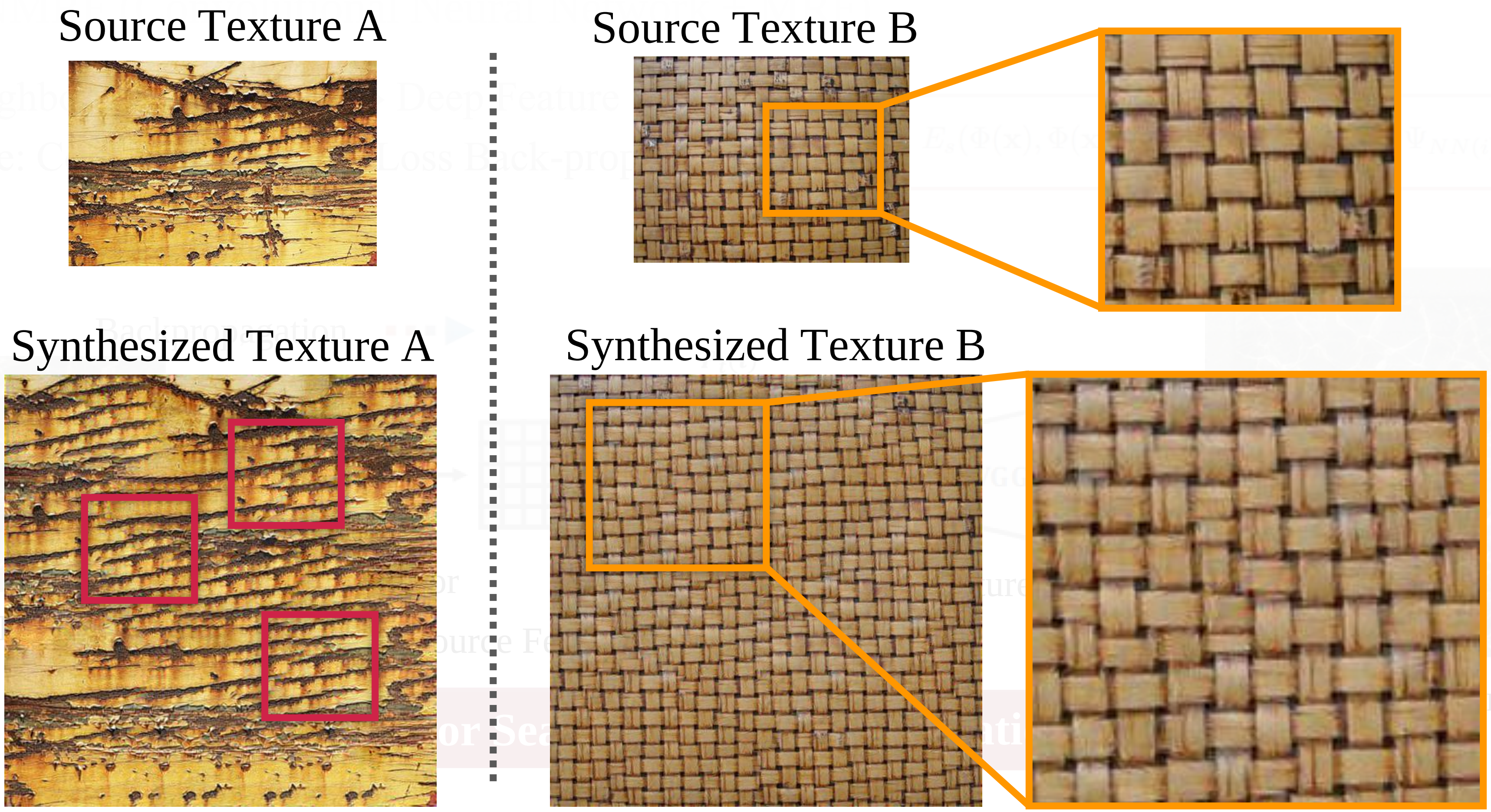
$$E_s(\Phi(\mathbf{x}), \Phi(\mathbf{x}_s)) = \sum_{i=1}^{m} ||\Psi_i(\Phi(\mathbf{x})) - \Psi_{NN(i)}(\Phi(\mathbf{x}_s))||^2$$



Nearest Neighbor Search **Loss calculation**

[Li et al. Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis. 2016]

CNNMRF



1) Repetition Issue!!! **2) Blurry Issue!!!**

Guided Correspondence Distance

➢Distance definition:

$$d_{ij} = d_{ij}^{VGG} + \lambda_{GC} * d_{ij}^{GC} + \lambda_{occ} * d_{j}^{occ}$$

Guided Correspondence Distance

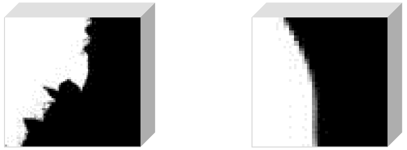
Deep Feature Distance

Guided Feature Distance

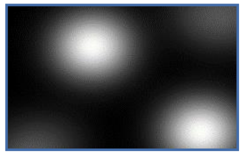
Occurrence Penalty

Target patch *i* and source patch *j*

Cosine similarity



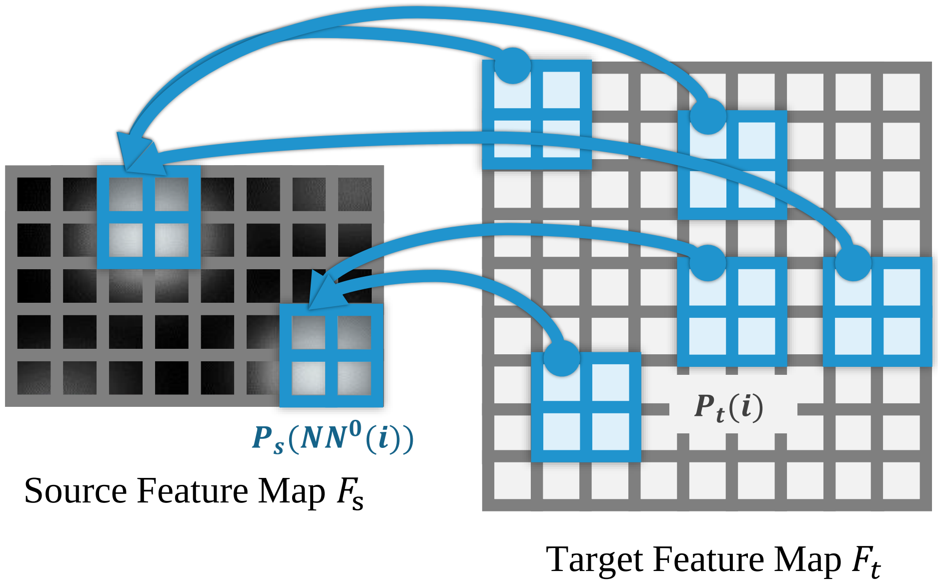
Guided similarity measure
(depends on the type of guidance channel)



Guided Correspondence Distance

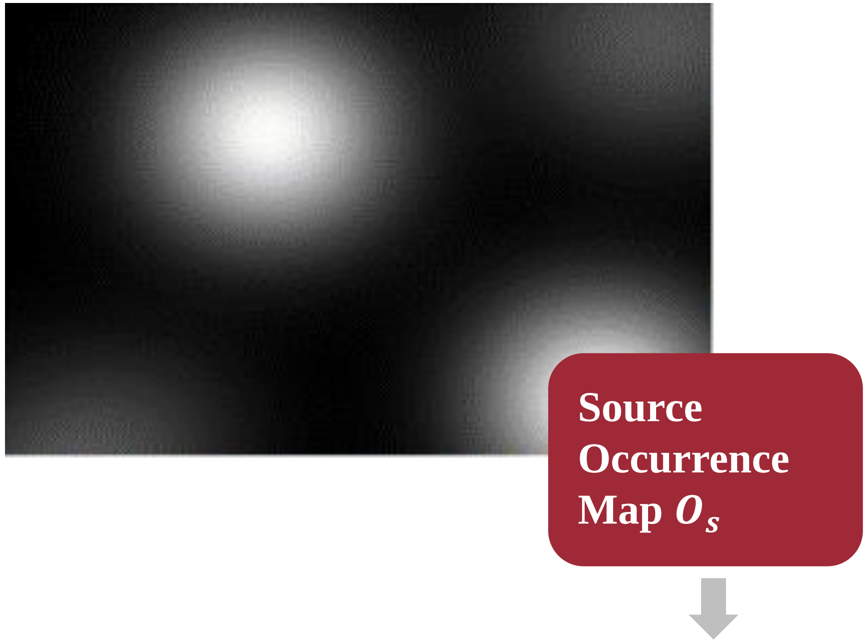
➢Occurrence penalty for solving the repetition issue.

- To prevent a source patch from being repeatedly selected as the correspondence.



$$d_{ij}^0 = d_{ij}^{VGG} + \lambda_{GC} * d_{ij}^{GC}$$

Calculate the **approximate nearest neighbor field**

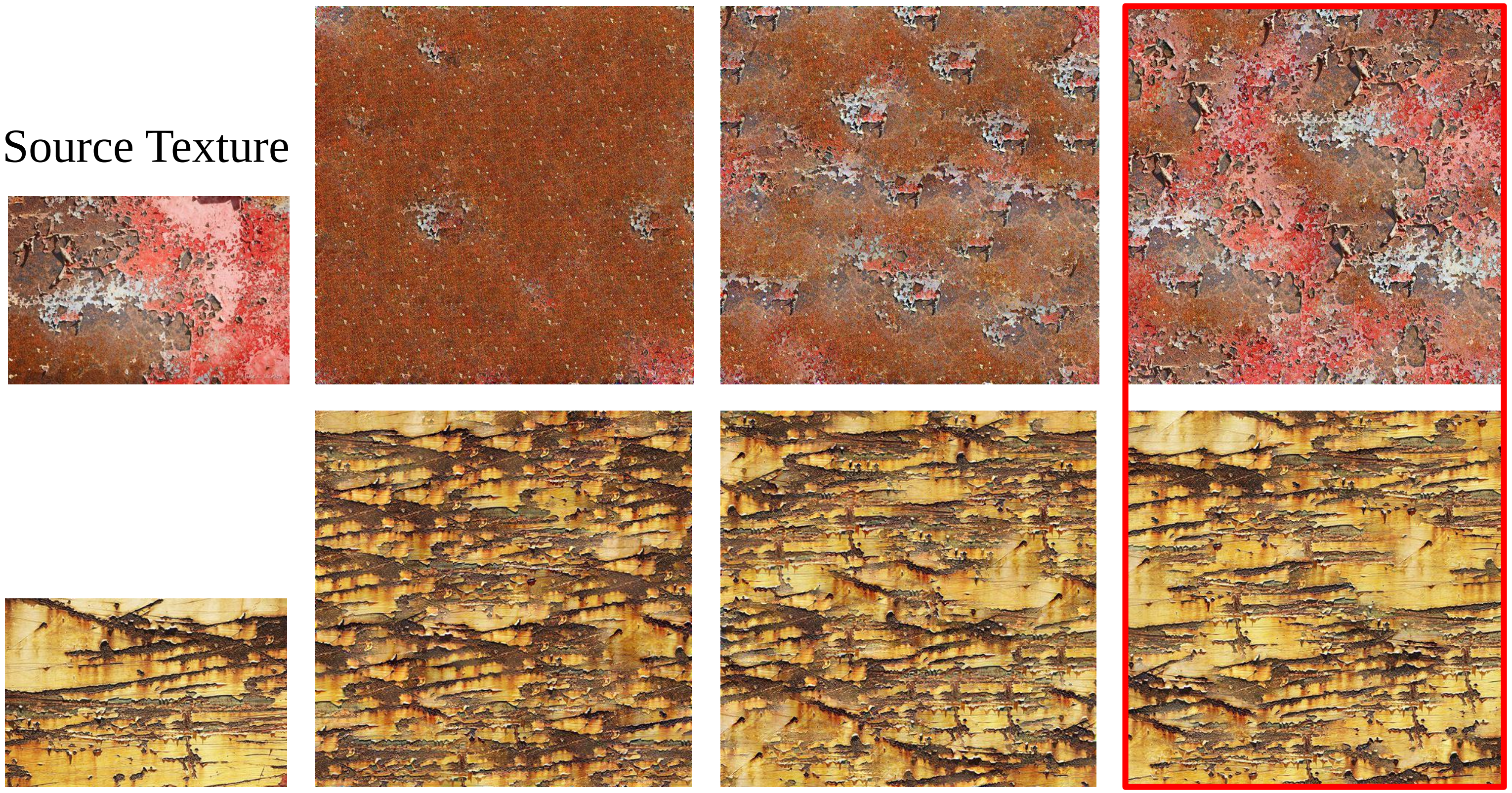


$$d_{ij} = d_{ij}^{VGG} + \lambda_{GC} * d_{ij}^{GC} + \lambda_{occ} * d_j^{occ}$$

