

# SGSST: Scaling Gaussian Splatting Style Transfer

Bruno Galerne<sup>1,2</sup>, Jianling Wang<sup>1</sup>, Lara Raad<sup>3</sup>, and Jean-Michel Morel<sup>4</sup>

<sup>1</sup>Institut Denis Poisson, Université d'Orléans, Université de Tours, CNRS    <sup>2</sup>Institut Universitaire de France (IUF)

<sup>3</sup>Instituto de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de la República    <sup>4</sup>City University of Hong Kong

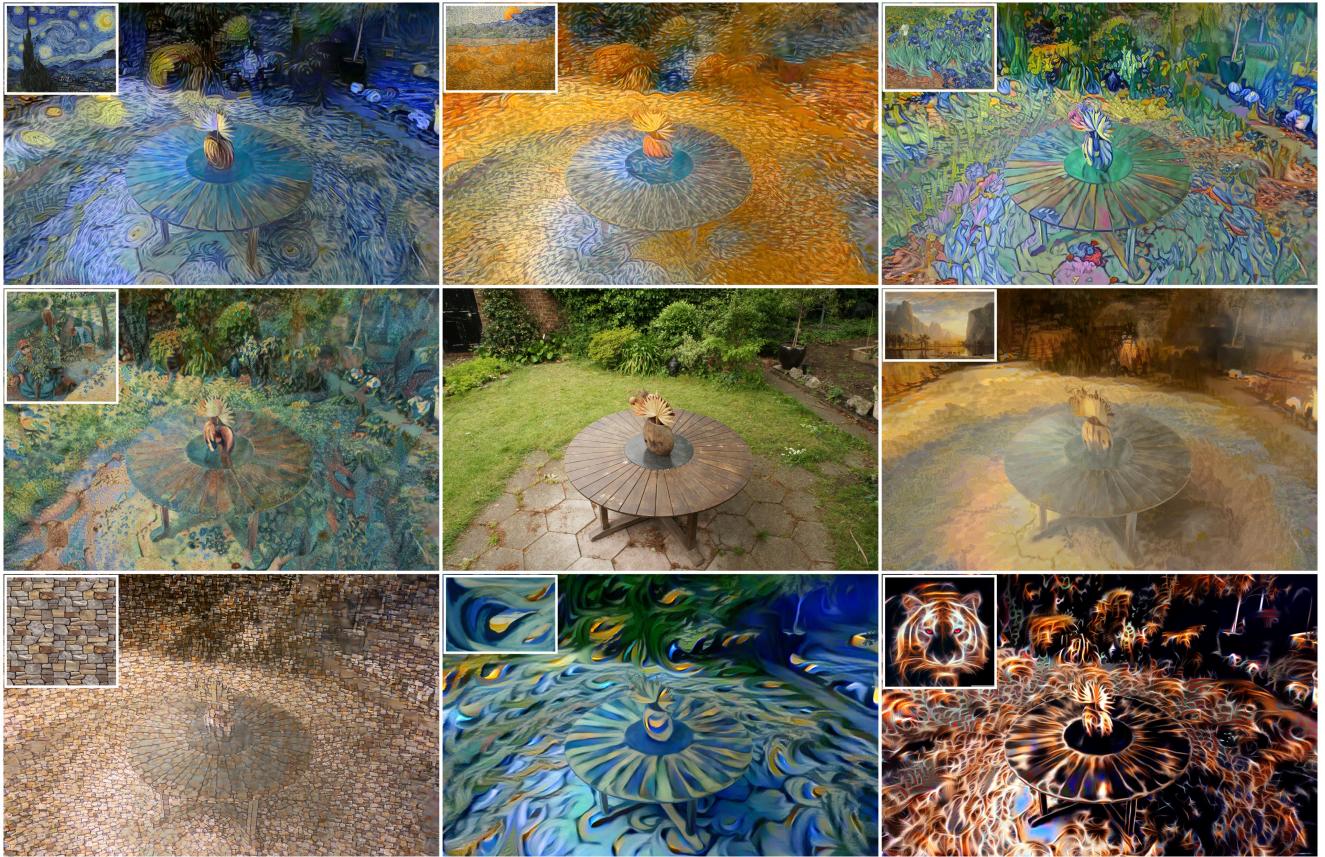


Figure 1. **Various ultra-high definition style transfers of a Gaussian splatting 3D scene.** SGSST transfers a very large set of global style statistics of an image to a 3DGS scene by minimizing a tailored multiscale SOS loss, yielding 3D style transfer of superior quality and at unprecedented high resolution (images have size  $5187 \times 3361$ ).

## Abstract

Applying style transfer to a full 3D environment is a challenging task that has seen many developments since the advent of neural rendering. 3D Gaussian splatting (3DGS) has recently pushed further many limits of neural rendering in terms of training speed and reconstruction quality. This work introduces SGSST: Scaling Gaussian Splatting Style Transfer, an optimization-based method to apply style transfer to pretrained 3DGS scenes. We demonstrate that

a new multiscale loss based on global neural statistics, that we name SOS for Simultaneously Optimized Scales, enables style transfer to ultra-high resolution 3D scenes. Not only SGSST pioneers 3D scene style transfer at such high image resolutions, it also produces superior visual quality as assessed by thorough qualitative, quantitative and perceptual comparisons.

## 1. Introduction

Dealing with ultra-high resolution (UHR) rendering is capital for AR/VR applications. Indeed, when navigating in a 3D environment the user only sees a partial field of view of the environment. This adds challenging issues for applying style transfer to 3D environments, that is, transferring the visual characteristics of an image such as a painting to a 3D scene. Indeed, while the general scene should convey the global color palette of the style painting, when getting closer to objects in a stylized environment the user should expect to see fine painting patterns such as brushstrokes. However, in the current state of the art, the user quickly encounters resolution-limited content in the form of blurry interpolated features.

Ever since the seminal work of Mildenhall *et al.* [28], neural radiance fields (NeRF) have seen many improvements. Several 3D scene representations have been proposed for improving the quality and resolution of the reconstructed scenes as well as easing the training, e.g. [2, 3, 9, 29]. The recent 3D Gaussian splatting (3DGS) [20] has introduced an efficient high-resolution (HR) scene representation which has stirred much interest [45]. Both frameworks have stirred attempts to 3D style transfer algorithms [6, 16, 22, 25, 26, 33, 46, 48], but these methods have so far produced medium resolution outputs. They do not faithfully transport high resolution multiscale textures such as those present in paintings. Motivated by a recent neural style transfer (NST) [11] contribution tailored for UHR images [10], i.e. with resolution larger than 4k images, we show that 3DGS can be leveraged for UHR style transfer.

The contributions of this work are the following:

- We introduce SOS, a Simultaneously Optimized Scales loss expressed in a single parameterless and explainable formula.
- By solely optimizing the SOS loss, we reach UHR for 3DGS style transfer and we scale Gaussian Splatting Style Transfer by a four times resolution gain.
- Superior quality transfer: By transferring a very large set of global style statistics, we obtain superior style transfer quality even at HR resolution, as confirmed by a comparative perceptual study.

In short, our approach is the first method that allows UHR style transfer directly to 3DGS. It produces high visual quality results and relies on optimizing a single multiscale loss. The simplicity of our approach ensures its reproducibility (source code is provided as supplementary material). Being optimization-based, SGSST’s main limitation is a fairly large training time that is two to eight times longer than the initial 3DGS training depending on the image resolution. Yet, considering the high quality of the results and their reproducibility, this contribution is valuable for AR/VR applications that require high visual quality for

their user experience.

## 2. Related work

**Neural style transfer** In the seminal work of Gatys *et al.* [11] NST is formulated as an optimization problem minimizing the distances between Gram matrices of VGG [37] features. Even though other VGG statistics have been considered, almost all subsequent style transfer and texture synthesis methods rely on VGG [8, 13, 14, 27, 32, 35, 42]. To accelerate style transfer, several methods [18, 40, 41] have attempted to train feed-forward networks approximately minimizing the Gram loss [11]. However, these approximate methods require learning a new model for each style type. This latter limitation has been addressed by fast Universal Style Transfer (UST) approaches [4, 7, 15, 23, 24, 30, 36] that use VGG feature decoders.

For HR images, coarse-to-fine multiscale strategies [12, 13, 38] have proved effective, but still face limitations due the high GPU memory usage of VGG statistics. Fast HR alternatives [1, 5, 39, 43, 44] do exist but generally suffer from artifacts and struggle to capture the full style complexity. Recently, SPST [10] proposed an implementation of the Gatys *et al.* method adapted to UHR (larger than 4k) images. The visual quality of SPST’s results is superior, at the cost of a long optimization time.

**Style transfer for neural radiance fields** NeRFs [28] have completely redefined the field of 3D scene modeling and novel view synthesis. Editing the visual aspect of NeRFs via style transfer has quickly been addressed [6, 16, 25, 48], usually by fine tuning a pretrained NeRF representation using a style transfer loss, or training an additional fast style transfer module. ARF [48] is a notable exception: It uses Nearest Neighbor Feature Matching (NNFM) for fine tuning a plenoxel radiance field [9], produces high-quality results at moderate resolution, and is the base model for other methods [19, 47]. While these works paved the way for radiance field style transfer, they are all limited in input and output image resolution.

**Style transfer for Gaussian splatting** A few recent contributions show that 3DGS is a promising framework for 3D scene style transfer. Saroha *et al* [33] propose a solution for universal style transfer of a given Gaussian splatting 3D scene. The method processes the colors of Gaussians with a tiny MLP trained using fast AdaIN [15] and relying on a multi-resolution hash grid representation [29]. StyleGaussian [26] is a concurrent approach that relies on transferring encoding of VGG features to each Gaussian and applying AdaIN to these features. The new Gaussian features are then decoded into an RGB color by processing the K-nearest neighbors of the Gaussians. After training,

both methods allow for instantaneous stylization with any style image, but the visual quality of the results is quite low, since high-resolution details such as brushstrokes patterns are not transferred.

StylizedGS [46] is an optimization based method that extends ARF [48] to 3DGS. It uses a training loss made of six terms combined with a color transfer preprocessing, the loss being designed to minimize changes in the Gaussian splatting geometry while letting the style evolve via VGG NN matching.  $\mathcal{G}$ -Style [22] also is an optimization based approach that further uses a CLIP-based loss [31]. While these approaches produce slightly better results than ARF [48], both methods are limited in content and style resolution due to the use of nearest-neighbor matching of VGG features.

None of the current state of the art deals with HR style transfer. The stylization is either fast but very approximate due to AdaIn [26, 33] or unable to use HR style images due to NNFMD [22, 46, 48]. To the best of our knowledge SGSST is the first procedure allowing high quality transfer for 3DGS that is trained and rendered at UHR.

### 3. Preliminaries

#### 3.1. Gaussian splatting representation

Starting from a multiview training set of images  $\{u_i\}_{i=1}^{N_{\text{views}}}$  of a static scene accompanied with corresponding calibrated cameras  $\{\mathcal{C}_i\}_{i=1}^{N_{\text{views}}}$  computed by structure from motion [34], the 3DGS algorithm [20] trains a set of colored 3D Gaussians  $\{\mathcal{G}_j\}_{j=1}^{N_{\text{Gaussians}}}$  so that they represent the 3D scene from any camera position. Each Gaussian  $\mathcal{G}_j$  is represented by a finite set of features: a center position  $\mu_j$ , a covariance matrix  $\Sigma_j$  (encoded by a scaling diagonal matrix and a rotation matrix), an opacity  $\alpha_j$  and a view dependent color function  $c_j$ . These parameters are used in a volumetric renderer that determines the color by summing the contribution of each Gaussian that intersects a ray with direction  $(\theta, \varphi)$  via  $\alpha$ -blending (see e.g. [28]). The contribution of a Gaussian is its color  $c_j(\theta, \varphi)$  multiplied by an opacity  $\sigma_j$  defined as the maximal opacity  $\alpha_j$  times the unnormalized Gaussian density at the ray position [20]. Thus, the resulting color  $C$  for the ray is

$$C = \sum_{j=1}^N c_j(\theta, \varphi) \sigma_j \prod_{k=1}^{j-1} (1 - \sigma_k) \quad (1)$$

where the sum is over all Gaussians intersecting the ray. The color function  $c_j(\theta, \varphi)$  associated with a Gaussian depends on the spherical direction  $(\theta, \varphi)$  through an order 3 spherical harmonics polynomial function given by

$$c_j(\theta, \varphi) = c_{j,0} + \sum_{\ell=1}^3 \sum_{m=1}^{2\ell+1} c_{j,\ell,m} Y_{\ell,m}(\theta, \varphi) \quad (2)$$

where the vectors  $c_{j,0}$  and  $c_{j,\ell,m}$  are in  $\mathbb{R}^3$  and the  $Y_{\ell,m}$  form a basis of the spherical harmonics polynomials of degree  $\ell$ . In short,  $c_{j,0} \in \mathbb{R}^3$  is the main color and the additional coefficients  $c_{j,\ell,m}$  encode smooth variations of this color when the viewing angle changes.

