Few-shot Neural Human Performance Rendering from Sparse RGBD Videos

Anqi Pang[1,2,3], **Xin Chen**[1,2,3], **Haimin Luo**[1,2,3], **Minye Wu**[1,2,3], **Jingyi Yu**[1], **Lan Xu**[1*]

[1] Shanghai Engineering Research Center of Intelligent Vision and Imaging, School of Information Science and Technology, ShanghaiTech University

[2] Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences

[3] University of Chinese Academy of Sciences

{pangaq, chenxin2, luohm, wumy, yujingyi, xulan1}@shanghaitech.edu.cn

Abstract

Recent neural rendering approaches for human activities achieve remarkable view synthesis results, but still rely on dense input views or dense training with all the capture frames, leading to deployment difficulty and inefficient training overload. However, existing advances will be ill-posed if the input is both spatially and temporally sparse. To fill this gap, in this paper we propose a few-shot neural human rendering approach (FNHR) from only sparse RGBD inputs, which exploits the temporal and spatial redundancy to generate photo-realistic free-view output of human activities. Our FNHR is trained only on the key-frames which expand the motion manifold in the input sequences. We introduce a two-branch neural blending to combine the neural point render and classical graphics texturing pipeline, which integrates reliable observations over sparse key-frames. Furthermore, we adopt a patch-based adversarial training process to make use of the local redundancy and avoids over-fitting to the key-frames, which generates fine-detailed rendering results. Extensive experiments demonstrate the effectiveness of our approach to generate high-quality free view-point results for challenging human performances under the sparse setting.

1 Introduction

The rise of virtual and augmented reality (VR and AR) to present information in an immersive way has increased the demand of the 4D (3D spatial plus 1D time) content generation. Further reconstructing human activities and providing photo-realistic rendering from a free viewpoint evolves as a cutting-edge yet bottleneck technique.

The early high-end volumetric solutions [Dou *et al.*, 2017; Joo *et al.*, 2018; Collet *et al.*, 2015] rely on multi-view dome-based setup to achieve high-fidelity reconstruction and rendering of human activities in novel views but are expensive and difficult to be deployed. The recent low-end approaches [Xu *et al.*, 2019; Su *et al.*, 2020] have enabled light-weight and template-less performance reconstruction by leveraging the RGBD sensors and modern GPUs but are still restricted by the limited mesh resolution and suffer from the uncanny texturing output.

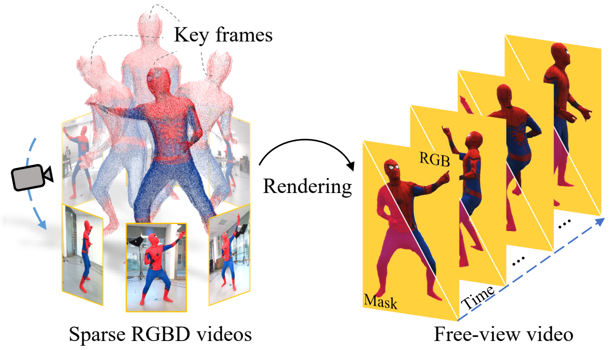

Figure 1: Our few-shot neural human rendering (FNHR) achieves photo-realistic free-view results from only six sparse RGBD inputs.

The recent neural rendering techniques [Wu *et al.*, 2020; Lombardi *et al.*, 2019a; Mildenhall *et al.*, 2020] bring huge potential for photo-realistic novel view synthesis and get rid of the heavy reliance on the reconstruction accuracy. However, for dynamic scene modeling, these approaches rely on dense spatial capture views and dense temporal training with all the capture frames, leading to deployment difficulty and inefficient training overload. On the other hand, few-shot or key-frame based strategy has been widely studied for human motion analysis [Mustafa *et al.*, 2016], revealing the temporal and spatial redundancy of human activities. However, the literature on few-shot neural human performance rendering remains sparse. Several recent works [Shysheya *et al.*, 2019] generated realistic neural avatars even based on key-frame inputs, but they rely on human body model and can hardly handle topology changes, leading to severe visual artifacts for complex performance.

In this paper, we attack these challenges and present *FNHR* – the first **F**ew-shot **N**eural **H**uman performance **R**endering approach using six sparse RGBD cameras surrounding the performer (see Fig. 1). Our approach generates photo-realistic texture of challenging human activities in novel views, whilst exploring the spatially and temporally sparse capture setup. Generating such a human free-viewpoint video

[*] Corresponding author.

