Neural Human Performer: Learning Generalizable Radiance Fields for Human Performance Rendering

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this paper, we aim at synthesizing a free-viewpoint video of an arbitrary human performance using sparse multi-view cameras. Recently, several works have proposed to use pixel-aligned radiance fields to enable generalization to arbitrary new scenes at test time. However, it is still highly challenging to reconstruct human performance from sparse input views due to the heavy occlusions and dynamic articulations of body parts. To tackle this, we propose Neural Human Performer, a novel approach that learns generalizable radiance fields based on a parametric human body model for robust performance capture. Specifically, we first introduce a temporal transformer that aggregates trackable visual features based on the skeletal body motions over video frames. Moreover, a multi-view transformer is proposed to perform cross-attention between the temporally-fused features and the pixel-aligned features at each time step to integrate observations on the fly from multiple views. Experiments on ZJU-MoCap and AIST datasets show that our method significantly outperforms recent generalizable NeRF methods on unseen identities and poses.

1 Introduction

Free-viewpoint video of a human performer has a variety of applications in the area of telepresence, mixed reality, gaming and so on. Conventional free-viewpoint video systems require extremely expensive setups such as dense camera rigs [4, 8, 41] or accurate depth sensors [6, 14], even for person-specific rendering. In this paper, we aim at a scalable solution for free-viewpoint human performance rendering that can generalize across different human performers and require only sparse camera views. This setting significantly reduces the cost of free-viewpoint systems and enables the creation of digital humans *at scale*. However, representing and rendering arbitrary human performances is extremely challenging when the observations are highly sparse (up to three to four views) due to heavy self-occlusion and dynamic articulations of the body parts. The main challenges we need to tackle are: how to represent arbitrary human motions in 3D? and how to obtain this representation *on-the-fly* from the multi-time and multi-view observations?

Recently, neural radiance fields (NeRF) [26] and the following works [12, 17, 30, 31, 33, 34, 45, 48, 50, 51, 53] have shown photo-realistic novel view synthesis results in per-scene optimization settings. To avoid the expensive training and improve the practicality, generalizable NeRFs [34, 50, 45] have proposed to use image-conditioned, pixel-aligned features and achieved feed-forward view synthesis from very sparse input views [34, 50]. However, direct application of these methods to complex and non-rigid human motion is extremely under-constrained and in turn, suffers inaccurate rendering of body parts as shown in Fig. 3. Furthermore, existing methods [35, 50] combine single-view features by simple average pooling, which will lead to over-smoothed outputs when different visible details are included in multi-view observations (*e.g.*, front and side views). To compensate for the

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

lack of observations for human rendering, several methods [23, 31] have proposed to learn global representations from all video frames. However, their representations are to *memorize* a per-person video and thus are not able to generalize to new human performers.

To address this, we propose Neural Human Performer, a novel approach that learns generalizable radiance fields based on a parametric body model for robust performance capture. Aside from a natural advantage of the parametric body model as a geometric prior, the core of our method is a combination of temporal and multi-view Transformers [44] which helps the best use of the spatio-temporal observations, to compute the density and color of a query point. First, the temporal transformer aggregates trackable visual features based on the input skeletal body motions over video frames. The following multi-view transformer performs cross-attention between the temporally-augmented skeletal features and the pixel-aligned features from each time step. The proposed modules collectively contribute to the adaptive aggregation of multi-time and multi-view information, and finally contribute to the significant improvement of synthesis results in different generalization settings.

We study the efficacy of Neural Human Performer on two multi-view human performance capture datasets, ZJU-MoCap [31] and AIST [16]. Experiments show that our method significantly outperforms recent generalizable rendering (NeRF) methods on unseen identities and poses of human performers. Furthermore, we compare with per-person optimization methods [31, 42, 47] that utilize a human body model. Surprisingly, our generalization onto unseen human performers achieves better rendering quality than the dedicated methods that are per-person optimized when tested on the person's novel poses. This confirms that our improvements are not merely from the use of the human body prior, but the contributions of the learnt generalizable representations and the transformer-based framework design are significant.

To summarize, our contributions are:

- We present a new feed-forward method of synthesizing novel-view video of arbitrary human performers from sparse camera views. We propose Neural Human Performer that learns generalizable neural radiance representations by leveraging the skeletal body motion prior.
- We design a combination of temporal and multi-view transformers that can aggregate information on the fly over video frames and from multiple views, to render each frame of the novel-view video.
- We show significant improvements over recent generalizable NeRF methods on novel identities and poses. Moreover, our generalization results can outperform even per-person methods on unseen poses of the person they are optimized with.