Guided Correspondence Distance

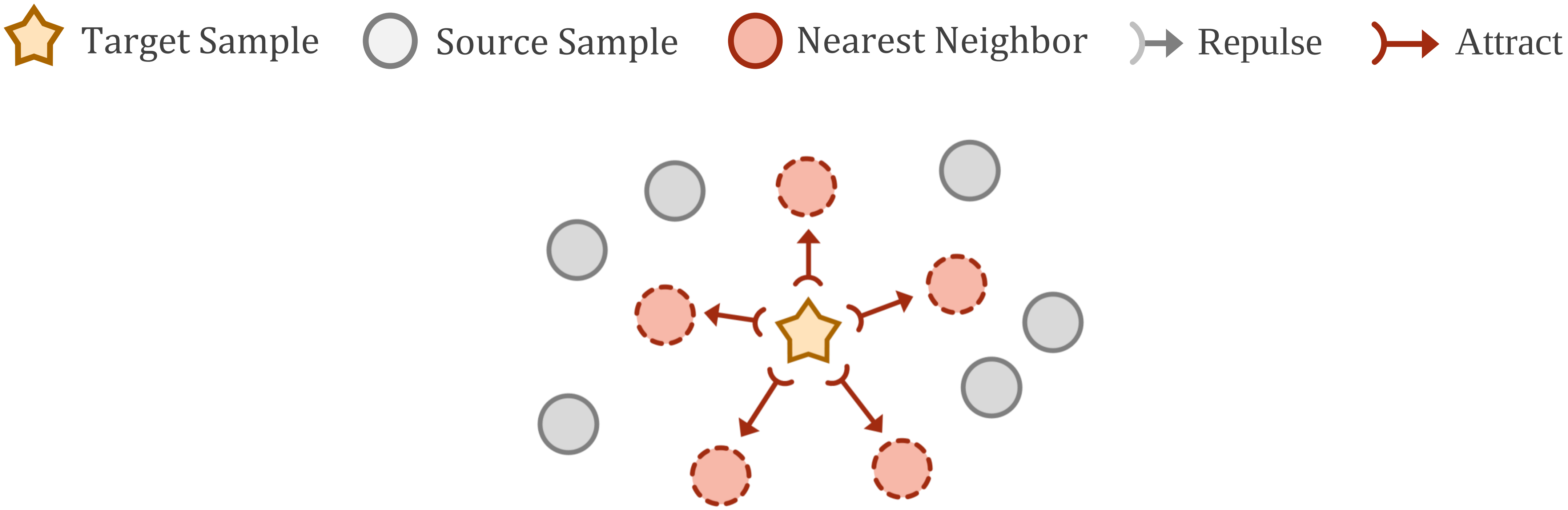
- Ablation study on occurrence penalty

Source Texture | $\lambda_{occ} = 0$ | $\lambda_{occ} = 0.005$ | $\lambda_{occ} = 0.05$



Guided Correspondence Loss

➢**Blurry issue** is caused by **nearest neighbor inconsistency** over the iterations

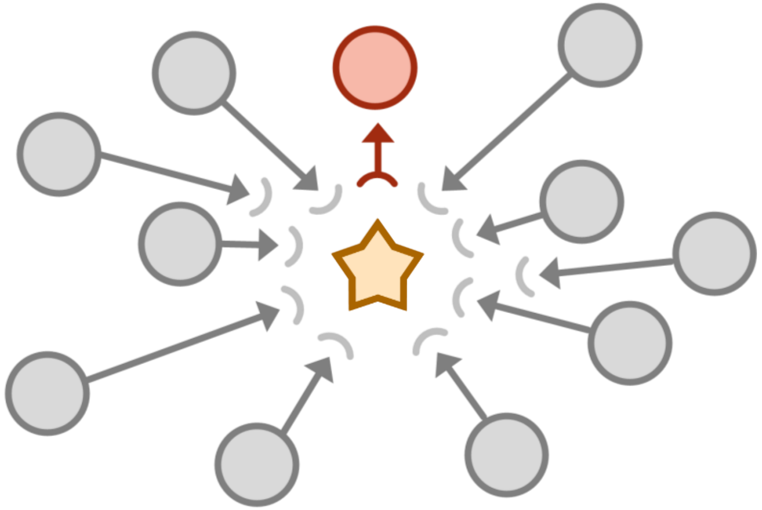


Minimizing the conventional L2-distance makes the optimization favor **the average of nearby samples**.

Guided Correspondence Loss

- **Contextual similarity** requires a target sample to be significantly closer to its nearest neighbor than to all other source samples.

Target Sample   Source Sample   Nearest Neighbor   Repulse   Attract



Contextual Similarity

[Mechrez et al. The contextual loss for image transformation with non-aligned data. 2018]

Guided Correspondence Loss

➢**Contextual similarity** requires a target sample to be significantly closer to its nearest neighbor than to all other source samples.

1. Normalize the distance and convert it to similarity:

$$w_{ij} = \exp\left(\frac{1 - d_{ij}/(\min_k d_{ik} + \epsilon)}{h}\right),$$

2. Introduce contextual information to obtain contextual similarity：

$$CX_{ij} = w_{ij}/\sum_k w_{ik}.$$

3. Calculate the Guided Correspondence loss ：

$$\mathcal{L}_{GC}(I_t, I_s) = \frac{1}{n_t}\sum_i -log(CX_{i,NN(i)}).$$

Guided Correspondence Loss

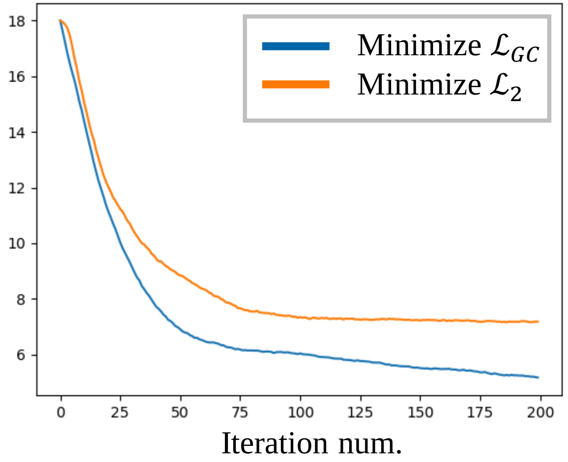
- Ablation study: $\mathcal{L}_2$-based loss vs. $\mathcal{L}_{GC}$ loss

