Computational Visual Media
https://doi.org/10.1007/s41095-022-0301-9 Vol. 9, No. 4, December 2023, 767–786

Research Article

STATE: Learning structure and texture representations for novel view synthesis

Xinyi Jing[1,*], Qiao Feng[1,*], Yu-Kun Lai[2], Jinsong Zhang[1], Yuanqiang Yu[1], and Kun Li[1] (✉)

Abstract Novel viewpoint image synthesis is very challenging, especially from sparse views, due to large changes in viewpoint and occlusion. Existing image-based methods fail to generate reasonable results for invisible regions, while geometry-based methods have difficulties in synthesizing detailed textures. In this paper, we propose STATE, an end-to-end deep neural network, for sparse view synthesis by learning *structure and texture* representations. Structure is encoded as a hybrid feature field to predict reasonable structures for invisible regions while maintaining original structures for visible regions, and texture is encoded as a deformed feature map to preserve detailed textures. We propose a hierarchical fusion scheme with intra-branch and inter-branch aggregation, in which spatio-view attention allows multi-view fusion at the feature level to adaptively select important information by regressing pixel-wise or voxel-wise confidence maps. By decoding the aggregated features, STATE is able to generate realistic images with reasonable structures and detailed textures. Experimental results demonstrate that our method achieves qualitatively and quantitatively better results than state-of-the-art methods. Our method also enables texture and structure editing applications benefiting from implicit disentanglement of structure and texture. Our code is available at `http://cic.tju.edu.cn/faculty/likun/projects/STATE`.

Keywords novel view synthesis; sparse views; spatio-view attention; structure representation; texture representation

1 Introduction

Given a single image of an object, or several images from different viewpoints, *novel view synthesis* aims to generate a further image seen from a new viewpoint. This has a wide range of applications in virtual reality, education, and movie production. It is a very challenging problem given sparse input views due to large appearance variations and occlusion.

Existing methods for novel view synthesis can be classified as image-based or geometry-based. Image-based methods warp a source image from the source viewpoint to the target viewpoint by estimating an affine transformation [1, 2] or an appearance flow field [3–5]. Flow-based methods can more flexibly deal with complex deformations than affine transformation methods. However, due to lack of geometric information, image-based methods tend to generate unsatisfactory results for invisible regions, especially given sophisticated objects or sparse views. Geometry-based methods first model the 3D structure of the object in an explicit [6–8] or implicit [9–11] manner, and then generate the target image by rotation and projection. Explicit representations use discrete volumes while implicit methods use continuous implicit functions. Along with neural rendering based methods [12], the latter can be trained without 3D supervision. Although geometry-based methods can ensure structural consistency and predict reasonable shapes for the invisible regions, results are poor for sparse views, and they may lose texture detail if the representation has limited resolution.

It is very important to find an effective way to make better use of multi-view information, especially for sparse views. Most works [10, 13–16] directly average the representations from all inputs, where all

* Xinyi Jing and Qiao Feng contributed equally to this work.
1 College of Intelligence and Computing, Tianjin University, Tianjin 300350, China. E-mail: X. Jing, jingxinyi@tju.edu.cn; Q. Feng, fengqiao@tju.edu.cn; J. Zhang, jinszhang@tju.edu.cn; Y. Yu, yuyuanqiang@tju.edu.cn; K. Li, lik@tju.edu.cn (✉).
2 School of Computer Science and Informatics, Cardiff University, Cardiff CF24 4AG, UK. E-mail: LaiY4@cardiff.ac.uk.
Manuscript received: 2022-02-15; accepted: 2022-06-16

locations of inputs are taken as valid values. However, not all locations of inputs have a positive impact on the target image. To solve this problem, Sun et al. [4] propose a self-learned confidence method to fuse the resulting images generated by each input at the pixel level. However, this fusion scheme requires a large amount of memory and cannot deal with the unavoidable misalignment problem.

The aforementioned methods encounter three challenges to synthesizing satisfactory images: (i) the coupling of shape and texture in the input images, (ii) potential uncertainties in invisible regions, and (iii) difficulty in achieving color, texture, and shape consistency.

To address these problems, in this paper, we propose an end-to-end deep neural network, STATE, for sparse view synthesis; it disentangles the input images into *STructure And TExture representations* to ensure both shape and texture consistency. Although our method does not explicitly control disentanglement, proper design of the two branches achieves effective disentanglement of structure and texture as we verify later through experimental results. In the structure-aware encoder, we represent structure as a hybrid feature field, which can predict reasonable structure for invisible regions. In the texture-aware encoder, we estimate an appearance flow field and warp the source image feature from the source viewpoint to the target viewpoint at the feature level. To make the best use of multiple images, we also propose *spatio-view attention* aggregation to adaptively fuse multi-view information at the feature level by regressing pixel-wise or voxel-wise confidence maps. The final image is delivered by decoding the aggregated feature of structure-aware representation and texture-aware representation. Our model works well for both single view and multi-view inputs. Experimental results demonstrate that our method works better than the state-of-the-art. We also validate our approach by comprehensive ablation studies. Figure 1 gives some examples of our results.

Our code is available at `http://cic.tju.edu.cn/faculty/likun/projects/STATE` to promote academic development.

Our main contributions are, in summary:

- STATE, an end-to-end deep neural network to disentangle sparse input images into two neural embedding representations of structure and texture; it can help predict reasonable regions for ones invisible in the source images, while also recovering detailed textures,
- a hierarchical fusion scheme with intra-branch and inter-branch aggregation; spatio-view attention provides multi-view fusion at the feature level to adaptively select important information by regressing pixel-wise or voxel-wise confidence maps, and
- a model which can realize texture or structure swapping without training due to effective disentanglement of structures and textures: our model can be easily and robustly trained with a hybrid loss such as cosine loss to achieve color, texture, and shape consistency, leading to state-of-the-art results.

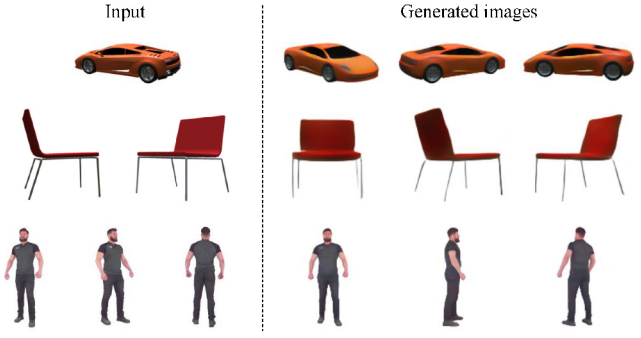

Fig. 1 Our STATE model can generate realistic images from sparse views or even a single image.

2 Related work

2.1 Scope

We next review existing work on novel view synthesis for objects or humans, from a single or multiple images; methods can be image-based or geometry-based. The former maintain appearance consistency by transferring pixels from the source images to the target image, while the latter maintain structural consistency by reconstructing the 3D object to render the novel image.

2.2 Image-based novel view synthesis

Image-based novel view synthesis methods directly generate pixels or move pixels from the source images to the target image. Tatarchenko et al. [1] and Yang et al. [2] generate pixels with affine transformation. Instead of learning to synthesize pixels from scratch, Zhou et al. [5] prove that the visual appearances of

the same instance from different viewpoints are highly correlated, and such correlation can be explicitly learned to predict appearance flow [3, 4, 17], i.e., 2D coordinate vectors specifying which pixels in the input view can be used to reconstruct the target view. To use features at different scales, Yin et al. [18] estimate appearance flows with different resolutions to warp the source view to the target view. Controlled by the appearance flow, bilinear sampling is used to move pixels from the source images to the target image [4, 5, 17, 19]. To avoid the poor gradient propagation of bilinear sampling, Ren et al. [3] propose a content-aware sampling method adopting a local attention mechanism. Most flow-based methods [4, 5] warp the input images pixel-wise, which prevents the network from generating new content for invisible pixels. Warping the input images at the feature level can solve this problem [3, 17, 20]. Other methods synthesize invisible pixels without warping the input features. Park et al. [21] use a completion network to hallucinate the empty parts. In summary, image-based methods can generate detailed textures by moving pixels from the source images to the target image, but the results generated by the above methods lack a consistent shape and so may have artifacts along the silhouette.

2.3 Geometry-based novel view synthesis

Geometry-based novel view synthesis methods determine the 3D structure of the object in an explicit or implicit manner, and then generate the target image by rotation and projection. Approaches may be based on depth maps or 3D models (textured occupancy volumes, colored point clouds, or neural scene representations). Depth-map-based approaches [6, 22, 23] typically generate a depth map for each input view as a 2.5D intermediate representation which captures hidden surfaces from one or multiple viewpoints. Point-cloud-based methods [8] generate a point cloud to be transformed into the target view. Several recent methods [7, 24–26] reconstruct an explicit occupancy volume from the input images, and render it using traditional rendering techniques. To overcome the memory limitation of volume representations, some methods leverage signed distance field encoded volumes [27, 28] or RGBα-encoded volumes [29, 30], with good results. Since explicit volumes are discrete, several methods [10, 31–33] based on implicit volume representations without any 3D supervision have been proposed. In order to have better understanding of the structure of objects, Galama and Mensink [34] propose IterGANs to iteratively learn an implicit 3D model of the object. Implicit volume representation has gained popularity due to its continuous shape and texture representation. Some methods [9, 11, 35, 36] predict continuous neural scene representations, and then use neural rendering to produce the novel view image. Geometry-based methods can keep structural consistency and predict reasonable shapes for invisible regions, but the generated textures tend to lack fine details.

In this paper, we propose an end-to-end deep neural network for sparse view synthesis by learning structure and texture representations. Structure is encoded as a hybrid feature field; texture is encoded as a deformed feature map. Each representation is generated by spatio-view attention aggregation for multi-view cases. The results generated by our approach have consistent structures and detailed textures.

3 Method

3.1 Overview

The inputs of novel view synthesis from N images are a target camera pose p_{t} and N source images each coupled with a camera pose $(I_{\mathrm{s}}^1, p_{\mathrm{s}}^1), \ldots, (I_{\mathrm{s}}^N, p_{\mathrm{s}}^N)$. Our goal is to synthesize the target image $\hat{I}_{\mathrm{t}}$ in the target camera pose p_{t}. I_{t} and $\hat{I}_{\mathrm{t}}$ denote the ground truth and synthetic target images, respectively. In order to generate a result with reasonable structure and fine texture, we propose a new network STATE that aggregates information from both structure and texture representations. As Fig. 2 shows, STATE consists of a two-branch encoder and a fusion decoder.