Like for NeRF, the key ingredient of the 3DGS parametrization is the differentiable rendering function  $\mathcal{R}(\mathcal{C}; \Theta)$  that renders a view of the scene for any camera  $\mathcal{C}$  given the scene parameters  $\Theta = \{(\mu_j, \Sigma_j, \alpha_j, c_{j,0}, (c_{j,\ell,m})_{\ell,m})\}_{j=1}^{N_{\text{Gaussians}}}$ . This differentiable rendering function allows to train the Gaussian parameters to minimize the reconstruction error

$$\min_{\Theta} \frac{1}{N_{\text{views}}} \sum_{i=1}^{N_{\text{views}}} E_{\text{reconstruction}}(\mathcal{R}(\mathcal{C}_i; \Theta); u_i) \quad (3)$$

where  $E_{\text{reconstruction}}$  is a 2D image comparison error (such as a combination of mean square error and SSIM [20]). The minimization is conducted using Adam [21] for 30k iterations by randomly picking one view at each iteration.

#### 3.2. Style transfer loss for UHR images

Our approach relies on optimizing VGG19 [37] feature statistics as initially proposed by Gatys *et al.* [11]. Content consistency is loosely monitored by preservation of the feature layer  $L_c = \text{ReLU\_4\_2}$  while style transfer is imposed by matching spatial statistics of five VGG19 layers, namely the set  $\mathcal{L}_s = \{\text{ReLU\_k\_1}, k \in \{1, 2, 3, 4, 5\}\}$ . The statistics of interest of the feature response  $V^L(w)$  of an image  $w$  at some layer  $L \in \mathcal{L}_s$  having  $n_c^L$  feature channels are its Gram matrix  $\text{Gram}(V^L(w)) \in \mathbb{R}^{n_c^L \times n_c^L}$ , its spatial mean vector  $\text{mean}(V^L(w)) \in \mathbb{R}^{n_c^L}$ , and its standard deviation vector  $\text{std}(V^L(w)) \in \mathbb{R}^{n_c^L}$ . Given a content image  $u$  and a style image  $v$ , we consider the loss function

$$E_{\text{transfer}}(x; u, v) = E_{\text{content}}(x; u) + E_{\text{style}}(x; v) \quad (4)$$

where  $L_{\text{content}}(x; u) = \lambda_c \|V^{L_c}(x) - V^{L_c}(u)\|^2$ , with  $\lambda_c > 0$  and the style loss is defined by

$$E_{\text{style}}(x; v) = \sum_{L \in \mathcal{L}_s} E_{\text{style}}^L(x; v). \quad (5)$$

where  $E_{\text{style}}^L(x; v)$  is a linear combination of the mean square error between the VGG19 statistics of  $V^L(x)$  and the one of the style features  $V^L(v)$  [10]. While only the Gram matrices were originally used [11], it has been shown that adding control for the mean and standard deviation corrects some style transfer color artefacts [10] previously identified in the litterature [14, 32, 35].

To obtain high-quality style transfer for HR images one needs to optimize the style transfer loss using several scales and a coarse-to-fine approach [12]. Indeed, if one applies

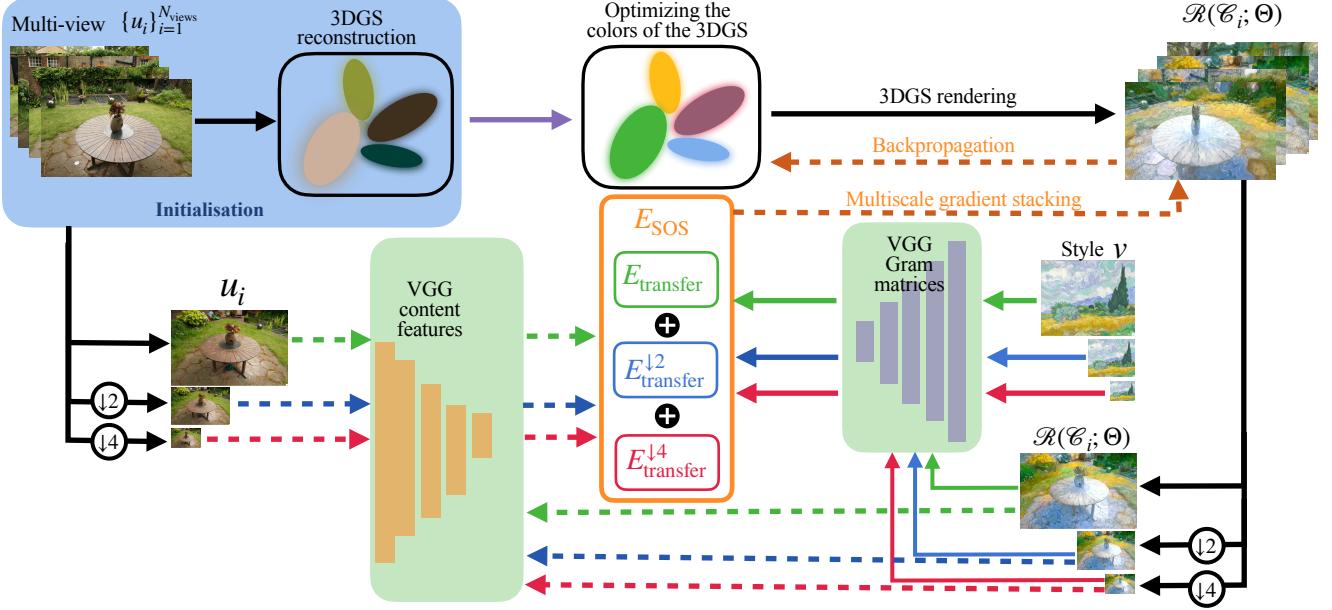


Figure 2. **Overview of SGSST.** Starting from a pretrained realistic 3DGS scene [20], we optimize the colors of each Gaussian using the new multiscale SOS loss (involving  $n_s = 3$  scales in the illustration). Computing the gradient wrt the loss is feasible for UHR images thanks to the SPST partition-based implementation [10]. Multiscale gradient stacking is used at the node of the rendered image to perform only one backpropagation per iteration through the 3DGS rendering pipeline.

style transfer on the highest resolution only, the changes within the content image are limited to local texture and the results does not convey a painting aspect. Due to the use of VGG19 features, computing the gradient of  $E_{\text{transfer}}(x; u, v)$  with respect to (wrt)  $x$  is a priori memory prohibiting for UHR images, however SPST [10] provides a constant memory algorithm for evaluating this gradient via a grid partition.

#### 4. Scaling Gaussian splatting style transfer

A complete overview of our SGSST algorithm is given in Figure 2. Starting from a realistic 3DGS representation [20], we optimize the colors of each Gaussian using a multiscale style transfer loss.

##### 4.1. Stylizing the 3DGS representation

By changing the reconstruction loss of Equation (3) with a style transfer loss for the input style image  $v$ , one can hope to stylize a realistic 3DGS. However, given the complexity of the 3DGS representation and the many interacting parameters, it is not such an easy task to alter the 3DGS aspects without loosing the content geometry. Our solution is to only optimize for the constant color components  $\Theta_{\text{color}} = \{c_{j,0}\}_{j=1}^{N_{\text{Gaussians}}}$  of the Gaussians and simply freeze all the other parameters  $\Theta_{\text{init}}$  given by the initial realistic 3DGS training.

We experimentally found that this robust approach en-

sures a rich style transfer and fully preserves the 3D geometry. Indeed, fixing all the Gaussian parameters except for the main color component  $c_{j,0}$  preserves the scene geometry, as the location and size of the Gaussians are being kept (see Section 5.3 for ablation experiments).

##### 4.2. Multiscale Style Transfer Loss

We introduce the Simultaneously Optimized Scales (SOS) loss defined as

$$E_{\text{SOS}}(x; u, v) = \frac{1}{n_s} \sum_{s=0}^{n_s-1} E_{\text{transfer}}(x^{\downarrow 2^s}; u^{\downarrow 2^s}, v^{\downarrow 2^s}) \quad (6)$$

where  $n_s \geq 1$  is the number of considered scales and  $u^{\downarrow 2^s}$  denotes an image  $u$  downsampled by a  $2^s$  factor. This SOS loss (6) enables style transfer simultaneously at all scales. A somewhat similar approach was proposed for 2D texture synthesis [38] but, as already mentioned, multiscale 2D style transfer is generally conducted using a coarse-to-fine strategy [12]. However, for 3DGS we observed that the initial configuration had only little influence on the final result, making the coarse-to-fine strategy ineffective for multiscale style transfer (See Section 5.3).

In the end, the stylization of UHR 3DGS is conducted by solving for

$$\min_{\Theta_{\text{color}}} \frac{1}{N_{\text{views}}} \sum_{i=1}^{N_{\text{views}}} E_{\text{SOS}}(\mathcal{R}(C_i; \Theta); u_i, v), \quad (7)$$

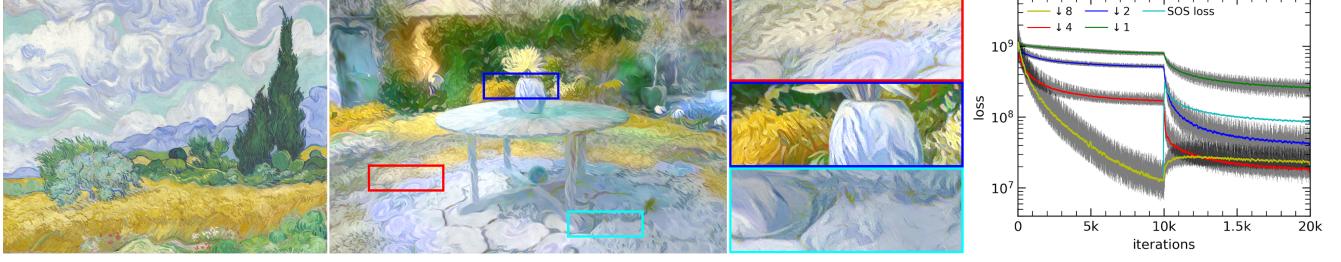


Figure 3. **UHR 3DGS style transfer.** SGSST allows for the multiscale style transfer of 3DGS scenes at UHR. From left to right: Style image, one UHR stylized view, three magnified details, and evolution of the SOS loss and each style transfer loss that contributes to it. We first optimize the transfer loss for the coarsest scale (yellow curve) for 10k iterations and then optimize for another 10k iterations the SOS loss (light blue curve), namely the mean of the four transfer losses. Images sizes are  $5187 \times 3361$  for content and  $4230 \times 3361$  for style.

where  $\Theta$  stands for the 3DGS parameters obtained by replacing the color components of  $\Theta_{\text{init}}$  by the values of the optimization variable  $\Theta_{\text{color}}$ .

#### 4.3. Implementation details

**Multiscale gradient stacking** The 2D SPST method [10] provides the gradient of each term  $E_{\text{transfer}}(x^{\downarrow 2^s}; u^{\downarrow 2^s}, v^{\downarrow 2^s})$  wrt to the input  $x^{\downarrow 2^s} = \mathcal{R}(\mathcal{C}_i; \Theta)^{\downarrow 2^s}$ . We first backpropagate each of these gradients through the downscaling operator and stack them at the level of the rendered image  $\mathcal{R}(\mathcal{C}_i; \Theta)$ . When each of the  $n_s$  scales has been treated, the stacked gradient is equal to the gradient of the full loss  $E_{\text{SOS}}(\mathcal{R}(\mathcal{C}_i; \Theta); u_i, v)$  wrt to the rendered image  $\mathcal{R}(\mathcal{C}_i; \Theta)$ . This gradient is then backpropagated through the Gaussian rendering pipeline to obtain the gradient wrt the Gaussian colors  $\Theta_{\text{color}}$  (see both orange arrows in Figure 2). Using this strategy we only backpropagate one time per iteration through the 3DGS rendering pipeline instead of  $n_s$  times.

#### Color transfer via style transfer at the coarsest scale

To speed up the color transfer, we first optimize for 10k Adam iterations with the loss restricted to the coarsest scale  $E_{\text{transfer}}(\mathcal{R}(\mathcal{C}_i; \Theta)^{\downarrow 2^s}; u_i^{\downarrow 2^s}, v^{\downarrow 2^s})$  with  $s = n_s - 1$ , then optimize the SOS loss for another 10k Adam iterations.

**Number of scales** The number of scales is chosen between 1 and 4 depending on the size of the original images (we always consider images with sides larger than 224 for VGG19 statistics to be reliable).

**Reproducibility** All our experiments have been conducted using the same SOS loss and training procedure, making our approach parameterless and fully reproducible. Our implementation is based on the public source codes<sup>1</sup>

<sup>1</sup><https://github.com/graphdeco-inria/gaussian-splatting>; [https://github.com/bgalerne/scaling\\_painting\\_style\\_transfer](https://github.com/bgalerne/scaling_painting_style_transfer)

for 3DGS [20] training and SPST [10]. Our PyTorch source code is available at <https://github.com/JianlingWANG2021/SGSST>.

## 5. Experiments

### 5.1. Ultra-high resolution results

Our multiscale stylization algorithm is able to transfer style details at UHR for both the content image resolution and the style image resolution. This results in unprecedentedly rich style transfer, as illustrated by Figure 3. As can be observed, minimizing the SOS loss indeed allows to decrease the style transfer loss for all scales (Figure 3 right). The approach is especially relevant when transferring the style of an UHR painting presenting style features at several scales, ranging from a specific color palette to a main curve style and to fine brushstrokes or canvas texture (see close-up views of Figure 3 and the first two lines of Figure 1). In addition, the approach is also efficient to transfer the style of a medium resolution style image to an UHR scene, as shown in the last row of Figure 1.