Source

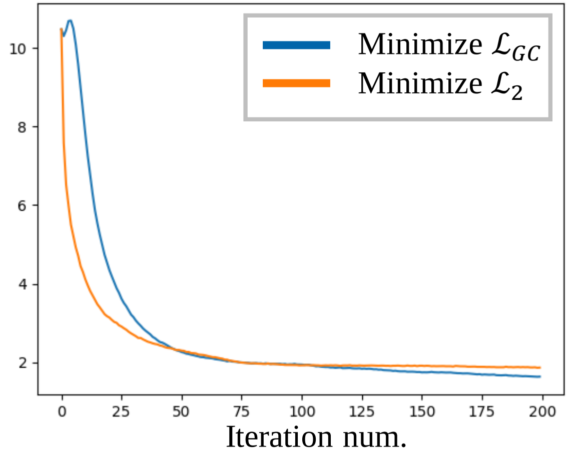
$\mathcal{L}_2$-based

Ours

Monitoring $\mathcal{L}_{GC}$ loss

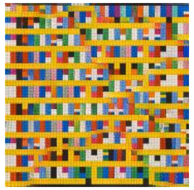


Monitoring $\mathcal{L}_2$-based loss

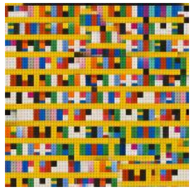


Source

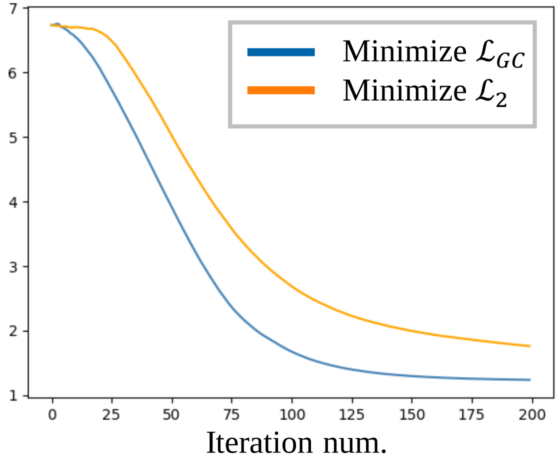
$\mathcal{L}_2$-based



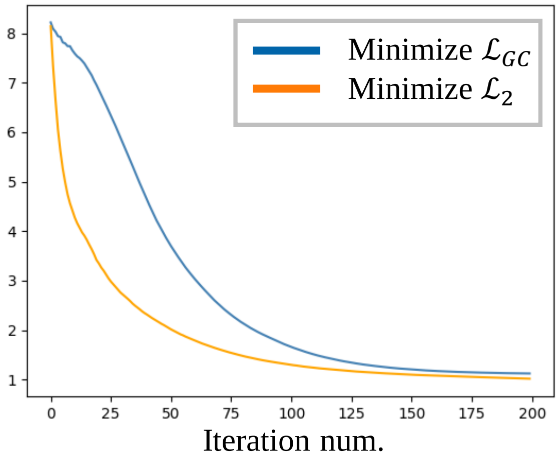
Ours



Monitoring $\mathcal{L}_{GC}$ loss

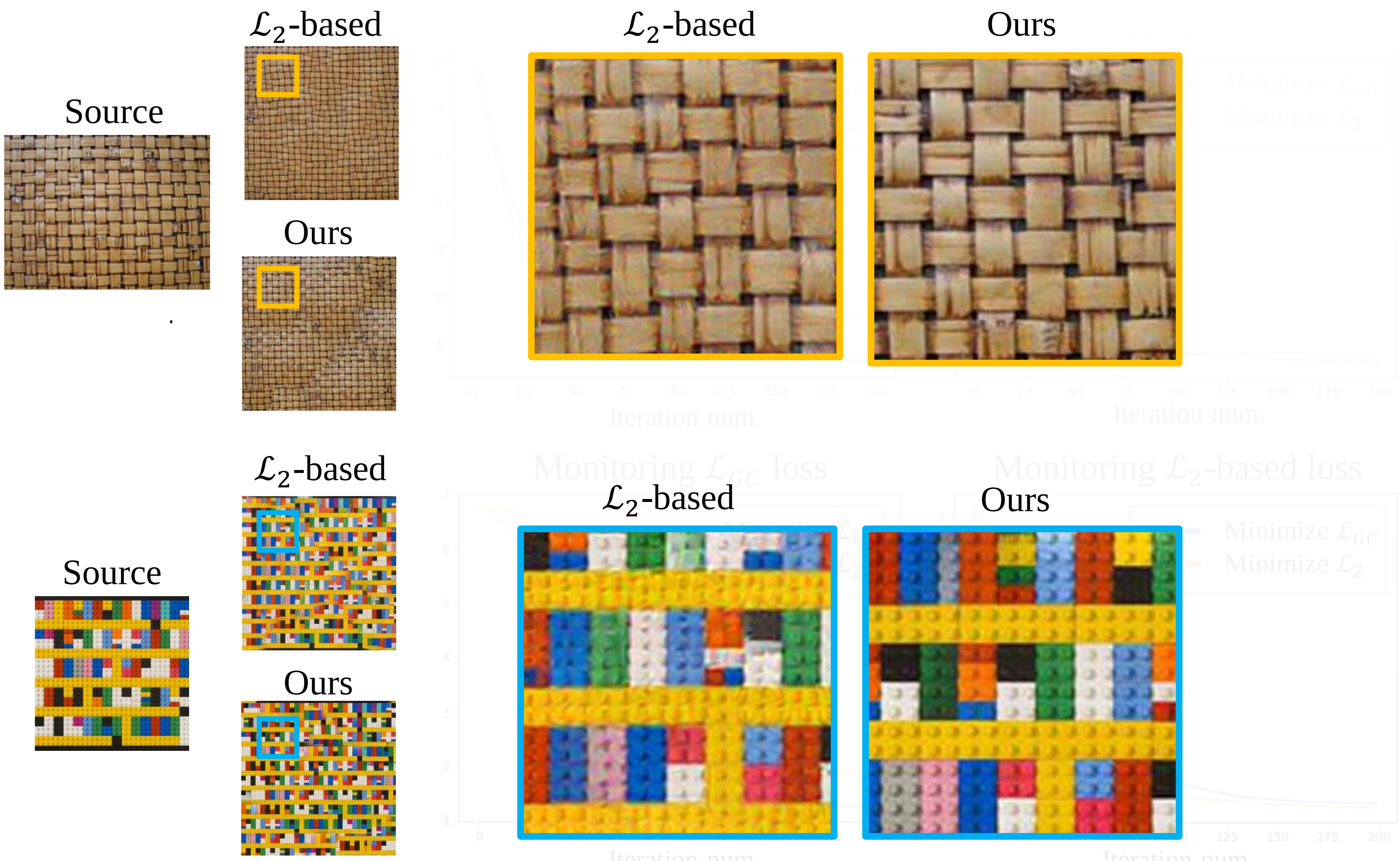


Monitoring $\mathcal{L}_2$-based loss



Guided Correspondence Loss

- Ablation study: $\mathcal{L}_2$-based loss vs. $\mathcal{L}_{GC}$ loss

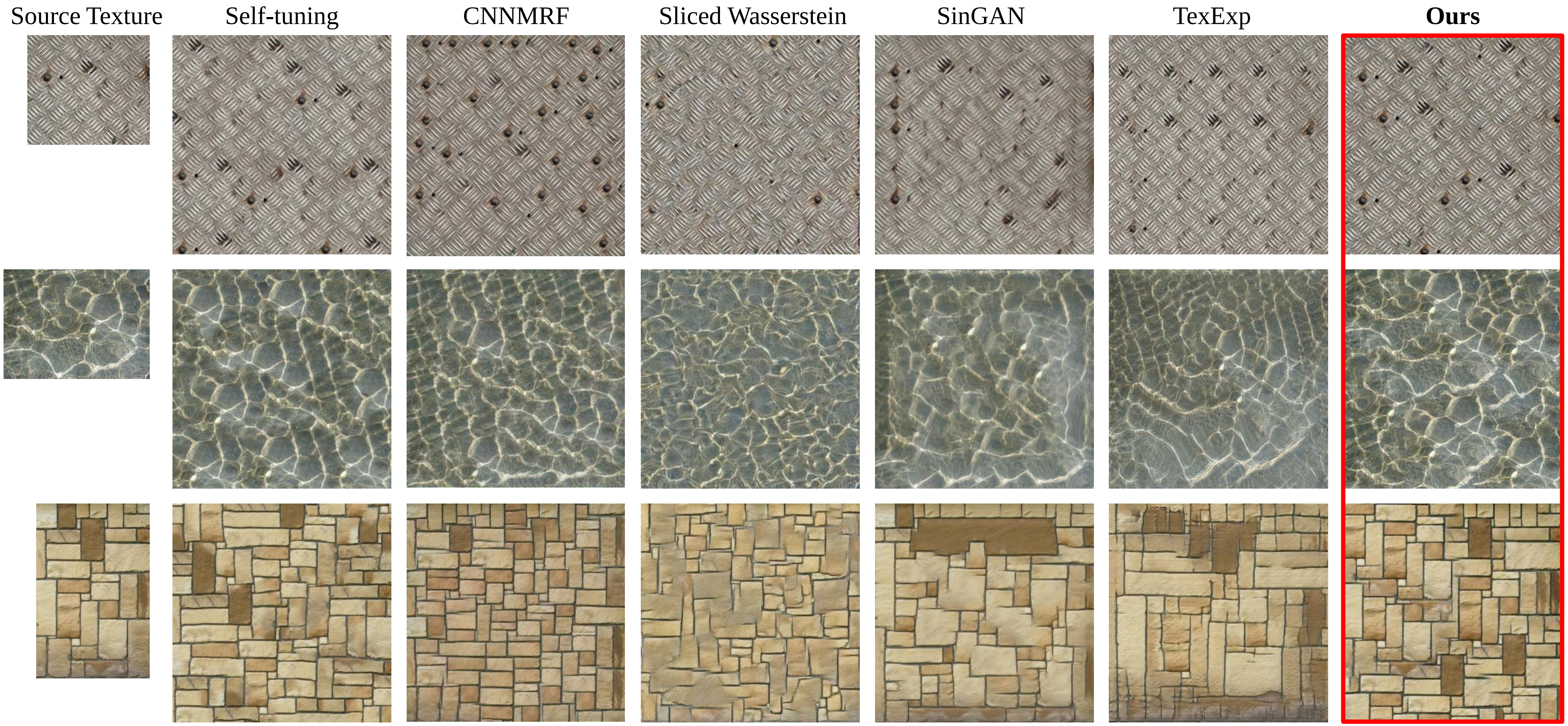


Experiments

Texture Dataset

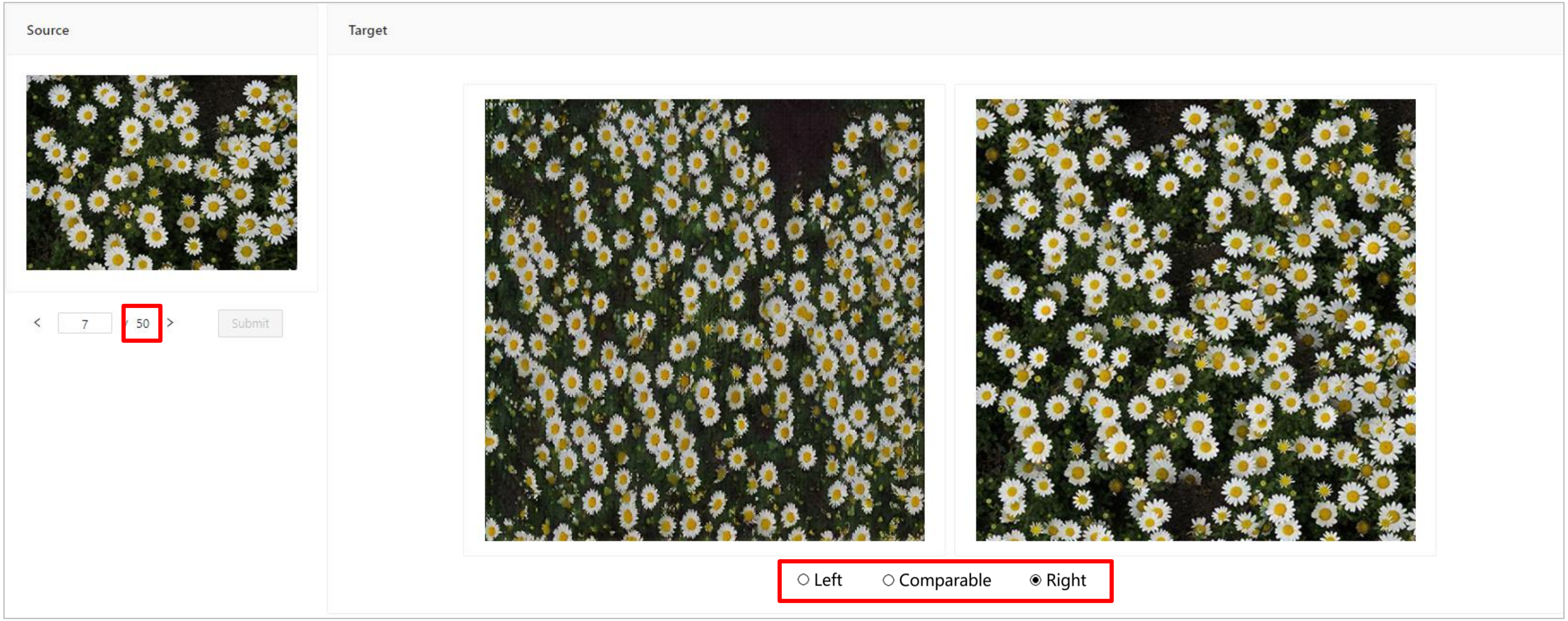
Uncontrolled Texture Synthesis

	Self-tuning	CNNMRF	SWD	SinGAN	TexExp	Ours
ColorDis	2.65	23.16	24.21	15.72	25.32	9.40



Uncontrolled Texture Synthesis

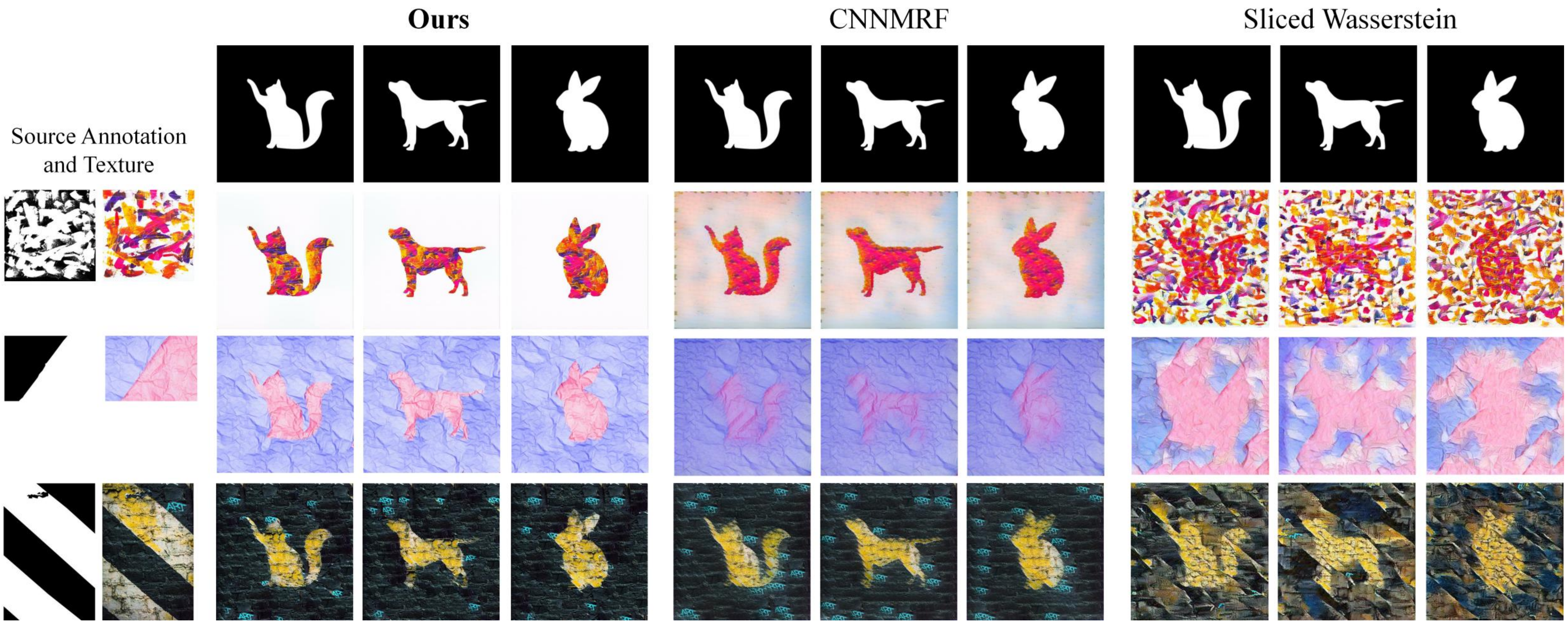
	Self-tuning	CNNMRF	SWD	SinGAN	TexExp	Ours
User Pref.	47.3%	33.1%	31.5%	9.57%	33.0%	-



