# Supplementary Material for "UV Volumes for Real-time Rendering of Editable Free-view Human Performance"

**ACM Reference Format:**
. 2022. Supplementary Material for "UV Volumes for Real-time Rendering of Editable Free-view Human Performance". In . ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/8888888.7777777

Here we provide more implementation details and experimental results. We encourage the reader to view the video results included in the supplementary materials for an intuitive experience of the editable free-view human performance.

## 1 NETWORK ARCHITECTURES

### 1.1 Volume Generator

We utilize the volume generator to construct UV volumes, which is presented in Figure 1. We take the human pose as input to the SMPL model and animate human point clouds in different poses. Then we follow the previous work [Peng et al. 2021b] to anchor a set of time-invariant latent codes to the posed human point cloud and voxelize the point cloud. We follow the network architecture of [Peng et al. 2021b] to model the 3D sparse CNN, and decrease the channels from 352 to 64, since the UV volumes only captures low-frequency semantic information.

### 1.2 Density, IUV and Color Network

We present architectures of density network $M_\sigma$, IUV network $M_{uv}$, color network $M_c$ and texture generator network $G$ in Figure 2, Figure 3, Figure 5 and Figure 4, respectively.

### 1.3 Texture Generator

Figure 4 shows the architecture of convolutional texture generator network $G$. For each human body part $i = 1 \cdots 24$, the texture generator generates corresponding neural texture stack $\mathcal{E}_i$. To predict the specific pose-dependent $\mathcal{E}_i$, we concatenate the human pose vector $\theta$ with a one-hot body part label vector $\mathbf{k}_i (i = 1 \cdots 24)$ as input to the texture generator. We forward propagate the generator network $G$ once to predict all the 24 neural textures with a batch size of 24. The CNN-based module is developed to extract the local relation of neural texture stack with respect to the human pose. The output spatial neural texture stacks (NTS) will be used for UV unwrapping subsequently.

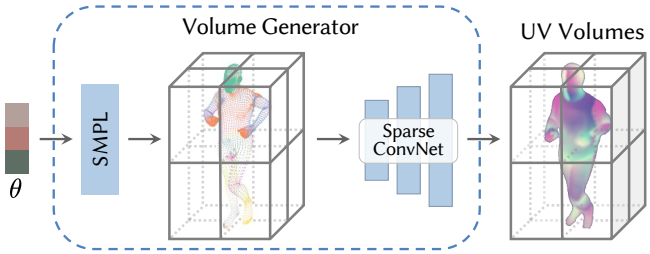



Fig. 1. Architecture of the volume generator. It takes human pose $\theta$ as input, drives the point clouds that parameterized by SMPL model under the control of $\theta$, and generates the UV volumes using a 3D sparse CNN to encode a set of latent codes $\mathbf{z}$ anchored on the posed point clouds.

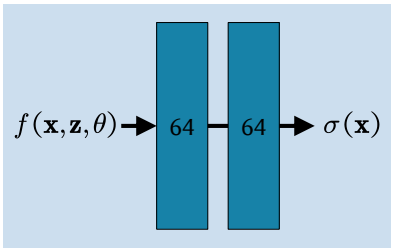



Fig. 2. Architecture of the density network. The network takes the feature vector $f(\mathbf{x}, \mathbf{z}, \theta)$ at point $\mathbf{x}$ interpolated by the generated UV volumes and outputs the density $\sigma(\mathbf{x})$ using relu activation. The shallow density MLP $M_\sigma$ consists of 2 fully-connected layers with 64 channels.

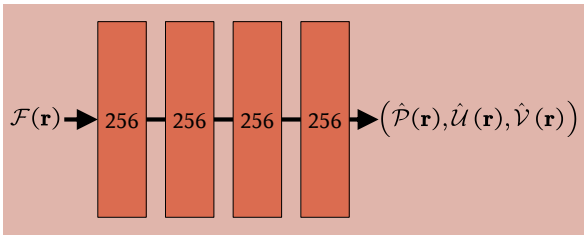



Fig. 3. Architecture of the IUV network. The network takes the rendered UV feature vector $\mathcal{F}(\mathbf{r})$ at the camera ray $\mathbf{r}$ and outputs the view-invariant texture coordinates $\left(\hat{\mathcal{P}}(\mathbf{r}), \hat{\mathcal{U}}(\mathbf{r}), \hat{\mathcal{V}}(\mathbf{r})\right)$ using sigmoid activation. The IUV MLP $M_{uv}$ is modeled by 4 fully-connected layers of 256 channels.


*SIGGRAPH Conference Proceedings, Dec 2022, Daegu*

ACM ISBN 978-1-4503-1234-5/22/07…$15.00
https://doi.org/10.1145/8888888.7777777

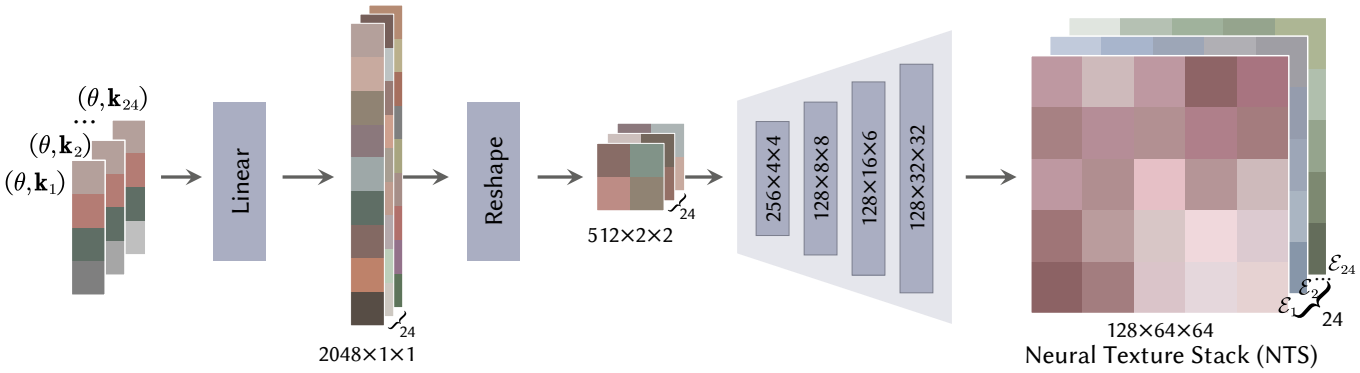



Fig. 4. Convolutional texture generator network $G$ consists of 5 convolution layers to get the neural texture stack $\mathcal{E}$ with 128 dimensions.

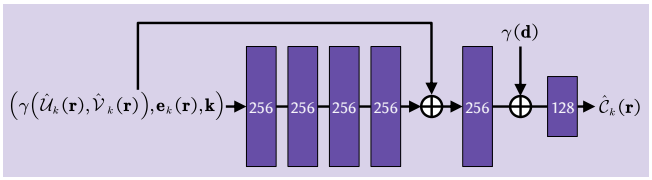
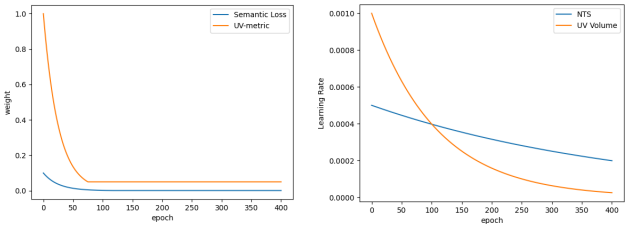



Fig. 5. Architecture of the color network. The network takes the positional encoding of the texture coordinates $\gamma(\hat{\mathcal{U}}_k(\mathbf{r}), \hat{\mathcal{V}}_k(\mathbf{r}))$ along with the sampled texture embeddings at locations $(\hat{\mathcal{U}}_k(\mathbf{r}), \hat{\mathcal{V}}_k(\mathbf{r}))$ and a one-hot part label vector $\mathbf{k}$. The color MLP $M_c$ is modeled by 5 fully-connected layers of 256 channels. It includes a skip connection that concatenates this input to the fourth layer's activation. The feature vector of the fifth layer is concatenated with the positional encoding of the input viewing direction $\gamma(\mathbf{d})$, and is processed by an additional fully-connected layer with 128 channels. A final layer with a sigmoid activation outputs the view-dependent RGB color $\hat{C}_k(\mathbf{r})$ of body part $\mathbf{k}$.




(a) Weight of warm-start loss. (b) Learning rate.

Fig. 6. (a) The blue represents the weight of semantic loss $\lambda_p$ and the orange represents the weight of UV-metric loss $\mathcal{L}_{uv}$. (b) The orange represents the learning rate of the UV volumes branch and the blue represents the learning rate of the NTS branch.

## 2 ADDITIONAL IMPLEMENTATION DETAILS

We set $\lambda_{vgg}$ to $5 \times 10^{-2}$ and $\lambda_s$ to $1 \times 10^{-1}$. The $\lambda_p$ and $\lambda_{uv}$ are exponential annealing from $1 \times 10^{-1}$ and $1 \times 10^{-0}$ to $1 \times 10^{-3}$ and $5 \times 10^{-2}$ with $k = 4 \times 10^{-2}$:

$$\begin{aligned} \lambda_p &= \max(1 \times 10^{-1} e^{-k \cdot \text{epoch}}, 1 \times 10^{-3}) \\ \lambda_{uv} &= \max(1 \times 10^{-0} e^{-k \cdot \text{epoch}}, 5 \times 10^{-2}) \end{aligned} \tag{1}$$

As show in Figure 6a, the weight of UV-metric are large at begining due to UV volumes require warm-start to statisfy the UV unwrap defined by DensePose[Güler et al. 2018] and then drops rapidly within 100 epochs, because DensePose outputs are not accurate. After 100 epochs, UV-metric becomes a regular term used to constrain the solution space of UV volume.

UV volumes should be learned primarily in the early stages of training, because The NTS makes sense only after UV volumes warm-start and a coarse geometry is constructed. Conversely, later in the training, we optimize NTS to fit high-frequency signal rather than UV coordinates. Therefore, we use two different optimization strategies to train the UV volumes branch and NTS branch. Their learning rates start from $1 \times 10^{-3}$ and $5 \times 10^{-4}$ with a decay rate of $\gamma = 1 \times 10^{-1}$, respectively, and decays exponentially along the optimization, as shown in Figure 6b.

$$\begin{aligned} l_{nts} &= 5 \times 10^{-4} \gamma^{\frac{\text{epoch}}{1000}} \\ l_{uv} &= 1 \times 10^{-3} \gamma^{\frac{\text{epoch}}{250}} \end{aligned} \tag{2}$$

In our experiments, we sample camera rays all over a full image and 64 points along each ray between near and far bounds. The scene bounds are estimated based on the SMPL model. We adopt the Adam optimizer[Kingma and Ba 2014] for training our model. We conduct the training on a single NVIDIA A100 GPU. The training on 26-view videos of 100 frames at $960 \times 540$ resolution typically takes around 200k iterations to converge (about 20 hours).