The two-branch encoder $E(\cdot)$, consisting of a structure-aware branch and a texture-aware branch, encodes the inputs into a structure feature volume f_{str} and a texture feature map f_{tex}. It can be written as

$$(f_{\mathrm{str}}, f_{\mathrm{tex}}) = E(p_{\mathrm{t}}, (I_{\mathrm{s}}^1, p_{\mathrm{s}}^1), \ldots, (I_{\mathrm{s}}^N, p_{\mathrm{s}}^N)) \quad (1)$$

The structure-aware branch produces a hybrid feature field for each view, and then rotates and adaptively aggregates them into a single feature volume f_{str} containing structure information. The texture-aware branch generates a single feature map f_{tex} containing texture information by adaptively fusing the flow-

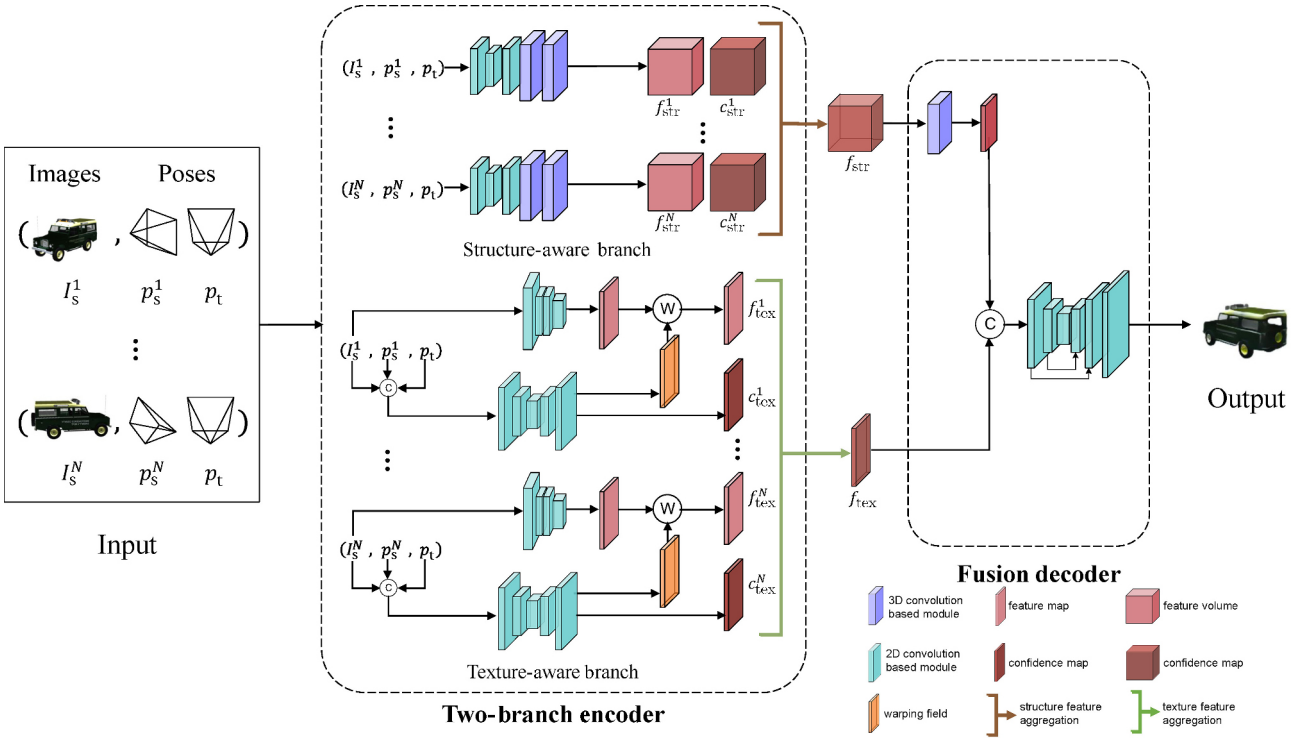

Fig. 2 Overview of our STATE model.

warped features from the N views.

The fusion decoder $D(\cdot)$ takes the feature volume f_{str} and the feature map f_{tex} as input and generates the target image:

$$\hat{I}_{\text{t}} = D(f_{\text{str}}, f_{\text{tex}}) \tag{2}$$

Adaptive fusion of multi-view inputs is explained in detail in Section 3.3. Note that our model can handle an arbitrary number of inputs for both training and testing without modifying the encoder or decoder.

3.2 Two-branch encoder

We use a two-branch encoder to disentangle texture and structure from the sparse input images; it includes a texture-aware branch and a structure-aware branch. For both branches, to cope with occlusion and large view differences, pixels in the input images should not have the same contributions. We thus use a spatio-view attention based on calculating confidence maps for multi-view images to obtain the final texture representation f_{tex} and structure representation f_{str}. See Section 3.3.

In the texture-aware branch (see Fig. 2), we use an hourglass network F_{warp} to predict a warping field w_i and a confidence map c^i_{tex} for each input view i, which takes the target pose p_{t}, the i-th source image I^i_{s}, and the i-th source pose p^i_{s} as inputs:

$$(w_i, c^i_{\text{tex}}) = F_{\text{warp}}(p_{\text{t}}, I^i_{\text{s}}, p^i_{\text{s}}) \tag{3}$$

The warping field w_i is represented by displacements between the source image and the target image. Camera poses p_{t} and p^i_{s} are represented by quaternions. We expand the dimensions of the quaternion to match the dimensions of the image, and then concatenate them to form the input. The confidence map c^i_{tex} is used to fuse the feature maps from different views. c^i_{tex} and w_i share all weights of F_{warp} except for their output layers. We use a fully convolutional network F_{tex} to extract features $\tilde{f}^i_{\text{tex}}$ from the source images, and then warp the features to get the target features f^i_{tex}:

$$\tilde{f}^i_{\text{tex}} = F_{\text{tex}}(I^i_{\text{s}}) \tag{4}$$

$$f^i_{\text{tex}} = \mathcal{W}(w_i, \tilde{f}^i_{\text{tex}}) \tag{5}$$

where $\mathcal{W}(\cdot)$ is the warping function; bilinear sampling is used in our network.

In the structure-aware branch, we use an encoder F_{str} [10] consisting of a series of 2D convolutions, reshaping, and 3D convolutions to extract a hybrid feature field represented as a structure feature volume for each image:

$$\tilde{f}^i_{\text{str}} = F_{\text{str}}(I^i_{\text{s}}) \tag{6}$$

where $\tilde{f}^i_{\text{str}}$ is the structure feature volume in the corresponding pose p^i_{s}. Each voxel in our 3D feature volume corresponds to a point in 3D space and represents information like its color, and whether it is inside the object or not. Such a 3D feature volume is more robust than a 2D feature map that represents depth information, and has been widely used in 3D reconstruction and novel view synthesis. It is also reasonable to reshape a 2D feature map to get a 3D feature volume. A feature map with $[c \times d]$-dimensional channels can be treated as a concatenation of d feature maps with c dimensional channels, each of which represents geometry and appearance information for a slice in 3D space. Thus, the feature map with $[c \times d]$-dimensional channels contains d slices in 3D space and can be reshaped to a 3D feature volume with a depth resolution of d. Next, we rotate $\tilde{f}^i_{\text{str}}$ from the source pose p^i_{s} to the target pose p_{t}:

$$f^i_{\text{str}} = \mathcal{R}(\tilde{f}^i_{\text{str}}, p^i_{\text{s}}, p_{\text{t}}), \quad c^i_{\text{str}} = \text{3DConv}(f^i_{\text{str}}) \tag{7}$$

where $\mathcal{R}(\cdot)$ is a rotation operation with trilinear sampling, f^i_{str} is the transformed feature volume having the same shape as $\tilde{f}^i_{\text{str}}$, and $\text{3DConv}(\cdot)$ represents 3D convolution. The confidence map c^i_{str} is used to fuse the feature maps from different views.

The texture representation f_{tex} and the structure representation f_{str} are decoded by a fusion decoder described in Section 3.4.

3.3 Spatio-view attention aggregation

Due to occlusions and large view variation, the texture representation f^i_{tex} of view i may be incomplete. Missing regions should not have the same weighting as other regions. Moreover, the visible view should have more impact on the final result. Similarly, the structure-aware branch requires different weights for different regions of f^i_{str} and different views. Therefore, instead of simply averaging the encoded feature maps, we apply adaptive aggregation with spatio-view attention for the texture-aware encoder and the structure-aware encoder by calculating a confidence map for each view, as shown in Fig. 3. The pixel-wise and voxel-wise confidence maps $\{c^i_{\text{tex}}\}_{1\leqslant i\leqslant N}$ and $\{c^i_{\text{str}}\}_{1\leqslant i\leqslant N}$ are used to fuse the texture features and structure features of all views using

$$f_{\text{tex}} = \sum_{i=1}^{N} f^i_{\text{tex}} \odot \text{Softmax}_i(c^1_{\text{tex}}, \ldots, c^N_{\text{tex}}) \tag{8}$$

$$f_{\text{str}} = \sum_{i=1}^{N} f^i_{\text{str}} \odot \text{Softmax}_i(c^1_{\text{str}}, \ldots, c^N_{\text{str}}) \tag{9}$$

We normalize the predicted confidence maps $\{c^i_{\text{tex}}\}_{1\leqslant i\leqslant N}$ and $\{c^i_{\text{str}}\}_{1\leqslant i\leqslant N}$ by applying $\text{Softmax}(\cdot)$ across them. The normalized confidence maps can then be used as the weights to aggregate the feature maps. This mechanism enables the weights to be automatically adjusted for any number of input views, which is very flexible. Moreover, fusion at the feature level needs less memory yet can produce a more continuous result.

3.4 Fusion decoder

The fusion decoder fuses the texture feature map and the structure feature volume, and then generates the final image. After several 3D convolutions, the

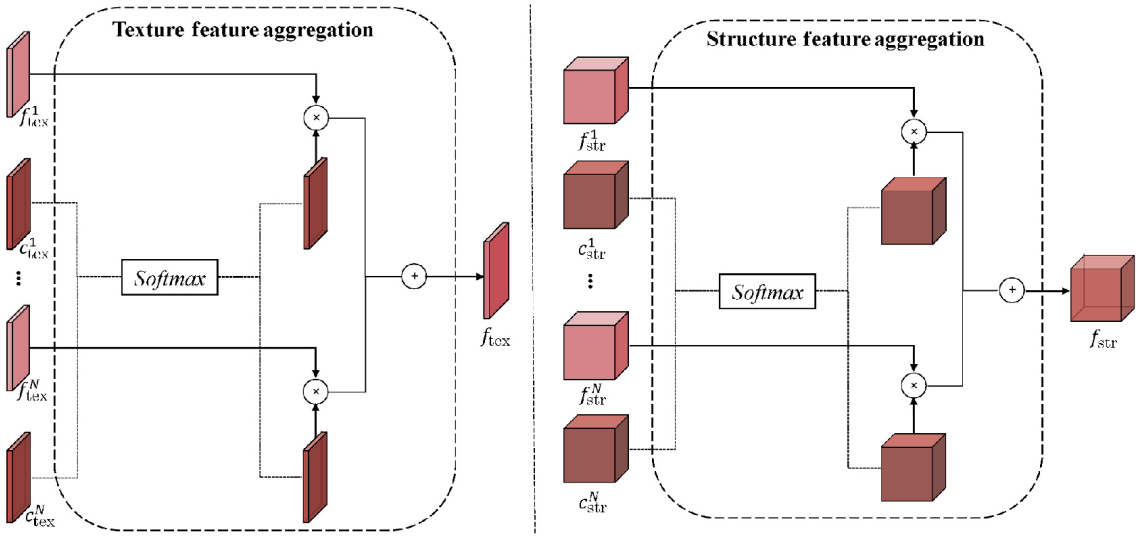

Fig. 3 Structures for spatio-view attention aggregation in the texture-aware branch (left) and structure-aware branch (right).

structure feature volume is turned into a structure feature map by merging the depth dimension into the channel dimension. We concatenate the structure feature map and the texture feature map, and then get the final image using a U-Net decoder. Instead of fusion at the pixel level, we fuse the structure representation and the texture representation at the feature level, for three reasons: (i) it is difficult to ensure the alignment of two-branch results, (ii) the features before the decoder contain more information than the decoded images, and (iii) fusion at the feature level enables the network to generate new content, especially for the invisible regions.