To the best of our knowledge, our approach is the first to work at UHR resolution for neural style transfer of neurally rendered 3D scenes. One of the main advantages of the Gram-based loss is that it does not depend on the style resolution and, when both the content and style images grow in  $O(N)$ , its complexity scales in  $O(N)$  while NNFM used in ARF [48] scales in  $O(N^2)$ .

Yet, applying style transfer at such resolutions remains computationally heavy: for the garden image of size  $5187 \times 3361$  (Figures 1 and 3) the style transfer takes 25.5 hours (VS 3 hours to train the initial 3DGS model [20]), that is an 8.5 overhead factor. For an image of moderate size (Figure 4 left and middle) the style transfer and the initial 3DGS training take respectively 22 minutes and 10 minutes, that is, the overhead factor is only 2.2. These time values were obtained using a single A100 GPU with 80 GB of memory. This could be accelerated by adapting the SOS loss implementation to a multi-GPU setting.

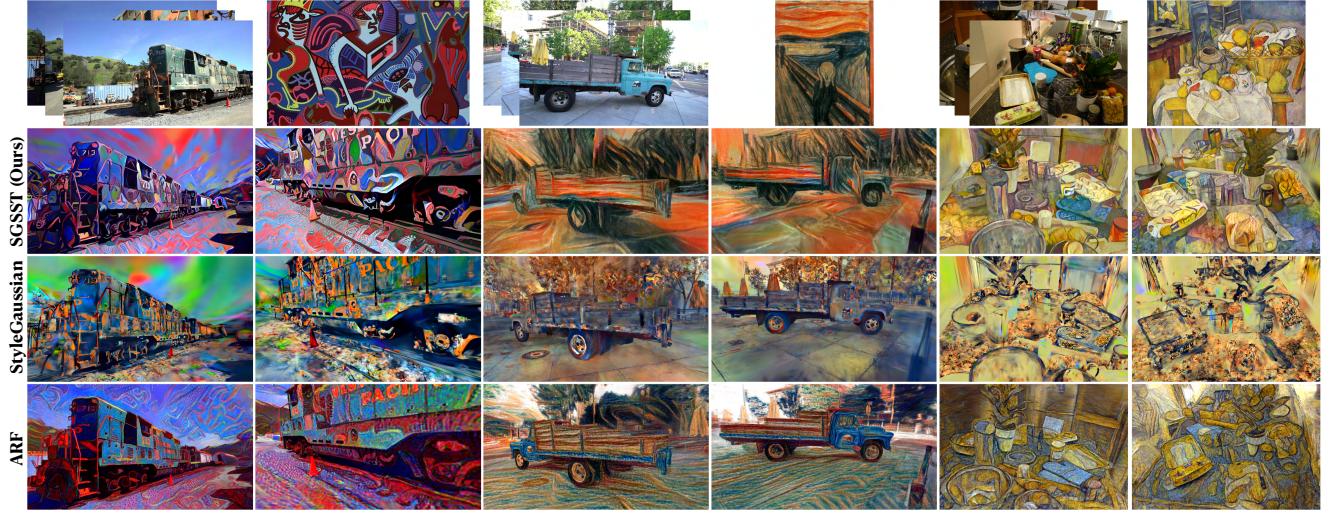


Figure 4. **Comparison of SGSST (ours, top) with StyleGaussian [26] (middle) and ARF [48] (bottom).** From left to right the content resolutions are  $980 \times 545$  (train),  $979 \times 546$  (truck), and  $3115 \times 2076$  (counter). For the first two examples, the various outputs keep the resolution of the content, but for the HR counter scene, the output sizes are  $3115 \times 2076$  for SGSST,  $1600 \times 1066$  for StyleGaussian and  $779 \times 519$  for ARF (see supp. mat. for ARF results without downscaling). Thanks to its multiscale global VGG statistics, SGSST is the most faithful method regarding style consistency.

## 5.2. Comparison

We perform a thorough comparative study using 40 3D style transfer experiments using 9 different scenes and various style images (see supp. mat.). We compare our results with the NeRF-based ARF [48] and the 3DGS-based StyleGaussian [26] algorithms, described in Section 2, using their public implementations<sup>2</sup>.

Comparing style transfer methods is challenging because each algorithm treats the style image differently. Following high quality 2D style transfer [10, 12], our loss uses up to four scales and each scale uses five VGG layers. The UHR style images were downsampled so that the style has the same size as the content images (no upscale was applied if the style image is smaller than the content image). In contrast, ARF uses a single VGG layer from a single scale and the style image is downsampled to be twice smaller than the content input, resulting in smaller local texture. StyleGaussian reduces the style images so that it has the size  $256 \times 256$ , a scale that is hardly sufficient to represent HR style images such as paintings.

**Qualitative comparison** Figure 4 shows three different comparative experiments. As one can observe, the results of StyleGaussian [26] are generally not satisfying since the method fails to transfer local texture or to preserve the style image’s color palette. ARF results better reproduce brush-stroke textures, but the transfer is limited, as the method

only involves a single VGG layer at a single scale. Also, NNFMD does not ensure the preservation of a global color palette. In comparison, SGSST preserves the style at all considered scales. This results in color palette preservation, as well as a verifiable transfer of features of any size, from large brushstrokes (Figure 4 middle) to local grain transfer (Figure 4 right). In addition, SGSST is the only method that performs style transfer at the original resolution of the HR example of Figure 4 right.

**Quantitative comparison** Even though there is no consensus for the quantitative evaluation of NST [17], following previous works, we report two different metrics for style transfer quality and texture consistency across views. Style transfer quality can be measured by the Gram loss [11]. A second metric proper to NeRF style transfer [25, 26, 33] is to check the short-term and long-term consistency of the radiance fields in terms of LPIPS [49] and RMSE between wrapped views. Since SGSST is the only method working with UHR images, when necessary we forcefully downgraded all the results to the resolution of ARF for comparison. The average of these two metrics over our 40 experiments is reported in Table 1. The Gram loss is the best for SGSST and, surprisingly, StyleGaussian achieves a lower Gram loss than ARF, which is not consistent with the qualitative evaluation, probably explained by ARF using use of a single VGG layer compared to five for the two other methods. In terms of consistency metrics, StyleGaussian reports to be the most stable approach followed by SGSST but note that this metric favors the lack of texture.

<sup>2</sup><https://github.com/Kai-46/ARF-svox2>; <https://github.com/Kunhao-Liu/StyleGaussian>

Method	Transfer quality	Short-range consistency		Long-range consistency	
		Gram $\downarrow$	LPIPS $\downarrow$	RMSE $\downarrow$	LPIPS $\downarrow$
SGSST	<b>2.59e8</b>	<b>0.030</b>	<u>0.032</u>	0.055	0.063
StyleGaussian	<u>4.61e8</u>	<u>0.033</u>	<b>0.029</b>	<b>0.050</b>	<b>0.056</b>
ARF	5.77e8	0.040	0.037	0.072	0.066

Table 1. **Quantitative comparison.** Style transfer quality and texture consistency metrics averaged over 40 experiments for SGSST, StyleGaussian, and ARF. Best results are in bold, second best underlined.

	SGSST	ARF	StyleGaussian
Voting results (%)	<b>66.3</b>	<u>31.6</u>	2.1

Table 2. **Perceptual study.** Summary of the 680 votes for the most style consistent result.

**Perceptual study** To further support our results, we conducted a perceptual study comparing the 40 3D style transfer experiments. For each example, four images were displayed (at a resolution of  $1280 \times 720$ ): the style image and, in a blind random order, the results of the three methods (SGSST, ARF and StyleGaussian) shown at a common viewpoint (also randomly selected). Each participant was shown ten random instances and was asked to select the result that was the most faithful to the style image. The study was conducted on the web with volunteer participants.

A total of 68 participants took part in the test, resulting in 680 votes summarized in Table 2. This perceptual study shows that the fast style transfer performed by StyleGaussian is consistently judged inferior in quality over ARF and SGSST. It also confirms that SGSST is far superior to ARF in terms of visual quality since 66% of the participants ranked our method first for its painting style transfer quality, even when presented with results downsampled in resolution.

### 5.3. Ablation study

**Influence of optimization parameters** Our SGSST algorithm stylizes a realistic 3DGS scene by tuning the constant color components of the 3DGS Gaussians using a single 2D style transfer loss function at multiple scales (Equation (6)). In contrast, Zhang *et al.* [46] optimize all 3DGS parameters. This, however, necessitates a complex loss made of six different terms to avoid artefacts: a loss term enforces consistency with the original geometry via depth preservation while a preprocessing of Gaussians floaters is necessary after color transfer.

Figure 5 shows that optimizing more parameters of the 3DGS for SGSST brings no benefit. Optimizing all the spherical harmonic coefficients does not improve the result, and letting all the parameters free like in [22, 46] leads to

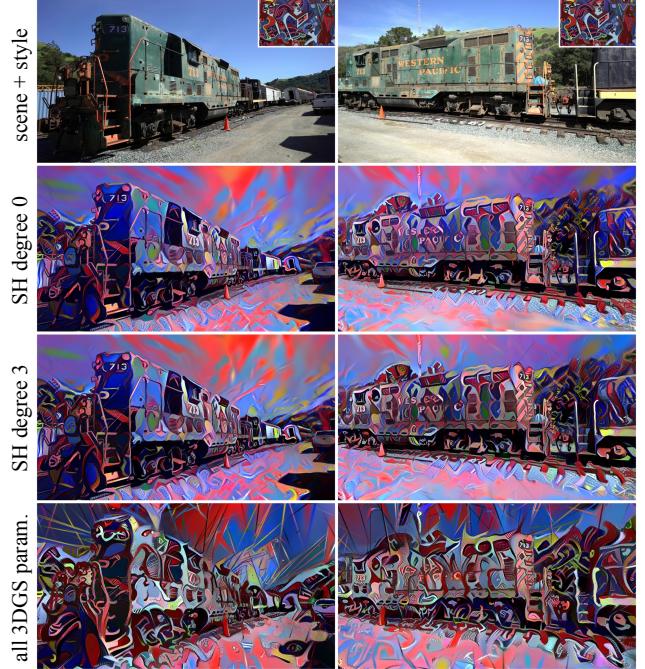


Figure 5. **Influence of optimization parameters:** Allowing more 3DGS parameters to be optimized when minimizing our SOS loss does not improve the stylization quality and can dramatically impact the geometry. From left to right: Style image, Content, SGSST default (optimization of colors), results when optimizing all spherical harmonics, results when optimizing all the parameters.

a strong degradation of the geometry. Note that two instantaneous 3DGS style transfer methods [26, 33] are based on modifying the color features via some neural networks, but their results are not comparable to optimization-based approaches in terms of visual quality.

**Failure of coarse-to-fine strategy** Our approach minimizes a style transfer loss simultaneously at all scales. This is different from the coarse-to-fine style transfer strategy that has proven successful for HR style transfer [12]. Figure 6 illustrates that this coarse-to-fine strategy fails in the context of 3DGS. Indeed, the gain obtained by optimizing at a given scale is quickly lost when optimizing the next one leading to the disappearance of large scale features. This can be explained by the fact that the 3DGS representation is a more constrained representation than pixel grids due to its sparsity.

### Color transfer via style transfer at the coarsest scale

As described in Section 4.3, we first optimize the style transfer loss at the lowest resolution for 10k iterations and then optimize the SOS loss for another 10k iterations. Figure 7 illustrates that these first iterations are necessary for a

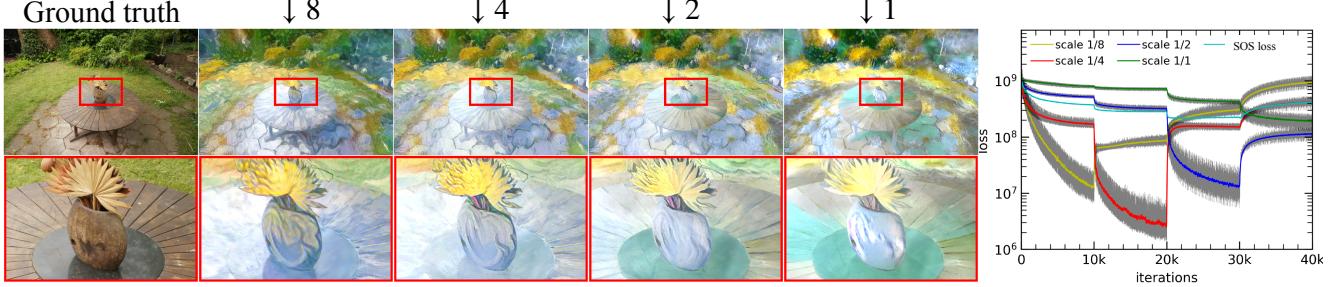


Figure 6. **Failure of coarse-to-fine strategy.** Results of coarse-to-fine style transfer for the example of Figure 3. Each scale is initialized with the output of the previous scale. As shown by the evolution of the losses (right), when training at a new scale, the loss of the previous scales increases quickly. This explains why the large scale painting features disappear progressively and are absent after training the target UHR (see close-up details).



Figure 7. **Color transfer via style transfer at the coarsest scale.** The first 10k iterations at the coarsest scale allow for quick color changes in the 3DGS scene. Removing this step may lead to color artifacts in the result (top) compared to our default two-step optimization (bottom).

faithful style color palette reproduction.