## 3 ADDITIONAL BASELINE METHOD DETAILS

**DyNeRF (DN) [Li et al. 2021].** Since DN is not open-sourced, we reimplement it by following the procedure referred in their paper to train the model on video sequences of a moving human.

**Neural Body (NB) [Peng et al. 2021b].** We use the NB code open-sourced by the authors at https://github.com/zju3dv/neuralbody and follow their procedure for training on video sequences of a moving human.

**Animatable NeRF (AN) [Peng et al. 2021a].** We use the AN code open-sourced by the authors at https://github.com/zju3dv/animatable and follow their procedure for training on video sequences of a moving human.

| Type | Novel View Synthesis | | | Novel Pose Generation | | |
|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
| Global NTS | 29.86 | 0.966 | 0.062 | 26.18 | 0.927 | 0.074 |
| Local NTS | 29.82 | 0.965 | 0.061 | 26.15 | 0.926 | 0.074 |
| Hyper NTS | 30.08 | 0.966 | 0.059 | 26.19 | 0.927 | 0.073 |
| Spatial NTS | 30.38 | 0.966 | 0.036 | 26.20 | 0.927 | 0.073 |

Table 1. Effect of different types of NTS on our model.

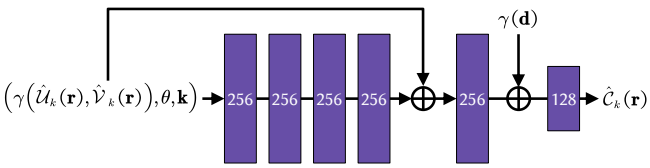



Fig. 7. The Global-NTS model takes directly pose $\theta$ as condition to generate the color. Its architecture is similar to our color MLP $M_c$ (in Figure 5) replacing texture embedding $\mathbf{e}_k(\mathbf{r})$ with pose $\theta$.

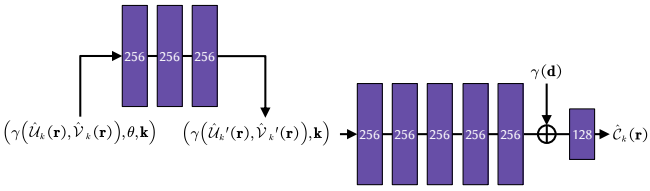



Fig. 8. The Local-NTS model transforms UV coordinates $\hat{\mathcal{U}}_t^k(\mathbf{r}), \hat{\mathcal{V}}_t^k(\mathbf{r})$ to $\hat{\mathcal{U}}_t^k{}'(\mathbf{r}), \hat{\mathcal{V}}_t^k{}'(\mathbf{r})$ using a deformation field which is conditioned on pose $\theta$ and modeled by 3 fully-connected layers of 256 channels. Then we use the transformed UV coordinates, part label vector $\mathbf{k}$ and viewing direction $\mathbf{d}$ as inputs to the subsequent MLP that is modeled by 5 fully-connected layers of 256 channels and 1 fully-connected layers of 128.

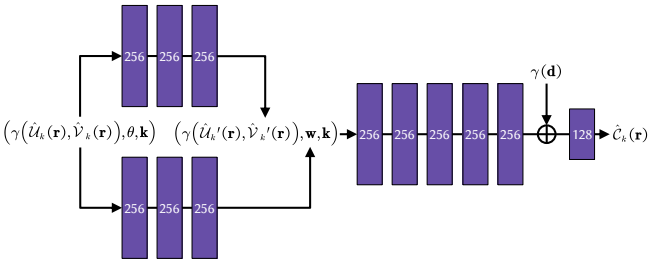



Fig. 9. The Hyper-NTS model adds an ambient MLP to the local-NTS model to parameterize the defomation field, which yields a warped UV coordinates $(\hat{\mathcal{U}}_t^k{}'(\mathbf{r}), \hat{\mathcal{V}}_t^k{}'(\mathbf{r}))$ and a coordinate $\mathbf{w}$ in the ambient space. Both outputs, part label vector $\mathbf{k}$ and viewing direction $\mathbf{d}$ are concatenated to the subsequent MLP to produce view-dependent colors. The ambient MLP has same architecture with the defomation MLP.

## 4 ADDITIONAL ABLATION STUDIES

We conduct ablation studies on performer p1 of CMU dataset. As shown in Table 1, Table 2, Table 3, Figure 10, Figure 11, and Figure 12, we analyze the effects of different losses for the proposed approach, different types of NTS, different resolutions of NTS, different methods to model the view-dependent color, different experimental setting of video frames and input views.

### 4.1 Effects of Different Losses

**Impact of warm-start loss.** No Warm-start Loss (w/o $\mathcal{L}_{\text{uv}}$) is an ablation that built upon our full model by eliminating the warm-start loss. As shown in Figure 10 and Figure 11, w/o $\mathcal{L}_{\text{uv}}$ suffers from ambiguity like the belt on pants and the meaningless texture. This comparison indicates that the warm-start loss yields a better information reuse of different frames by transforming the observation XYZ coordinates to canonical UV coordinates defined by the consistent semantic and UV-metric loss.

**Impact of perceptual loss.** No Perceptual Loss (w/o $\mathcal{L}_{\text{vgg}}$) is an ablation that uses the same model but training without the perceptual loss. As shown in Figure 10 and Figure 11, w/o $\mathcal{L}_{\text{vgg}}$ suffers from blur like the number 9 on the shirt and distorted number 9 on the texture. This comparison illustrates that perceptual loss can improve the visual quality of synthesized images by supervising the structure of the renderings from local to global during training.

**Impact of silhouette loss.** No Silhouette Loss (w/o $\mathcal{L}_{\text{s}}$) is an ablation that built upon our full model by eliminating the silhouette loss. As shown in Figure 10, w/o $\mathcal{L}_{\text{s}}$ suffers from artifacts around the performance because the there dose not exist the warm-start supervision of semantic and UV-metric labels around the boundary. This comparison demonstrate that silhouette loss if significant for us to model a fine-grained geometry.

### 4.2 Neural Texture Stacks

We performed two ablations: 1) different types of NTS; 2) NTS at different resolutions to illuminate the design decisions for the proposed *Neural Texture Stacks*.

**Different Types of NTS.** We evaluate our proposed CNN-based Spatial NTS against three ablations: Global NTS, Local NTS, and Hyper NTS. Global NTS (in Figure 7) builds upon our full model by replacing local texture embedding $\mathbf{e}_k(\mathbf{r})$ with global pose $\theta$. Local NTS (in Figure 8) transforms observation UV coordinates $\hat{\mathcal{U}}_t^k(\mathbf{r}), \hat{\mathcal{V}}_t^k(\mathbf{r})$ to canonical UV coordinates $\hat{\mathcal{U}}_t^k{}'(\mathbf{r}), \hat{\mathcal{V}}_t^k{}'(\mathbf{r})$ using a deformation field. Hyper NTS (in Figure 9) adds an ambient MLP to the local-NTS model to model a slicing surface in hyperspace, which yields a coordinate $\mathbf{w}$ in an ambient space.

Table 1 shows the quantitative results on different types of NTS (i.e., global, local, hyper and spatial). It can be seen that local-NTS model has the worst performance, which is the most limited among these methods. Local NTS only allows coordinate transformation but cannot generate new topologically space, which can not model the topologically varying texture given different pose, as shown in Figure 10, it can not reconstruct the belt on pants due to the topologically variation (the belt appears when the performer raises his hand and disappears when the performer puts his hand down because it is covered by the shirt).

Since the texture of Global-NTS directly condition on global pose without restriction, it is easy for Global-NTS to generate a new topologically space, as shown in Figure 10, Global-NTS successfully reconstructs the belt on pants. However, the method that globally

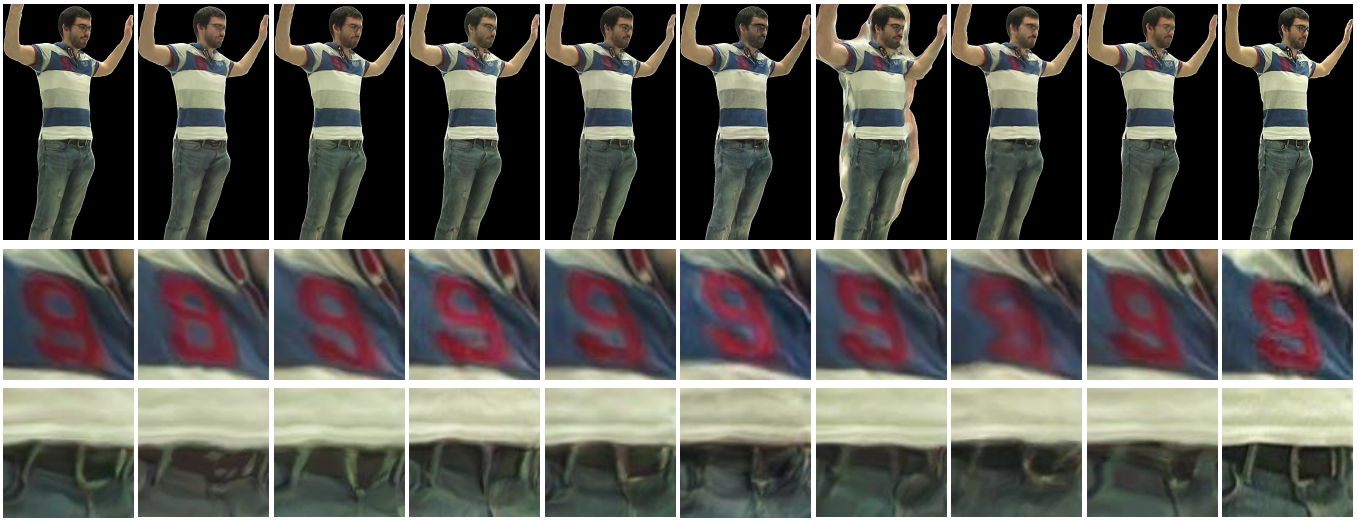



Fig. 10. Renderings of our model against ablations.

models the texture variation is hard to reuse the information of different observation spaces, which leads to ambiguous textures. As shown in Figure 10, the outline of the number 9 on the shirt is not clear and even connected to each other makes it looks like 8. The ambiguous textures are shown in Figure 11, where the number is just a red spot. The lacks of local mapping makes a relatively poor performance of Global-NTS as demonstrated in Table 1.