User Study

Controlled Synthesis: Annotation

$$d_{ij}^{GC} = \frac{1}{k*k}\sum_{x=0}^{k*k} ||G_t(i)(x) - G_s(j)(x)||_0$$



Controlled Synthesis: Annotation

$$d_{ij}^{GC} = \frac{1}{k*k}\sum_{x=0}^{k*k} ||G_t(i)(x) - G_s(j)(x)||_0$$

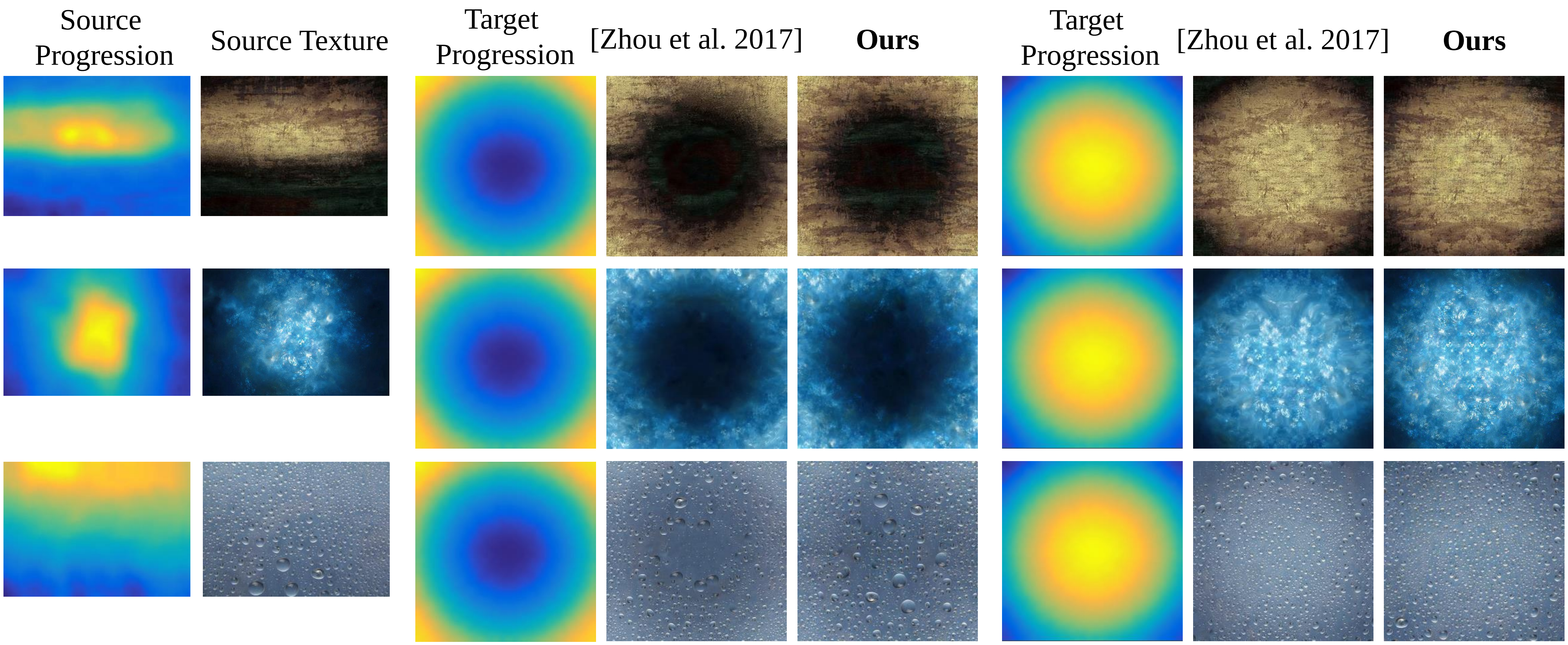
Ours

CNNMRF

Sliced Wasserstein

Controlled Synthesis: Progression

$$d_{ij}^{GC} = \frac{1}{k*k}\sum_{x=0}^{k*k} ||G_t(i)(x) - G_s(j)(x)||_2^2.$$



Controlled Synthesis: Progression

$$d_{ij}^{GC} = \frac{1}{k*k}\sum_{x=0}^{k*k}||G_t(i)(x) - G_s(j)(x)||_2^2.$$

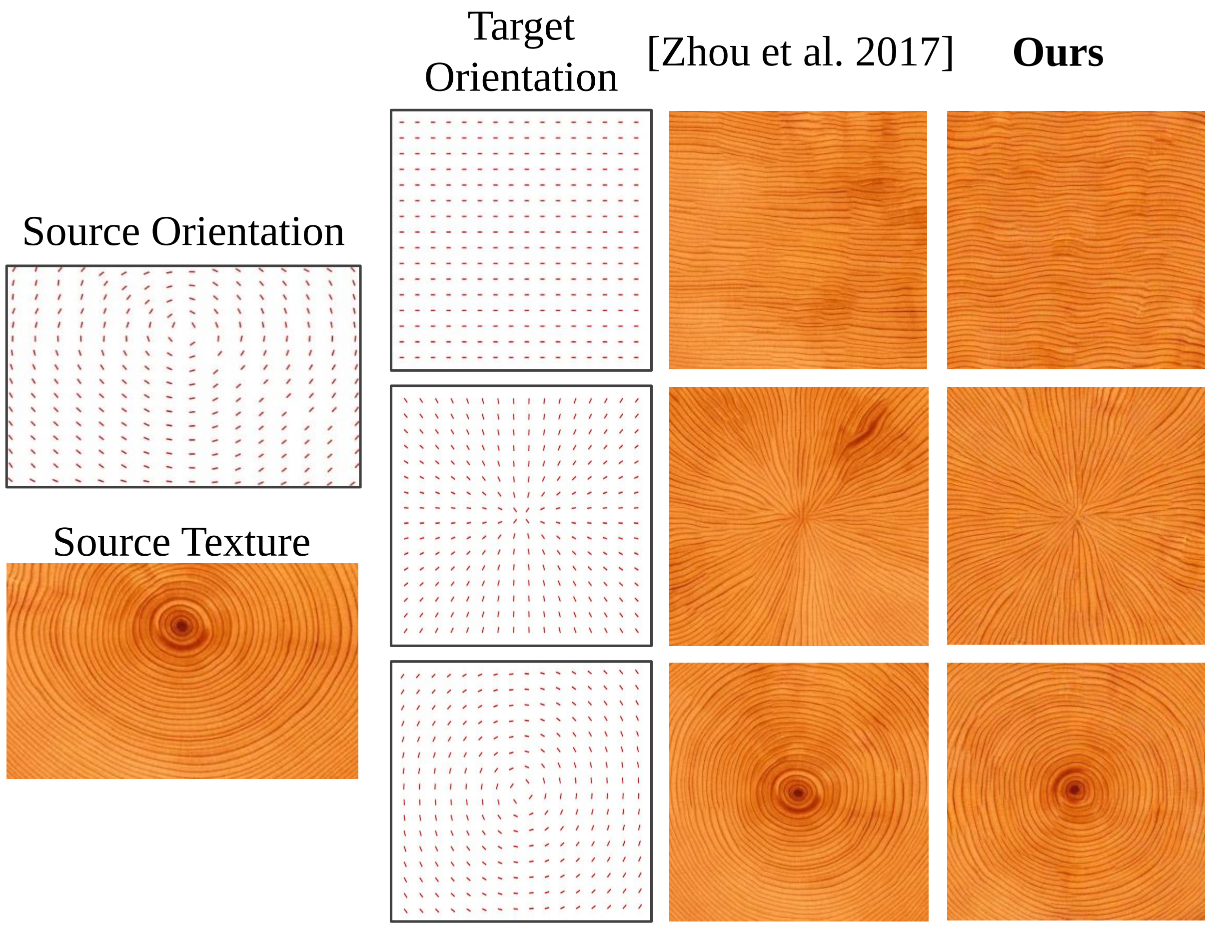
Source Texture

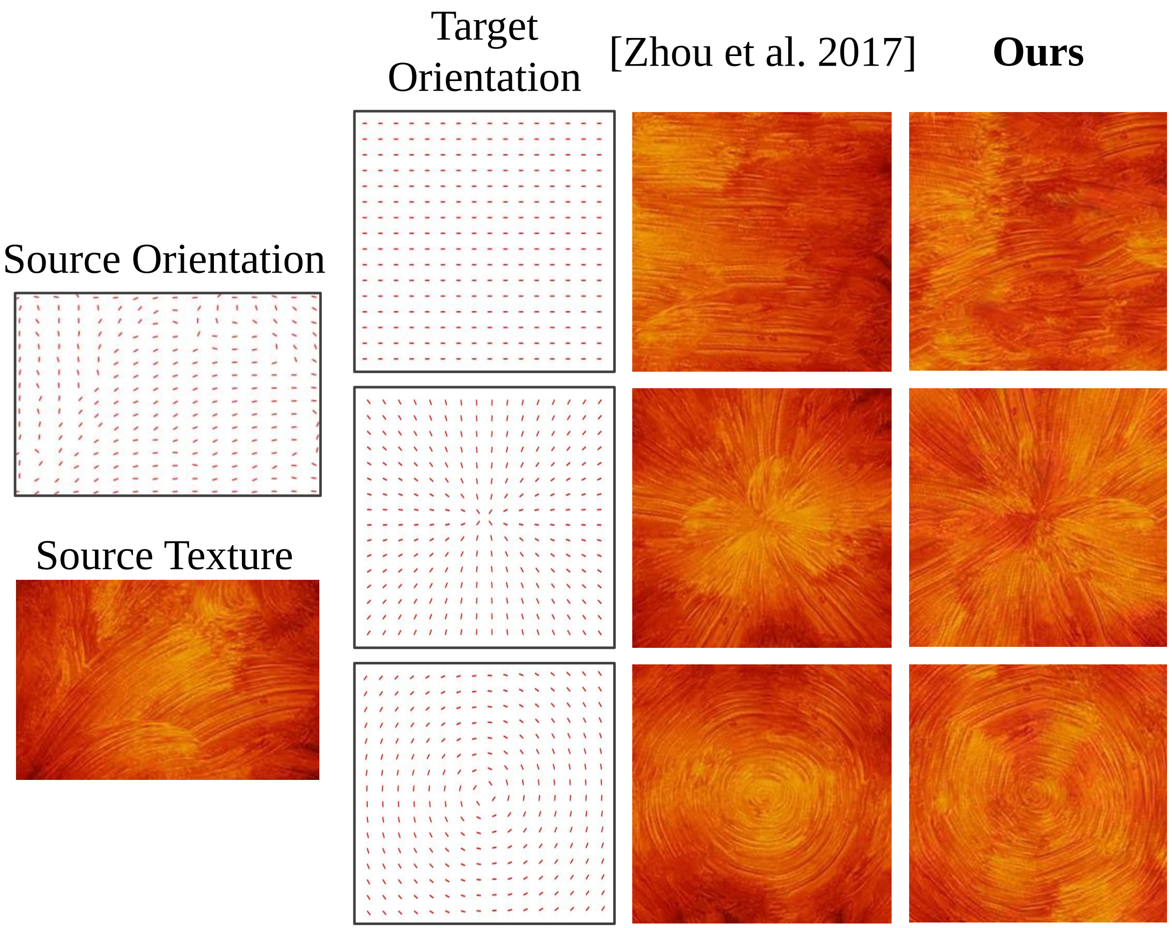
[Zhou et al. 2017]

Ours

Controlled Synthesis: Orientation

$$d_{ij}^{GC} = \frac{1}{k*k}\sum_{x=0}^{k*k}\left(1 - \frac{|G_t(i)(x)\cdot G_s(j)(x)|}{|G_t(i)(x)|\cdot|G_s(j)(x)|}\right)$$