by training on only spatially and temporally sparse input in an end-to-end manner is non-trivial. To this end, our key idea is to explore effective neural render design to encode the spatial, temporal, and local similarities across all the inputs, besides utilizing the inherent global information from our multi-view setting. We first formulate the key-frame selection as a pose-guided clustering problem to generate key-frames which expand the motion manifold in the input sequences. Then, based on these key-frames with coarse geometry proxy, a novel two-branch neural rendering scheme is proposed to integrates reliable observations over sparse key-frames, which consists of a neural point renderer and a classical graphics texturing renderer in a data-driven fashion. Finally, we introduce a patch-based training process in an adversarial manner to make use of the local redundancy, which not only avoids over-fitting to the key-frames, but also generates fine-detailed photo-realistic texturing results. To summarize, our main technical contributions include:

- We present the first few-shot neural human rendering approach, which can generate photo-realistic free-view results from only sparse RGBD inputs, achieving significant superiority to existing state-of-the-art.
- We propose a two-branch hybrid neural rendering design to integrate reliable observations over the sparse key-frames generated via an effective pose-guide clustering process.
- We introduce a novel patch-wise training scheme in an adversarial manner to exploit the local similarities and provide fine-detailed and photo-realistic texture results.

2 Related Work

Human Performance Capture. Markerless human performance capture techniques have been widely adopted to achieve human free-viewpoint video or reconstruct the geometry. The high-end solutions [Dou *et al.*, 2017; Joo *et al.*, 2018; Chen *et al.*, 2019] require studio-setup with the dense view of cameras and a controlled imaging environment to generate high-fidelity reconstruction and high-quality surface motion, which are expensive and difficult to deploy. The recent low-end approaches [Xiang *et al.*, 2019; Chen *et al.*, 2021] enable light-weight performance capture under the single-view setup. However, these methods require a naked human model or pre-scanned template. Recent method [Xu *et al.*, 2019; Su *et al.*, 2020] enable light-weight and template-less performance reconstruction using RGBD cameras, but they still suffer from the limited mesh resolution leading to uncanny texturing output. Comparably, our approach enables photo-realistic human free-viewpoint video generation using only spatially and temporally sparse input.

Neural Rendering. Recent work have made significant process on 3D scene modeling and photo-realistic novel view synthesis via differentiable neural rendering manner based on various data representations, such as point clouds [Wu *et al.*, 2020; Aliev *et al.*, 2019], voxels [Lombardi *et al.*, 2019b], texture meshes [Thies *et al.*, 2019] or implicit functions [Park *et al.*, 2019; Mildenhall *et al.*, 2020; Suo *et al.*, 2021]. These methods bring huge potential for photo-realistic novel view synthesis and get rid of the heavy reliance on reconstruction accuracy.

For neural rendering of dynamic scenes, Neural Volumes [Lombardi *et al.*, 2019a] adopts a VAE network to transform input images into a 3D volume representation and can generalize to novel viewpoints. NHR [Wu *et al.*, 2020] models and renders dynamic scenes through embedding spatial features with sparse dynamic point clouds. Recent work [Park *et al.*, 2020; Li *et al.*, 2020; Xian *et al.*, 2020; Tretschk *et al.*, 2020] extend the approach NeRF [Mildenhall *et al.*, 2020] using neural radiance field into the dynamic setting. They decompose the task into learning a spatial mapping from a canonical scene to the current scene at each time step and regressing the canonical radiance field. However, for all the methods above, dense spatial capture views or dense temporal training with all the capture frames are required for high fidelity novel view synthesis, leading to deployment difficulty and inefficient training overload. On the other hand, recent works [Shysheya *et al.*, 2019] generate realistic neural avatars even based on key-frame inputs, but they rely on human body model and can hardly handle topology changes, leading to severe visual artifacts for complex performance.

Comparably, our approach explores the spatially and temporally sparse capture setup and generates photo-realistic texture of challenging human activities in novel views.

3 Methods

As illustrated in Fig. 2, our FNHR explores neural human performance rendering under the few-shot and sparse-view setting to generate photo-realistic free-view results of human activities. During training, the beauty of our approach lies in its light-weight reliance on only spatial and temporal key-frames to encode the appearance information of the whole human motion sequence, which breaks the deployment difficulty and inefficient training overload of previous methods. To this end, we generate the key-frames to expand the motion manifold based on pose-guided clustering (Sec. 3.1). Then, a two-branch neural rendering scheme is proposed to take a hybrid path between the recent neural point renderer and classical graphics pipeline (Sec. 3.2). Our scheme extracts both the implicit and explicit appearance similarities of the performance between frames to overcome the spatial and temporal shortage of capture viewpoints. We also propose a patch-based adversarial re-renderer in FNHR to utilize the local redundancy, which avoids over-fitting to the key-frames and generates photo-realistic texturing details (Sec. 3.3). Our approach takes only 6 RGBD streams from stereo cameras or Azure Kinect sensors surrounding the performer as input, and the human foreground mask is extracted using DeepLab v3.