Hyper-NTS can model local texture changes by coordinate transformation and generate new topologically spaces at the same time, so it performs better than Global-NTS and Local-NTS. However, it is hard to tune the dimension of coordinate $\mathbf{w}$. If the dimension of the coordinates $\mathbf{w}$ is too high, it is equivalent to Global-NTS, and if it is too low, it is equivalent to Local-NTS.

In contrast, as shown in Table 1, our CNN-based Spatial NTS outperforms all other NTS, which benefits from the nature of convolution operation capturing local 2D texture changes. At the same time, the MLP only needs to model the local mapping between the neural texture stack and RGB color.

As demonstrated in Figure 10 and Figure 11, our CNN-based Spatial NTS can accurately capture the high-frequency details like numbers and wrinkles on shirts, glasses on eyes and the belt on pants.

**NTS at Different Resolutions.** As shown in Figure 4, we generate a NTS at the resolution of $64 \times 64$. Choosing the resolution of NTS provides a trade-off between quality and memory. We analyze the impact of resolution in Figure 12, where we report test quality vs. resolution for the dataset of CMU-171204p4s6 on PSNR, SSIM and LPIPS metrics. Due to memory limitations, NTS have a maximum resolution of 128. It was observed that the larger the resolution of NTS, the better the model performed on novel view synthesis and the novel pose generalization tasks. In this analysis, we found $64 \times 64$ to be a favorable optimum in our applications, where NTS at $128 \times 128$ resolution is not much better than that at $64 \times 64$ resolution but cost more memory and time, so we use $64 \times 64$ resolution in all other experiments and recommend them as the default to practitioners.

| Type | Novel view synthesis | | | Novel pose generation | | |
|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
| SH1 | 29.41 | 0.966 | 0.052 | 26.00 | 0.926 | 0.078 |
| SH2 | 30.02 | 0.966 | 0.051 | 26.15 | 0.927 | 0.076 |
| SH3 | 29.43 | 0.965 | 0.051 | 26.14 | 0.926 | 0.075 |
| SH4 | 27.83 | 0.962 | 0.054 | 26.10 | 0.926 | 0.076 |
| w/o noise | 28.56 | 0.962 | 0.058 | 25.90 | 0.924 | 0.078 |
| Ours | 30.38 | 0.966 | 0.036 | 26.20 | 0.927 | 0.073 |

Table 2. Comparison of different methods to model the view-dependent RGB color.

### 4.3 View-dependent Color

**Ray Direction Noise.** To model the view-dependent RGB color of human performances, we apply the positional encoding $\gamma(\cdot)$ [Rahaman et al. 2019] to viewing direction, and pass the encoded viewing direction and UV map with the sampled texture embedding into the color network $M_c$ to decode the view-dependent color $\hat{C}_k(\mathbf{r})$ of camera ray $\mathbf{r}$.

Since the UV map is generated in a learning-based fashion instead of using direct sampling locations, the color network tends to overfit to the training viewing direction which is directly sampled during training. To improve the generalisability of color network, we apply a sub-pixel noise to the ray direction. Here, instead of shooting in the pixel centres, a noise $\psi$ is used as follows:

$$x_i = \mathbf{o} + t_i(\mathbf{d} + \psi), \tag{3}$$

where noise $\psi$ can be interpreted as a locality condition, i.e., in similar view conditions, RGB color should not be too different. This allows model to learn smoother transitions between different views.

The ablation of ours w/o noise is presented in Table 2, it demonstrates the effectiveness of the proposed ray direction noise. Figure 10 and Figure 11 shows the qualitative results of ours w/o noise testes on the novel views, it is obvious that ours w/o noise tends to exhibit artifacts in the rendering and textures.

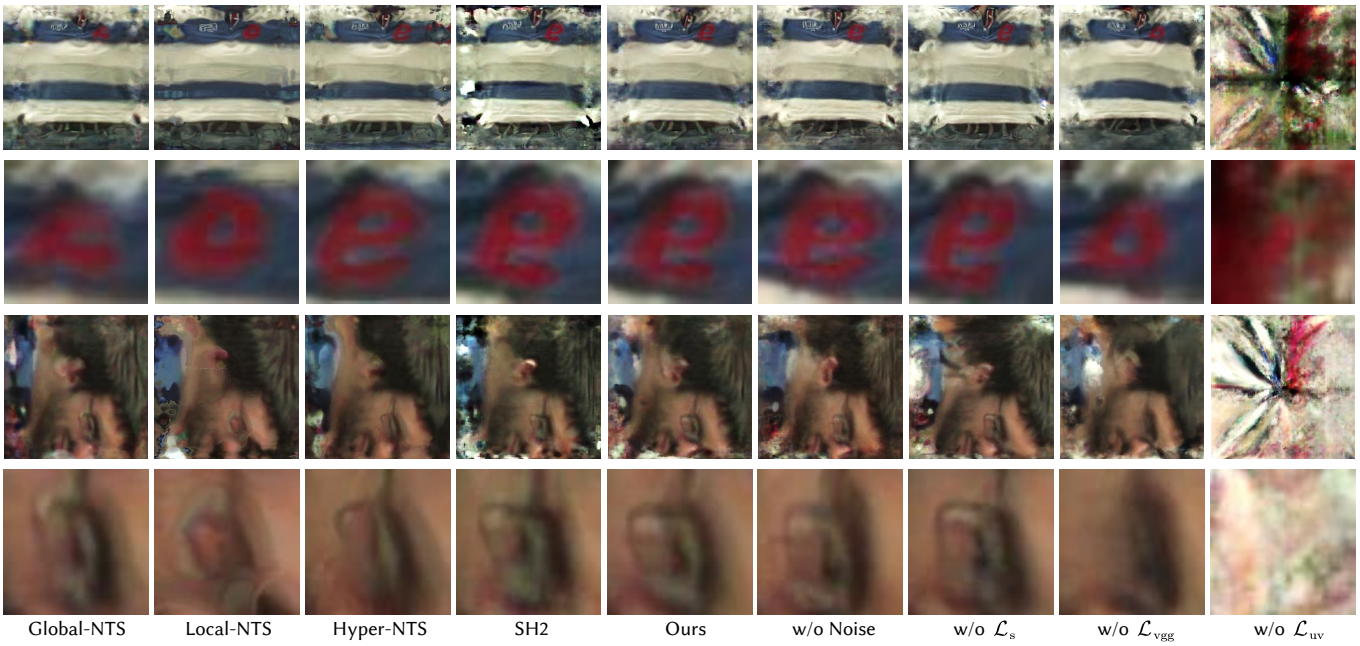
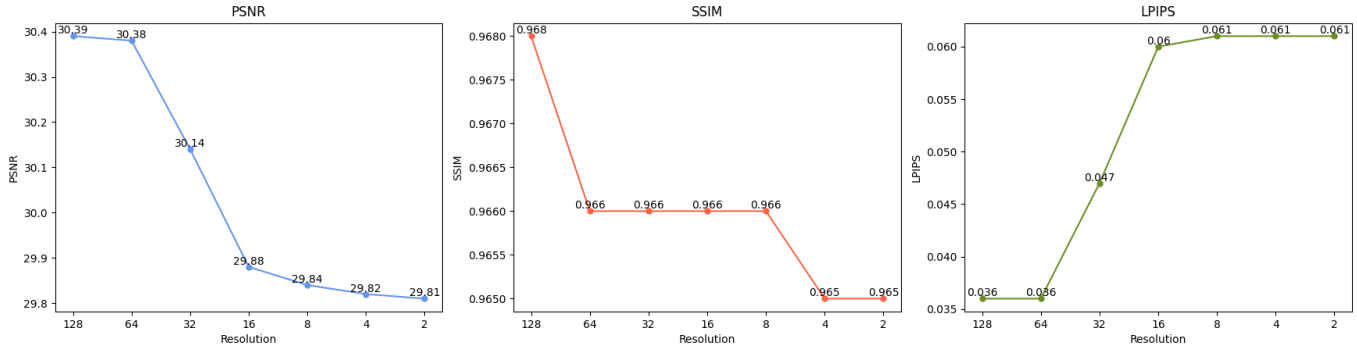
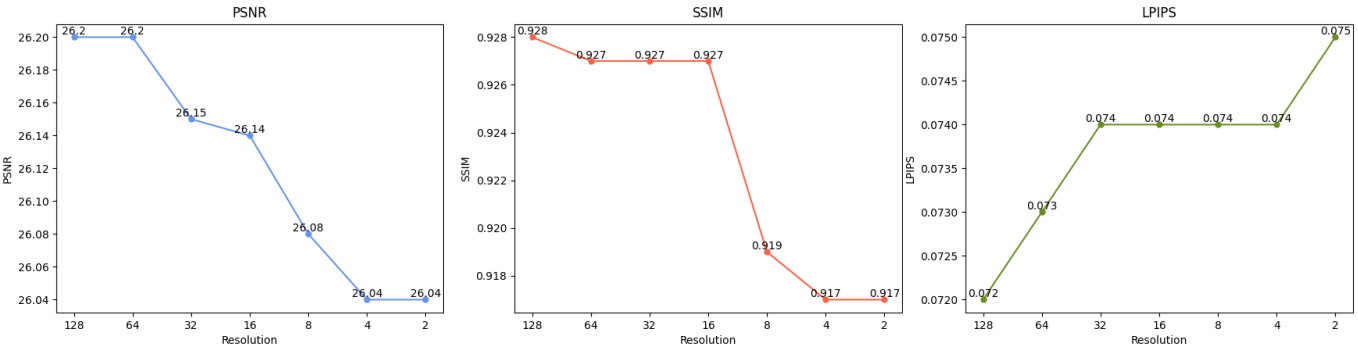



Fig. 11. Textures of our model against ablations.




(a) Novel view synthesis




(b) Novel pose generalization

Fig. 12. Impact of NTS at different resolutions.

| Task | 4 Views | | | 20 Views | | |
|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
| 60 Frames | 29.56 | 0.967 | 0.045 | 31.12 | 0.975 | 0.038 |
| 300 Frames | 29.23 | 0.963 | 0.048 | 29.90 | 0.968 | 0.046 |
| 600 Frames | 29.37 | 0.964 | 0.049 | 29.50 | 0.966 | 0.048 |
| 1200 Frames | 28.96 | 0.961 | 0.052 | 29.26 | 0.963 | 0.053 |

Table 3. Ablation study on the number of training frames and views.