## 6. Discussion and limitations

**Texture representation** The texture representation within a 3DGS scene depends on the density of Gaussians and may be limited in low density areas. Isolated Gaussians can sometimes be spotted as illustrated by Figure 8.

**Large computation time** Our method is optimization-based and requires from several minutes to several hours of computation time depending on the resolution, as discussed in Section 5.1. On the other hand, very fast stylization for Gaussian splatting proposed by some concurrent approaches [26, 33] does not reach a satisfactory visual quality.

**Content-style mismatch** As said earlier, the Gram loss has the advantage of being independent of the style’s image resolution. Also, it enables a faithful transfer of global statistics of the style image, such as its color palette. This

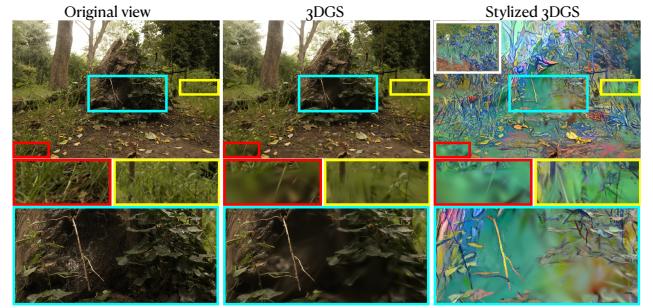


Figure 8. **Limited texture representation due to low Gaussian density of the initial 3DGS.** From left to right: Original scene, 3DGS reconstruction, stylization of the 3DGS scene.

important feature is, nevertheless, counterproductive when the style and content images strongly mismatch, leading to color bleeding or texturing of flat areas. Other controls can be added to mitigate these artefacts [12] and it was shown that these controls are effective for 3DGS scenes [46].

## 7. Conclusion

In this work we presented SGSST, a method that, for the first time, enables UHR 3DGS style transfer. To that aim, among other innovations, we introduced the simultaneously optimized scales (SOS) loss. Our qualitative, quantitative and perceptual studies show that SGSST obtains superior style transfer quality than state of the art, even after reducing our results’ resolution for a fair comparison with methods that do not reach UHR. Such high quality UHR results necessitate a large computation time that, nevertheless, remains comparable with that of UHR 3DGS training.

This work opens the way to several research directions. A first challenge is to produce equally high quality style transfer with a faster algorithm based on UST. A second more exploratory direction is to investigate geometry style transfer for 3DGS by designing adapted regularization to avoid the caveats depicted by Figure 5.

## References

- [1] Jie An, Tao Li, Haozhi Huang, Li Shen, Xuan Wang, Yongyi Tang, Jinwen Ma, Wei Liu, and Jiebo Luo. Real-time universal style transfer on high-resolution images via zero-channel pruning. *arXiv preprint arXiv:2006.09029*, 2020. 2
- [2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5470–5479, 2022. 2, 11
- [3] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Computer Vision – ECCV 2022*, pages 333–350, Cham, 2022. Springer Nature Switzerland. 2
- [4] Tian Qi Chen and Mark Schmidt. Fast patch-based style transfer of arbitrary style. *arXiv preprint arXiv:1612.04337*, 2016. 2
- [5] Zhe Chen, Wenhui Wang, Enze Xie, Tong Lu, and Ping Luo. Towards ultra-resolution neural style transfer via thumbnail instance normalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 393–400, 2022. 2
- [6] Pei-Ze Chiang, Meng-Shiun Tsai, Hung-Yu Tseng, Wei-Sheng Lai, and Wei-Chen Chiu. Styling 3d scene via implicit representation and hypernetwork. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1475–1484, 2022. 2
- [7] Tai-Yin Chiu and Danna Gurari. Iterative feature transformation for fast and versatile universal style transfer. In *European Conference on Computer Vision*, pages 169–184. Springer, 2020. 2
- [8] Valentin De Bortoli, Agnès Desolneux, Alain Durmus, Bruno Galerne, and Arthur Leclaire. Maximum entropy methods for texture synthesis: Theory and practice. *SIAM Journal on Mathematics of Data Science*, 3(1):52–82, 2021. 2
- [9] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5501–5510, 2022. 2, 11
- [10] Bruno Galerne, Lara Raad, José Lezama, and Jean-Michel Morel. Scaling Painting Style Transfer. *Computer Graphics Forum*, 2024. 2, 3, 4, 5, 6, 11
- [11] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016. 2, 3, 6
- [12] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling perceptual factors in neural style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3, 4, 6, 7, 8
- [13] Nicolas Gonthier, Yann Gousseau, and Saïd Ladjal. High-resolution neural texture synthesis with long-range constraints. *Journal of Mathematical Imaging and Vision*, 64(5):478–492, 2022. 2
- [14] Eric Heitz, Kenneth Vanhoey, Thomas Chambon, and Laurent Belcour. A sliced wasserstein loss for neural texture synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9412–9420, 2021. 2, 3
- [15] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2
- [16] Yi-Hua Huang, Yue He, Yu-Jie Yuan, Yu-Kun Lai, and Lin Gao. Stylizednerf: Consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18342–18352, 2022. 2
- [17] Eleftherios Ioannou and Steve Maddock. Evaluation in neural style transfer: A review. *Computer Graphics Forum*, 43(6):e15165, 2024. 6
- [18] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711, 2016. 2
- [19] Hyunyoung Jung, Seonghyeon Nam, Nikolaos Sarafianos, Sungjoo Yoo, Alexander Sorkine-Hornung, and Rakesh Ranjan. Geometry transfer for stylizing radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8565–8575, 2024. 2
- [20] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d Gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4), 2023. 2, 3, 4, 5, 11
- [21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 3
- [22] Áron Samuel Kovács, Pedro Hermosilla, and Renata G. Raidou.  $\mathcal{G}$ -style: Stylized gaussian splatting. *Computer Graphics Forum*, n/a(n/a):e15259, 2024. 2, 3, 7
- [23] Xuetong Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast image and video style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3809–3817, 2019. 2
- [24] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. *Advances in neural information processing systems*, 30:386–396, 2017. 2
- [25] Kunhao Liu, Fangneng Zhan, Yiwen Chen, Jiahui Zhang, Yingchen Yu, Abdulmotaleb El Saddik, Shijian Lu, and Eric P. Xing. Stylerf: Zero-shot 3d style transfer of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8338–8348, 2023. 2, 6
- [26] Kunhao Liu, Fangneng Zhan, Muyu Xu, Christian Theobalt, Ling Shao, and Shijian Lu. StyleGaussian: Instant 3d style transfer with Gaussian splatting, 2024. 2, 3, 6, 7, 8, 11
- [27] Yang Lu, Song-chun Zhu, and Ying Nian Wu. Learning frame models using cnn filters. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016. 2

- [28] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Computer Vision – ECCV 2020*, pages 405–421, Cham, 2020. Springer International Publishing. [2](#), [3](#)
- [29] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4), 2022. [2](#)
- [30] Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5880–5888, 2019. [2](#)
- [31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. [3](#)
- [32] Eric Risser, Pierre Wilmot, and Connally Barnes. Stable and controllable neural texture synthesis and style transfer using histogram losses, 2017. [2](#), [3](#)
- [33] Abhishek Saroha, Mariia Gladkova, Cecilia Curreli, Tarun Yenamandra, and Daniel Cremers. Gaussian splatting in style, 2024. [2](#), [3](#), [6](#), [7](#), [8](#)
- [34] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [3](#)
- [35] Omry Sendik and Daniel Cohen-Or. Deep correlations for texture synthesis. *ACM Trans. Graph.*, 36(5), 2017. [2](#), [3](#)
- [36] Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8242–8250, 2018. [2](#)
- [37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [2](#), [3](#)
- [38] Xavier Snelgrove. High-resolution multi-scale neural texture synthesis. In *SIGGRAPH Asia 2017 Technical Briefs*, New York, NY, USA, 2017. Association for Computing Machinery. [2](#), [4](#)
- [39] Ondřej Texler, David Futschik, Jakub Fišer, Michal Lukáč, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. Arbitrary style transfer using neurally-guided patch-based synthesis. *Computers & Graphics*, 87:62–71, 2020. [2](#)
- [40] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*, pages 1349–1357, 2016. [2](#)
- [41] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [2](#)
- [42] Jonathan Vacher, Aida Davila, Adam Kohn, and Ruben Coen-Cagli. Texture interpolation for probing visual perception. In *Advances in Neural Information Processing Systems*, pages 22146–22157. Curran Associates, Inc., 2020. [2](#)
- [43] Huan Wang, Yijun Li, Yuehai Wang, Haoji Hu, and Ming-Hsuan Yang. Collaborative distillation for ultra-resolution universal style transfer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [2](#)
- [44] Zhizhong Wang, Lei Zhao, Zhiwen Zuo, Ailin Li, Haibo Chen, Wei Xing, and Dongming Lu. Microast: Towards super-fast ultra-resolution arbitrary style transfer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(3): 2742–2750, 2023. [2](#)
- [45] Tong Wu, Yu-Jie Yuan, Ling-Xiao Zhang, Jie Yang, Yan-Pei Cao, Ling-Qi Yan, and Lin Gao. Recent advances in 3d gaussian splatting. *Computational Visual Media*, 10(4):613–642, 2024. [2](#)
- [46] Dingxi Zhang, Zhuoxun Chen, Yu-Jie Yuan, Fang-Lue Zhang, Zhenliang He, Shiguang Shan, and Lin Gao. StylizedGS: Controllable stylization for 3d Gaussian splatting, 2024. [2](#), [3](#), [7](#), [8](#)
- [47] Deheng Zhang, Clara Fernandez-Labrador, and Christopher Schroers. CoARF: Controllable 3D Artistic Style Transfer for Radiance Fields . In *2024 International Conference on 3D Vision (3DV)*, pages 612–622, Los Alamitos, CA, USA, 2024. IEEE Computer Society. [2](#)
- [48] Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. ARF: Artistic radiance fields. In *ECCV 2022*, 2022. [2](#), [3](#), [5](#), [6](#), [11](#)
- [49] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [6](#)

# SGSST: Scaling Gaussian Splatting Style Transfer

## Supplementary Material

**Supplementary material description** The supplementary material consists of the following elements:

- This present appendix with additional details and figures.
- A project website with rendered videos, including one video showing the stylized scenes of the main paper’s teaser figure and the 40 comparison videos: <https://www.idpoisson.fr/galerne/sgsst/>.
- The source code used for all experiments available at <https://github.com/JianlingWANG2021/SGSST>. It is based on the public source codes<sup>3</sup> for 3DGS [20] training and SPST [10].

Note that due to space constraints the images have been compressed and/or reduced in size.

### A. UHR style transfers of the teaser figure

Due to space limitation, style images of the main paper’s teaser figure have been displayed as tiny images regardless of their resolution. Figures 9 to 16 show the eight pairs of images of this figure in full size to better appreciate the multiscale details of the style images and their corresponding stylized results. Each style image is displayed at the same resolution as the rendered view so that one can observe that the style features are reproduced with the same size (see e.g. the stone wall of Figure 14).

### B. Comparison experiments

As said in the main paper, we performed a thorough comparative study using 40 3D style transfer experiments using 9 different scenes from previous works [2, 9, 20] and various style images. We compare our results with the NeRF-based ARF [48] and the 3DGS-based StyleGaussian [26] algorithms using their public implementations<sup>4</sup>.

Figures 17 to 25 display a rendered view for each of these 40 experiments. Let us recall that for the HR scenes (Figures 17 to 21) our approach is the only one working at high-resolution. While SGSST produces outputs having the content size, StyleGaussian outputs are limited in resolution to a maximal width of 1600 or maximal height of 1200, and for ARF the content images have been downsampled by a factor 4 to obtain a low-resolution input suitable for ARF (see Section C below).

<sup>3</sup><https://github.com/graphdeco-inria/gaussian-splatting>; [https://github.com/bgalerne/scaling\\_painting\\_style\\_transfer](https://github.com/bgalerne/scaling_painting_style_transfer)

<sup>4</sup><https://github.com/Kai-46/ARF-svox2>; <https://github.com/Kunhao-Liu/StyleGaussian>

### C. ARF and high resolution inputs

ARF [48] uses Nearest Neighbor Feature Matching (NNFM) of a single layer of VGG features for fine tuning a plenoxel radiance field [9]. It produces high-quality results at moderate resolution. While comparing our results with ARF, we observed that this algorithm does not produce visually satisfying results for high-resolution scenes. This is illustrated by Figure 26 where one can observe that the style transfer quality decreases as the input size increases. To allow a fair comparison we decided to downscale images by a factor 4 for the high-resolution scene as a preprocess for ARF.

Although it has been shown that NNFM is superior to Gram feature matrix optimization for NeRF style transfer when optimizing for a single VGG layer [48], our results show that optimizing for a (slightly corrected [10]) Gram-based loss using several image scales and five VGG layers for each scale is an effective solution for applying high quality style transfer at UHR.

### D. Details on the perceptual study

As described in the main paper, a comparative perceptual study was conducted using the 40 3D style transfer experiments presented in Section B (Figures 17 to 25). For each experiment, they were asked to pick the image that appeared to be the most faithful to the style image among the three displayed results. Each participant was shown ten random experiments and participation was voluntary. Figure 26 is an example of such an experiment displaying the style image (top left image) and three views of the scenes stylized by SGSST, ARF and StyleGaussian respectively and displayed in random order. To choose between one of the three results, the participant had to press the left arrow key to select the bottom left result, the up arrow key to select the top right result and the down arrow key to select the bottom right result.