3.1 Pose-guided Key-frame Selection

Here, we introduce an effective scheme to select the representative key-frames to encode the information of the whole motion sequence, so as to avoid the heavy training process due to spatially and temporally superfluous training data. Specifically, we first apply the OpenPose [Cao *et al.*, 2017] followed by the triangulation on the six RGBD images to obtain a 3D human pose of the frame. For triangulation, we optimize a

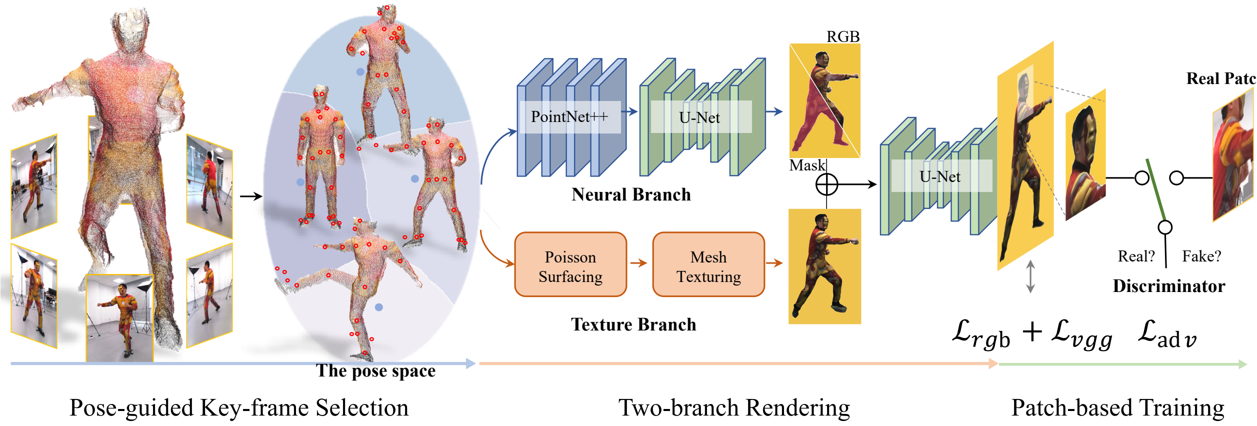

Figure 2: The pipeline of FNHR. Given the video inputs from six RGBD cameras surrounding the performer, our approach consists of key-frame selection (Sec. 3.1), two-branch rendering (Sec. 3.2), and patch-based training (Sec. 3.3) to generate free-view rendering results.

global 3D skeleton with the depth to lift it from 2D to 3D. The per-joint confidences from all six views are utilized to filter out the wrong estimation of occlude joints.

Let $\mathcal{J}_t = \{\mathbf{J}_t^1, \mathbf{J}_t^2, ..., \mathbf{J}_t^V\}$ denote the predicted 3D pose of frame t, where $V = 25$ is the number of body joints, while each element $\mathbf{J}_t^i = [x_i, y_i, z_i], i \in [1, V]$ corresponds to the 3D position of the body joint. Note that all these detected 3d poses are normalized into the same coordinates, and their root joints are aligned at the origin.

Next, the key-frames selection is formulated as a pose-guided clustering problem, which ensures these key-frames to be representative in the motion manifold so that our neural rendering networks can be generalized to the entire sequence. Thus, we conduct K-means in the pose space, the unsupervised clustering algorithm, with k cluster centers, and each cluster center represents the 3D pose of a key-frame. Since each 3D pose $\mathcal{J}_t$ is a set of 3D joint positions, we define the following distance function to measure the difference between two poses $\mathcal{J}_x$ and $\mathcal{J}_y$ of various timestamps:

$$dist(\mathcal{J}_x, \mathcal{J}_y) = \sum_{i=1}^{V} \|\mathbf{J}_x^i - \mathbf{J}_y^i\|_2. \tag{1}$$

Then, we calculate the numerical mean pose of each cluster and assign the frame with the nearest 3d pose such numerical central pose using the same distance function defined in Eqn. 1. The poses of these assigned frames are set to be the new cluster centers to ensure that the cluster centers are always located on exactly the frames from the sequence.

After several clustering iterations until convergence, the key-frames of a sequence are selected from all the cluster centers. For multiple people scenarios, we concatenate the 3D human poses under the same timestamps and extend the Eqn. 1 to accumulate the differences of corresponding joint pairs. In our neural renderer training, we set k to be 20 for a typical motion sequence with about 500 frames, leading to 4% sparsity of capture view sampling.

3.2 Two-branch Rendering

Here we introduce a novel neural render design to encode the self-similarities across the spatially and temporally sparse key-frames. Our key observation is that existing dynamic neural point renderers like [Wu *et al.*, 2020] and the graphics texturing pipeline are complementary to each other under our challenging sparse setting. The former one leads to photo-realistic texture in the input views but suffers from strong artifacts in between, while the latter one provides spatially consistent rendering but the result suffers from reconstructed geometry error. Thus, we propose a two-branch neural renderer to take a hybrid path between the recent neural point renderer and classical graphics pipeline, which utilizes both the implicit neural features and explicit appearance features to integrate reliable observations over sparse key-frames.