**Spherical Harmonic Functions.** Another way to reconstruct view-dependent color of human performances is using the spherical harmonic (SH) functions. We pass the encoded UV map with the sampled texture embedding into the color network $M_c$ to decode the spherical harmonic coefficients $\eta$ for each color channel:

$$\left(\hat{\eta}_k^0(\mathbf{r}), \hat{\eta}_k^1(\mathbf{r}), \cdots, \hat{\eta}_k^n(\mathbf{r})\right) = M_c\left(\gamma(\hat{\mathcal{U}}_k(\mathbf{r}), \hat{\mathcal{V}}_k(\mathbf{r})), \mathbf{e}_k(\mathbf{r}), \mathbf{k}\right), \quad (4)$$

where spherical harmonics $(\eta^0, \eta^1, \cdots, \eta^n)$ form an orthogonal basis for functions defined over the sphere, with zero degree harmonics $\eta^0$ encoding diffuse color and higher degree harmonics encoding specular effects. The view-dependent color $\hat{C}_k(\mathbf{r})$ of camera ray $\mathbf{r}$ can be determined by querying the specular spherical functions $SH$ at the desired viewing direction $\mathbf{d}$:

$$\hat{C}_k(\mathbf{r}) = S\left(\frac{\hat{\eta}_k^0(\mathbf{r})}{2}\sqrt{\frac{1}{\pi}} + \sum_{m=1}^{n} SH^m(\hat{\eta}_k^m(\mathbf{r}), \mathbf{d})\right), \quad (5)$$

where $S$ is the sigmoid function for normalizing the colors.

Higher degree of harmonics results in higher capability to model the high-frequency color but is more prone to overfit the training viewing direction.

The ablation of different harmonics degree is presented in Table 2, it demonstrates that harmonics degree of 2 achieves the best performance among all the SH models but still could not reach the performance of ours. Figure 10 illustrates the qualitative results of SH2 testes on the novel views, it shows that SH2 suffers from global color shifts (the head of performance) and artifacts (the belt on pants), while ours does not.

### 4.4 Video Frames and Input Views

To analysis the impact of the number of camera views and video length, we show the results of our models trained with different numbers of camera views and video frames in Table 3. We conduct the experiments on performer 313 of ZJU dataset. All the results are evaluated on the rest two views of the first 60 frame video. The results show that although the number of training views improves the performance on novel view synthesis, sparse four views are good enough for our model to reconstruct dynamic human performances. In addition, the ablation study of frame numbers indicates that training on too many frames may decrease the performance as the network cannot fit such a long video, which is also mentioned in NeuralBody [Peng et al. 2021b].

## 5 ADDITIONAL RESULTS

### 5.1 Novel View Synthesis

For comparison, we synthesize images of training poses in hold-out test-set views. More qualitative results of novel view synthesis are shown in Figure 13, Figure 14 and Figure 15. Our method produces photo-realistic images with sharp details, particularly letters on clothes (in Figure 13), stripes on T-shirts and wrinkles in clothes (in Figure 14), which benefits from our proposed spatial neural texture stacks (NTS) that encode high-frequency appearance information.

Figure 15 are the results of comparisons on ZJU Mocap and H36M dataset which are training on sparse-views video sequences. Here, we use four training views on ZJU Mocap dataset and three for the most challenging H36M dataset. Our model obviously perform well in details and sharpness than all other baselines. Furthermore, DyNeRF fails to render plausible results with sparse training views because taking time-varying latent codes as the conditions is hard to reuse information among frames.

### 5.2 Novel View Synthesis of Dynamic Humans

We present more results on novel view synthesis of dynamic humans in Figure 16. As presented, our model can handle dynamic human with rich textures and challenging motions, and preserve sharp image details like letters and wrinkles, while keeps inter-view consistency and inter-frame consistency. Note that the last row is the result of our model on the H36M dataset, which demonstrates that our model can still recover high-fidelity free-view videos under sparse training views.

In addition, we also show the intermediate UV images and final RGB images rendered by our model varying with views and human poses in Figure 17, which demonstrates that our model can synthesise photo-realistic view-consistent RGB images that condition on view-consistent UV images rendered by UV volumes.

### 5.3 Novel Pose Generalization

More qualitative results of novel pose generalization are shown in Figure 18 and Figure 19, where the latter are the results of comparisons on H36M dataset where only three cameras are available for training.

### 5.4 Reshaping

By changing the SMPL parameters, we can conveniently deform the human performer. We present the performer whose size is getting smaller and shoulder-to-waist ratio is getting smaller from left to right in Figure 20. With the help of view-consistent UV coordinates generated by UV volumes, our model still renders view-consistent images with challenging shape parameters. These rendered images maintain highly appearance consistency across changing shapes thanks to the neural texture stacks.

### 5.5 Visualization of NTS

In contrast to [Peng et al. 2021b] learning a radiance filed in the 3D volumes, we decompose a 3D dynamic human into 3D UV volumes and 2D neural texture stacks, as illustrated in Figure 21. The disentanglement of appearance from geometry enables us to achieve real-time rendering of free-view human performance. We learn a

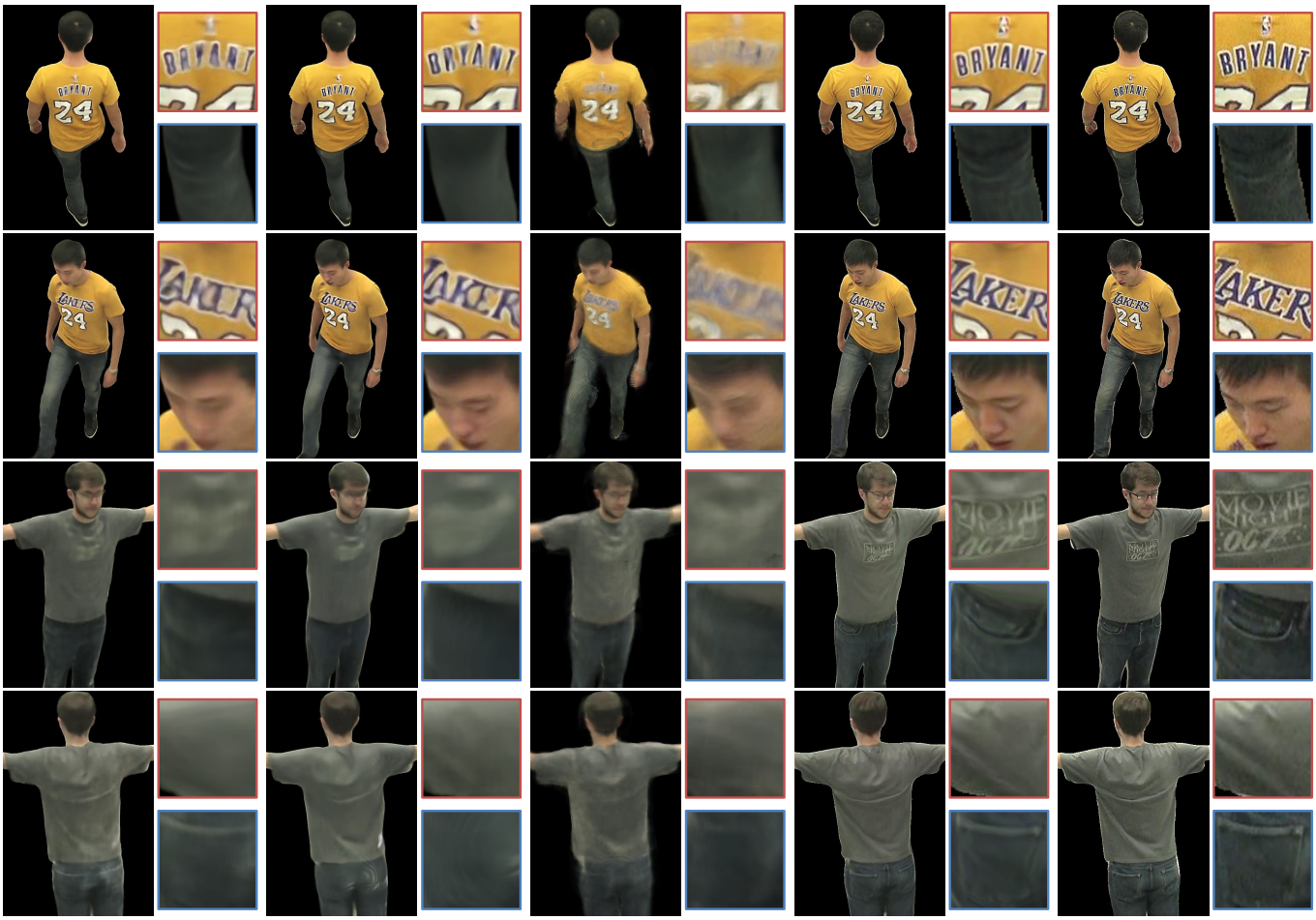



Fig. 13. Comparisons on test-set views for performers from CMU Panoptic dataset with 960×540 images. Our model generates photo-realistic appearance images even with rich textures, particularly letters on the performers' clothes. By contrast, baselines gives blurry results while misses a lot of high-frequency details. Here, we present results on two different test views at the same time for each performer.

view-consistent UV field to transfer neural texture embeddings to colors, which guarantees view-consistent human performance. Details like the folds of clothing vary from motion to motion, as does the topology, so we require a dynamic texture representation. Referring to Figure 22, we visualize the pose-driven neural texture stacks to describe appearance at different times, which enables us to handle dynamic 3D reconstruction tasks and to generalize our model to unseen poses. It is obvious that our learned NTS preserve rich textures and high-frequency details which varying from different poses.

### 5.6 Retexturing

With the learned dense correspondence of 3D UV volumes and 2D neural texture stacks, we can edit performers' 3D cloth by user-provided 2D textures. As shown in Figure 23, given any arbitrary artistic paintings, we can produce cool stylized dynamic humans leveraging stylizations transferred from the original texture stacks by the network [Ghiasi et al. 2017]. Visually inspected, the new texture are well painted onto the performer's T-shirt under different poses at different viewing directions. Besides, we perform some interesting applications of our model in Figure 24 and Figure 25, which includes a 3D virtual try-on implemented by replace original texture stacks with user-provided appearance. The visualization results demonstrates that our model can conveniently edit textures preserving rich appearance and various style, which benefits from our proposed Neural Texture Stacks, and can render retextured human performance with view consistency well using 3D UV volumes.

## REFERENCES

Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. 2017. Exploring the structure of a real-time, arbitrary neural artistic stylization network. *arXiv preprint arXiv:1705.06830* (2017).

Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. 2018. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7297–7306.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, and Zhaoyang Lv. 2021.

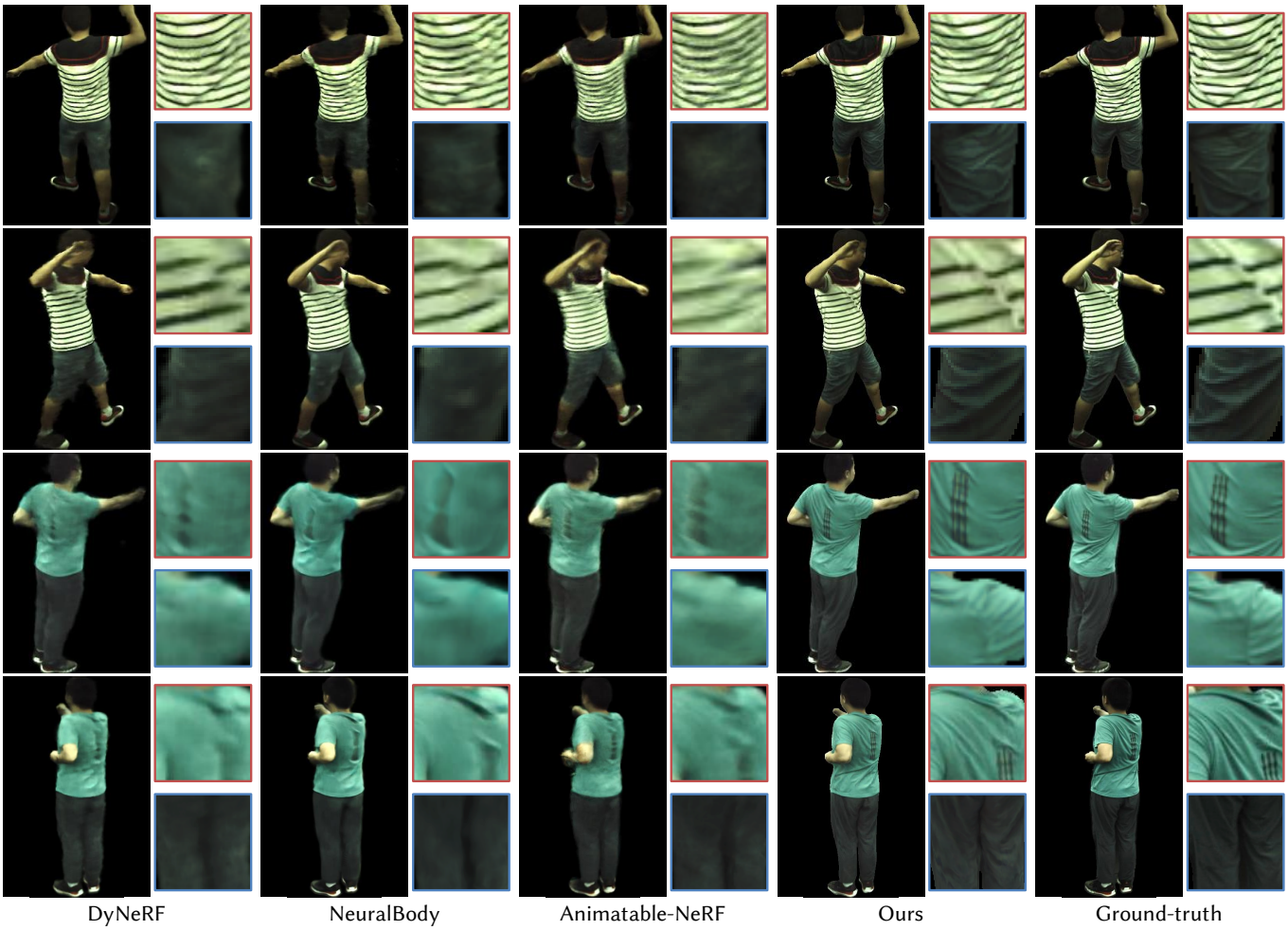



Fig. 14. Comparisons on test-set views for performers from ZJU Mocap dataset. Our model obviously perform well in details (e.g., stripes on T-shirts and wrinkles in clothes) and sharpness than all other baselines, which benefits from our proposed spatial *neural texture stacks (NTS)* that encode high-frequency appearance information. Other methods gives plausible but blurry and rough synthesized images. Here, we present results on two different test views at the same time for each performer.

Neural 3d video synthesis. *arXiv preprint arXiv:2103.02597* (2021).

Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. 2021a. Animatable Neural Radiance Fields for Modeling Dynamic Human Bodies. In *ICCV*.

Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2021b. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9054–9063.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. 2019. On the spectral bias of neural networks. In *International Conference on Machine Learning*. PMLR, 5301–5310.

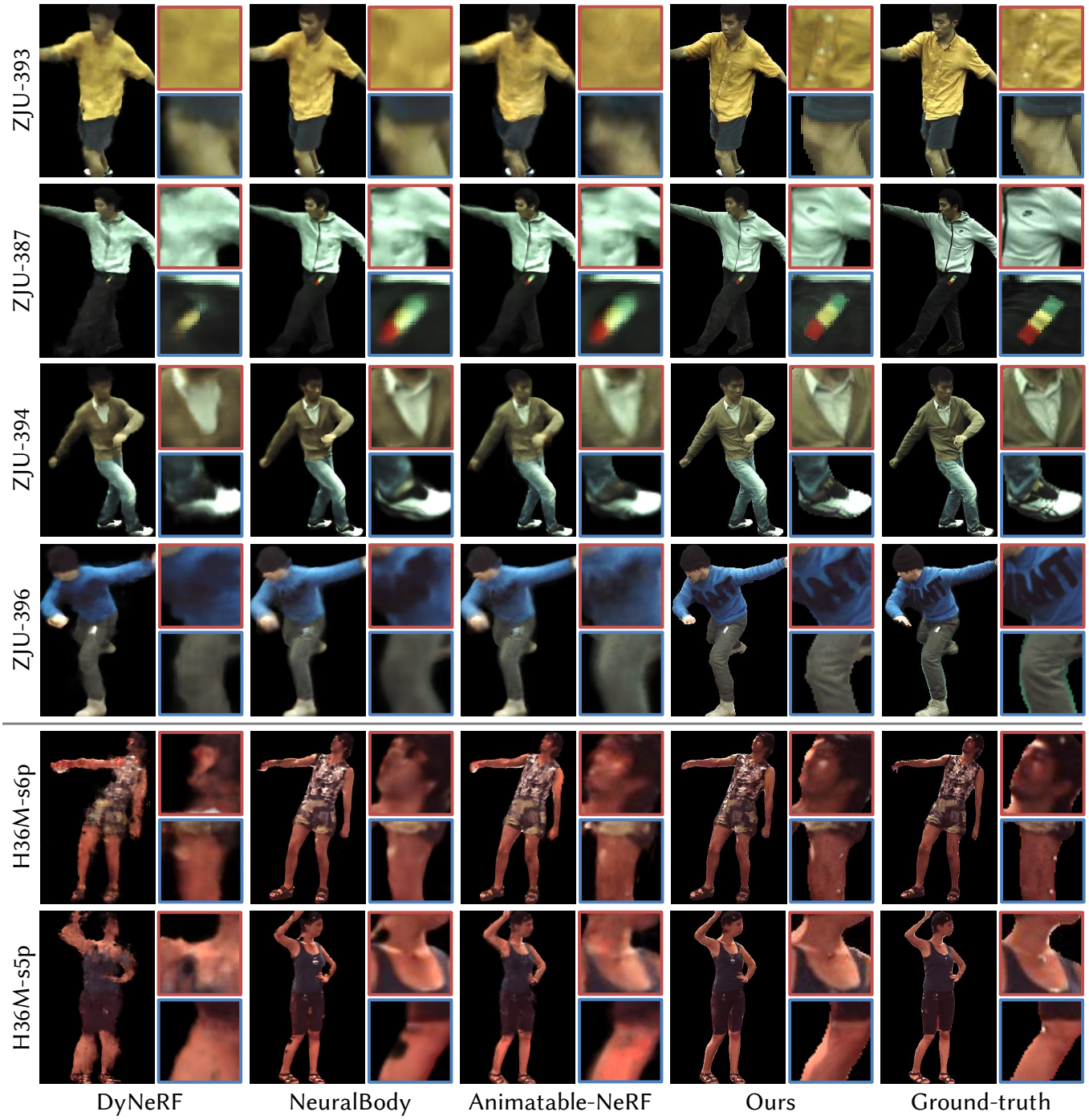



Fig. 15. Comparisons on test-set views from ZJU Mocap dataset that has four training views and the most challenging H36M dataset that only has three available views for training. Our model generates high-definition results even with rich textures and challenging motions. DyNeRF fails to render plausible results with sparse training views because taking time-varying latent codes as the conditions is hard to reuse information among frames.


Fig. 16. The rendering of our method on different sequences. Our model can handle dynamic human with rich textures and challenging motions preserving sharp image details like letters and wrinkles, which benefits from our proposed spatial *neural texture stacks (NTS)* that encode high-frequency appearance information, while keeping inter-view consistency and inter-frame consistency, which benefits from our proposed *UV Volumes*. Note that the last row is the result of our model on the H36M dataset, which demonstrates that our model can still recover high-fidelity free-view videos under sparse training views.