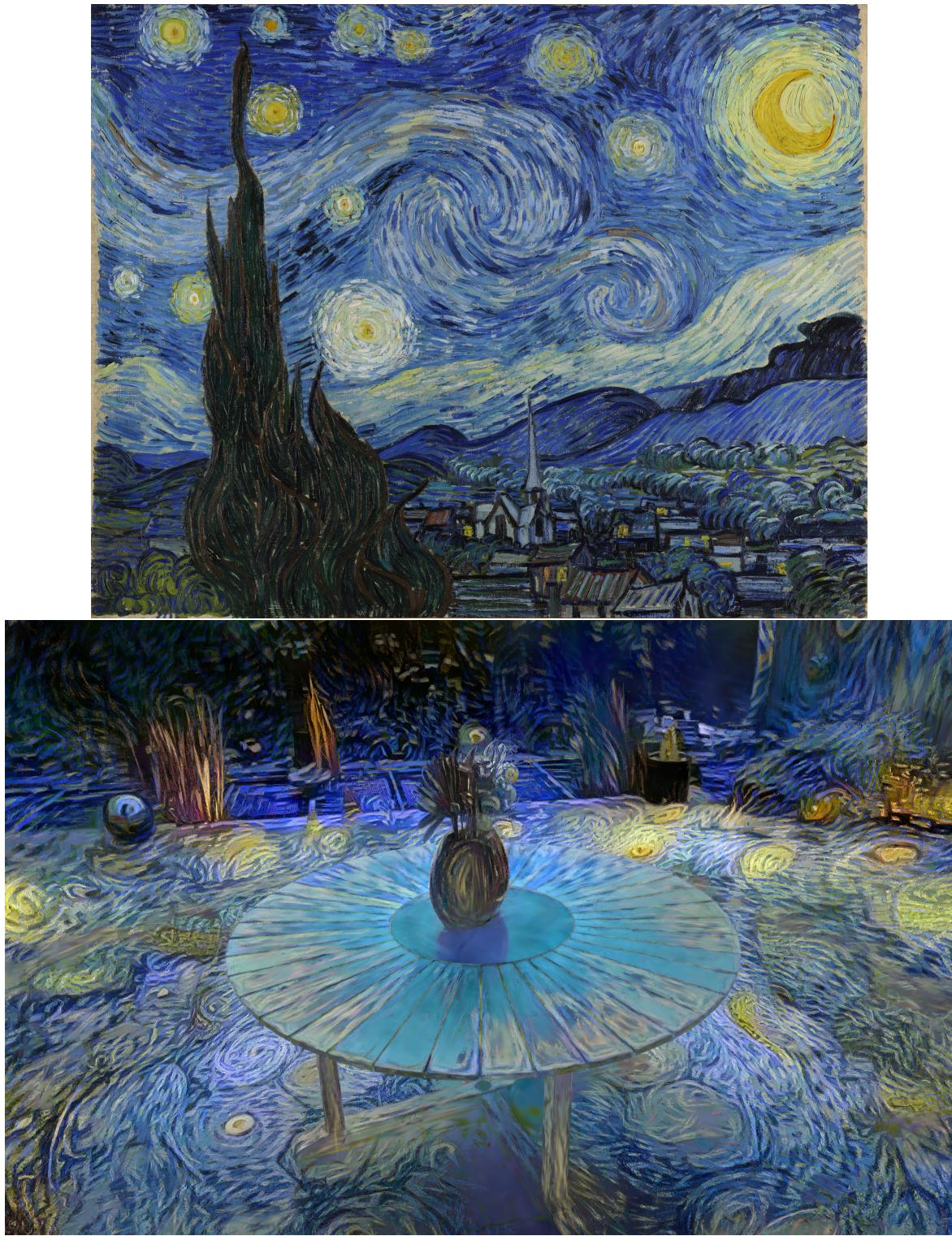


Figure 9. Full view display of the example 1/8 of the teaser figure with the style image (size  $4244 \times 3361$ ) displayed at the same scale as the rendered image (size  $5187 \times 3361$ ). Images have been downscaled by a factor 2 and compressed using jpeg.

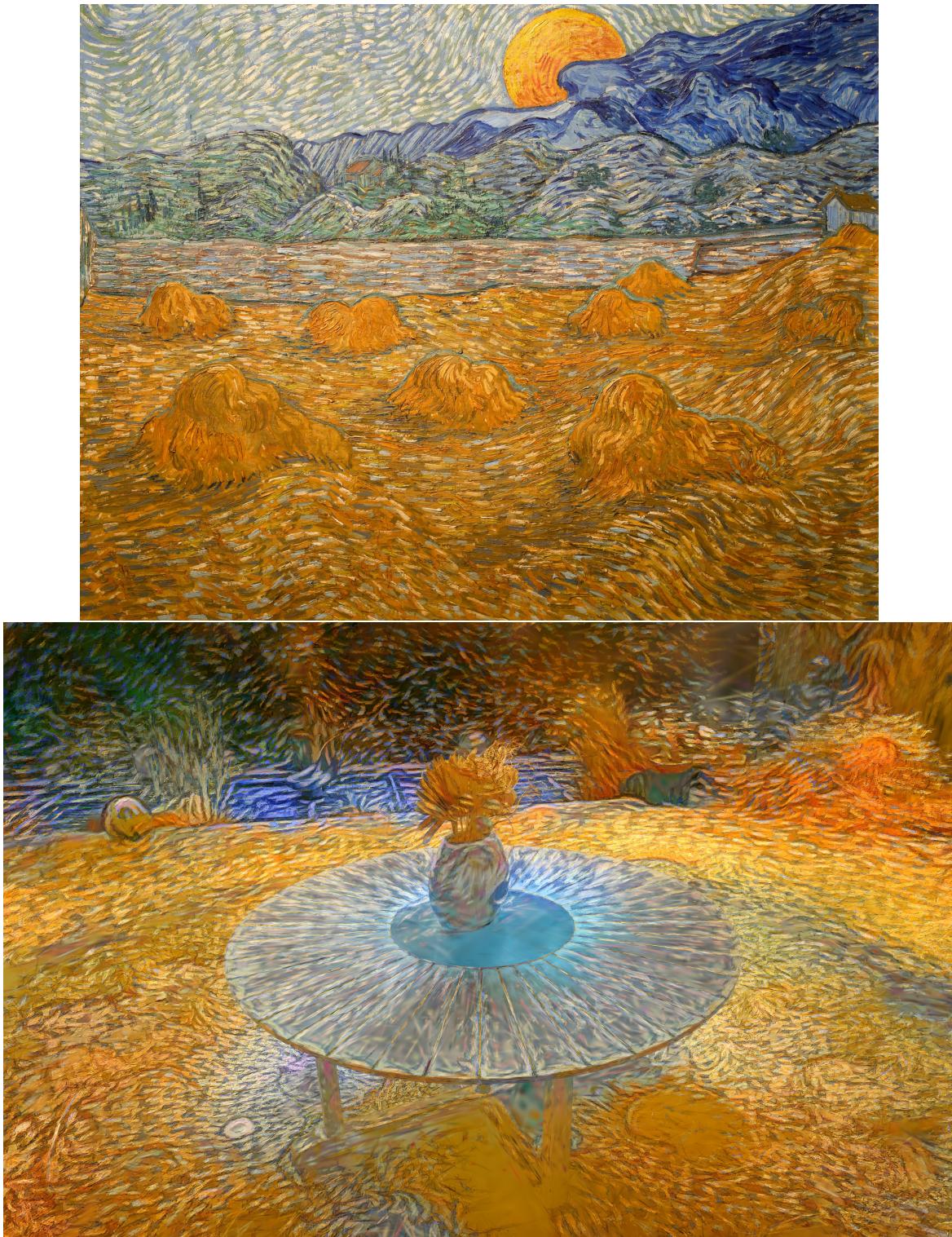


Figure 10. Full view display of the example 2/8 of the teaser figure with the style image (size  $4351 \times 3361$ ) displayed at the same scale as the rendered image (size  $5187 \times 3361$ ). Images have been downscaled by a factor 2 and compressed using jpeg.

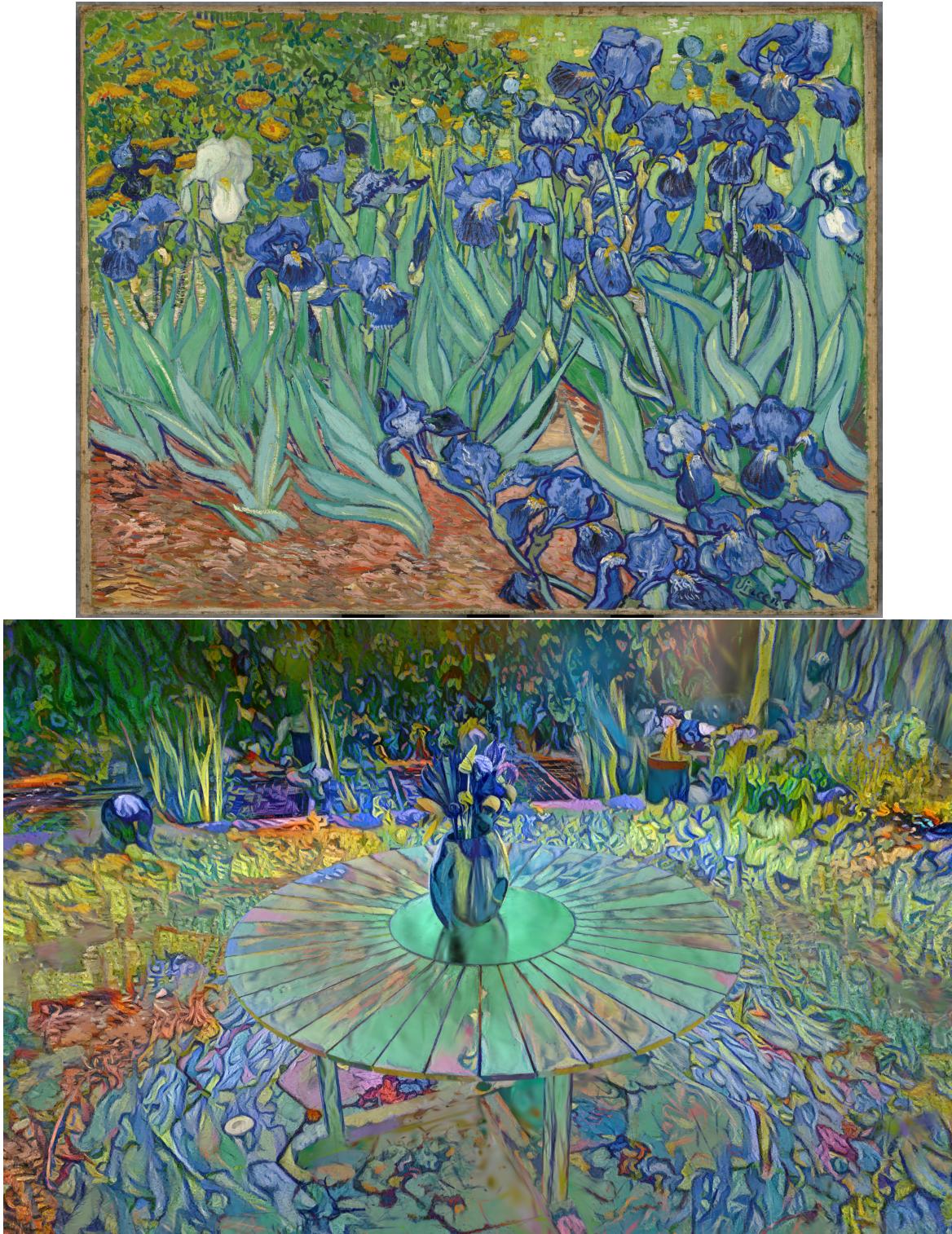


Figure 11. Full view display of the example 3/8 of the teaser figure with the style image (size  $4398 \times 3361$ ) displayed at the same scale as the rendered image (size  $5187 \times 3361$ ). Images have been downscaled by a factor 2 and compressed using jpeg.



Figure 12. Full view display of the example 4/8 of the teaser figure with the style image (size 4398×3361) displayed at the same scale as the rendered image (size 5187×3361). Images have been downsampled by a factor 2 and compressed using jpeg.