Graphics Texturing Branch. To provide consistent rendering as a good anchor under our sparse setting, for the t-th frame, we use a similar fuse strategy to DynamicFusion to fuse the six depth images into a texture mesh $\mathbf{P}_t$ via Poisson reconstruction. Then, classic graphics texturing mapping with mosaicing is adopted to generate a textured mesh and corresponding rendering image $\mathbf{I}_{\text{tex}}$. Note that $\mathbf{I}_{\text{tex}}$ suffers from texturing artifacts due to sparse-view reconstruction, but it preserves view coherent information and serves as an appearance prior for the following neural branch.

Neural Point Renderer Branch. We further adopt a neural point renderer to implicitly encode the appearance similarities between sparse key-frames in a self-supervised manner. Similar to the previous dynamic neural point rendering approach [Wu *et al.*, 2020], we adopt a share-weighted PointNet++ on all the fused textured point clouds to extract point features and then the renderer splats point features into pixel coordinates on the target image plane with depth order to form the feature map. To avoid over-fitting to the key-frames due to our sparse setting, we randomly drop out 20% points before feeding the point clouds into PointNet++. Then, a modified U-Net with the gated convolution is applied on the feature map to generate the texture output with a foreground mask. Let ψ_{Neural} denote our neural renderer as follows:

$$\mathbf{I}_{\text{Neural}}, \mathbf{M}_{\text{Neural}} = \psi_{\text{Neural}}(\mathbf{P}_t, \mathbf{K}, \mathbf{T}), \tag{2}$$

where $\mathbf{K}$ and $\mathbf{T}$ are the intrinsic and extrinsic matrices of the target view. $\mathbf{I}_{\text{Neural}}$ and $\mathbf{M}_{\text{Neural}}$ are rendered color image

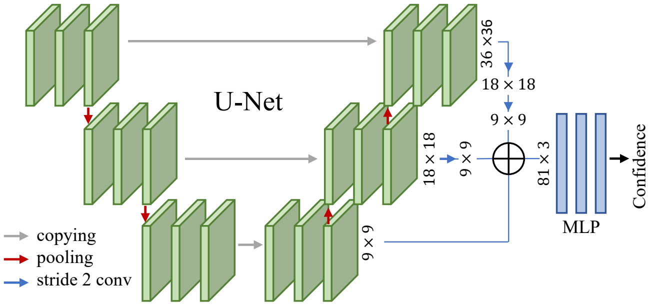

Figure 3: The network architecture of our multi-scale discriminator in patch-based adversarial training.

and foreground mask, encoding the inherent appearance information from the key-frames into the render view.

Two-branch Blending. Finally, another U-Net based network ψ_{fuse} is adopted to fuse the above neural rendering result $\mathbf{I}_{\text{Neural}}$ and the appearance prior $\mathbf{I}_{\text{tex}}$ from our two branches, so as to obtain the final texture output as follows:

$$\mathbf{I}_* = \psi_{\text{fuse}}(\mathbf{I}_{\text{Neural}}, \mathbf{I}_{\text{tex}}), \tag{3}$$

where $\mathbf{I}_{\text{Neural}}$ and $\mathbf{I}_{\text{tex}}$ are concatenated and fed into fuse net. Our blending scheme jointly utilizes both the implicit and explicit appearance similarities to overcome the spatial and temporal shortage of capture viewpoints.

3.3 Patch-based Adversarial Training

To handle the lack of diversity of training samples in our few-shot and sparse-view setting, we introduce a novel patch-based adversarial training scheme. Our scheme improves the generalization of our neural render and the fine-grained details in the texture output significantly. To this end, we randomly sample rectangular 36×36 patches from $\mathbf{I}_*$ and assign a discriminator network D to predict if a patch comes from the real image or not. Such patch-wise strategy narrows the scope of the real image distribution down and provides sufficiently diverse training samples to avoid overfitting to our small number of key-frames. Moreover, we mask background pixels of $\mathbf{I}_*$ out using the foreground masks $\mathbf{M}$ and define a patch as valid if the patch has 10% of pixels belongs to foreground objects in our adversarial training.

Multi-Scale Discriminator Network. To distinguish the patches effectively, we design a multi-scale discriminator network as illustrated in Fig. 3. Our discriminator D adopts a 3-level U-Net to extract multi-scale features as well as a multiple layer perceptron (MLP) to aggregate features from different levels for final classification. Specifically, at each level in the upsampling stream of U-Net, we use fully convolutional layers with a stride of 2 to downsample the feature map into a unified size of 9×9, and then flatten the feature map to a one dimension feature vector. All feature vectors from three levels are concatenated together as the input of the MLP. The MLP has three layers followed by ReLU activation with widths of 256, 128, 1, respectively. Such multi-scale features enhance the discriminative sensing ability to multi-scale details for our discriminator D. Besides, our two-branch renderer serves as the generator where the Pointnet++ and U-Net in ψ_{Neural} share the same architecture from [Wu *et al.*, 2020] while the U-Net in ψ_{fuse} has 5 level convolutional layers with skip connections similar to [Ronneberger *et al.*, 2015].