3.5 Loss functions

Because STATE is an end-to-end trainable network, we directly define several losses in image space to train our network. Our full training loss consists of a reconstruction term, a structural term, a perceptual term, a cosine term, and an adversarial term. The full loss is formulated as

$$\mathcal{L} = \lambda_{\mathrm{r}}\mathcal{L}_{\mathrm{R}} + \lambda_{\mathrm{s}}\mathcal{L}_{\mathrm{S}} + \lambda_{\mathrm{p}}\mathcal{L}_{\mathrm{P}} + \lambda_{\mathrm{c}}\mathcal{L}_{\mathrm{C}} + \lambda_{\mathrm{a}}\mathcal{L}_{\mathrm{A}} \quad (10)$$

where λ_{r}, λ_{s}, λ_{p}, λ_{c}, and λ_{a} weight the five loss terms.

The reconstruction loss directly guides the similarity between the generated image $\hat{I}_{\mathrm{t}}$ and the ground-truth image I_{t} at the pixel level, accelerating convergence. $\mathcal{L}_{\mathrm{R}}$ is defined as the ℓ_1 distance:

$$\mathcal{L}_{\mathrm{R}} = \left\| \hat{I}_{\mathrm{t}} - I_{\mathrm{t}} \right\|_1 \quad (11)$$

We use the structural similarity (SSIM) loss $\mathcal{L}_{\mathrm{S}}$ [37] with a window size of 11×11 to improve the structural similarity, and to improve consistency with human perception. The structural dissimilarity between the generated image $\hat{I}_{\mathrm{t}}$ and the ground-truth image I_{t} is given by

$$\mathcal{L}_{\mathrm{S}} = 1 - \mathrm{SSIM}(\hat{I}_{\mathrm{t}}, I_{\mathrm{t}}) \quad (12)$$

In addition to low-level constraints at the pixel level, we adopt perceptual loss [38] to compute the difference between the deep features of the generated image $\hat{I}_{\mathrm{t}}$ and the ground-truth image I_{t} at a perceptual level; this is formulated as

$$\mathcal{L}_{\mathrm{P}} = \sum_i \left\| \phi_i(\hat{I}_{\mathrm{t}}) - \phi_i(I_{\mathrm{t}}) \right\|_2 \quad (13)$$

where ϕ_i is the output of the i-th layer of VGG-19 [39] pre-trained on ImageNet [40]. We use layers 1, 6, 11, and 16 to supervise our network.

To ensure color consistency, we calculate the cosine similarity between the generated image $\hat{I}_{\mathrm{t}}$ and the ground-truth image I_{t}. Cosine similarity measures the similarity between two vectors by measuring the cosine of the angle between them:

$$\mathcal{L}_{\mathrm{C}} = 1 - \cos(\hat{I}_{\mathrm{t}}, I_{\mathrm{t}}) \quad (14)$$

We adopt the discriminator from generative adversarial networks [41], which has achieved great progress in image synthesis. It constrains the distance between the distributions of the generated image $\hat{I}_{\mathrm{t}}$ and the ground-truth image I_{t}. The discriminator loss is defined as

$$\mathcal{L}_{\mathrm{A}} = \mathbb{E}[\log(1 - D(\hat{I}_{\mathrm{t}}))] + \mathbb{E}[\log D(I_{\mathrm{t}})] \quad (15)$$

where $D(\cdot)$ is a patch discriminator, $\log(\cdot)$ is the base 2 logarithm, and $\mathbb{E}[\cdot]$ is the expectation.

3.6 Implementation details

Our framework is implemented in PyTorch. The hyper-parameters $[\lambda_{\mathrm{r}}, \lambda_{\mathrm{s}}, \lambda_{\mathrm{p}}, \lambda_{\mathrm{c}}, \lambda_{\mathrm{a}}]$ were set to $[1, 10, 0.5, 1, 1]$ for training. The Adam optimizer [42] was used to optimize our network with the default parameters ($\beta_1 = 0.9$ and $\beta_2 = 0.999$) and learning rate 2×10^{-4}. We trained our model with four source view images until convergence on the training data, which took approximately 7 days using a single GeForce GTX 2080 Ti GPU. During testing, generating an image takes about 90 ms using a single GeForce GTX 2080 Ti GPU.

4 Experiments

4.1 Datasets

To evaluate the performance of our view synthesis approach, we conducted experiments on ShapeNet (using *Chair* and *Car*) [43], in which the camera poses are represented by the rotation components around the object's central axis. We used the same training and testing splits as Refs. [4, 5, 10, 21] (80% of models for training and the remaining 20% for testing). Each model was rendered as 256×256 RGB images at 18 azimuth angles sampled at 20° intervals and 3 elevations (0°, 10°, 20°), for a total of 54 viewpoints per model.

We also synthesized a dataset *Human* from 496 real scanned 3D human models from `https://web.twindom.com`. Each model was rendered as 256×256 RGB images at 18 azimuth angles sampled at 20° intervals and 3 elevations (0°, 10°, 20°), for a total of 54 viewpoints per model. We used 80% of the models for training and the remaining 20% for testing.

Models in the test images were not included in the training set.

4.2 Metrics

We used two popular metrics, *learned perceptual image patch similarity* (LPIPS) [44] and *Fréchet inception distance* (FID) [45], which are generally considered to be closer to human perception, to assess the reconstruction errors. LPIPS computes the distance between the generated image and the ground-truth image in the perceptual domain. FID calculates the Wasserstein-2 distance between the distributions of the generated images and the ground-truth images, which measures the realism of the generated images.

4.3 Ablation study

We first evaluate our method against four alternative models to determine the factors that contribute to achieving reasonable view synthesis from sparse input images. These models use the same setup, training schedule, and sequence of input images as STATE. We used the same training and test scheme as Refs. [4, 10] for *Chair*, *Car*, and *Human* datasets: training with 4 views and testing with 1–4 views.

The alternative models were as follows.

w/o Tex. This model omits the texture-aware branch but retains the multi-view adaptive weighting. It is designed to assess the importance of the texture-aware branch, and to verify the necessity of the combination of both texture representation and structure representation.

w/o Str. The model omits the structure-aware branch but retains the multi-view adaptive weighting. It is designed to assess the importance of the structure-aware branch, and to verify the necessity of the combination of both texture representation and structure representation.

w/o SVA. This model is trained with multi-view averaging fusion, to assess the importance of spatio-view attention.

w/o Cos. This model omits cosine loss to assess the importance of cosine loss.

Full. Our full model includes the two-branch encoder and multi-view fusion at the feature level with adaptive weighting.

Table 1 gives quantitative results for *Chair*, *Car*, and *Human* datasets. Our full model outperforms all the alternatives on *Chair* and *Car* datasets in terms of LPIPS and FID. Note that spatio-view attention aggregation is not used when the test input is a single view. Therefore, the LPIPS values of the w/o SVA model and the Full model are similar on *Human* dataset. On the other hand, all the models in the ablation study are trained on input with four views, and different confidences are assigned to different views due to the SVA module of the full model. However, when the test input is a single view with low confidence, the results may be affected. Furthermore, the clothed posed human has complex color and is asymmetric, which influences the learning of structures. Therefore, the FID of the Full model is slightly worse than that of the w/o Cos. model for four views input for the *Human* dataset. Various visual results are presented in Figs. 4–7. It can be seen that the w/o Tex. model can generate correct

Table 1 Quantitative comparison of four alternative designs

Dataset	Method	1 view		2 views		3 views		4 views	
		LPIPS↓	FID↓	LPIPS↓	FID↓	LPIPS↓	FID↓	LPIPS↓	FID↓
Car	w/o Tex.	0.139	79.143	0.104	57.997	0.096	54.261	0.092	52.961
	w/o Str.	0.127	64.788	0.098	44.501	0.089	39.765	0.084	37.901
	w/o SVA	0.118	62.619	0.090	42.023	0.081	38.642	0.078	37.258
	w/o Cos.	0.136	82.208	0.104	57.810	0.096	53.844	0.092	52.462
	Full	**0.117**	**60.387**	**0.089**	**39.052**	**0.080**	**34.472**	**0.075**	**32.290**
Chair	w/o Tex.	0.250	64.584	0.113	21.622	0.096	19.488	0.092	18.898
	w/o Str.	0.166	33.330	0.141	26.628	0.133	25.145	0.129	24.443
	w/o SVA	0.209	48.731	0.100	19.228	0.086	17.336	0.081	16.730
	w/o Cos.	0.246	62.418	0.109	20.006	0.093	17.998	0.088	17.461
	Full	**0.159**	**30.936**	**0.096**	**18.486**	**0.080**	**16.547**	**0.074**	**15.881**
Human	w/o Tex.	0.118	70.431	0.087	64.174	0.082	64.860	0.081	65.550
	w/o Str.	0.106	82.642	0.088	76.567	0.081	75.137	0.078	75.357
	w/o SVA	**0.102**	61.274	0.078	57.386	0.072	57.710	0.069	58.330
	w/o Cos.	0.110	62.791	0.082	56.604	0.077	56.487	0.076	**56.525**
	Full	0.105	**60.056**	**0.076**	**55.802**	**0.070**	**56.469**	**0.068**	57.055