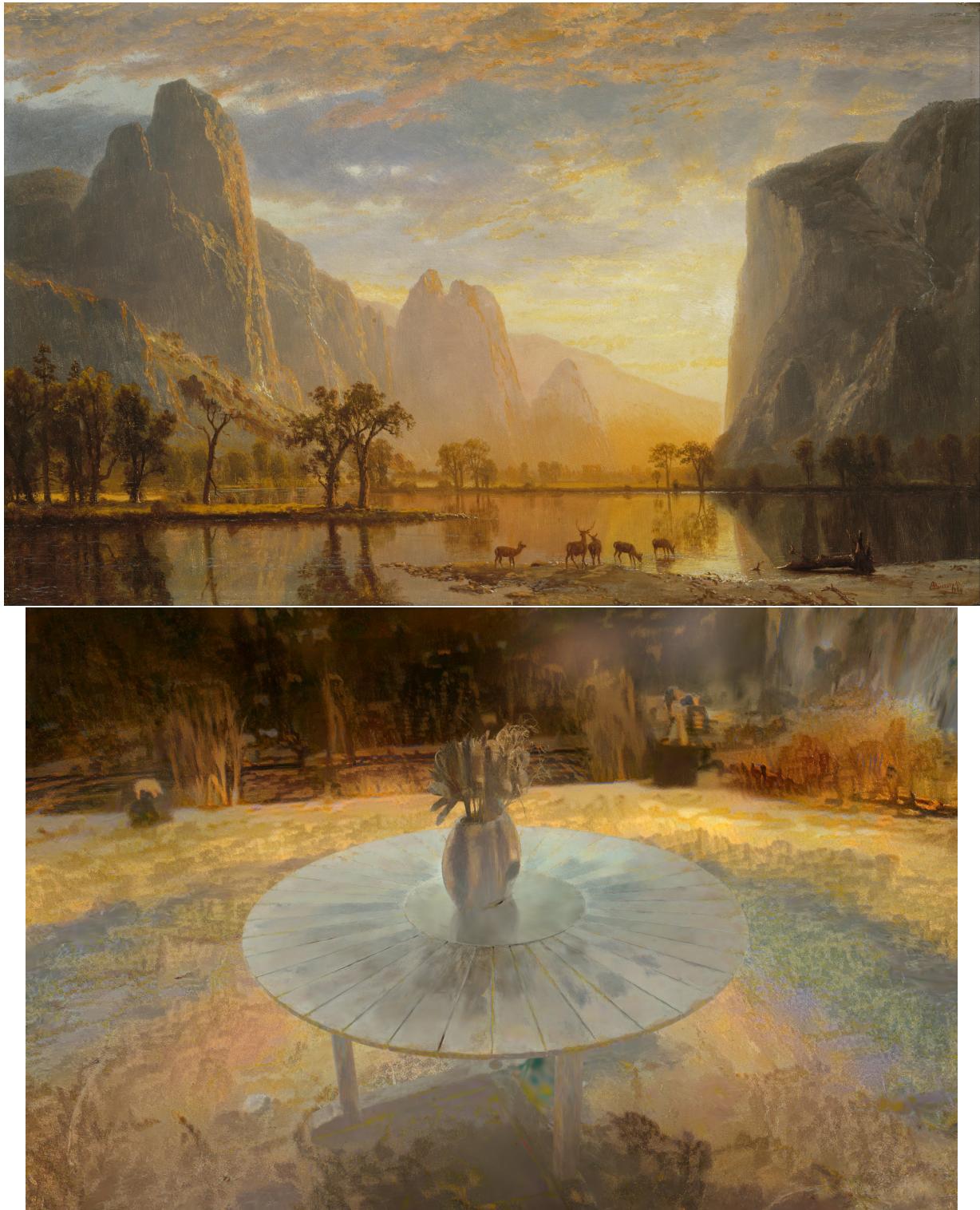


Figure 13. Full view display of the example 5/8 of the teaser figure with the style image (size 5433×3361) displayed at the same scale as the rendered image (size 5187×3361). Images have been downsampled by a factor 2 and compressed using jpeg.

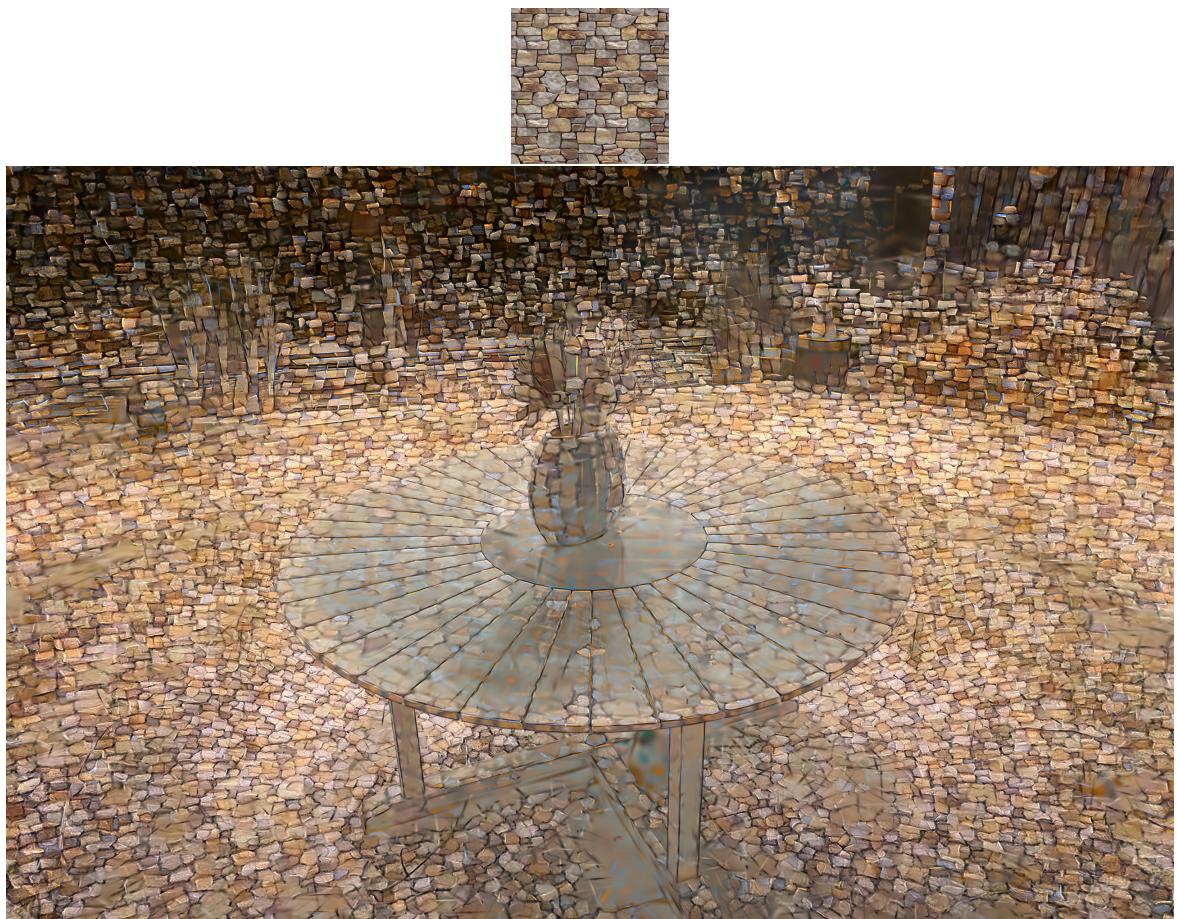


Figure 14. Full view display of the example 6/8 of the teaser figure with the style image (size  $700 \times 692$ ) displayed at the same scale as the rendered image (size  $5187 \times 3361$ ). Images have been downscaled by a factor 2 and compressed using jpeg.

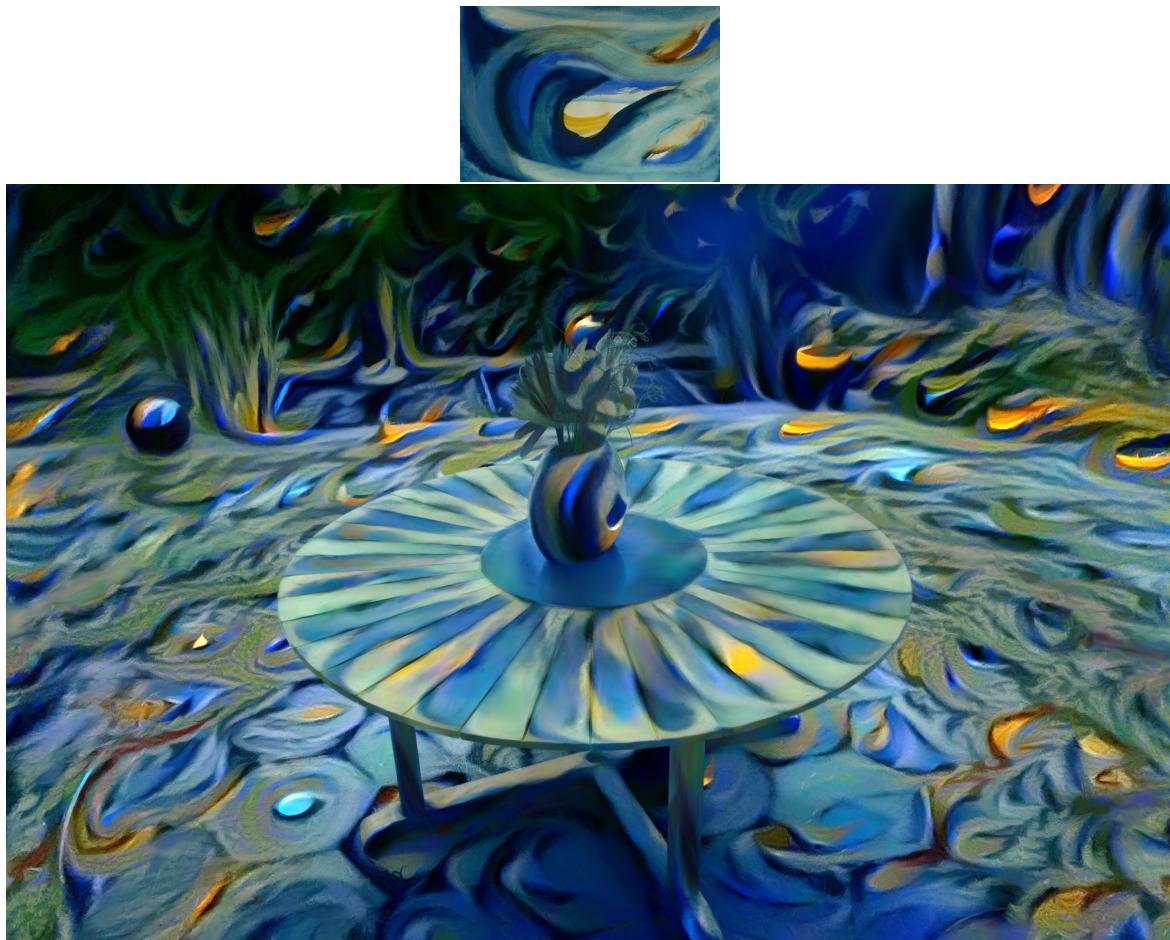


Figure 15. Full view display of the example 7/8 of the teaser figure with the style image (size 1152×781) displayed at the same scale as the rendered image (size 5187×3361). Images have been downscaled by a factor 2 and compressed using jpeg.



Figure 16. Full view display of the example 8/8 of the teaser figure with the style image (size  $1024 \times 1024$ ) displayed at the same scale as the rendered image (size  $5187 \times 3361$ ). Images have been downsampled by a factor 2 and compressed using jpeg.

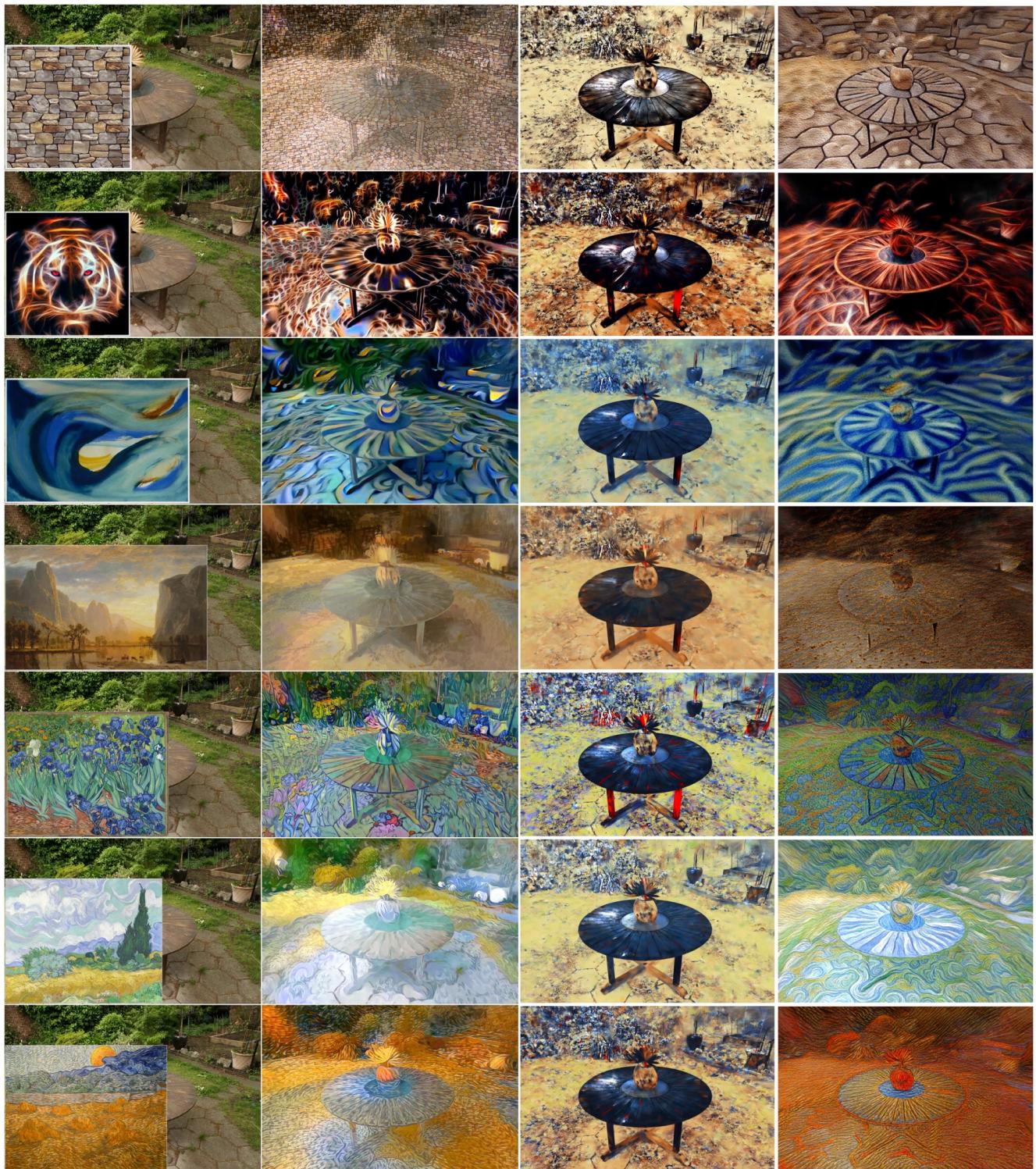


Figure 17. Comparative experiments using the garden scene. From left to right: Content and style, SGSST (ours), StyleGaussian, ARF. Content image size is 5187×3361.