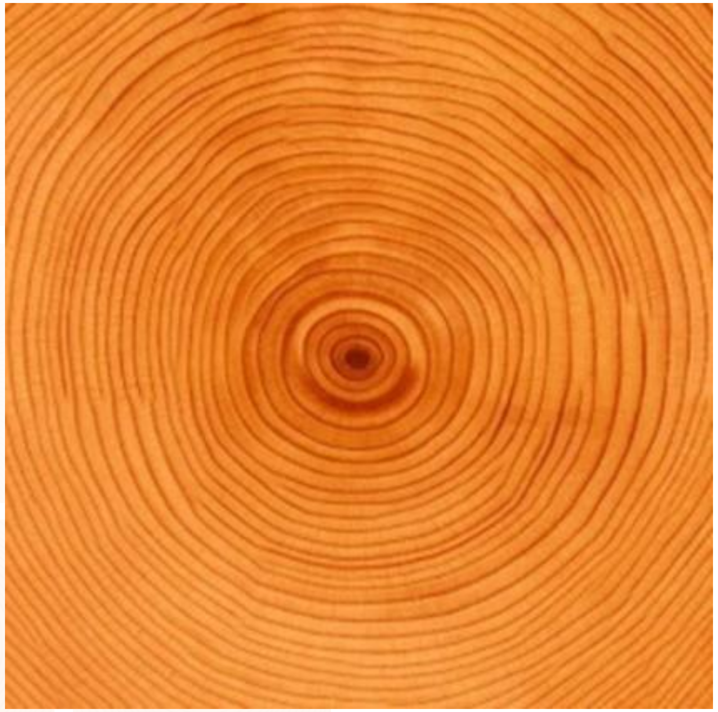


Controlled Synthesis: Orientation

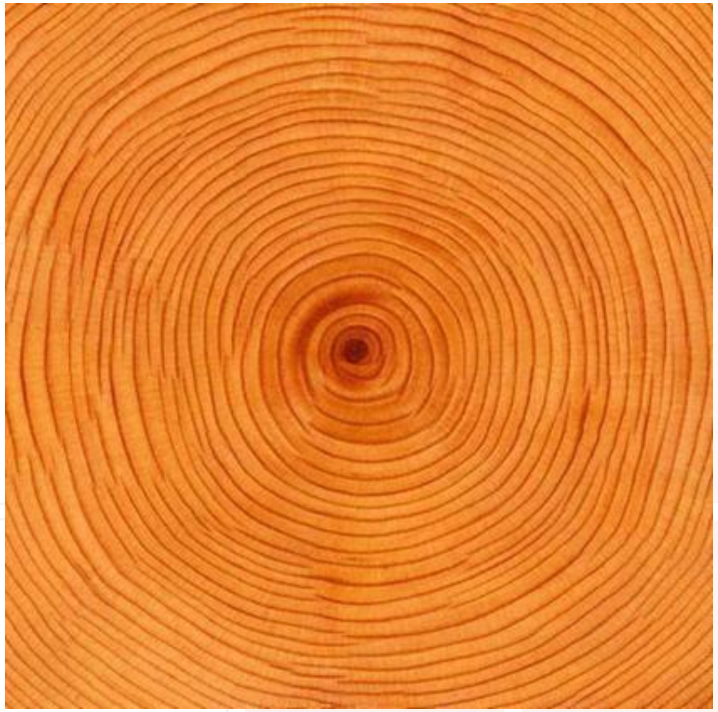
$$d_{ij}^{GC} = \frac{1}{k*k}\sum_{x=0}^{k*k}\left(1 - \frac{|G_t(i)(x)\cdot G_s(j)(x)|}{|G_t(i)(x)|\cdot|G_s(j)(x)|}\right)$$

Source Texture

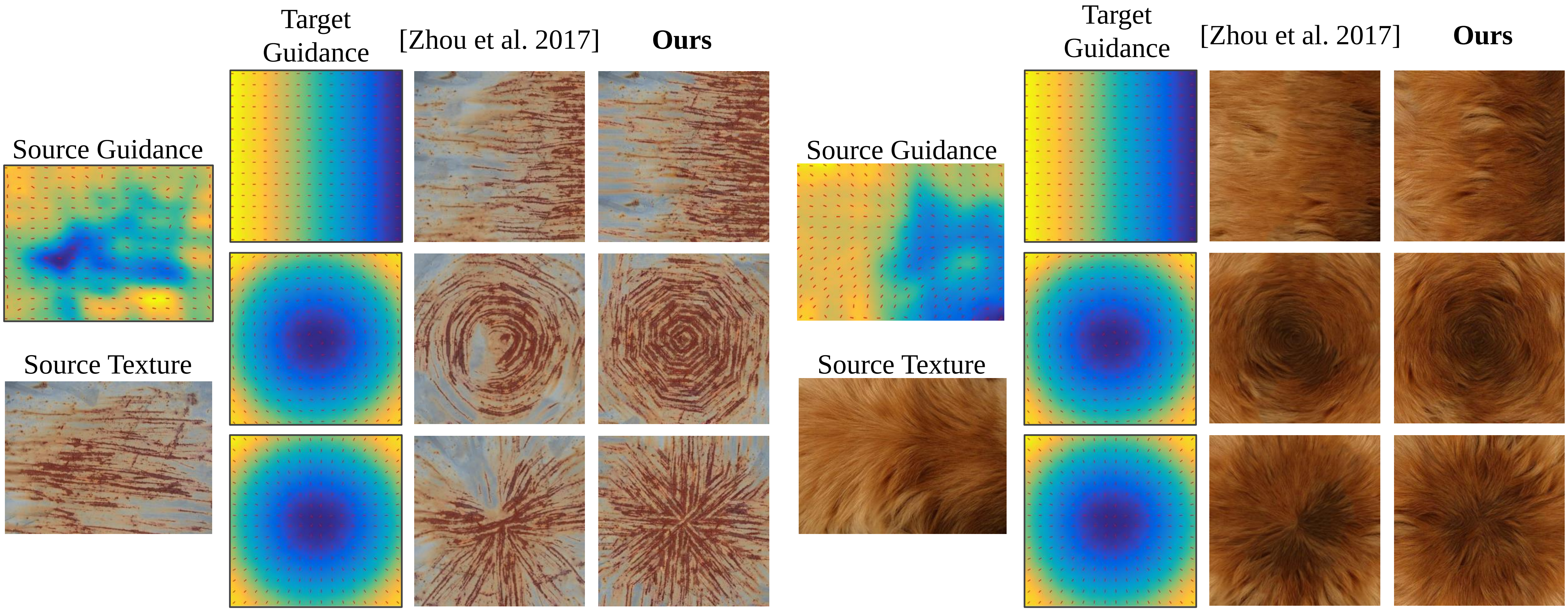
[Zhou et al. 2017]



Ours



Controlled Synthesis: Progression + Orientation



Controlled Synthesis: Progression + Orientation

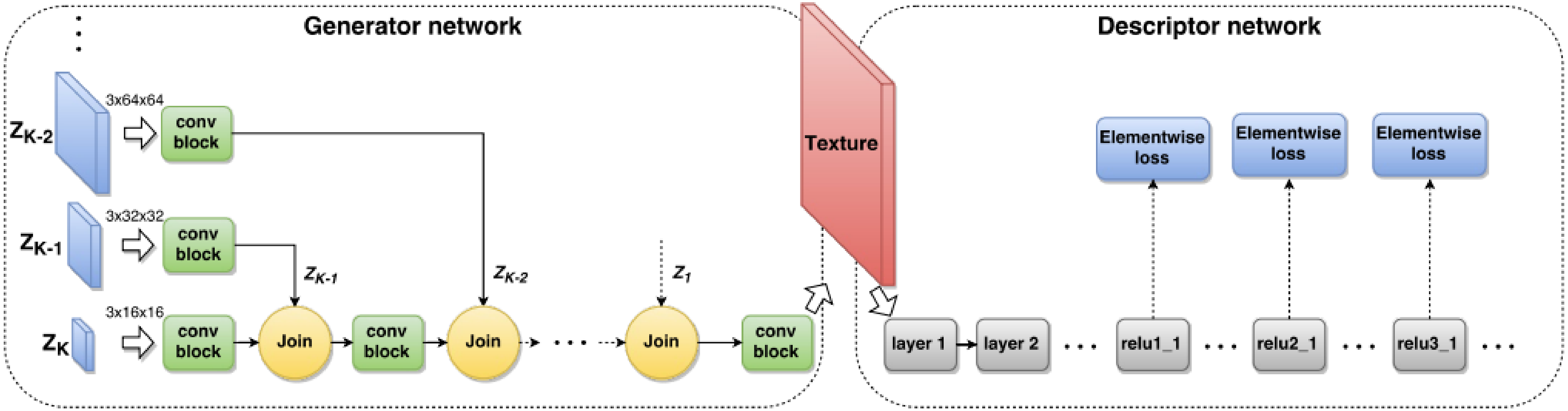
Source Texture

[Zhou et al. 2017]

Ours

Real-time Texture Synthesis

➢ Train TextureNets [Ulyanov et al., 2016] using Guided Correspondence loss.



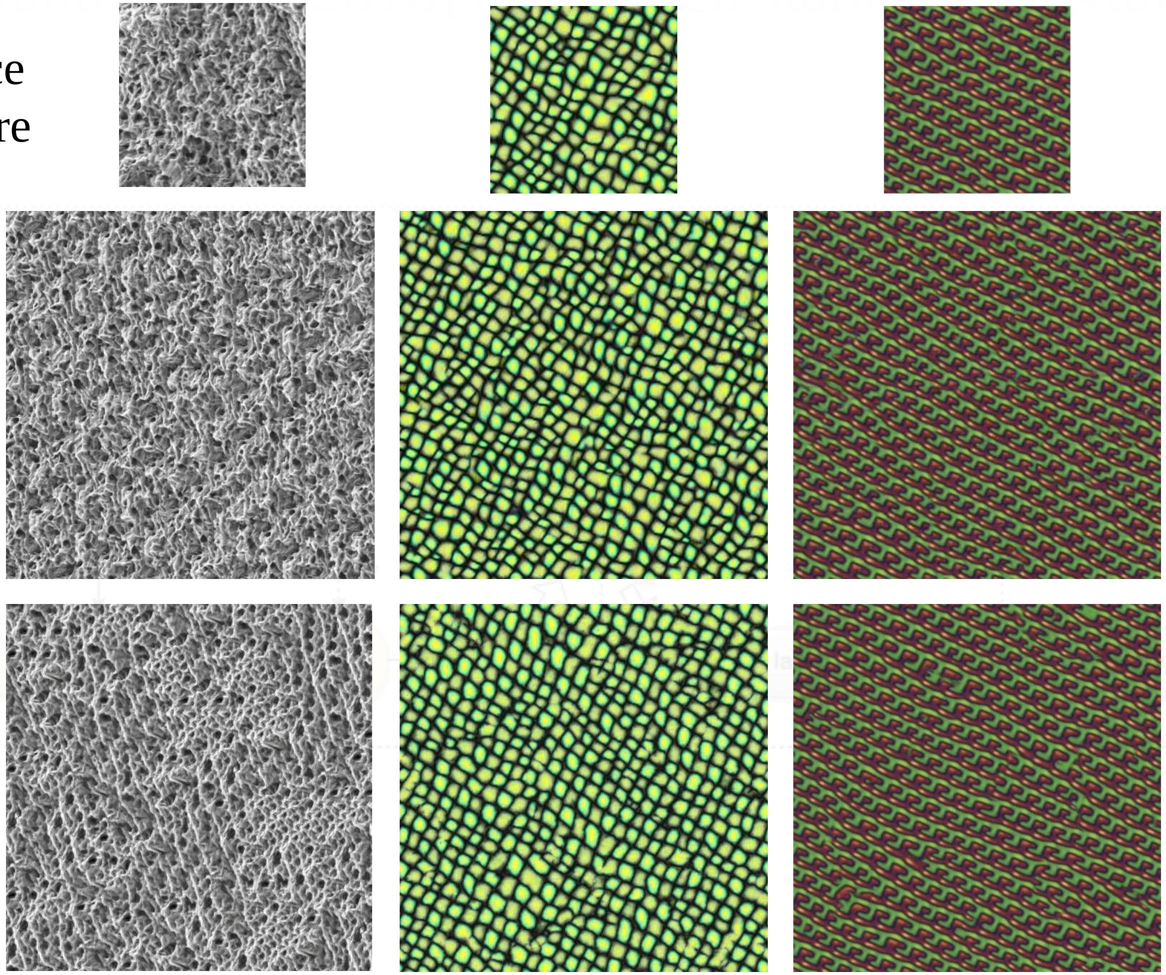
[Ulyanov et al. Texture Networks: Feed-forward Synthesis of Textures and Stylized Images. 2016]

Real-time Texture Synthesis

Source Texture

Sliced Wasserstein

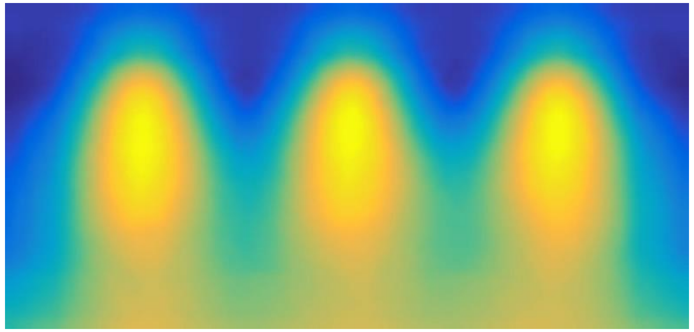
Ours



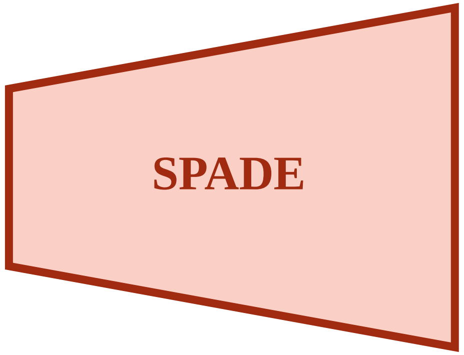
[Ulyanov et al. Texture Networks: Feed-forward Synthesis of Textures and Stylized Images. 2016]

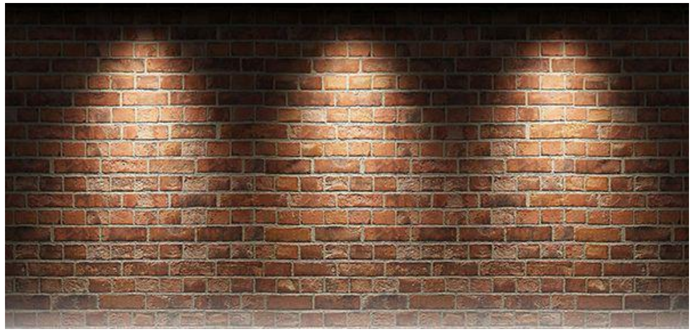
Real-time Controlled Synthesis

➢ Train SPADE [Park et al., 2019] using Guided Correspondence loss.