Training Details. To enable few-shot and sparse-view neural human rendering, we need to train the fusion net ψ_{fuse} and discriminator D simultaneously. But before that, we firstly bootstrap the neural point renderer branch on key-frame data set with 10 training epochs.

Next, we exploit the following loss functions to supervise the training of the fusion net a and the discriminator. As described above, we apply patch-based training using the adversarial losses:

$$\begin{aligned} \mathcal{L}_{adv_D} &= \frac{1}{\mid \mathcal{B}_r \mid} \sum_{x \in \mathcal{B}_r} (1 - D(x))^2 + \frac{1}{\mid \mathcal{B}_* \mid} \sum_{y \in \mathcal{B}_*} D(y)^2 \\ \mathcal{L}_{adv_G} &= -\frac{1}{\mid \mathcal{B}_* \mid} \sum_{y \in \mathcal{B}_*} D(y)^2, \end{aligned} \tag{4}$$

where $\mathcal{B}_r$ and $\mathcal{B}_*$ are the set of valid patches from real images and rendering results respectively; $\mathcal{L}_{adv_D}$ is only for updating the discriminator, and $\mathcal{L}_{adv_G}$ is only for updating the renderer; $\mathcal{L}_{adv} = \mathcal{L}_{adv_D} + \mathcal{L}_{adv_G}$. Here we use the L2 norm for better training stability in practice. Then gradients of $\mathbf{I}_*$ are accumulated from random sampled patches'.

In addition to the patch-based adversarial loss $\mathcal{L}_{adv}$, we also apply L1 loss $\mathcal{L}_{rgb}$ and perceptual loss $\mathcal{L}_{vgg}$ on the output of fusion net as:

$$\begin{aligned} \mathcal{L}_{rgb} &= \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{I}_i - \mathbf{I}_i^*\|_1, \\ \mathcal{L}_{vgg} &= \frac{1}{n} \sum_{i=1}^{n} \|\Phi_{\text{VGG}}(\mathbf{I}_i) - \Phi_{\text{VGG}}(\mathbf{I}_i^*)\|_2, \end{aligned} \tag{5}$$

where n is the number of one batch samples; $\mathbf{I}$ is the masked ground truth image; $\Phi_{\text{VGG}}(\cdot)$ extracts feature maps of input from the 2th and 4th layer of the VGG-19 network which is pretrained on ImageNet.

We linearly combine these three losses and have the total loss:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{adv} + \lambda_2 \mathcal{L}_{rgb} + \lambda_3 \mathcal{L}_{vgg}, \tag{6}$$

where λ_1, λ_2 and λ_3 are weights of importance. We set them to 0.3, 5, and 0.7 respectively in our experiments. We use the Adam optimizer with a learning rate of 0.0002 to optimize the network parameters. The batch size is 4, and the number of one batch sample is 40.

We also augment our training data with random translation, random scaling, and random rotation. These transformations are applied on 2D images and camera parameters. Benefiting from these loss functions and training strategies, our FNHR can synthesize photo-realistic free-view video under the few-shot and sparse-view setting.

4 Experiment

In this section, we evaluate our FNHR approach on a variety of challenging scenarios. We run our experiments on a PC with 2.2 GHz Intel Xeon 4210 CPU 64GB RAM, and