2 Related works

Human performance capture. Novel view synthesis of human performance has a long history in computer vision and graphics. Traditional methods rely on complicated hardware such as dense camera rigs [4, 8, 41] or accurate depth sensors [6, 14]. To enable free-view video from sparse camera views, template-based methods [3, 5, 11, 40] exploit pre-scanned human models to track the motion of a person. However, their synthesis results are not photo-realistic and pre-scanned human models are not available in most cases. Instead of optimizing a single network per scene, recent learning-based methods [27, 35, 36, 52] learn human geometry prior from ground-truth 3D data, enabling the 3D human reconstruction from even a single image. However, these methods often suffer under complex human poses that are never seen during training.

Neural 3D representations. Recently, there has been great progress in learning neural networks to represent the shape and appearance of scenes. The 3D representations are learned from 2D images via differentiable rendering networks. Convolutional neural networks are used to predict volumetric representations via 3D voxel-grid features [38, 23, 29, 25, 15], point clouds [1, 47], textured meshes [18, 21, 42] and multi-plane images [10, 54]. The learnt representations are projected by a 3D-2D operation to synthesize images. However, these methods often have difficulty in scaling to higher resolution due to memory restrictions, and in rendering multi-view consistent images.

To eschew these problems, implicit function-based methods [20, 22, 28, 39] learn a multi-layer perceptron that directly translates a 3D positional feature into a pixel generator. The more recent NeRF [26] learns implicit fields of density and color with a volume rendering technique and achieves

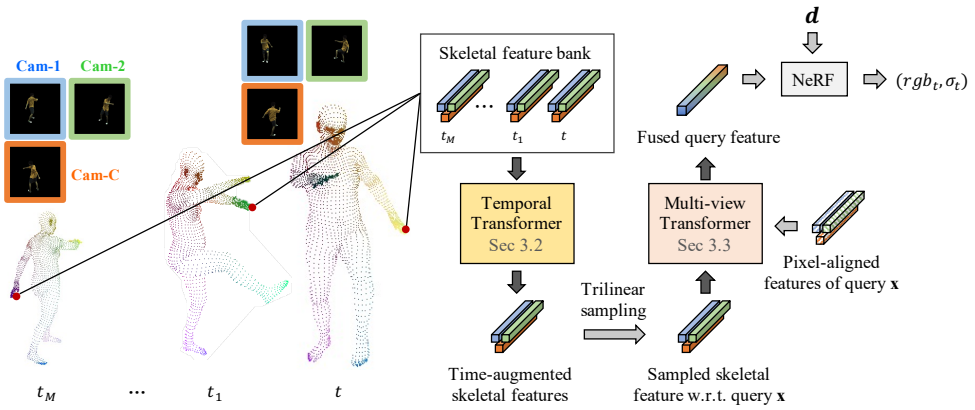

Figure 1: **Overview of Neural Human Performer.**

photo-realistic view synthesis. Among many following NeRF extensions, [30, 33, 51, 17] focus on dynamic scenes. While they can handle some dynamic scenes, it's an extremely under-constrained problem to jointly learn NeRF and deformation fields. To regularize the training, Neural Body [31] combines NeRF with a human body model (*e.g.*, SMPL [24]). However, SMPL models are inherently not accurate in 3D space (the gap between naked model and real complex geometry). Despite the promising results, these video NeRF [17, 51, 53] and human NeRF [12, 30, 31, 33, 48] methods must be optimized for each new video separately, and generalize poorly on novel scenarios. Generalizable NeRFs [34, 45, 50] try to avoid the expensive per-scene optimization by image-conditioning using pixel-aligned features. However, modeling complex and dynamic 3D humans is an extremely challenging problem when available observations are highly sparse. Unlike existing works, our method exploits temporal and multi-view information on-the-fly and achieves free-viewpoint human rendering in a *feed-forward* manner, also generalizing to new, unseen human identities and poses.

3 Method

3.1 Neural Human Performer

Problem definition. Given a sparse set of multi-view cameras $c = 1, \ldots, C$, input videos of an arbitrary human performance $I_{c,1:T} := \{I_{c,1}, I_{c,2}, ..., I_{c,T}\}$ are captured for each camera view c defined by $\{\mathbf{K}_c, [\mathbf{R}|\mathbf{t}]_c\}$. Our problem to solve is to synthesize a novel view video $\hat{I}_{q,1:T}$ for a query viewpoint q defined by $\{\mathbf{K}_q, [\mathbf{R}|\mathbf{t}]_q\}$.

Overview. To compensate for the sparsity of available input views, we propose to exploit temporal information across video frames. In practice, we sample M memory frames from the original input videos to augment each queried timestep t. Our goal is to learn generalizable 3D representations of human performers from multi-time (M) and multi-view (C) observations.

To this end, the Neural Human Performer is proposed with two main components. The overview is illustrated in Fig. 1. First, we construct the time-augmented skeletal features $\{s\}$. We exploit a human body model (SMPL [24]) to construct 3D skeletal features by projecting all the SMPL vertices onto each memory frame and picking up the pixel-aligned image features [34, 35, 50] at the projected 2D locations. The skeletal features are sampled from all memory frames to construct the skeletal feature bank. Inspired by Transformers[2, 44, 46], we propose a temporal Transformer that aggregates these memory features into the time-augmented skeletal features $\{s'\}$.