Source Guidance Map



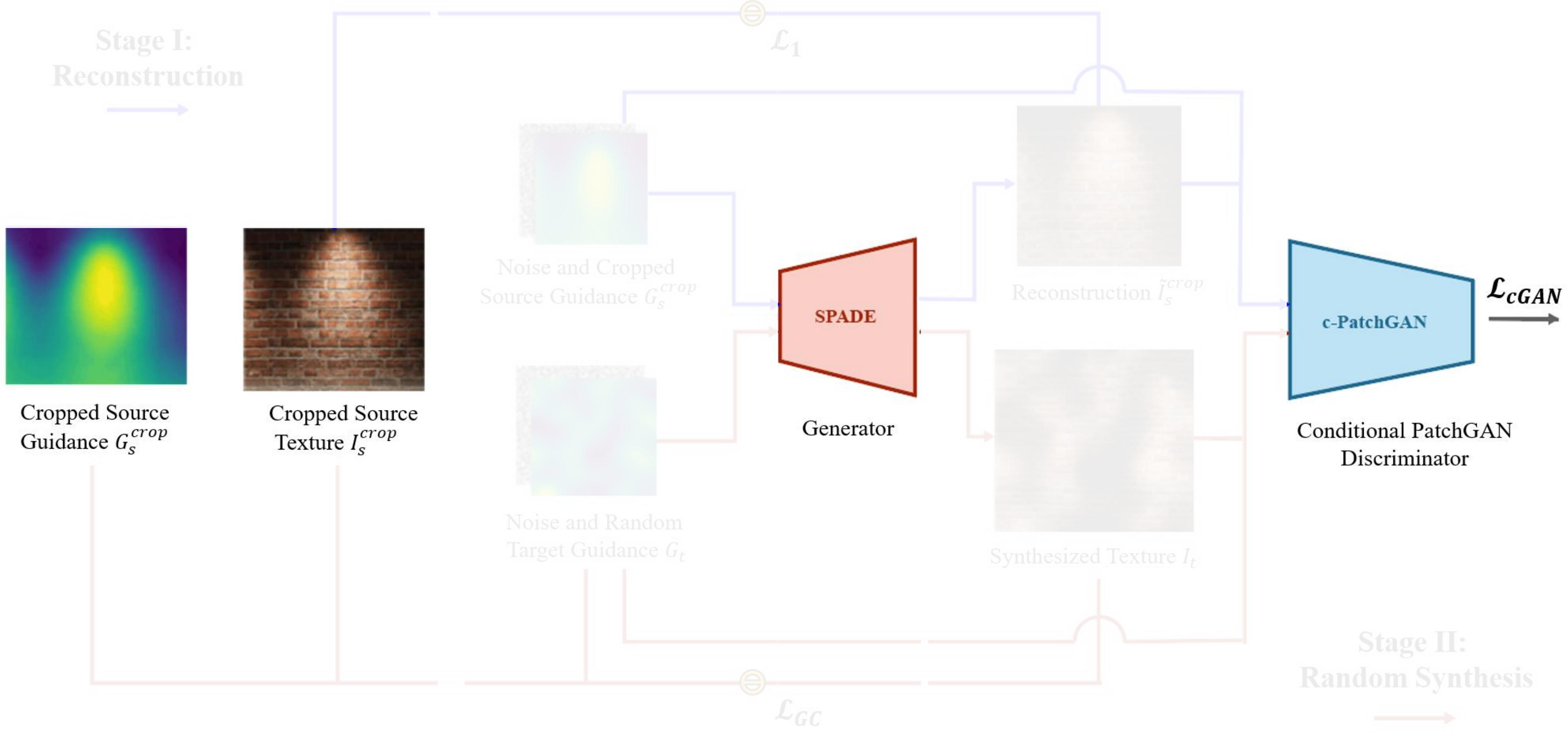


Source Texture

[Park et al. Semantic Image Synthesis with Spatially-Adaptive Normalization. 2019]

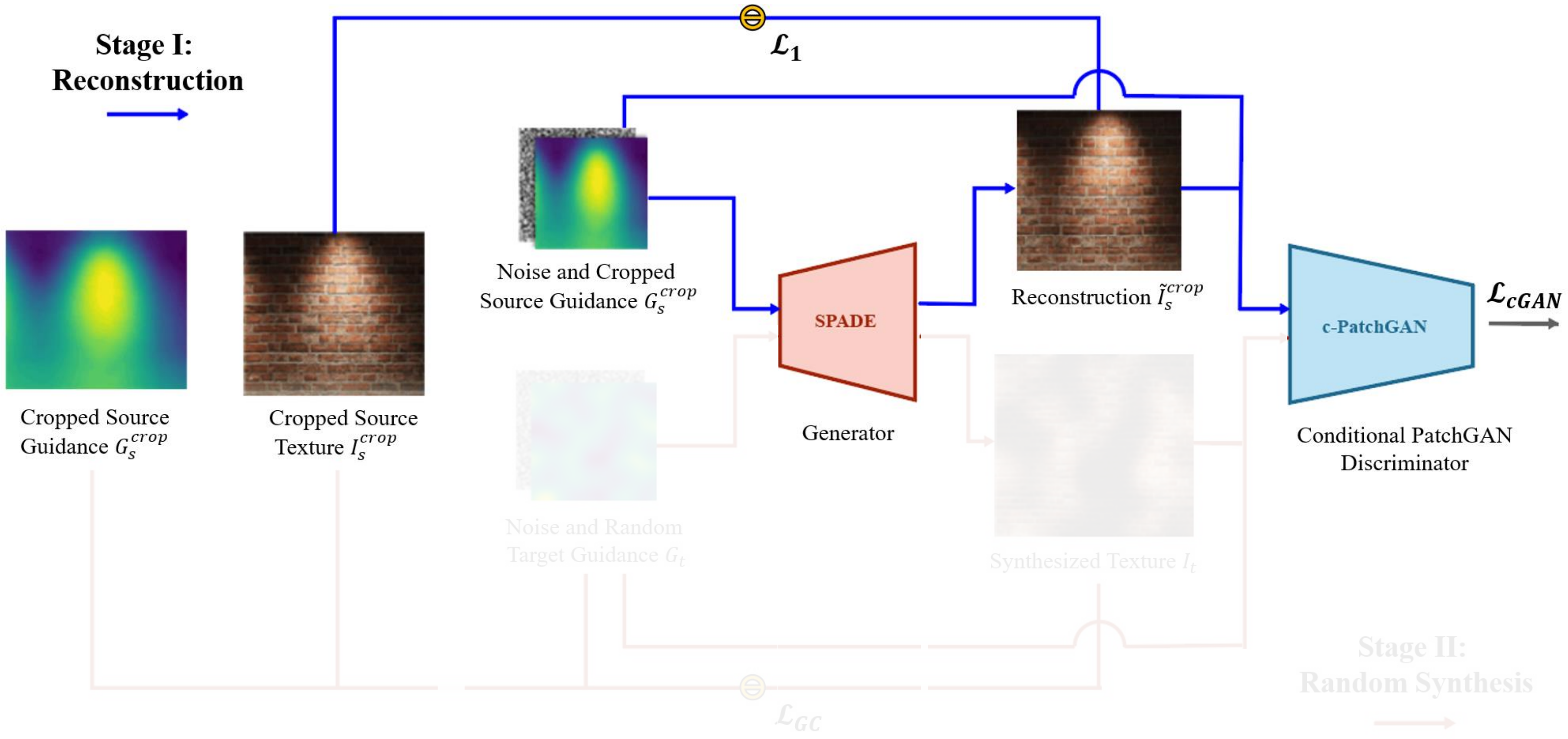
Real-time Controlled Synthesis

- Train SPADE [Park et al., 2019] using Guided Correspondence loss.



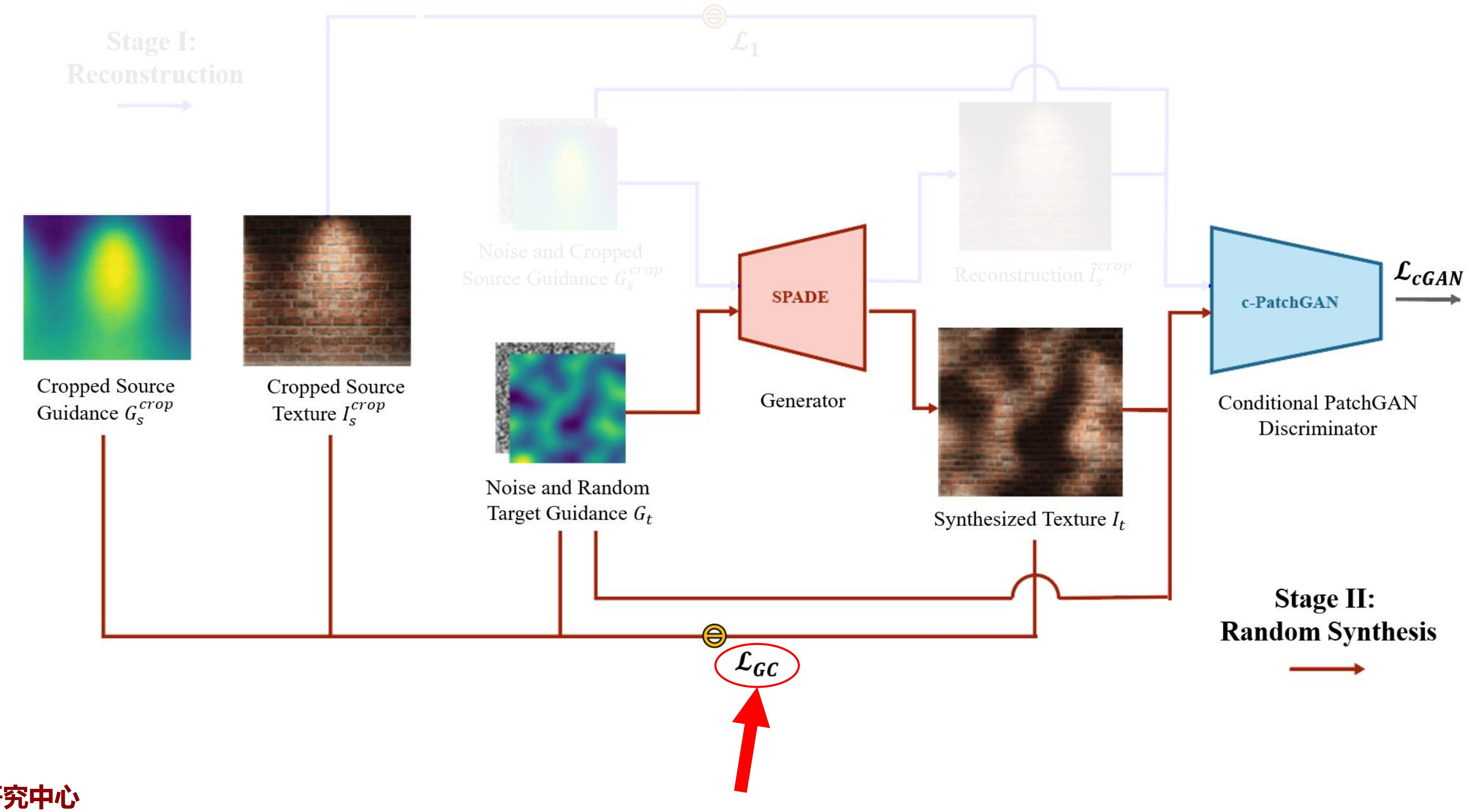
Real-time Controlled Synthesis

➢ Train SPADE [Park et al., 2019] using Guided Correspondence loss.