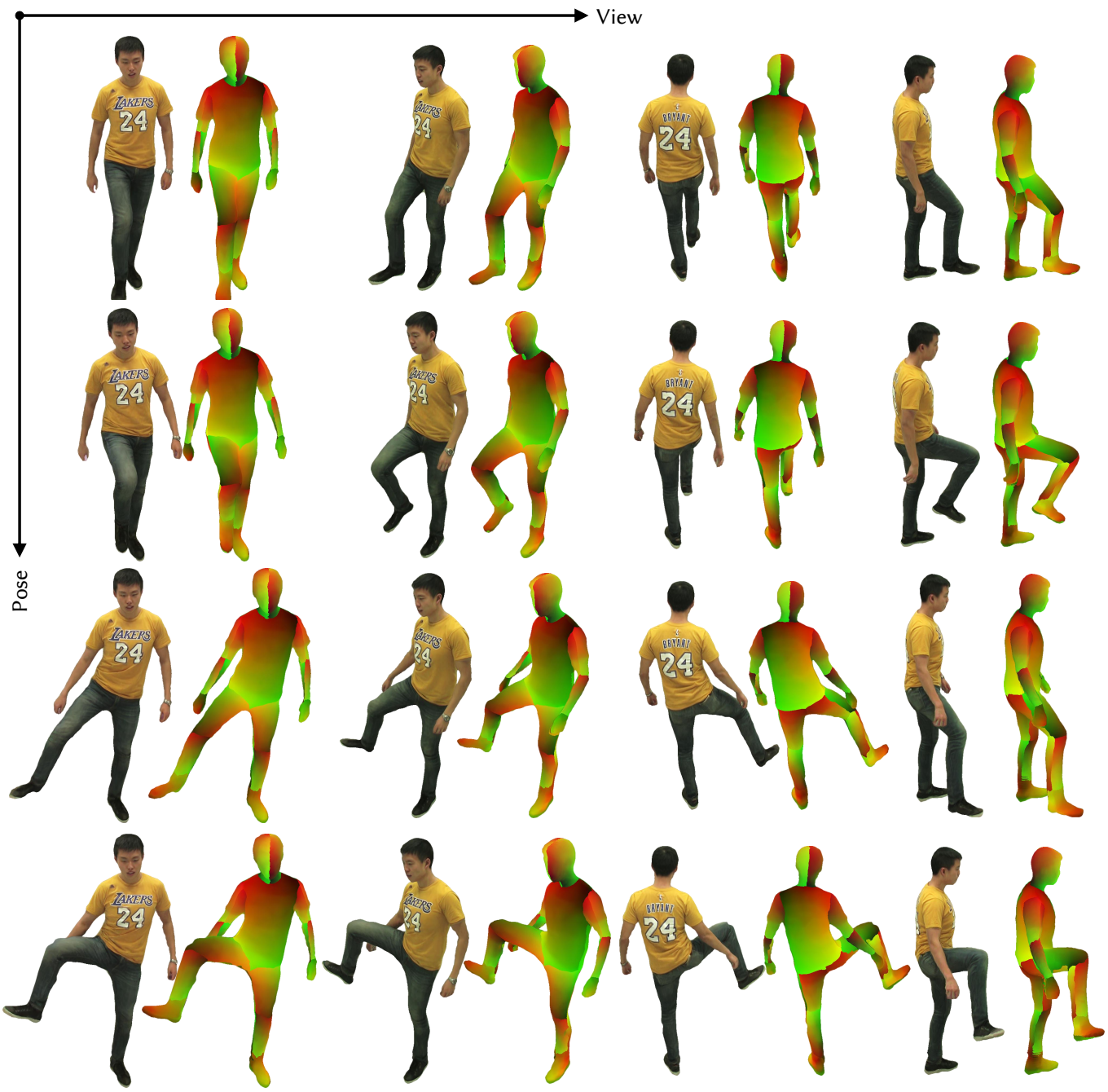



Fig. 17. Novel view synthesis of the dynamic human. Our model can synthesise photo-realistic view-consistent RGB images that condition on view-consistent UV images rendered by UV volumes. Here, The horizontal axis shows the change in novel views and the vertical axis shows the change in human poses. All results are rendered from novel views in the training pose sequence.

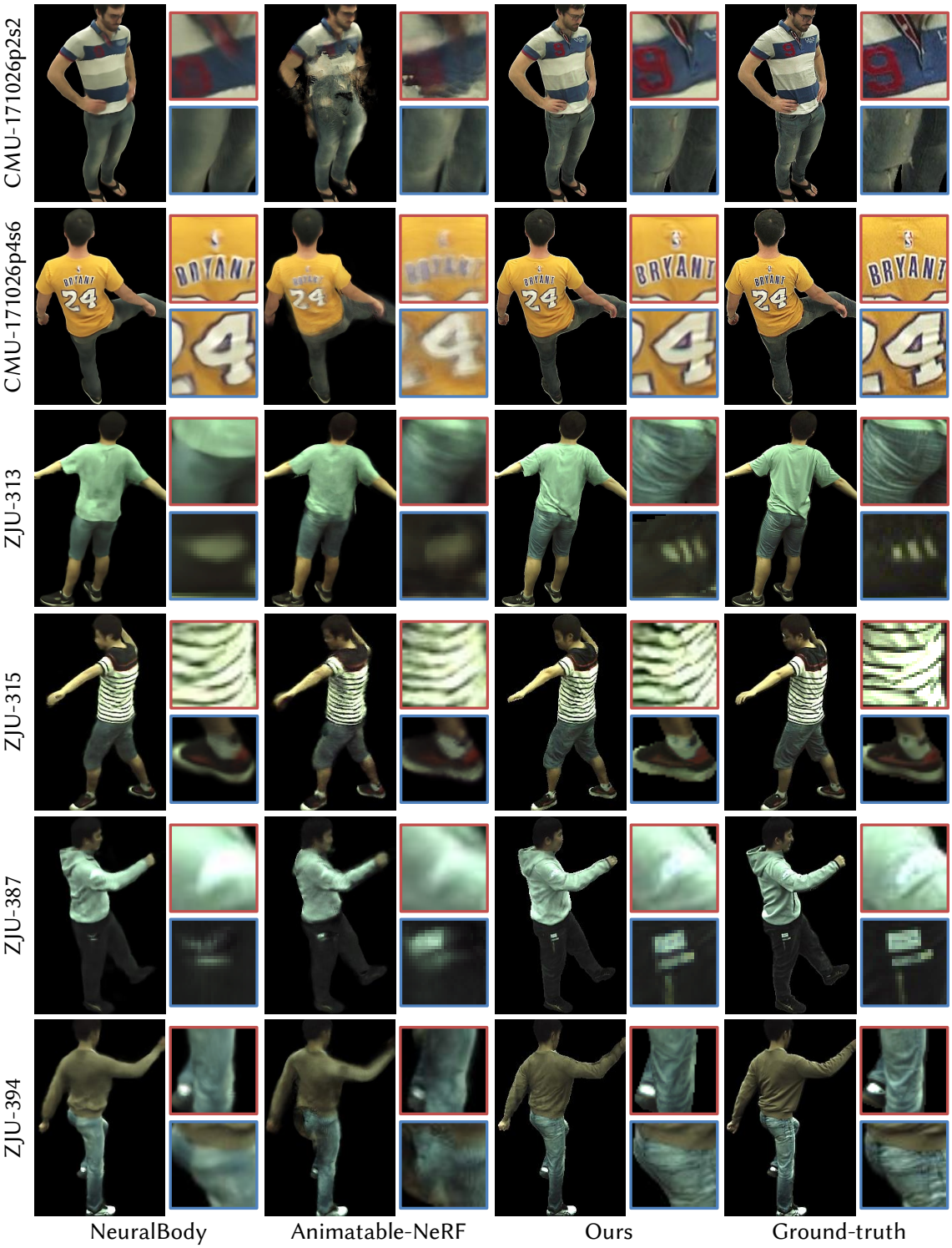



Fig. 18. Comparisons on test-set poses for performers from CMU Panoptic and ZJU Mocap dataset.

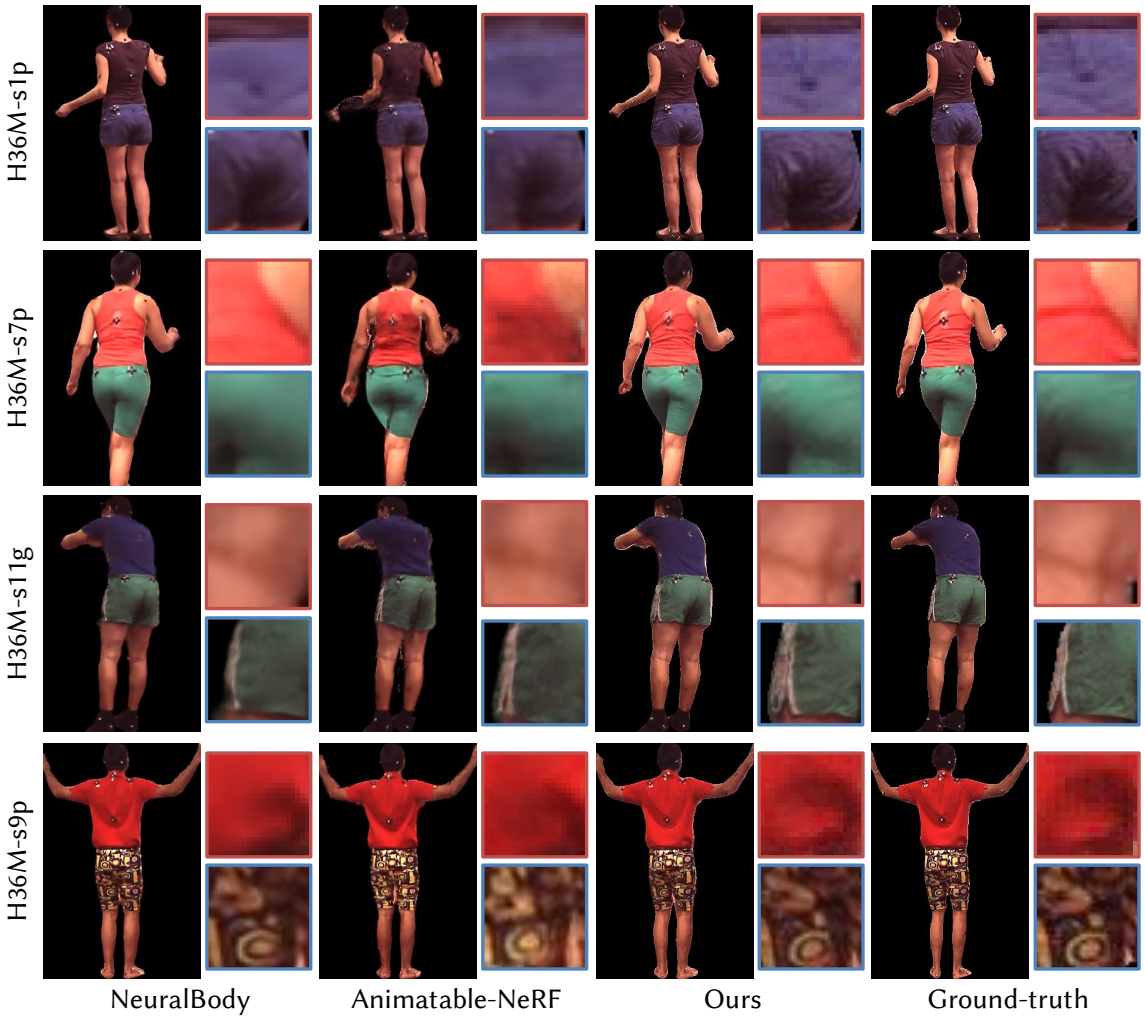



Fig. 19. Comparisons on test-set poses for performers from the most challenging H36M dataset where only three cameras are available for training.