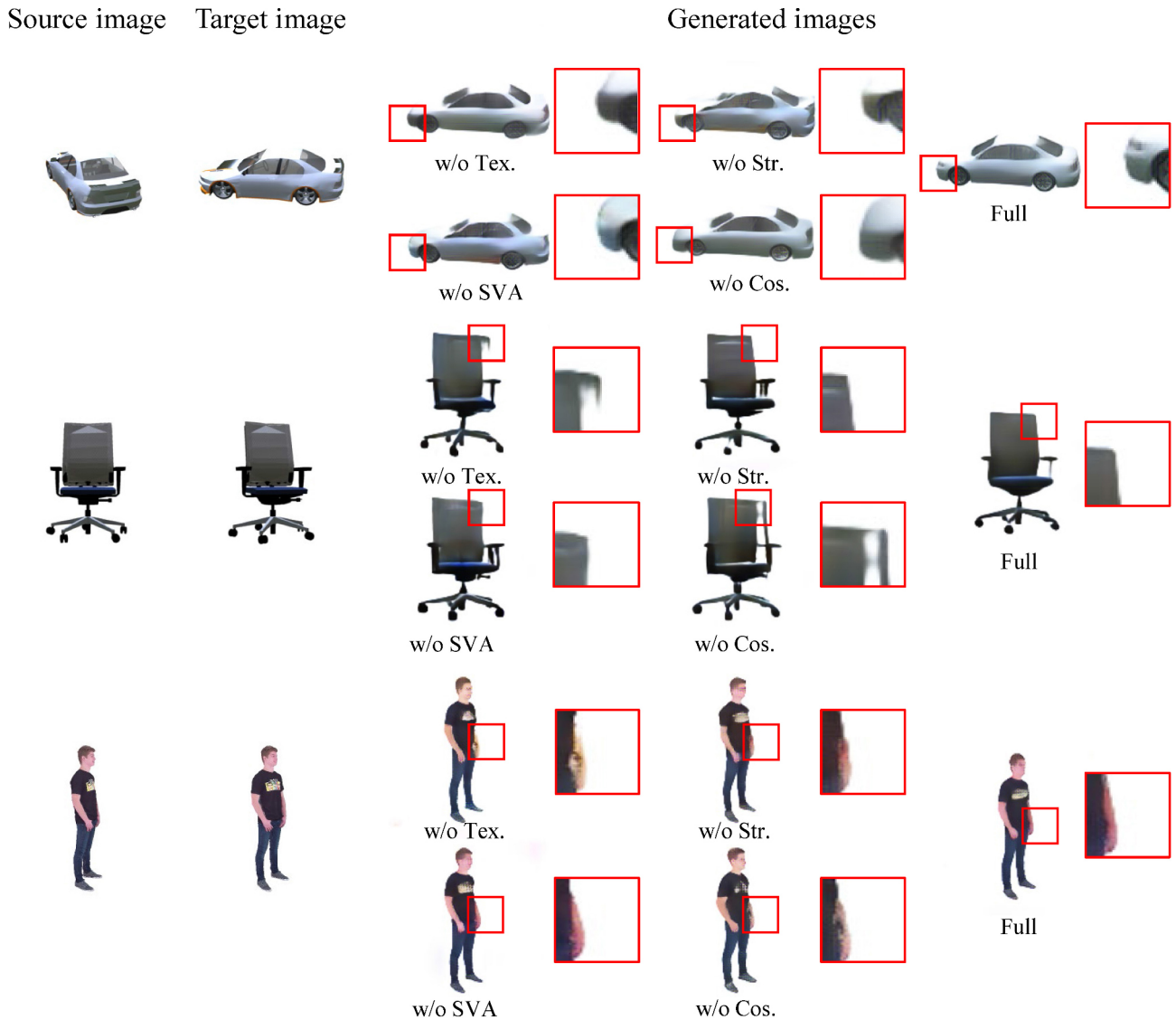

Fig. 4 Qualitative comparison with four alternative designs for single-view inputs.

structures, but the textures in the source images are not well maintained, e.g., the bonnet of the car. The w/o Str. model can recover detailed textures, especially for *Car* and *Human* datasets, but fails to maintain shape consistency. The w/o SVA model fails to effectively fuse the results of the two branches, and thus the results lose some textures or structures, such as the texture of the car, the back and the legs of the chair, and the human's arms. The w/o Cos. model cannot ensure color consistency, e.g., on the bonnet of the car. However, our full model can achieve consistency of color, texture, and structure.

To verify the disentanglement of textures and structures, we visualize the results of the two branches: we output the result of one branch by zeroing out the features of the other branch. Figure 8 demonstrates that our method can effectively disentangle textures and structures to generate realistic images with correct shapes and textures.

We visualize the confidence maps to demonstrate the effect of spatio-view attention aggregation in Fig. 9, taking novel view synthesis from two views as an example. The first two columns are the source images, the third column is the generated image, and the last two columns multiply the confidence map by the generated image. As can be seen, the generated image obtains more texture information from source image 2 due to its similarity to the target view, showing that our spatio-view attention aggregation can select more relevant information from different input views.

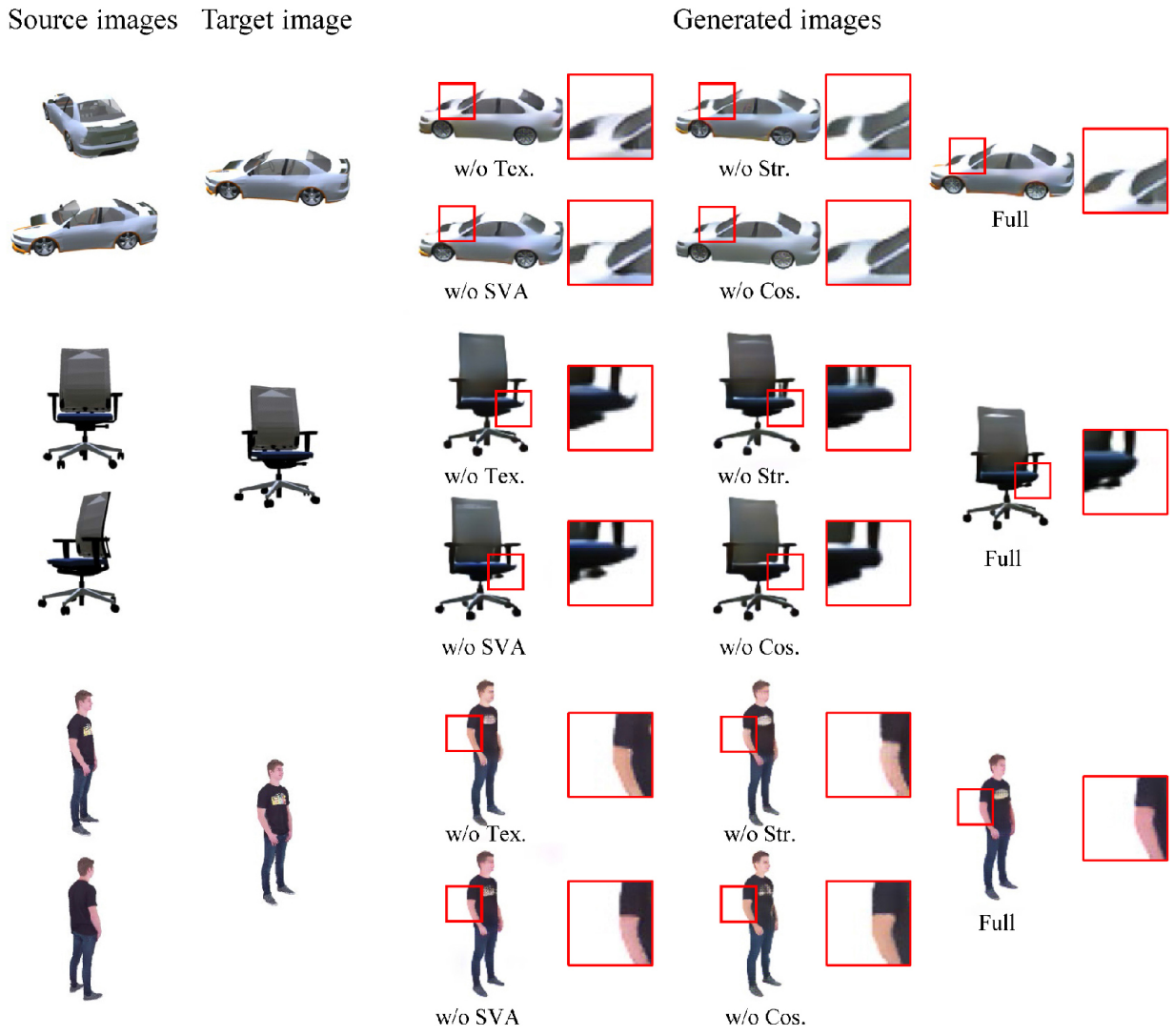

Fig. 5 Qualitative comparison with four alternative designs for two-view inputs.

Further results are given in the Electronic Supplementary Material (ESM).

4.4 Comparison to other methods

We compare our method to TBN [10] and pixelNeRF [11]. For simplicity, we omit comparisons to earlier works [4, 9] that have already been compared to TBN or pixelNeRF, and the methods that do not work well for sparse views [30, 35, 46, 47]. We use the same training and test scheme as TBN [10] for *Chair* and *Car* datasets: training with 4 views and testing with 1–4 views. For the case of single-view input, we use a single view for training, as multi-view adaptive weighting is not used. Pre-trained TBN [10] models for *Chair* and *Car* datasets were used and we re-trained TBN [10] on the *Human* dataset for a fair comparison, using the same training and test scheme: training with 4 views and testing with any number of views. We also re-trained pixelNeRF [11] on the *Car*, *Chair*, and *Human* datasets for fair comparison: training with 4 views and testing with 2–4 views. For single-view input, we used single view for training as suggested by the author.