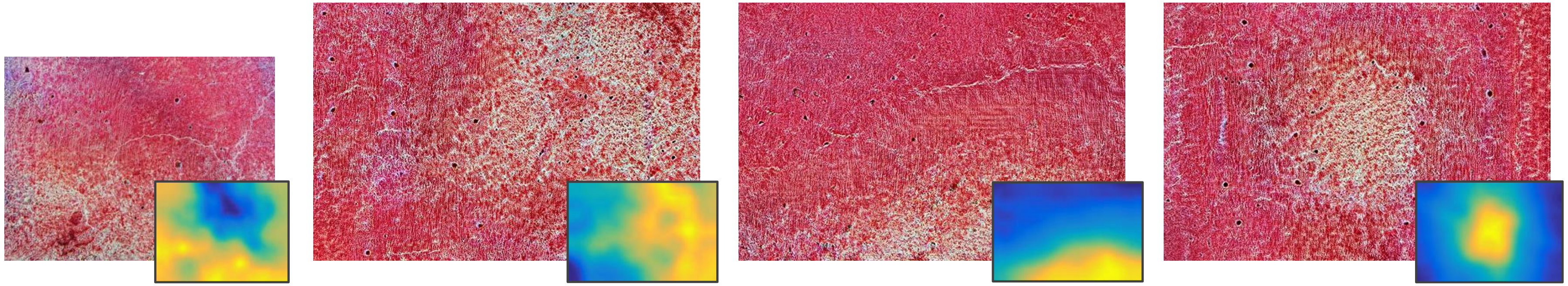
Real-time Controlled Synthesis

- Train SPADE [Park et al., 2019] using Guided Correspondence loss.

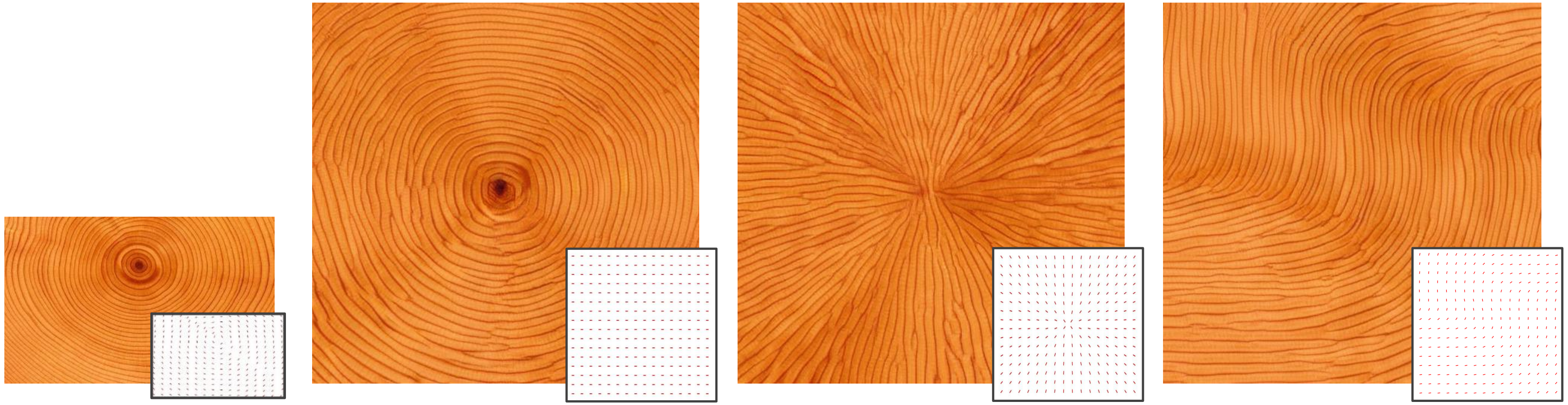


Real-time Controlled Synthesis

➢ Train SPADE [Park et al., 2019] using Guided Correspondence loss.



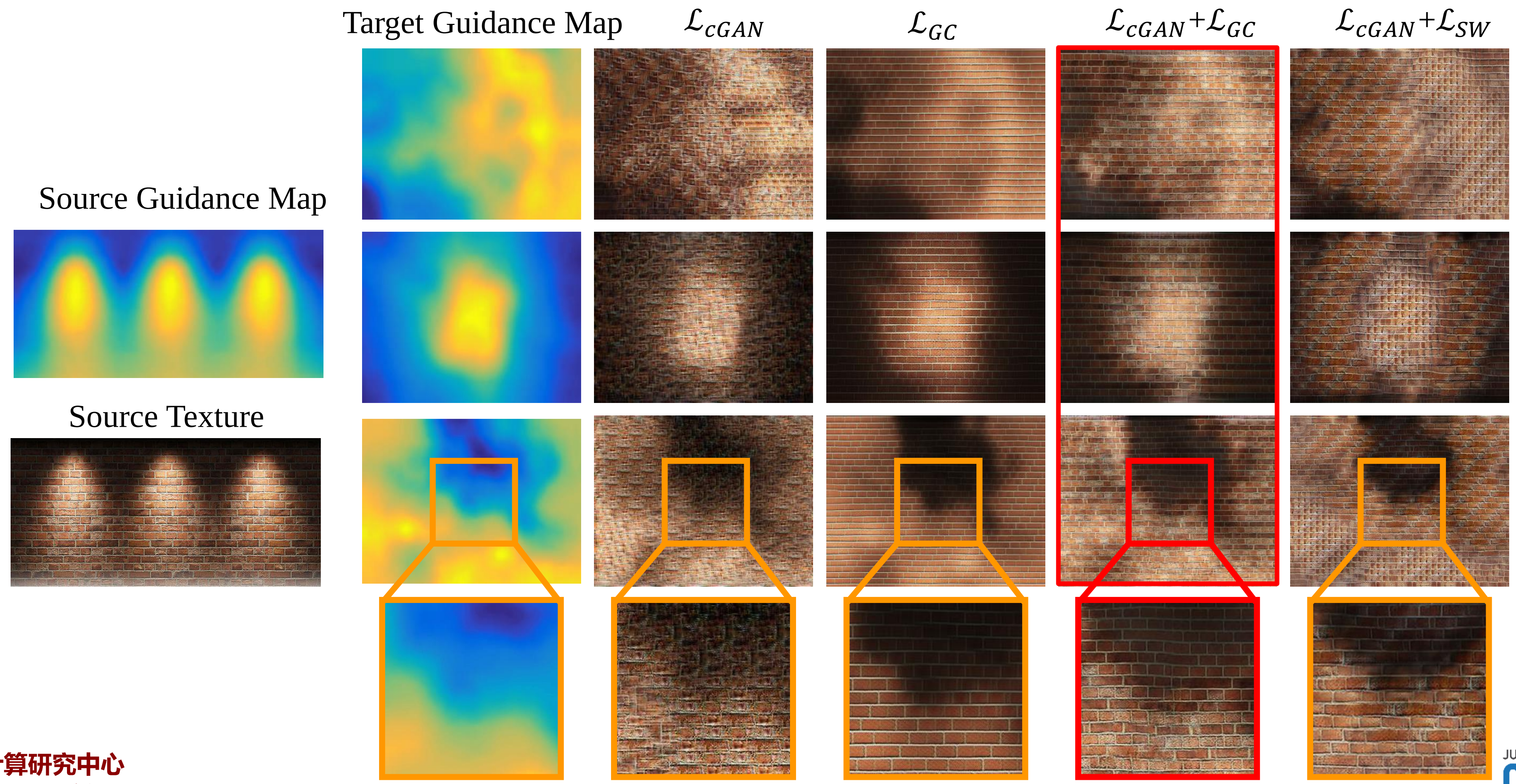
Based on Progression Map



Based on Orientation Map

Real-time Controlled Synthesis

➢ Train SPADE [Park et al., 2019] using Guided Correspondence loss.



Applications

Texture Transfer

- Replace the Gram loss with our Guided Correspondence loss.

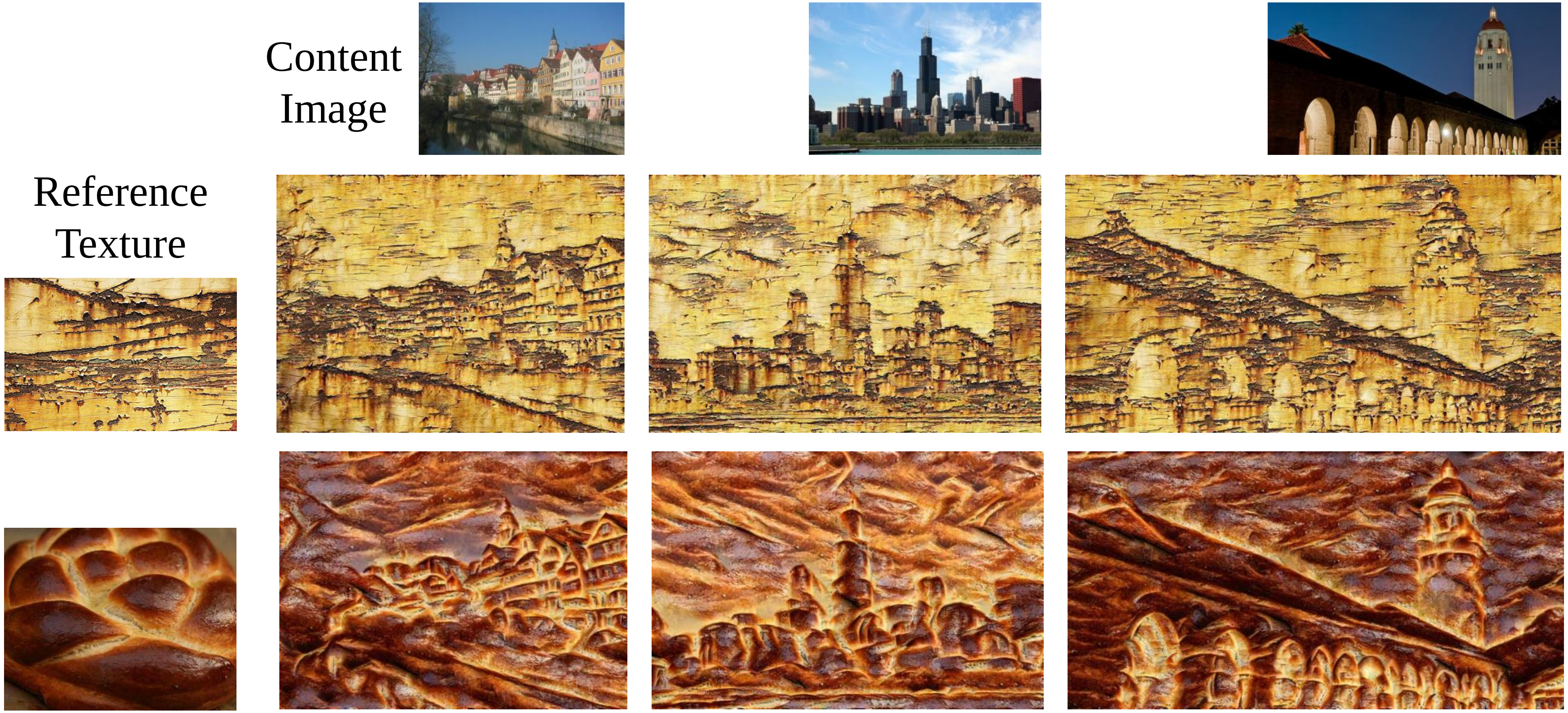
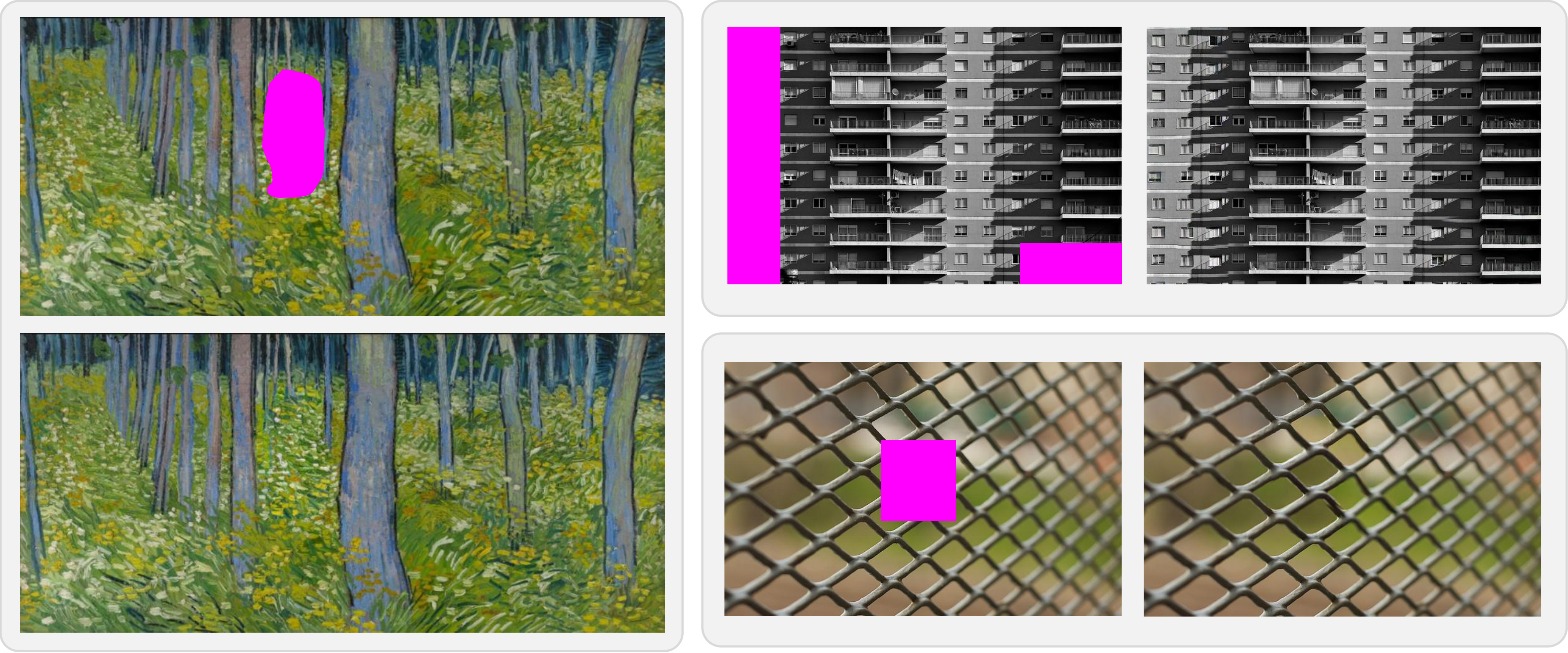