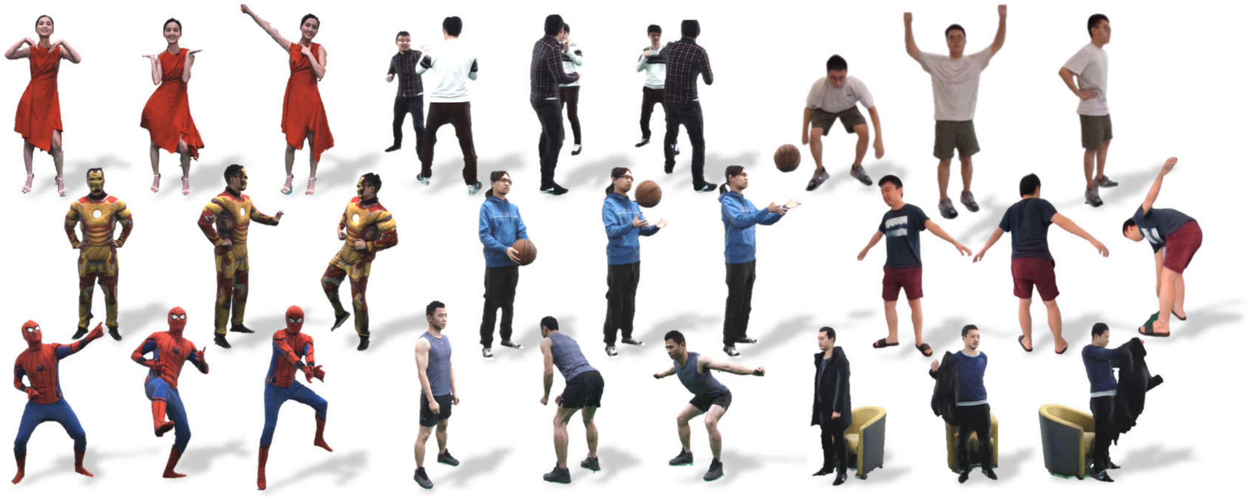

Figure 4: The photo-realistic free-view rendering texture results of the proposed few-shot neural rendering approach.

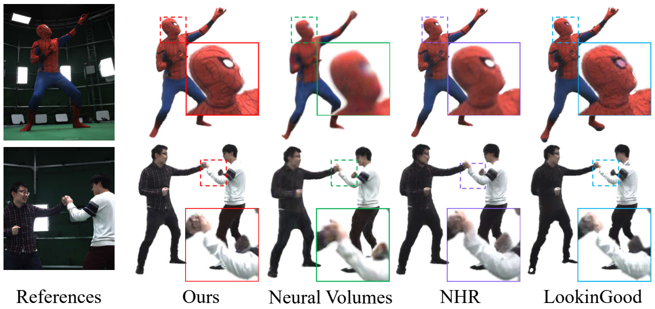

Figure 5: Qualitative comparison. Our approach generates more photo-realistic texture details than other methods.

Nvidia TITAN RTX GPU. It takes 541 ms to render a 720 × 1280 frame. As demonstrated in Fig. 4 our approach generates high-quality texture results under the few-shot setting and even handles human-object or multi-human interaction scenarios with topology changes, such as playing basketball, removing clothes, or fighting.

4.1 Comparison

To the best of our knowledge, our approach is the first few-shot neural human rendering approach. Therefore, we compare to existing dynamic neural rendering methods, including the voxel-based **Neural Volumes** [Lombardi *et al.*, 2019a], point-based **NHR** [Wu *et al.*, 2020], mesh-based **LookinGood** [Martin-Brualla *et al.*, 2018] using the same sparse training data for fair comparison. As shown in Fig. 5, all the other methods with various data representation suffer from uncanny texture results due to the shortage of training viewpoints. In contrast, our approach achieves significantly better rendering results in terms of sharpness and realism, under the challenging sparse setting. Then, we compare to the implicit method **NeRF** [Mildenhall *et al.*, 2020] based on the neural radiance field. Since NeRF only handles static scene, we use the six RGB images of the target frame as input and use the fused geometry from depths as the prior of initial density during the training of NeRF for a fair comparison. As illustrated in Fig.6, our approach reconstruct more sharp texture ouptut than NeRF under the few-shot setting. Moreover, NeRF takes 2 hours for one frame training, leading to 800 hours for a sequence with frames, while our approach takes about 8 hours for training this sequence, achieving about 100× speedup.

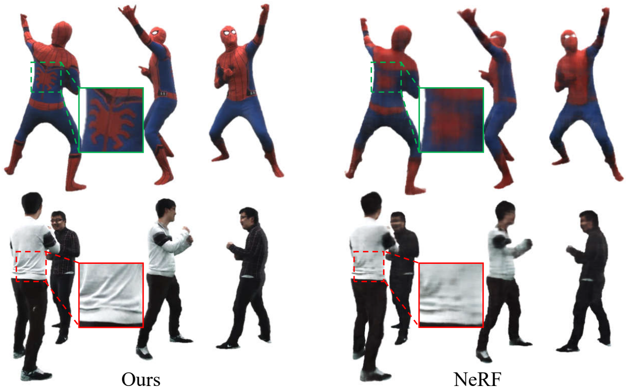

Figure 6: Qualitative comparison against NeRF. Using few-shot training data, our method achieves more sharp free-view rendering.

For quantitative comparison, we adopt the peak signal-to-noise ratio (**PSNR**), structural similarity index (**SSIM**), the L1 loss (**Photometric Error**) and the mean squared error (**MSE**) as metrics similar to previous methods. Note that all the quantitative results are calculated in the reference captured views. As shown in Tab. 1, our approach consistently outperforms the other baselines in terms of all these metrics above, illustrating the effectiveness of our approach to handle the sparse setting and provide high-quality rendering results.