Table 2 provides a quantitative comparison on *Chair*, *Car*, and *Human* datasets. It can be seen that our proposed method outperforms the other methods in terms of FID by a significant margin on the *Chair* dataset, even in the challenging case of single-view input. For the *Car* dataset, benefiting from spatio-view attention, our method achieves the best performance for multi-view inputs. The cars are left–right symmetric, but not front–back. As a result, our texture-aware branch finds it difficult to provide reasonable textures when there is heavy occlusion in

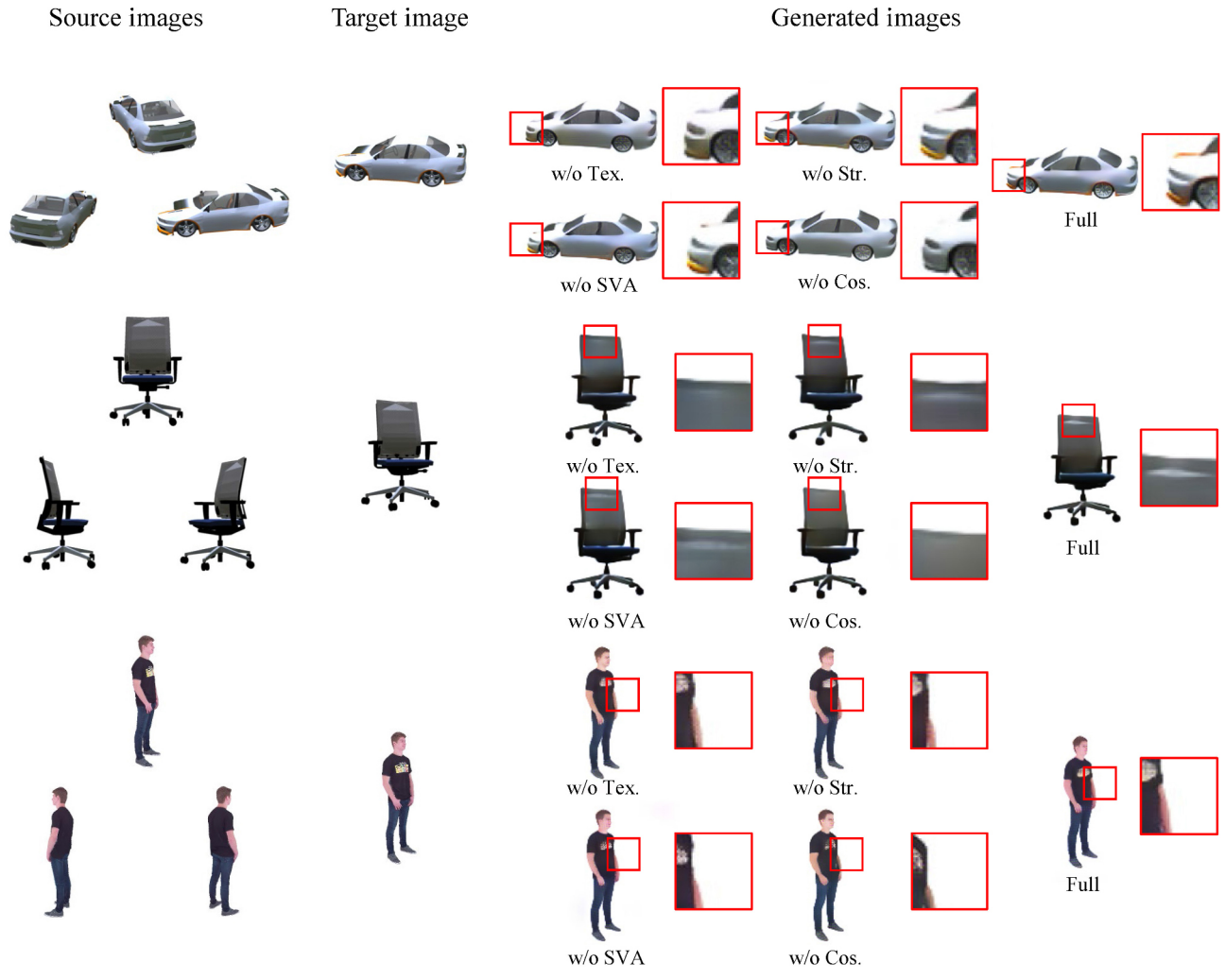

Fig. 6 Qualitative comparison with four alternative designs for three-view inputs.

Table 2 Quantitative comparison on the *Chair*, *Car*, and *Human* datasets

Dataset	Method	1 view		2 views		3 views		4 views	
		LPIPS↓	FID↓	LPIPS↓	FID↓	LPIPS↓	FID↓	LPIPS↓	FID↓
Chair	TBN [10]	0.182	38.446	0.109	21.159	0.093	18.891	0.086	18.051
	pixelNeRF [11]	0.183	40.515	0.181	71.560	0.095	28.588	**0.068**	18.118
	Ours	**0.159**	**30.936**	**0.096**	**18.486**	**0.080**	**16.547**	0.074	**15.881**
Car	TBN [10]	**0.112**	**46.401**	0.091	40.404	0.084	38.841	0.080	38.129
	pixelNeRF [11]	0.155	91.252	0.145	89.553	0.101	55.887	0.083	41.496
	Ours	0.117	60.387	**0.089**	**39.052**	**0.080**	**34.472**	**0.075**	**32.290**
Human	TBN [10]	0.187	92.368	0.093	**51.535**	0.083	**51.573**	0.080	**52.262**
	pixelNeRF [11]	0.137	84.211	0.102	67.718	0.078	60.250	**0.068**	61.453
	Ours	**0.105**	**60.056**	**0.076**	55.802	**0.070**	56.469	**0.068**	57.055
Average	TBN [10]	0.160	59.072	0.098	**37.699**	0.087	36.435	0.082	36.147
	pixelNeRF [11]	0.158	71.993	0.143	76.277	0.091	48.242	0.073	40.256
	Ours	**0.127**	**50.460**	**0.087**	37.780	**0.077**	**35.829**	**0.072**	**35.075**

front of or behind the car for a single view, leading to some faults in the final textures, even if the shape estimated by the structure-aware branch is accurate.

For the *Human* dataset, our method achieves the best LPIPS scores for all cases. The clothed posed human has complex color and is asymmetric, which influences

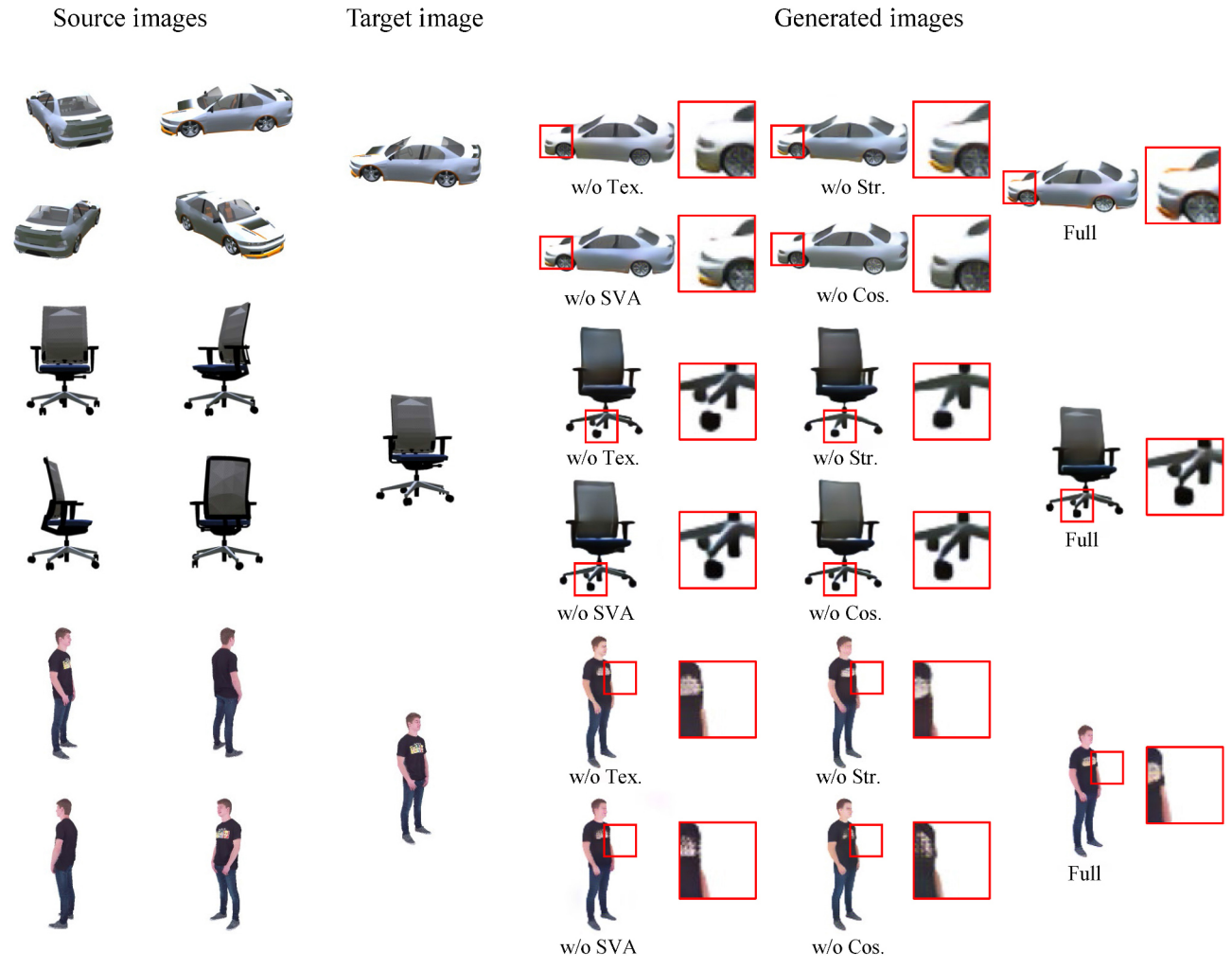

Fig. 7 Qualitative comparison with four alternative designs for four-view inputs.

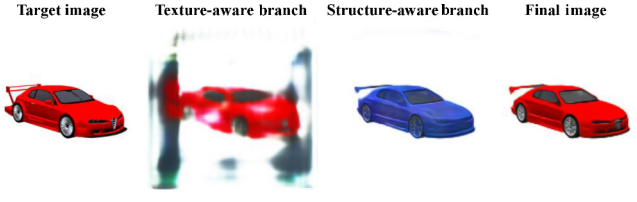

Fig. 8 Disentanglement of textures and structures.

structure learning. Therefore, our FID scores are not the best for multi-view inputs on the *Human* dataset. Nevertheless, considering the average results over all datasets, our method achieves the best results for all views except for FID scores for two views.

Visual results for several challenging examples with large viewpoint transformations from *Chair*, *Car*, and *Human* datasets are shown in Figs. 10–12. Due to the representation's limited resolution, TBN [10] finds it difficult to recover image details, such as chair legs, and textures of cars and people. PixelNeRF [11] generates certain artifacts along structural edges. In contrast, our method provides detailed textures while maintaining the structures of objects: e.g., see the stripes on the car and the suit on the person. Thanks to the disentangled learning of the structure representation and the texture representation, invisible regions and detailed textures are successfully recovered by our method for any number of input views. By fusing and decoding the two representations, our method does not suffer from missing pixels: our method can generate visually better and more realistic images.

Further results are given in the ESM.