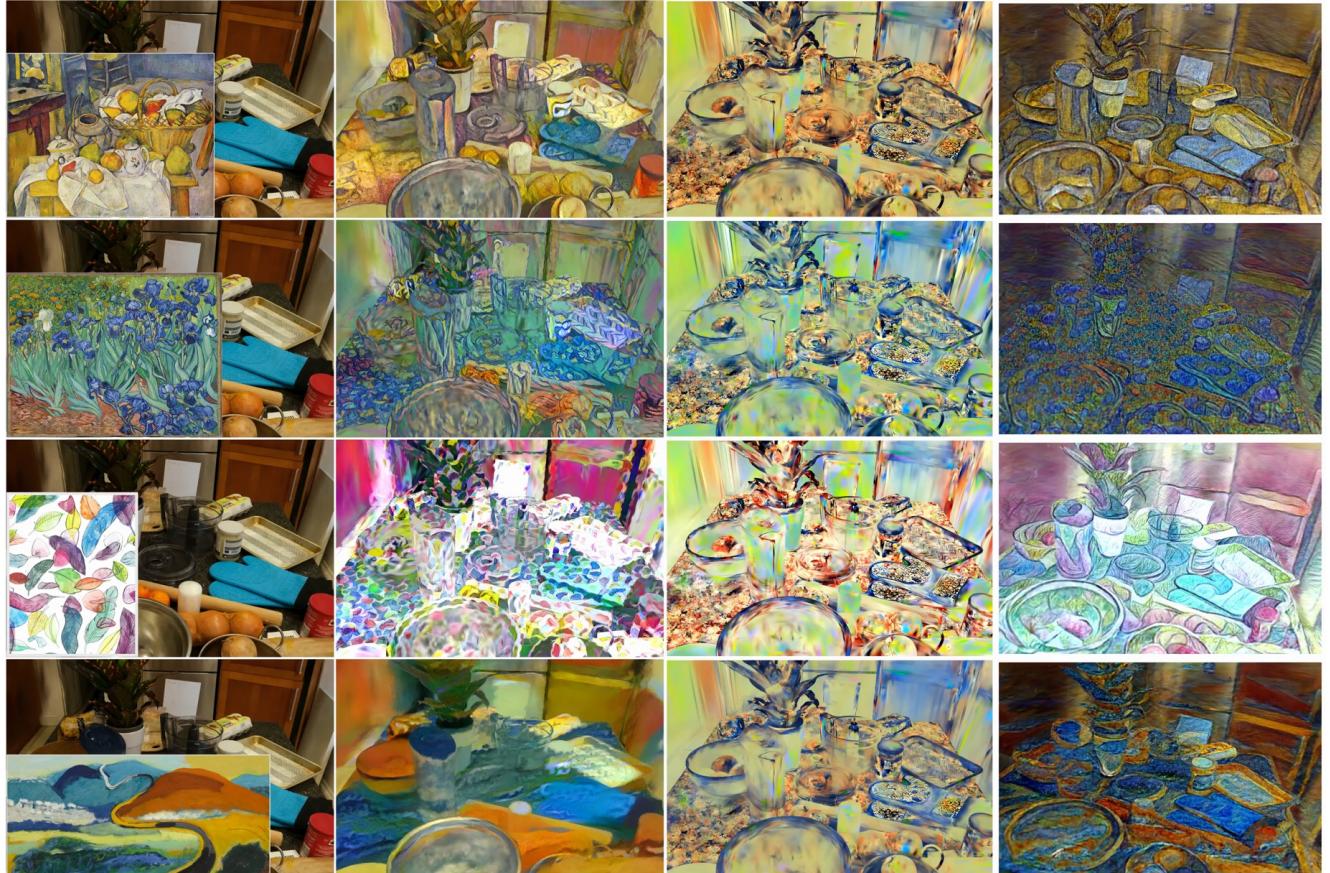


Figure 18. Comparative experiments using the counter scene. From left to right: Content and style, SGSST (ours), StyleGaussian, ARF. Content image size is  $3115 \times 2076$ .

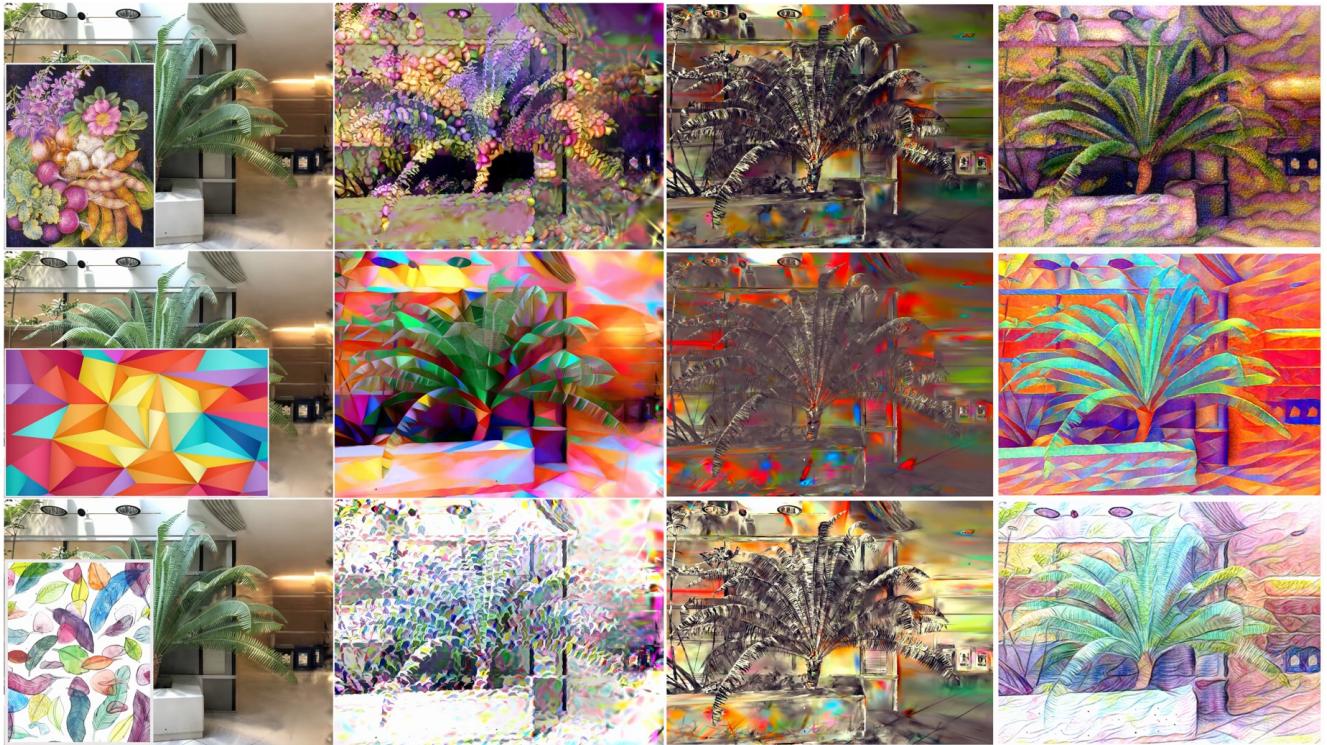


Figure 19. Comparative experiments using the fern scene. From left to right: Content and style, SGSST (ours), StyleGaussian, ARF. Content image size is  $4032 \times 3024$ .

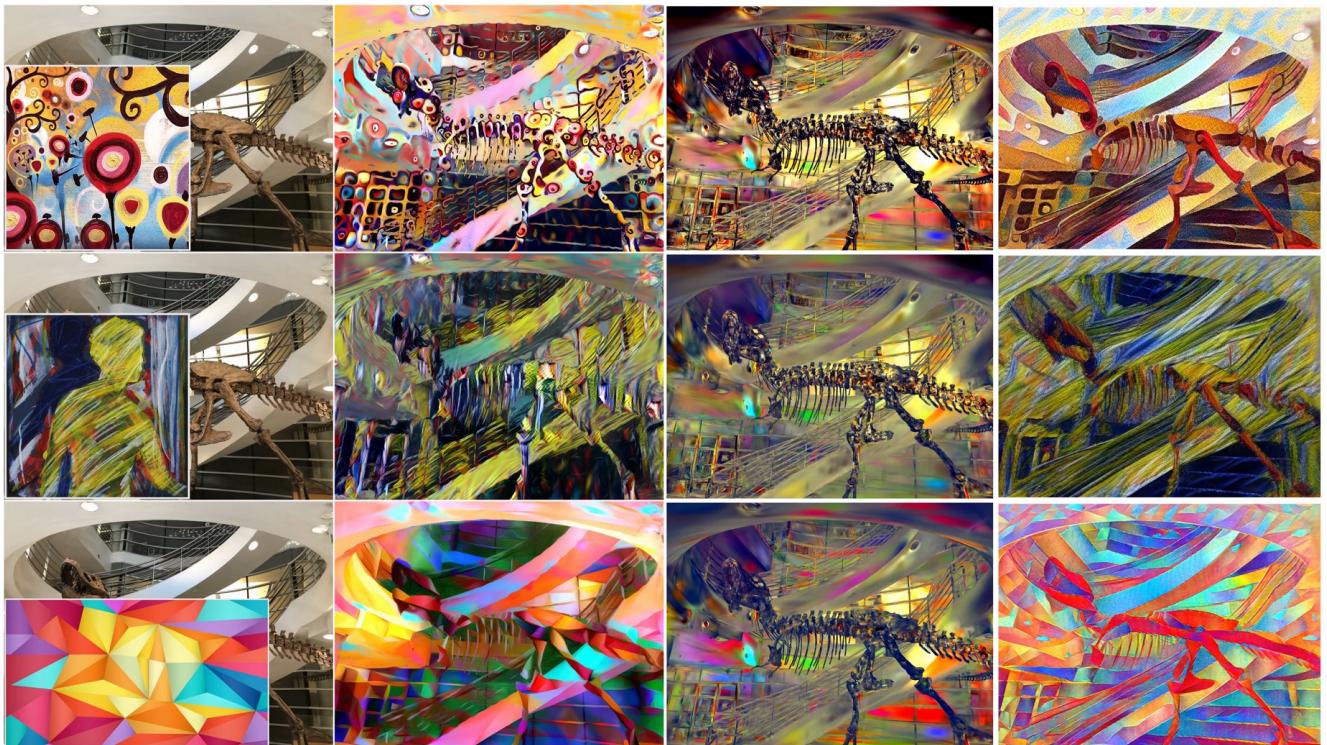


Figure 20. Comparative experiments on the t-rex scene. From left to right: Content and style, SGSST (ours), StyleGaussian, ARF. Content image size is  $4032 \times 3024$ .