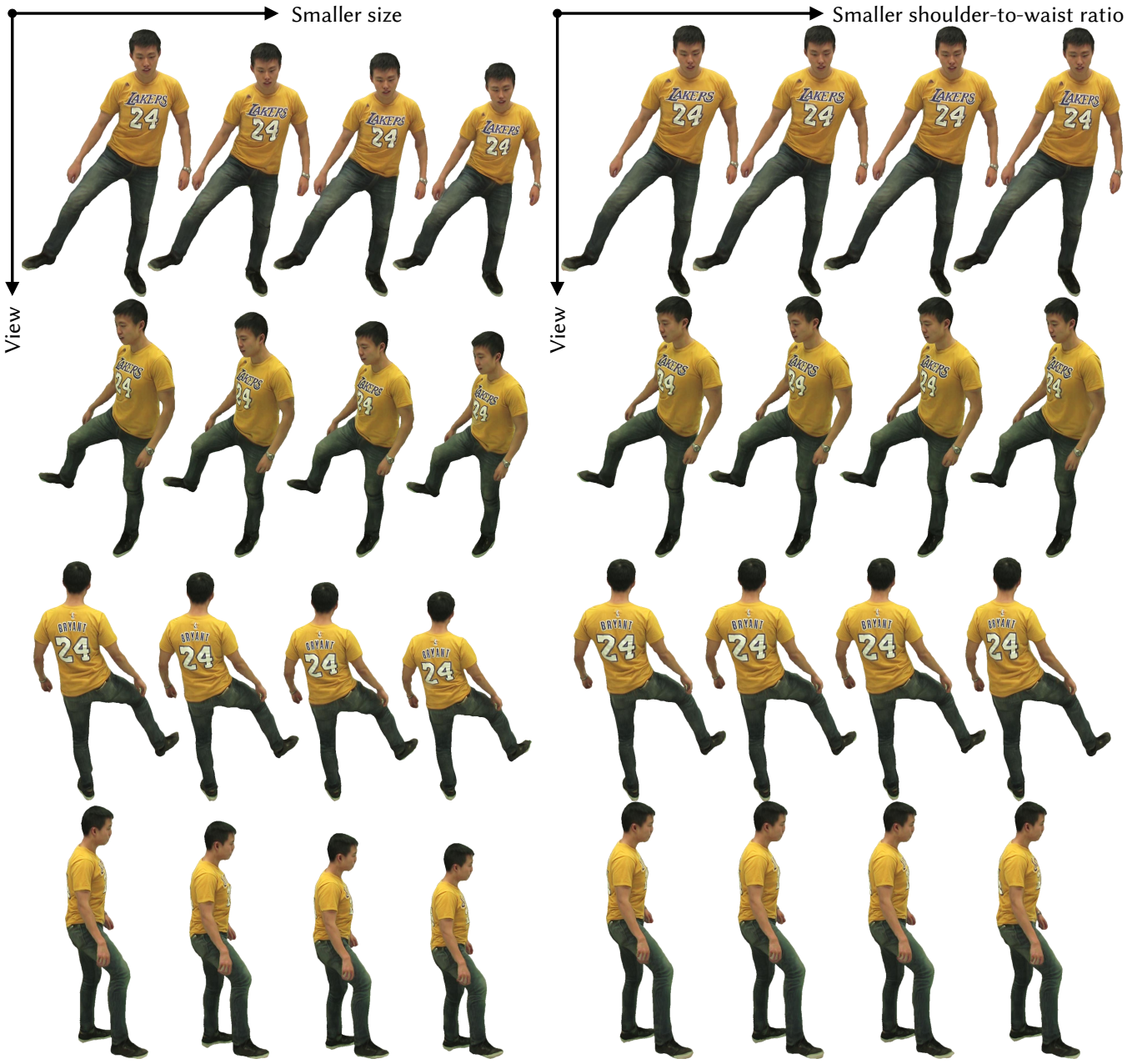



Fig. 20. Novel view synthesis results of reshaping. By changing the SMPL parameters, we can conveniently deform the human performer. We present the performer whose size is getting smaller and shoulder-to-waist ratio is getting smaller from left to right. With the help of view-consistent UV coordinates encoded by UV volumes, our model still renders view-consistent images with challenging shape parameters. Here, The horizontal axis shows shape changing and the vertical axis shows views changing. All results are rendered from novel views.

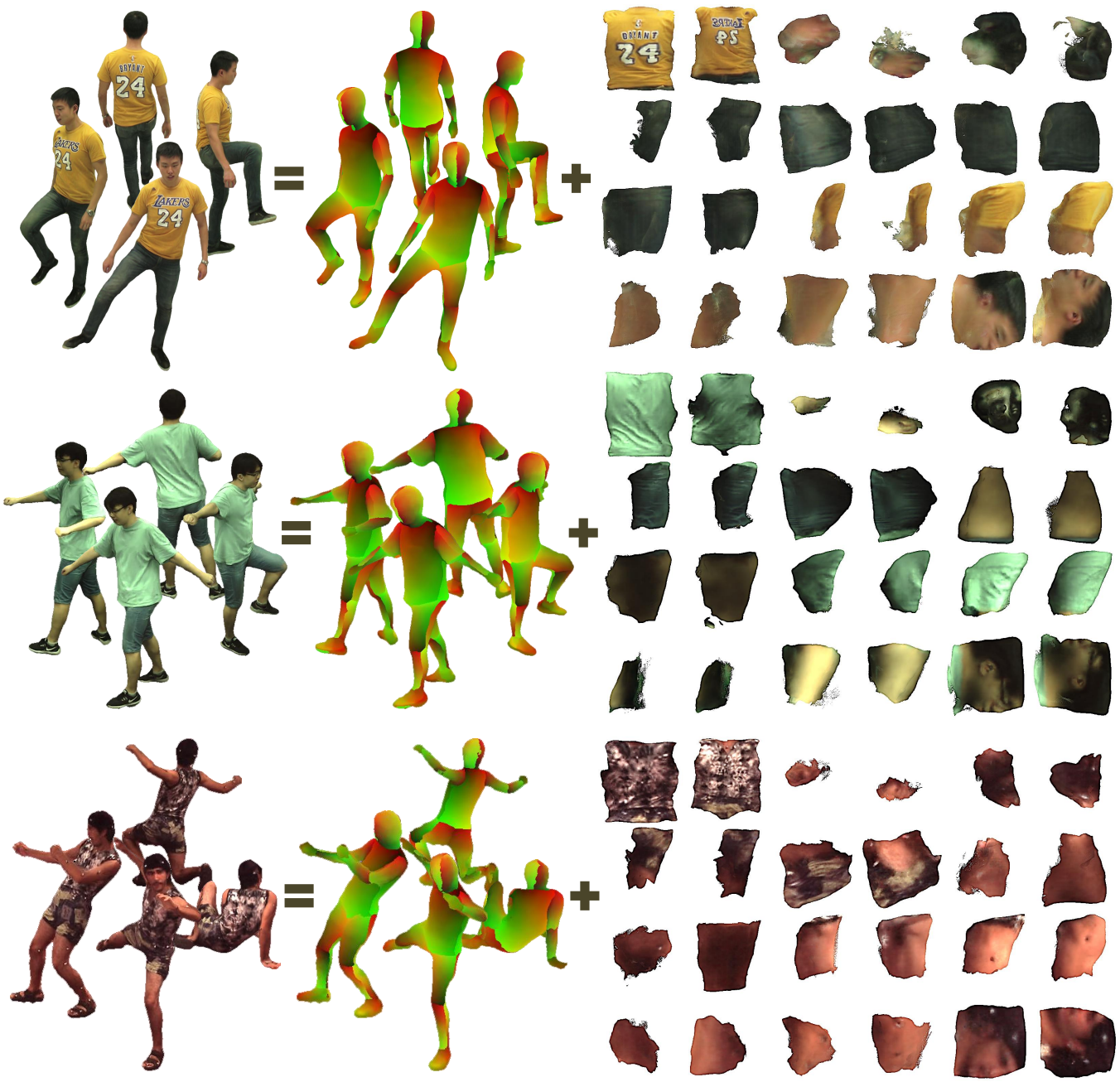



Fig. 21. We decompose (a) the dynamic human into (b) 3D UV volumes and (c) 2D neural texture stacks. The disentanglement of appearance from geometry enables us to achieve real-time rendering of free-view human performance. We show performers and their UV avatars with four different poses at four different viewing directions from CMU Panoptic, ZJU-Mocap and H36M datasets. Their neural texture stacks that preserve human appearance with high-frequency details under one of these poses are visualized in the last column. Our method takes smooth UV coordinates to sample neural texture stacks for corresponding RGB value.