4.5 User study

To better evaluate our method, we performed a perceptual evaluation with a user study, including a comparison to state-of-the-art methods. We showed results from TBN [10] (Method A), pixelNeRF [11]

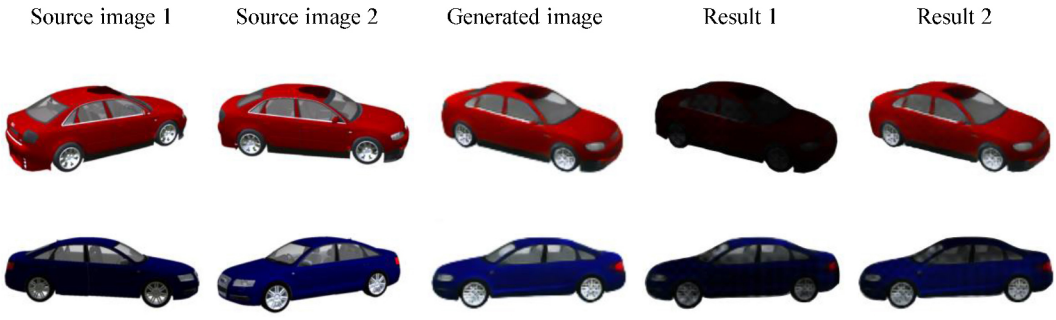

Fig. 9 Confidence maps for different views.

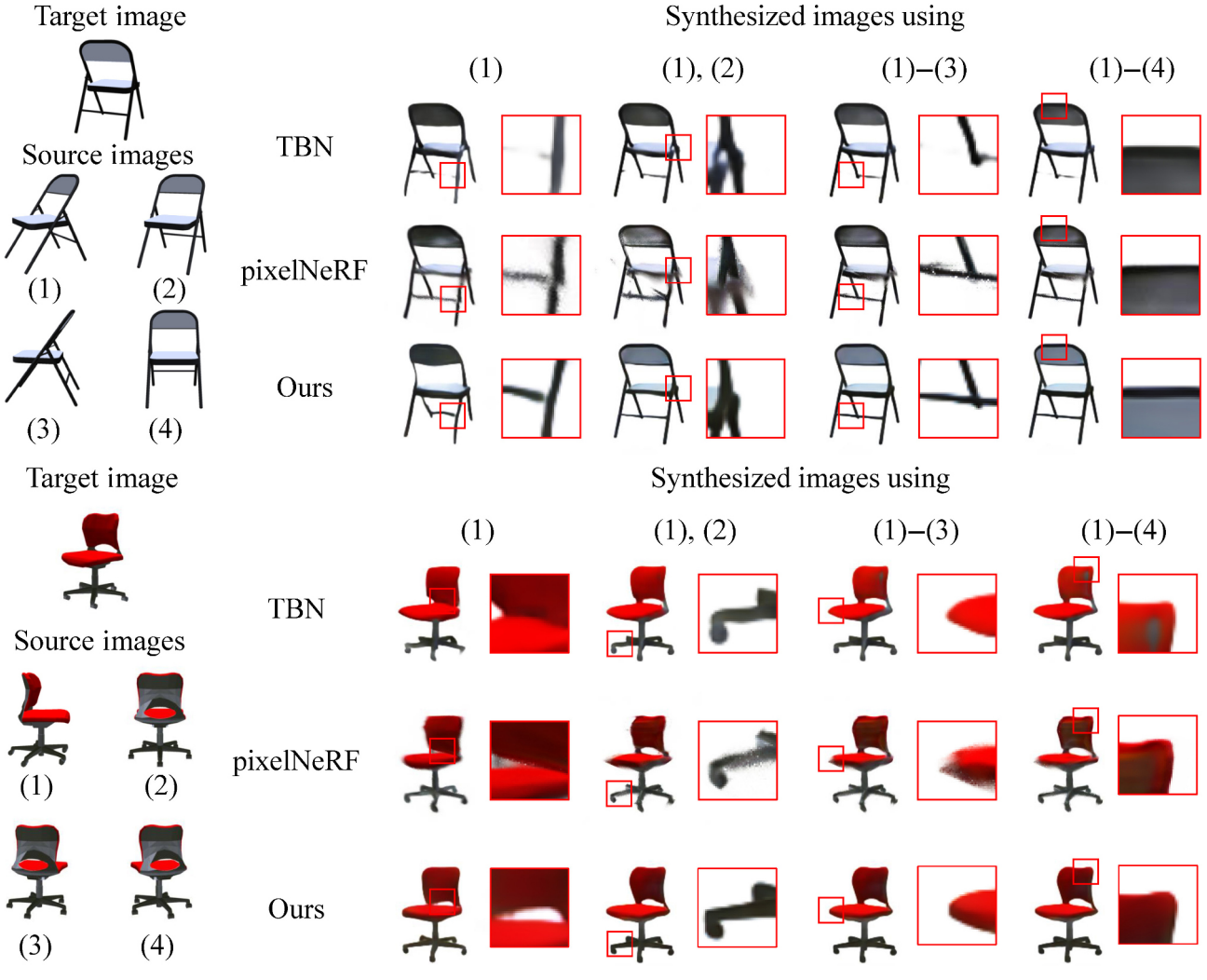

Fig. 10 Qualitative comparison on the *Chair* dataset.

(Method B), our method (Method C), and the ground-truth for the same input images, for twelve cases, with three questions per case (38 questions in total including asking the gender and age of the participant): 1–4 views as input on the *Car*, *Chair*, and *Human* datasets. The results shown were randomly selected, and the users are required to choose from A, B, and C the one closest to the ground-truth in terms of texture, structure, and overall quality for each case. We collected 111 sets of answers, from 59 females and 52 males, with 108 users aged between 18 and 40, 1 user between 40 and 60, and 2 users over 60. Table 3 presents results of the user study. For each gender, we give the percentage of participants

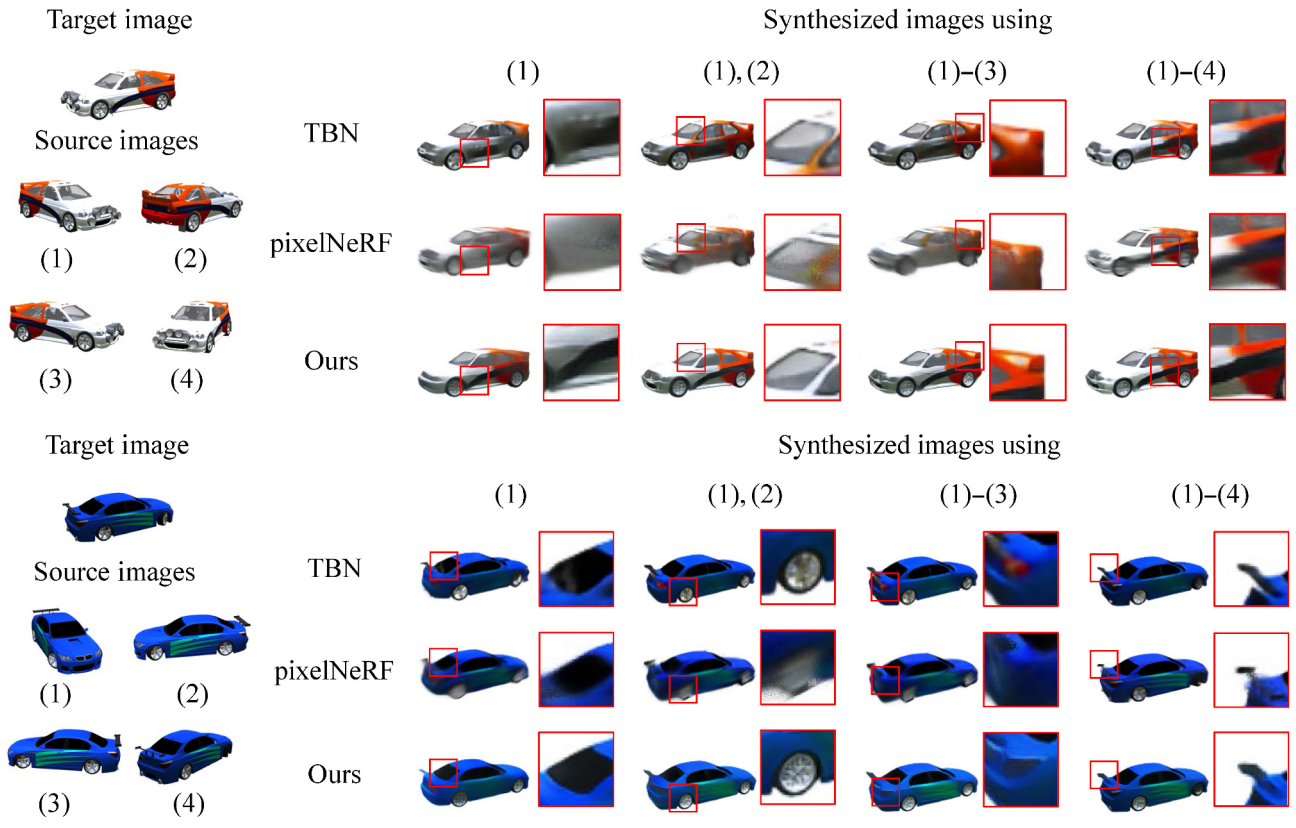

Fig. 11 Qualitative comparison on the *Car* dataset.

Table 3 User study results

Case	Females			Males			Independent T test	
	Method A	Method B	Method C	Method A	Method B	Method C	t	p
Case 1	27.12%	16.95%	**55.93%**	20.51%	9.62%	**69.87%**	−1.487	0.140
Case 2	16.38%	15.82%	**67.80%**	6.41%	15.38%	**78.21%**	−1.875	0.064
Case 3	13.56%	14.12%	**72.32%**	8.97%	11.54%	**79.49%**	−1.072	0.286
Case 4	20.34%	12.99%	**66.67%**	19.23%	14.10%	**66.67%**	−0.086	0.932
Case 5	16.38%	13.00%	**70.62%**	18.59%	7.69%	**73.72%**	−0.067	0.946
Case 6	9.61%	10.73%	**79.66%**	13.46%	8.33%	**78.21%**	0.510	0.611
Case 7	16.95%	9.60%	**73.45%**	7.05%	11.54%	**81.41%**	−1.685	0.095
Case 8	15.25%	14.13%	**70.62%**	8.33%	9.62%	**82.05%**	−1.774	0.079
Case 9	15.82%	14.69%	**69.49%**	10.90%	8.33%	**80.77%**	−1.437	0.154
Case 10	16.95%	15.82%	**67.23%**	16.67%	10.25%	**73.08%**	−0.489	0.626
Case 11	14.69%	14.69%	**70.62%**	9.62%	13.46%	**76.92%**	−0.979	0.330
Case 12	12.99%	9.61%	**77.40%**	12.82%	8.97%	**78.21%**	−0.087	0.931
Average	16.40%	13.47%	**70.13%**	12.89%	10.76%	**76.35%**	−1.115	0.267

who chose the result from a particular method for each case, as well as average results over the twelve cases. In addition to the percentage, we also carried out an independent T test [48] between the result and gender: t is a statistical variable calculated from the results and p is found from a table according to t. A p value greater than 0.05 means there is no significant difference between the results for the two genders. We use 1, 2, and 3 to represent Methods A, B, and C, respectively, and average the results of the three questions in each case. Table 3 shows that the user study results do not depend on gender. Overall, our method achieves the best results in the user study.