4.2 Evaluation

Here, we first evaluate the individual components of the proposed FNHR. Let **w/o classic** and **w/o neural** denote the variations of FNHR without the graphic texturing branch and the neural point rendering branch in Sec. 3.2, respectively. We

Method	PSNR↑	SSIM↑	Photometric error ↓	MSE ↓
Neural Volumes	25.69	0.9145	0.056	0.010
LookinGood	27.11	0.9518	0.052	0.008
NeRF	28.35	0.9657	0.041	0.006
NHR	30.08	0.9626	0.038	0.004
Ours	**33.83**	**0.9807**	**0.025**	**0.002**

Table 1: Quantitative comparison against various methods under various metrics. Our method achieve consistently better results.

Method	PSNR↑	SSIM↑	Photometric error ↓	MSE ↓
w/o classic	28.72	0.9107	0.039	0.005
w/o neural	27.71	0.8743	0.048	0.006
w/o adversarial	30.56	0.9218	0.035	0.003
Ours	**33.82**	**0.9602**	**0.025**	**0.002**

Table 2: Quantitative evaluation for our rendering modules. The result shows that adopt all the branch can generate better result than using either. And with the well designed final renderer the result achieves less errors

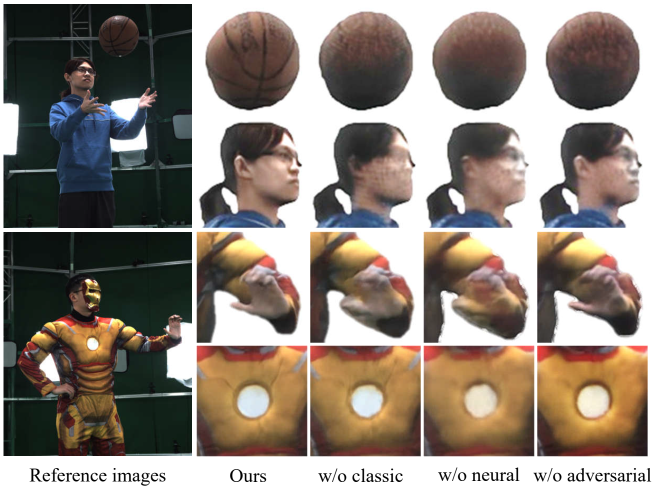

Figure 7: Ablation study for the various components of our approach. Our full pipeline achieves more realistic rendering results.

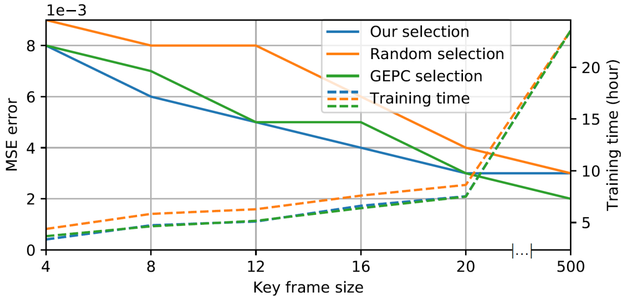

Figure 8: Evaluation of our key-frame selection. Our pose-guided selection scheme consistently outperforms the random one.

also compare against the variation which is supervised directly using the L1 loss and perception loss without the patch-based adversarial training, denoted as **w/o adversarial**. As shown in Fig. 7, only a single-branch rendering suffers from uncanny texture artifacts in novel views, while the lack of adversarial training leads to severe blur texturing output. In contrast, our full pipeline enables photo-realistic texture reconstruction in novel views. For further analysis of the individual components, we utilize the same four metrics as the previous subsection, as shown in Tab. 2. This not only highlights the contribution of each algorithmic component but also illustrates that our approach can robustly render texture details.

We further evaluate our key-frame selection strategy under various numbers of key-frames. As shown in Fig. 8, our pose-guided strategy consistently outperforms the random one in terms of accuracy and training efficiency. The Graph Embedded Pose Clustering (GEPC) method [Markovitz *et al.*, 2020] performs sililar to our strategy. Besides, it can be seen from Fig. 8 that using more key-frames will increase the training time but improve the accuracy. Empirically, the setting with 20 key-frames for a sequence with about 500 frames serves as a good compromising settlement between effectiveness and efficiency, which achieves reasonable MSE error and reduces nearly 70% training time.

4.3 Limitation and Discussion

We have demonstrated compelling neural rendering results of a variety of challenging scenarios. Nevertheless, as the first trial of a few-shot neural human performance rendering approach, the proposed FNHR is subject to some limitations. First, the training process of FNHR to provide a visually pleasant rendering result for about 500 frames takes about 6 to 8 hours, which is not suitable for online applications. Besides, our approach is fragile to challenging self-occlusion motions and severe segmentation error, leading to body-part missing during rendering. How to handle such challenging scenarios remains an open problem for future research. Furthermore, our approach relies on a consistent lighting assumption. It's promising to handle complex changing lighting conditions for view-dependent rendering under our challenging sparse setting. From the system aspect, our FNHR still relies on six RGBD images to provide a good appearance prior. It's an interesting direction to use fewer RGB inputs for data-driven in-the-wild rendering.