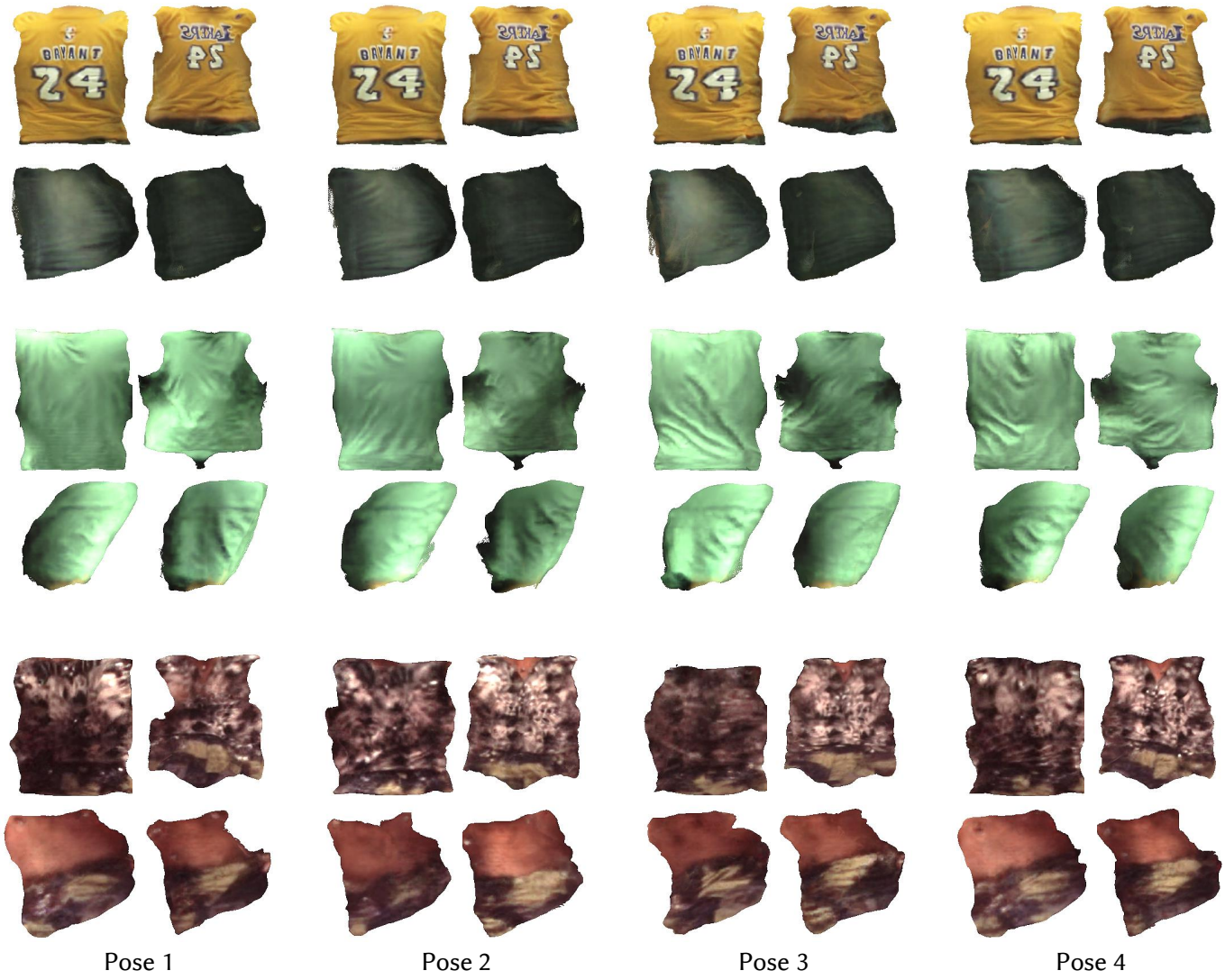



Fig. 22. Visualization of neural texture stacks under different poses. Details like the folds of clothing vary from motion to motion, as does the topology. Therefore, we propose pose-driven neural texture stacks to describe textures at different times, which enables us to handle dynamic 3D reconstruction tasks and to generalize our model to unseen poses.

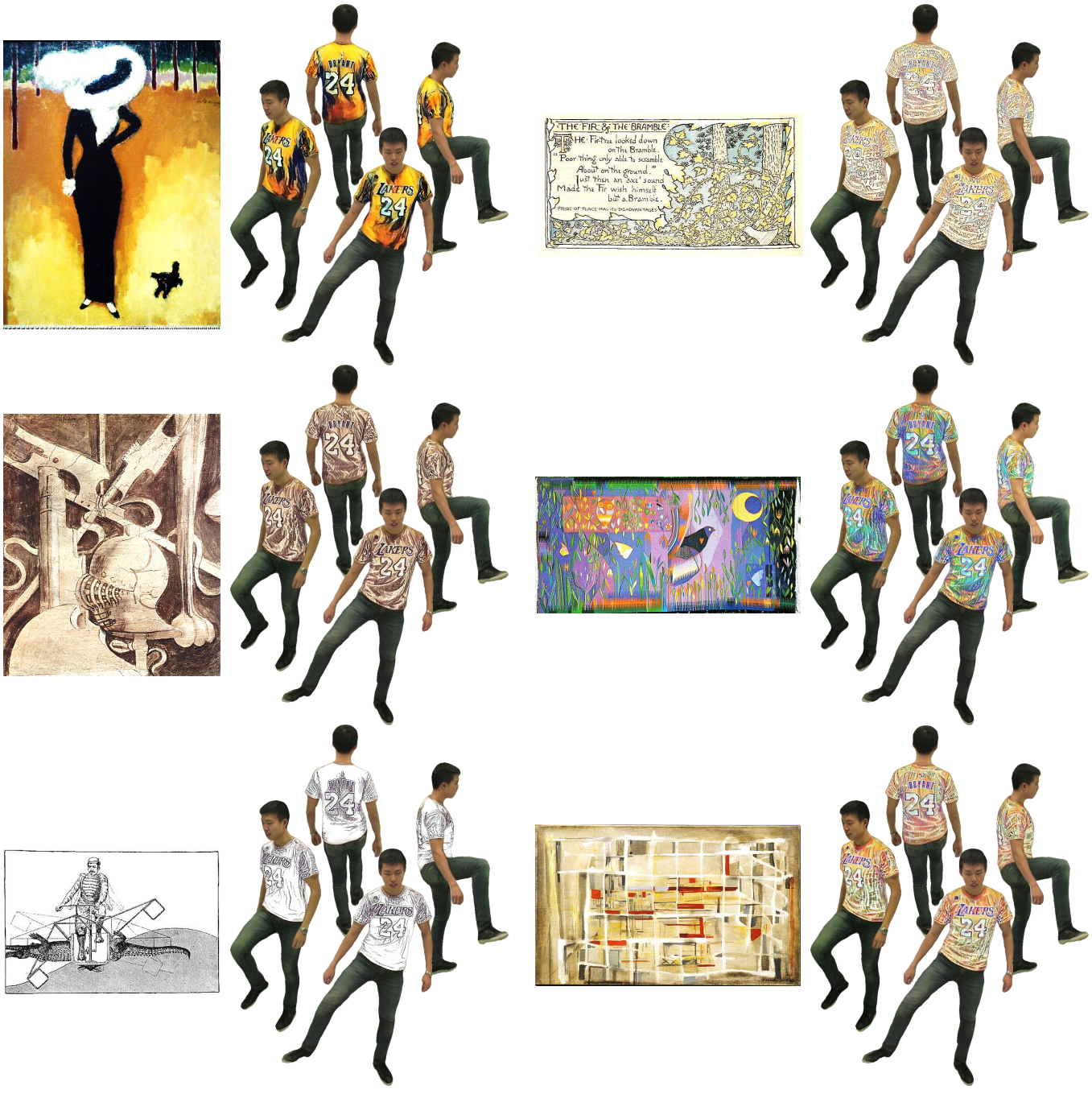



Fig. 23. Our model supports rendering a stylized dynamic human with any arbitrary artistic painting, which can be applied in controllable 3D style transfer with multi-view consistency.

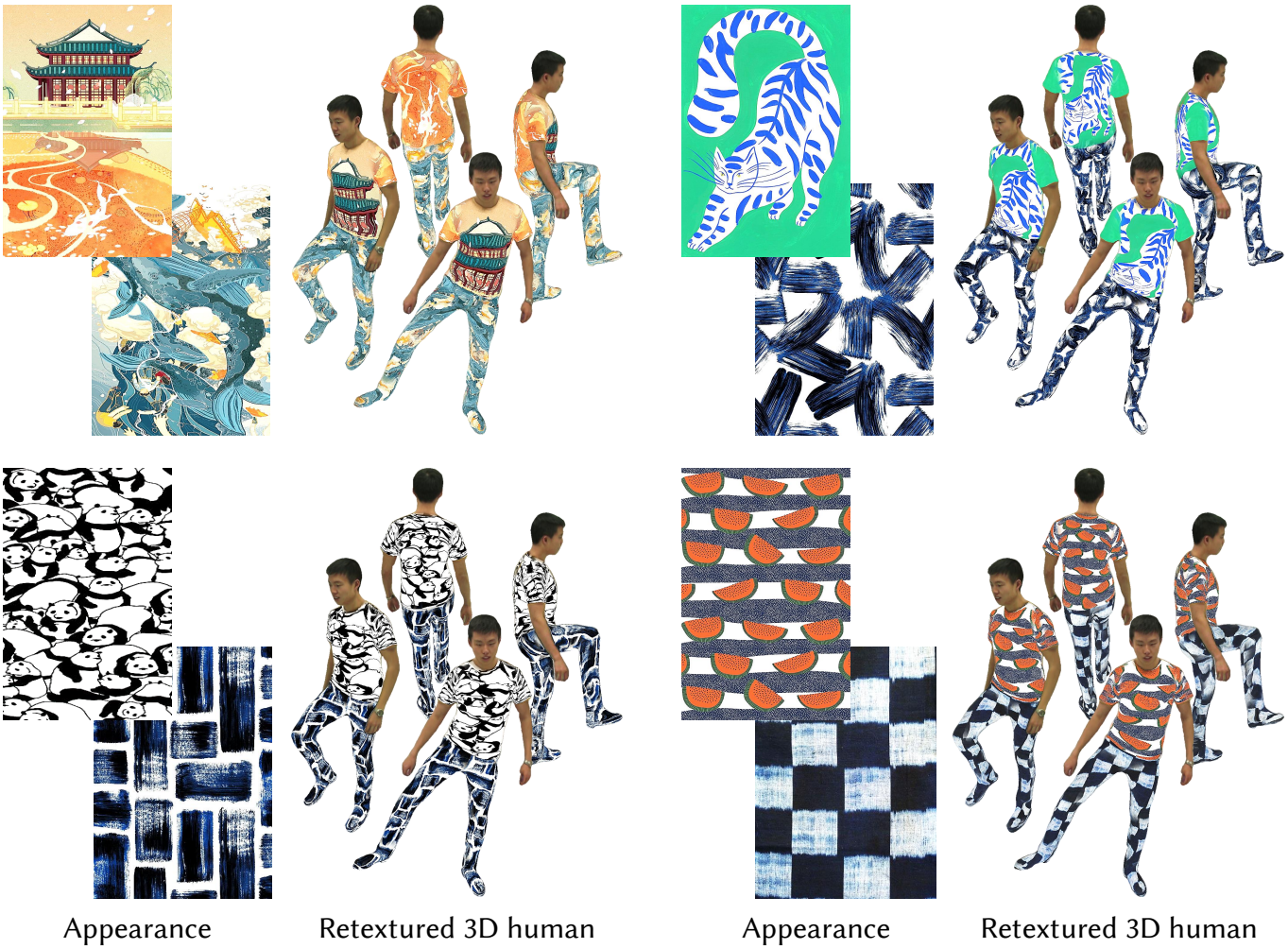



Fig. 24. Our model allows us to generate free-view human performance with a user-provided cloth texture image, which enables some interesting applications such as real-time 3D virtual try-on. We collect these appearance images from the Internet.


Fig. 25. Our model allows us to generate free-view human performance with a user-provided cloth texture image, which enables some interesting applications such as real-time 3D virtual try-on. We collect these appearance images from the Internet.