Further results are given in the ESM.

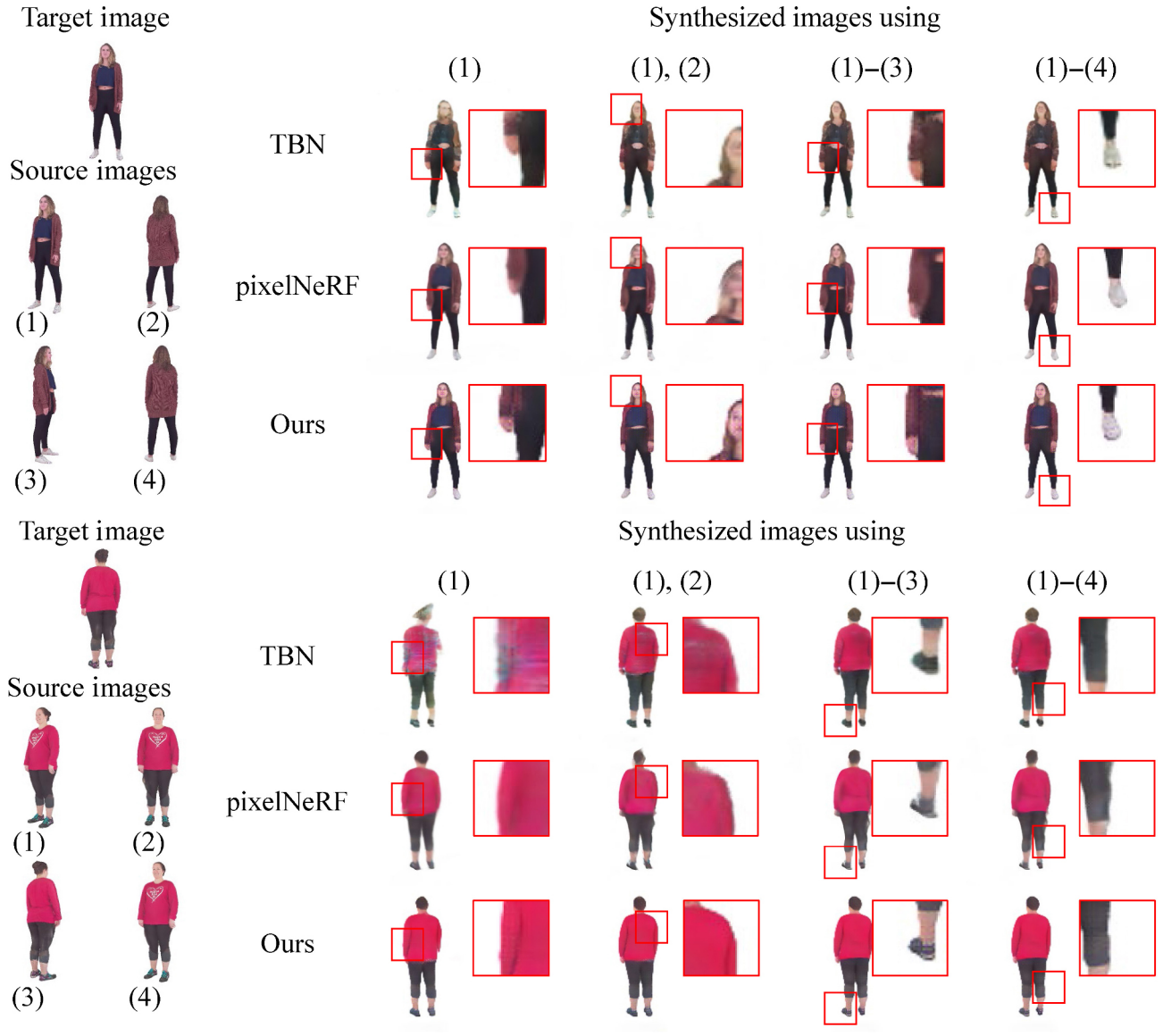

Fig. 12 Qualitative comparison on the *Human* dataset.

4.6 Applications

Our method does not explicitly constrain texture and structure, but as the branches are capable of generating better structure and texture respectively, this implicitly leads to disentanglement. We may also achieve texture or structure swapping with trained models for novel view synthesis.

Using the texture and structure branches, we can easily edit the texture and the structure by changing the inputs to each branch. Figures 13 and 14 show some disentangled results on the *Car* and *Chair* datasets. The first row provides the texture information and the first column gives the structure information. Each result in other positions i, j uses a decoded result of that combination of structure representation and texture representation. It can be

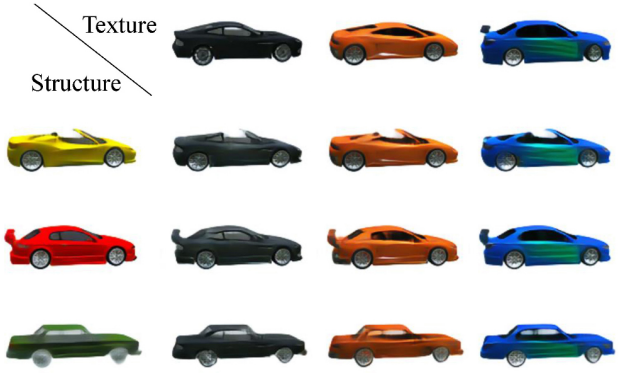

Fig. 13 Results of texture or structure swapping on the *Car* dataset.

Fig. 14 Results of texture or structure swapping on the *Chair* dataset.

seen that the structure of the result in each case is consistent with that of the first item in the row, and the texture of the result is consistent with the top item in this column. Figure 15 shows some disentangled results for various views, showing that our method achieves the disentanglement of texture and structure.

4.7 Failure cases

Although our method generates realistic images with reasonable structures and detailed textures in most cases, it cannot cope well with the structures and textures that deviate greatly from the training set distribution. The neural network predicts outputs by interpolation within in the manifold built on the training data. Therefore, it is difficult to predict reasonable results for some challenging cases, especially those with extremely complex structures and textures. Figure 16 shows examples in which our method fails to predict correct textures and shapes for extremely complex cases.

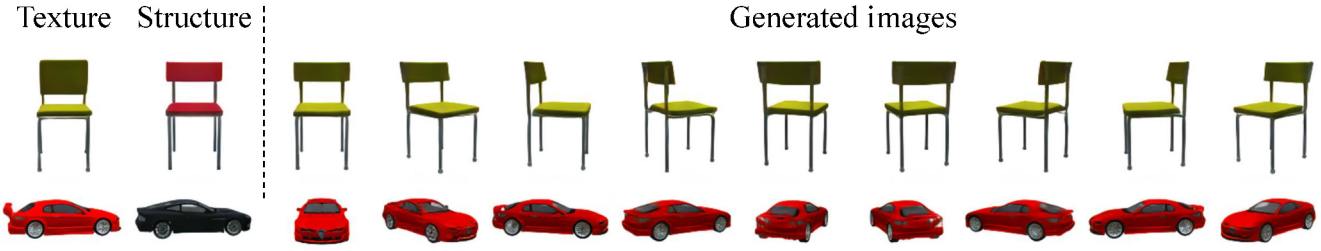

Fig. 15 Results of texture or structure swapping for various views.

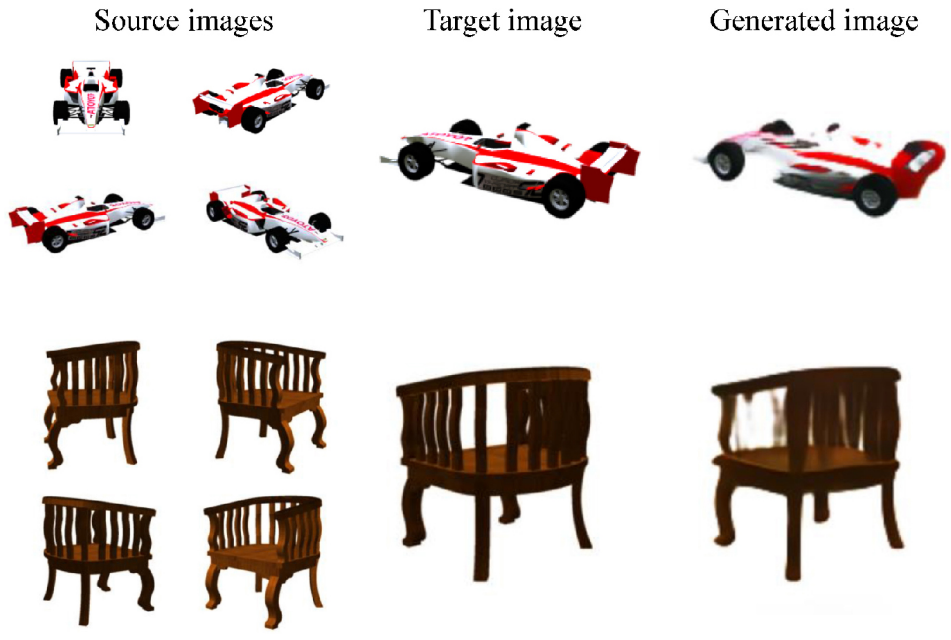

Fig. 16 Failures.

5 Conclusions

In this paper, we propose STATE, an end-to-end deep neural network, for view synthesis from sparse input images by learning structure and texture representations. Specifically, we propose a two-branch encoder to extract implicit structure representation and deformed texture representation. We also propose spatio-view attention to adaptively fuse multi-view information at the feature level by regressing pixel-wise or voxel-wise confidence maps. By decoding the aggregated feature, STATE can generate realistic images with reasonable structures and detailed textures. Experimental results demonstrate that our method works better than current state-of-the-art methods. We have validated our approach via a comprehensive ablation study. Our method enables texture and structure editing applications benefiting from implicit disentanglement of structures and textures.

Despite its good novel view synthesis results, the training efficiency of our method is not high. Our method is implemented in PyTorch, and it takes approximately 7 days to train the model for four source images using a single GeForce GTX 2080 Ti GPU. In future, we hope to improve training efficiency using a Jittor model [49, 50], which is 2.26 times faster than the equivalent PyTorch model on average.

Availability of data and materials

Our code and further results are available at `http://cic.tju.edu.cn/faculty/likun/projects/STATE`.

Funding

This work was supported in part by the National Natural Science Foundation of China (62171317 and 62122058).

Acknowledgements

We are grateful to the Associate Editor and anonymous reviewers for their help in improving this paper.

Author contributions

Xinyi Jing: theoretical development, experiment implementation, paper writing, approving the final version of the article publication, including references.
Qiao Feng: theoretical development, experiment implementation, paper writing, approving the final version of the article for publication, including references.
Yu-Kun Lai: guidance, theoretical development, experimental design, paper revision, approving the final version of the article for publication, including references.
Jinsong Zhang: theoretical development, experimental design, paper revision, approving the final version of the article for publication, including references.
Yuanqiang Yu: theoretical development, experiment implementation, paper writing, approving the final version of the article for publication, including references.
Kun Li: guidance, theoretical development, experimental design, paper writing, approving the final version of the article for publication, including references.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

Electronic Supplementary Material

Supplementary material is available in the online version of this article at `https://doi.org/10.1007/s41095-022-0301-9`.