5 Conclusion

We have presented the first few-shot neural human performance rendering approach to generate photo-realistic textures of human activities in novel views with only spatially and temporally sparse RGBD training inputs. Our experimental results show the effectiveness of our approach for challenging human performance rendering with various poses, clothing types, and topology changes under sparse setting. We believe that it is critical step for neural human performance analysis, with many potential applications in VR/AR like gaming, entertainment and immersive telepresence.

Acknowledgements

This work was supported by NSFC programs (61976138, 61977047), the National Key Research and Development Program (2018YFB2100500), STCSM (2015F0203-000-06) and SHMEC (2019-01-07-00-01-E00003)

References

[Aliev *et al.*, 2019] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. *arXiv preprint arXiv:1906.08240*, 2019.

[Cao *et al.*, 2017] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[Chen *et al.*, 2019] Xin Chen, Anqi Pang, Yang Wei, Lan Xui, and Jingyi Yu. Tightcap: 3d human shape capture with clothing tightness. *arXiv preprint arXiv:1904.02601*, 2019.

[Chen *et al.*, 2021] Xin Chen, Anqi Pang, Wei Yang, Yuexin Ma, Lan Xu, and Jingyi Yu. Sportscap: Monocular 3d human motion capture and fine-grained understanding in challenging sports videos. *arXiv preprint arXiv:2104.11452*, 2021.

[Collet *et al.*, 2015] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (TOG)*, 34(4):69, 2015.

[Dou *et al.*, 2017] Mingsong Dou, Philip Davidson, Sean Ryan Fanello, Sameh Khamis, Adarsh Kowdle, Christoph Rhemann, Vladimir Tankovich, and Shahram Izadi. Motion2fusion: Real-time volumetric performance capture. *ACM Trans. Graph.*, 36(6):246:1–246:16, November 2017.

[Joo *et al.*, 2018] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[Li *et al.*, 2020] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. *arXiv preprint arXiv:2011.13084*, 2020.

[Lombardi *et al.*, 2019a] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Trans. Graph.*, 38(4), July 2019.

[Lombardi *et al.*, 2019b] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019.

[Markovitz *et al.*, 2020] Amir Markovitz, Gilad Sharir, Itamar Friedman, Lihi Zelnik-Manor, and Shai Avidan. Graph embedded pose clustering for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10539–10547, 2020.

[Martin-Brualla *et al.*, 2018] Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, and et al. Lookingood: Enhancing performance capture with real-time neural re-rendering. *ACM Trans. Graph.*, 37(6), December 2018.

[Mildenhall *et al.*, 2020] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 405–421, Cham, 2020. Springer International Publishing.

[Mustafa *et al.*, 2016] Armin Mustafa, Hansung Kim, and Adrian Hilton. 4d match trees for non-rigid surface alignment. In *European Conference on Computer Vision*, pages 213–229. Springer, 2016.

[Park *et al.*, 2019] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[Park *et al.*, 2020] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo-Martin Brualla. Deformable neural radiance fields. *arXiv preprint arXiv:2011.12948*, 2020.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[Shysheya *et al.*, 2019] Aliaksandra Shysheya, Egor Zakharov, Kara-Ali Aliev, Renat Bashirov, Egor Burkov, Karim Iskakov, Aleksei Ivakhnenko, Yury Malkov, Igor Pasechnik, Dmitry Ulyanov, et al. Textured neural avatars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2397, 2019.

[Su *et al.*, 2020] Zhuo Su, Lan Xu, Zerong Zheng, Tao Yu, Yebin Liu, and Lu Fang. Robustfusion: Human volumetric capture with data-driven visual cues using a rgbd camera. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 246–264, Cham, 2020. Springer International Publishing.

[Suo *et al.*, 2021] Xin Suo, Yuheng Jiang, Pei Lin, Yingliang Zhang, Kaiwen Guo, Minye Wu, and Lan Xu. Neuralhuman-fvv: Real-time neural volumetric human performance rendering using rgb cameras. *arXiv preprint arXiv:2103.07700*, 2021.

[Thies *et al.*, 2019] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019.

[Tretschk *et al.*, 2020] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a deforming scene from monocular video. *arXiv preprint arXiv:2012.12247*, 2020.

[Wu *et al.*, 2020] Minye Wu, Yuehao Wang, Qiang Hu, and Jingyi Yu. Multi-view neural human rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[Xian *et al.*, 2020] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. *arXiv preprint arXiv:2011.12950*, 2020.

[Xiang *et al.*, 2019] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular total capture: Posing face, body, and hands in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[Xu *et al.*, 2019] L. Xu, Z. Su, L. Han, T. Yu, Y. Liu, and L. FANG. Unstructuredfusion: Realtime 4d geometry and texture reconstruction using commercialrgbd cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019.