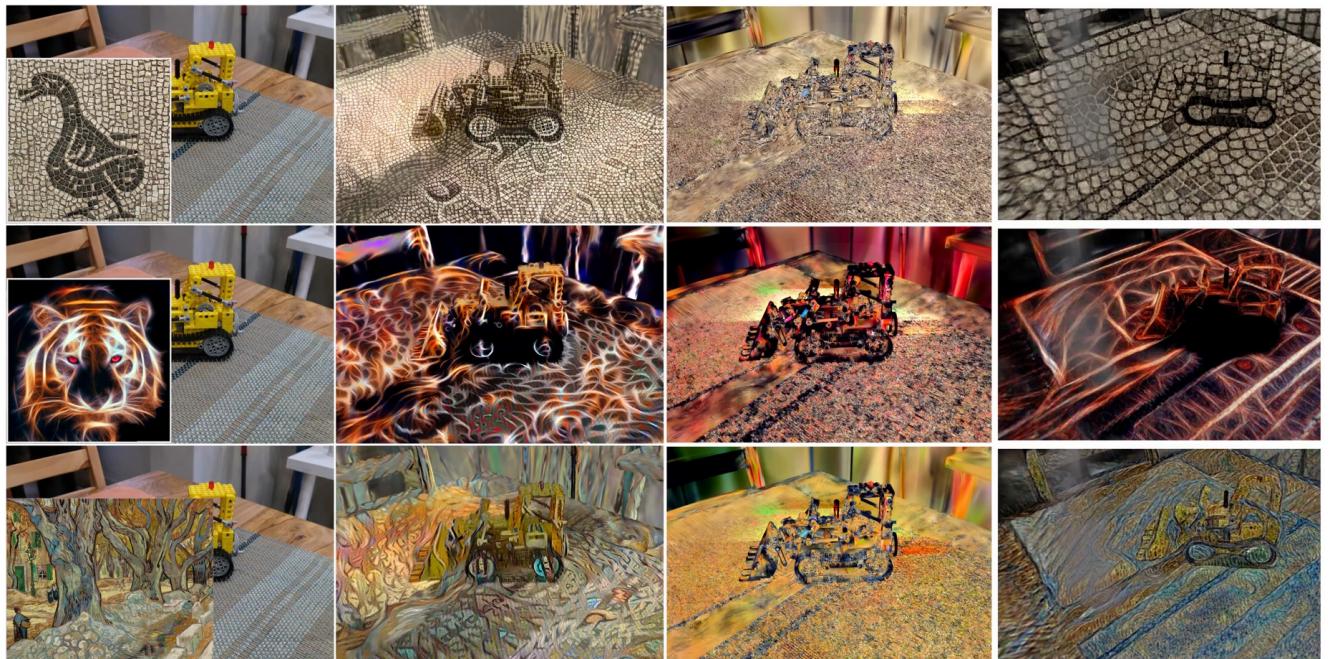


Figure 21. Comparative experiments on the kitchen scene. From left to right: Content and style, SGSST (ours), StyleGaussian, ARF. Content image size is  $3115 \times 2078$ .

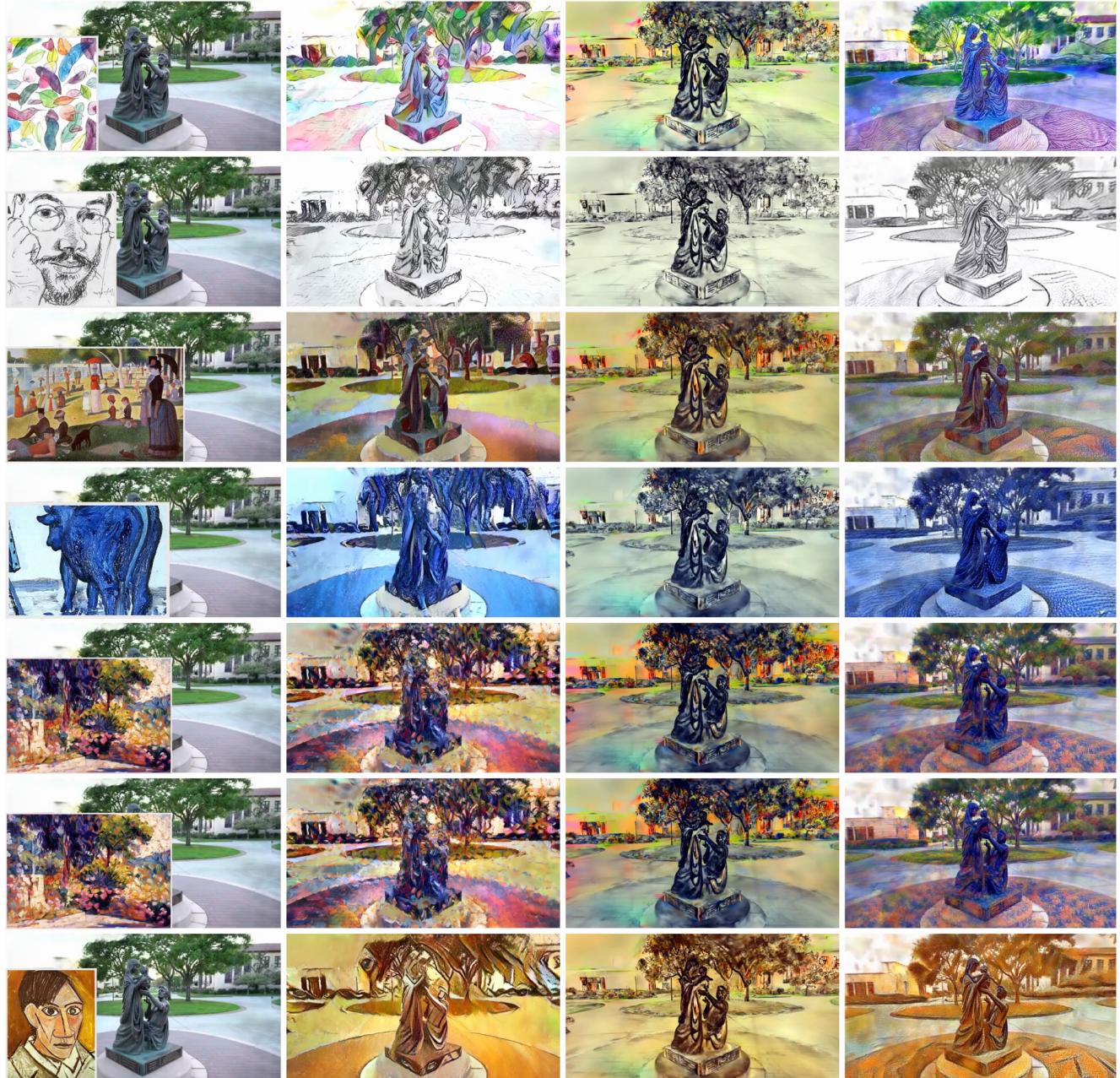


Figure 22. Comparative experiments on the family scene. From left to right: Content and style, SGSST (ours), StyleGaussian, ARF. Content image size is  $977 \times 544$ .



Figure 23. Comparative experiments on the horse scene. From left to right: Content and style, SGSST (ours), StyleGaussian, ARF. Content image size is 976×544.



Figure 24. Comparative experiments on the train scene. From left to right: Content and style, SGSST (ours), StyleGaussian, ARF. Content image size is  $980 \times 545$ .

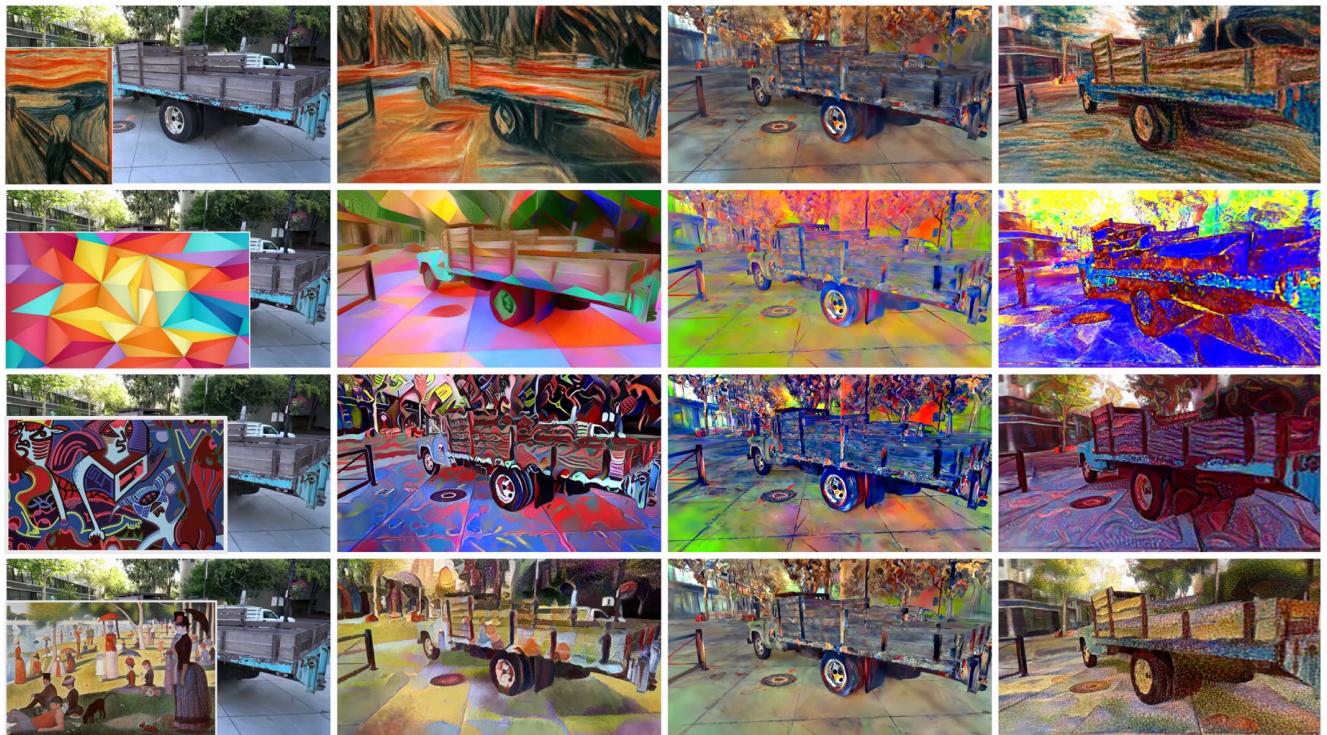


Figure 25. Comparative experiments on the truck scene. From left to right: Content and style, SGSST (ours), StyleGaussian, ARF. Content image size is  $979 \times 546$ .



Figure 26. ARF outputs for the HR style transfer: example of the main paper with various downscaling factors. ARF produces good stylization results for inputs of moderate resolution only. From left to right: Scene and style (input size is  $3115 \times 2076$ ), ARF result with input down-scaled by 4 (size  $779 \times 519$ ), ARF result with input down-scaled by 2 (size  $1557 \times 1038$ ), ARF result with original HR (size  $3115 \times 2076$ ).

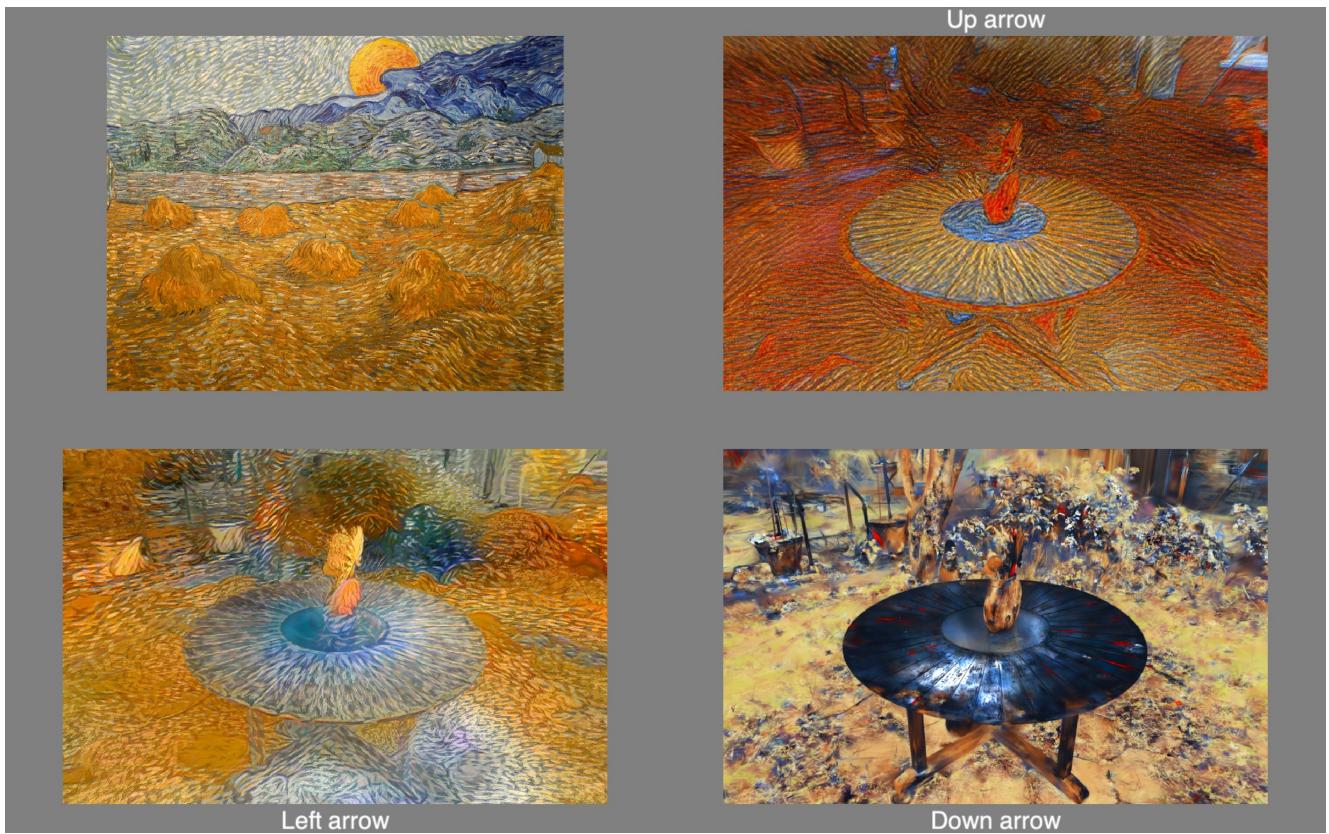


Figure 27. Perceptual study. The style input image is presented on the top left and the results of each stylization algorithm (SGSST, ARF and StyleGaussian) are presented in a random order. To select the best result, the participant has to press the key indicated next to it.