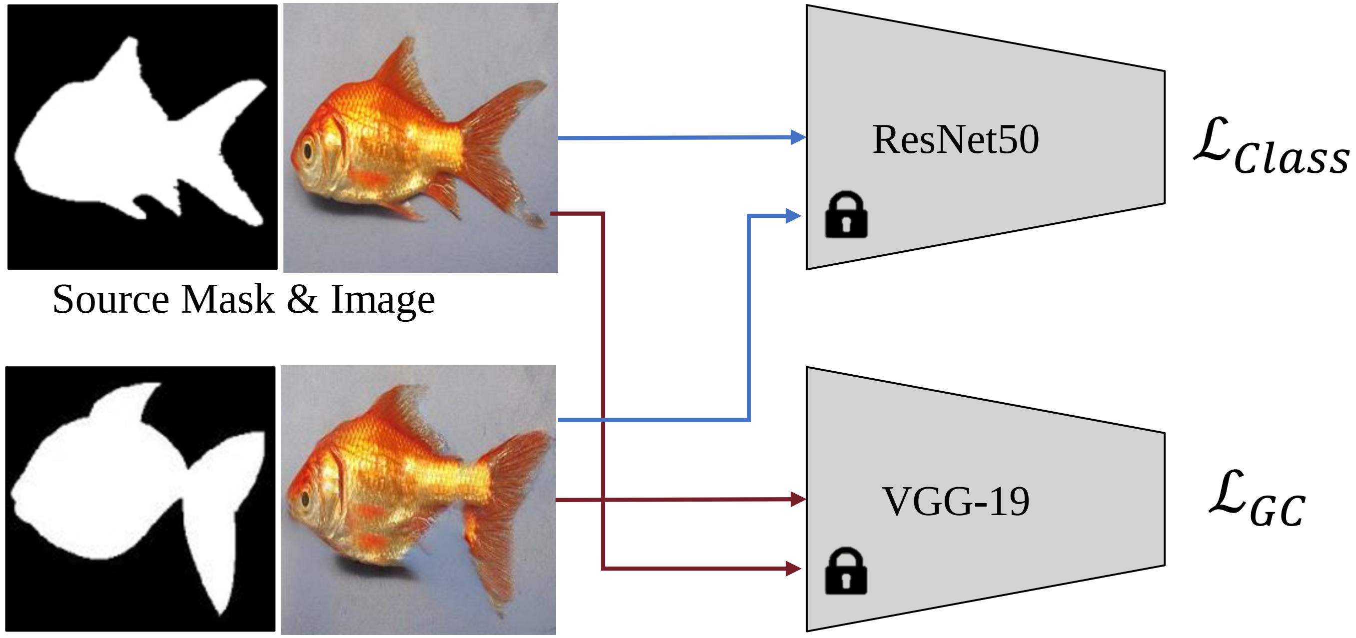
Image Inpainting

➢ Constrain the texture optimization to fill the holes only using source patches from the remaining area of the same image.



Single-image Editing

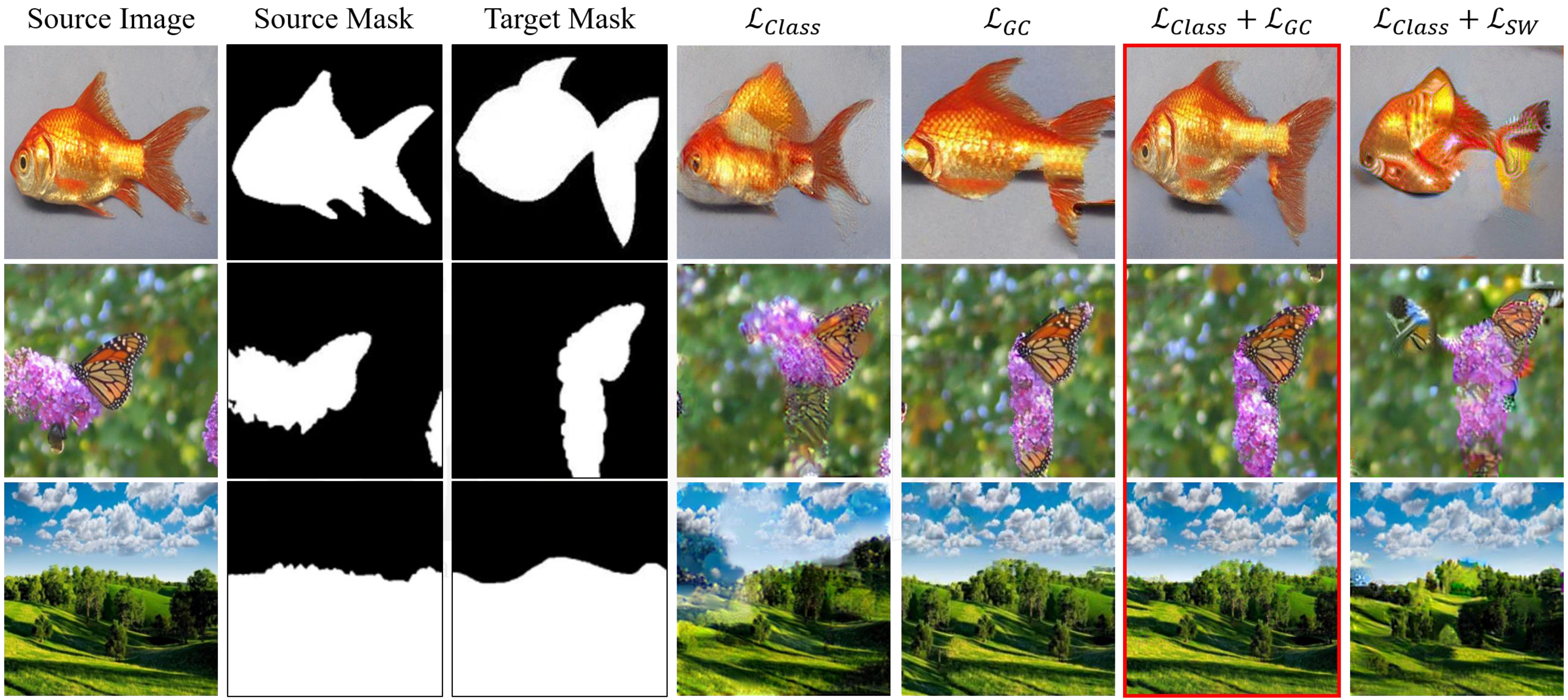
➢ Add the Guided Correspondence loss to the inversion-based image editing.



[Wang et al., IMAGINE: image synthesis by image guided model inversion. 2021]

Single-image Editing

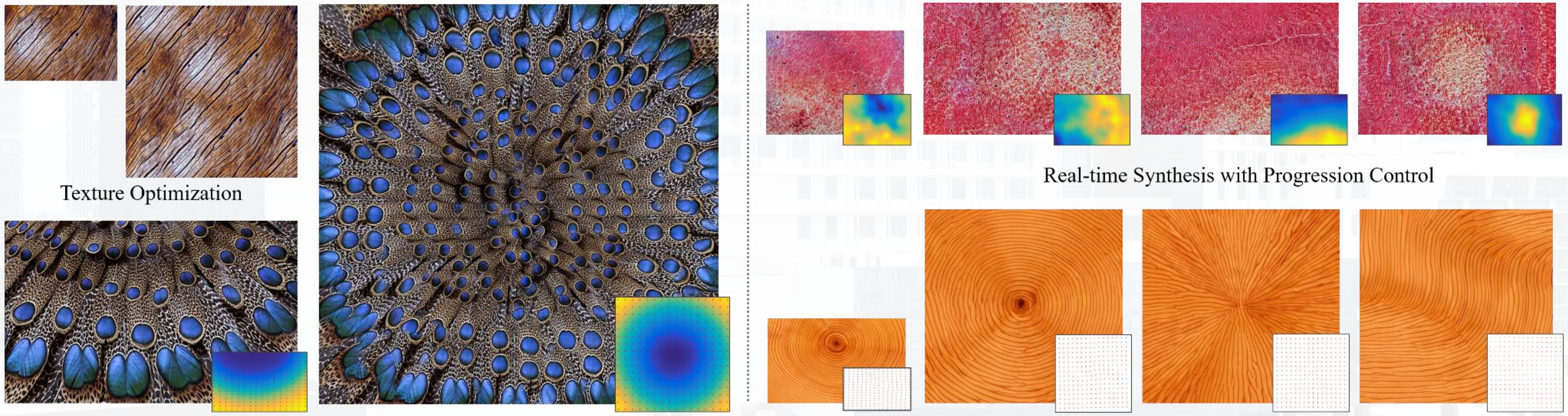
- Add the Guided Correspondence loss to the inversion-based image editing.



[Wang et al., IMAGINE: image synthesis by image guided model inversion. 2021]

Thank you

Neural Texture Synthesis with Guided Correspondence



Texture Optimization

Real-time Synthesis with Progression Control

Controlled Texture Optimization with Guidance Maps

Real-time Synthesis with Orientation Control

https://vcc.tech/research/2023/DeepTex