References

[1] Tatarchenko, M.; Dosovitskiy, A.; Brox, T. Multi-view 3D models from single images with a convolutional network. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9911*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 322–337, 2016.

[2] Yang, J.; Reed, S. E.; Yang, M.-H.; Lee, H. Weakly-supervised disentangling with recurrent transformations for 3D view synthesis. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, Vol. 1, 1099–1107, 2015.

[3] Ren, Y. R.; Yu, X. M.; Chen, J. M.; Li, T. H.; Li, G. Deep image spatial transformation for person image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7687–7696, 2020.

[4] Sun, S. H.; Huh, M.; Liao, Y. H.; Zhang, N.; Lim, J. J. Multi-view to novel view: Synthesizing novel views

with self-learned confidence. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11207*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 162–178, 2018.

[5] Zhou, T. H.; Tulsiani, S.; Sun, W. L.; Malik, J.; Efros, A. A. View synthesis by appearance flow. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9908*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 286–301, 2016.

[6] Flynn, J.; Neulander, I.; Philbin, J.; Snavely, N. Deep stereo: Learning to predict new views from the world's imagery. In: Proceedings of the IEEE Conference on Computer Vision and Pattern, 5515–5524, 2016.

[7] Tulsiani, S.; Zhou, T. H.; Efros, A. A.; Malik, J. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 209–217, 2017.

[8] Lê, H. Â.; Mensink, T.; Das, P.; Gevers, T. Novel view synthesis from single images via point cloud transformation. In: Proceedings of the British Machine Vision Conference, 2020.

[9] Sitzmann, V.; Zollhoefer, M.; Wetzstein, G. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, Article No. 101, 1121–1132, 2019.

[10] Olszewski, K.; Tulyakov, S.; Woodford, O.; Li, H.; Luo, L. J. Transformable bottleneck networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 7647–7656, 2019.

[11] Yu, A.; Ye, V.; Tancik, M.; Kanazawa, A. pixelNeRF: Neural radiance fields from one or few images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4576–4585, 2021.

[12] Ali Eslami, S. M.; Jimenez Rezende, D.; Besse, F.; Viola, F.; Morcos, A. S.; Garnelo, M.; Ruderman, A.; Rusu, A. A.; Danihelka, I.; Gregor, K.; et al. Neural scene representation and rendering. *Science* Vol. 360, No. 6394, 1204–1210, 2018.

[13] Liu, X. F.; Guo, Z. H.; You, J.; Vijaya Kumar, B. V. K. Dependency-aware attention control for image set-based face recognition. *IEEE Transactions on Information Forensics and Security* Vol. 15, 1501–1512, 2020.

[14] Liu, X. F.; Kumar, B. V. K. V.; Yang, C.; Tang, Q. M.; You, J. Dependency-aware attention control for unconstrained face recognition with image sets. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11215*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 573–590, 2018.

[15] Trevithick, A.; Yang, B. GRF: Learning a general radiance field for 3D representation and rendering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 15162–15172, 2021.

[16] Yan, X.; Yang, J.; Yumer, E.; Guo, Y.; Lee, H. Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In: Proceedings of the 30th International Conference on Neural Information Processing Systems, 1704–1712, 2016.

[17] Kim, J.; Kim, Y. M. Novel view synthesis with skip connections. In: Proceedings of the IEEE International Conference on Image Processing, 1616–1620, 2020.

[18] Yin, M. Y.; Sun, L.; Li, Q. L. ID-unet: Iterative soft and hard deformation for view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7216–7225, 2021.

[19] Kwon, Y.; Petrangeli, S.; Kim, D.; Wang, H. L.; Fuchs, H.; Swaminathan, V. Rotationally-consistent novel view synthesis for humans. In: Proceedings of the 28th ACM International Conference on Multimedia, 2308–2316, 2020.

[20] Jaderberg, M.; Simonyan, K.; Zisserman, A.; Kavukcuoglu, K. Spatial transformer networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, Vol. 2, 2017–2025, 2015.

[21] Park, E.; Yang, J. M.; Yumer, E.; Ceylan, D.; Berg, A. C. Transformation-grounded image generation network for novel 3D view synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 702–711, 2017.

[22] Song, J.; Chen, X.; Hilliges, O. Monocular neural image based rendering with continuous view control. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 4089–4099, 2019.

[23] Hou, Y. X.; Solin, A.; Kannala, J. Novel view synthesis via depth-guided skip connections. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 3118–3127, 2021.

[24] Choy, C. B.; Xu, D. F.; Gwak, J.; Chen, K.; Savarese, S. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9912*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 628–644, 2016.

[25] Girdhar, R.; Fouhey, D. F.; Rodriguez, M.; Gupta, A. Learning a predictable and generative vector representation for objects. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9910*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 484–499, 2016.

[26] Kar, A.; Häne, C.; Malik, J. Learning a multi-view stereo machine. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, 364–375, 2017.

[27] Park, J. J.; Florence, P.; Straub, J.; Newcombe, R.; Lovegrove, S. DeepSDF: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 165–174, 2019.

[28] Saito, S.; Huang, Z.; Natsume, R.; Morishima, S.; Li, H.; Kanazawa, A. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2304–2314, 2019.

[29] Guo, P. S.; Bautista, M. A.; Colburn, A.; Yang, L.; Ulbricht, D.; Susskind, J. M.; Shan, Q. Fast and explicit neural view synthesis. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 11–20, 2022.

[30] Lombardi, S.; Simon, T.; Saragih, J.; Schwartz, G.; Lehrmann, A.; Sheikh, Y. Neural volumes: Learning dynamic renderable volumes from images. *ACM Transactions on Graphics* Vol. 38, No. 4, Article No. 65, 2019.

[31] Nguyen-Phuoc, T.; Li, C.; Theis, L.; Richardt, C.; Yang, Y. L. HoloGAN: Unsupervised learning of 3D representations from natural images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 7587–7596, 2019.

[32] Nguyen-Phuoc, T.; Richardt, C.; Mai, L.; Yang, Y. L.; Mitra, N. BlockGAN: Learning 3D object-aware scene representations from unlabelled images. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, Article No. 568, 6767–6778, 2020.

[33] Niemeyer, M.; Mescheder, L.; Oechsle, M.; Geiger, A. Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3501–3512, 2020.

[34] Galama, Y.; Mensink, T. IterGANs: Iterative GANs to learn and control 3D object transformation. *Computer Vision and Image Understanding* Vol. 189, 102803, 2019.

[35] Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; Ng, R. NeRF: Representing scenes as neural radiance fields for view synthesis. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12346*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 405–421, 2020.

[36] Tewari, A.; Fried, O.; Thies, J.; Sitzmann, V.; Lombardi, S.; Sunkavalli, K.; Martin-Brualla, R.; Simon, T.; Saragih, J.; Nießner, M.; et al. State of the art on neural rendering. *Computer Graphics Forum* Vol. 39, No. 2, 701–727, 2020.

[37] Wang, Z.; Bovik, A. C.; Sheikh, H. R.; Simoncelli, E. P. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* Vol. 13, No. 4, 600–612, 2004.

[38] Johnson, J.; Alahi, A.; Li, F. F. Perceptual losses for real-time style transfer and super-resolution. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9906*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 694–711, 2016.

[39] Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint* arXiv:1409.1556, 2014.

[40] Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S. A.; Huang, Z. H.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision* Vol. 115, No. 3, 211–252, 2015.

[41] Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Communications of the ACM* Vol. 63, No. 11, 139–144, 2020.

[42] Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference for Learning Representations, 2015.

[43] Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q. X.; Li, Z. M.; Savarese, S.; Savva, M.; Song, S. R.; Su, H.; et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint* arXiv:1512.03012, 2015.

[44] Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 586–595, 2018.

[45] Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, 6629–6640, 2017.

[46] Chibane, J.; Bansal, A.; Lazova, V.; Pons-Moll, G. Stereo radiance fields (SRF): Learning view synthesis for sparse views of novel scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7907–7916, 2021.

[47] Riegler, G.; Koltun, V. Free view synthesis. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12364*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 623–640, 2020.

[48] Gretton, A.; Fukumizu, C.; Teo, H.; Song, L.; Schölkopf, B.; Smola, A. J. A kernel statistical test of independence. In: Proceedings of the 20th International Conference on Neural Information Processing Systems, 585–592, 2007.

[49] Hu, S. M.; Liang, D.; Yang, G. Y.; Yang, G. W.; Zhou, W. Y. Jittor: A novel deep learning framework with meta-operators and unified graph execution. *Science China Information Sciences* Vol. 63, No. 12, 222103, 2020.

[50] Zhou, W. Y.; Yang, G. W.; Hu, S. M. Jittor-GAN: A fast-training generative adversarial network model zoo based on Jittor. *Computational Visual Media* Vol. 7, No. 1, 153–157, 2021.

Xinyi Jing received her B.E. degree from the School of Computer Science, Shaanxi Normal University, Xi'an, China, in 2020. She is currently pursuing an M.E. degree in the College of Intelligence and Computing, Tianjin University, China. Her research interests are in computer vision and computer graphics.

Qiao Feng received his B.E. degree from the College of Intelligence and Computing, Tianjin University in 2021. He is currently pursuing a master degree in the College of Intelligence and Computing, Tianjin University. His research interests include machine learning and computer graphics.

Yu-Kun Lai received his bachelor and Ph.D. degrees in computer science from Tsinghua University in 2003 and 2008, respectively. He is currently a professor in the School of Computer Science & Informatics, Cardiff University, UK. His research interests include computer graphics, geometry processing, image processing, and computer vision. He is on the editorial boards of *Computer Graphics Forum* and *The Visual Computer*.

Jinsong Zhang received his B.E. and M.E. degrees from Tianjin University in 2018. He is currently pursuing a Ph.D. degree in computer science in Tianjin University. His interests are mainly in computer vision and image synthesis.

Yuanqiang Yu received his B.E. degree from the School of Computer Science and Technology, Tiangong University, Tianjin, in 2020. He is currently pursuing an M.E. degree in the College of Intelligence and Computing, Tianjin University. His research interests are in deep reinforcement learning, transfer learning, and computer vision.

Kun Li received her B.E. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2006, and master and Ph.D. degrees from Tsinghua University, Beijing, in 2011. She visited the École Polytechnique Fédérale de Lausanne, Switzerland, in 2012 and 2014–2015. She is currently an associate professor in the College of Intelligence and Computing, Tianjin University. Her research interests include dynamic scene 3D reconstruction, and image and video processing.