In the second stage, a query 3D point $\mathbf{x}$ is given, and the corresponding skeletal features are trilinearly sampled from the queried location. In addition, pixel-aligned features $\{p\}$ at each time t are sampled by directly projecting $\mathbf{x}$ onto the input images $\{I\}_t$. The multi-view Transformer is proposed to learn the correlation between the pixel-aligned features $\{p\}$ and the time-augmented skeletal features $\{s'\}$, and to adaptively fuse multi-view information.

Finally, the output representation of the query point $\mathbf{x}$ is fed into the radiance field module to become the color and density values.

3.2 Construction of time-augmented skeletal features

For video inputs of moving characters, compared to the static scenes, there are inherently more visual cues as the occluded regions in a frame may be visible in other (potentially distant) frames. To take advantage of the temporal information, we first build up the skeletal feature bank from memory frames by leveraging a parametric body model (see Fig. 1). Then, we propose a temporal Transformer module that aggregates the collected time information.

For each view c and time t, we first build frame-level skeletal features $s_{c,t} \in \mathbb{R}^{L\times d}$ by sampling image features at the SMPL vertices' $\mathbb{R}^{3\to 2}$ projected locations on $I_{c,t} \in \mathbb{R}^{H\times W\times 3}$. L denotes the number of SMPL vertices and d the dimension of image features.

After collecting all the skeletal features from all memory frames, we propose to aggregate the time information in an attention-aware manner, instead of using simple average pooling.

For any i^{th} skeletal feature vertex $s^i_{c,t} \in \mathbb{R}^d$, the proposed temporal Transformer casts attention over all other features contained in the vertex's memory bank $s^i_{c,t_1:t_M} = \{s^i_{c,t_1}, s^i_{c,t_2}, ..., s^i_{c,t_M}\} \in \mathbb{R}^{M\times d}$. In particular, soft weights of all memory feature vertices are computed in a non-local manner with respect to the current timestep t. Then, the values of the memory features are weighted summed as

$$\begin{aligned} t_att^i_{c,t} &= \psi\big(\frac{1}{\sqrt{d_0}} q(s^i_{c,t}) \cdot k(s^i_{c,t_1:t_M})^T\big), \quad t_att^i_{c,t} \in \mathbb{R}^{1\times M} \\ {s'_{c,t}}^i &= t_att^i_{c,t} \cdot v(s^i_{c,t_1:t_M}) + s^i_{c,t}, \qquad s'_{c,t} \in \mathbb{R}^{L\times d} \quad \forall i \end{aligned} \tag{1}$$

where ψ represents the softmax operator along the second axis, $q(\cdot)$, $k(\cdot)$ and $v(\cdot)$ are learnable query, key and value embedding functions $\mathbb{R}^{d\to d_0}$ of the temporal Transformer.

In other words, the representation $s^i_{c,t}$ of each skeletal vertex at time t is computed through a dynamically weighted combination of all its previous and next representations in the memory frames. This allows our network to incorporate helpful information and ignore irrelevant ones from other timesteps. In practice, the temporal Transformer operation in Eq. (1) is performed by a batch matrix multiplication for all skeletal vertices L and all available viewpoints C.

3.3 Multi-view aggregation of skeletal and query features

Given a query 3D point $\mathbf{x} \in \mathbb{R}^3$, we retrieve the corresponding (time-augmented) skeletal feature ${s'_{c,t}}^{\mathbf{x}} \in \mathbb{R}^d$ at the queried location via trilinear interpolation in the SMPL space with SparseConvNet [19], following [37, 32, 49, 31].

In addition, we sample pixel-aligned image feature $p^{\mathbf{x}}_{c,t}$ via direct $\mathbb{R}^{3\to 2}$ projection of the query point $\mathbf{x}$ on $I_{c,t}$. It is important to note that the pixel-aligned feature $p^{\mathbf{x}}_{c,t}$ is time-specific and represents the exact query location of $\mathbf{x}$, while the skeletal feature ${s'_{c,t}}^{\mathbf{x}}$ is time-augmented (w.r.t t) and contains inherent geometric deviations in the SMPL vertices and the following trilinear interpolations. We propose to combine these two complementary features, which will be shown to be effective in Sec. 4.3.

Given the two sets of multi-view features, skeletal ${s'_{1:C,t}}^{\mathbf{x}} = \{{s'_{c,t}}^{\mathbf{x}} | c = 1, ..., C\} \in \mathbb{R}^{C\times d}$ and pixel-aligned $p^{\mathbf{x}}_{1:C,t} = \{p^{\mathbf{x}}_{c,t} | c = 1, ..., C\} \in \mathbb{R}^{C\times d}$, we propose a multi-view Transformer that performs cross-attention from skeletal to pixel-aligned features. Specifically, the values of pixel-aligned features from all viewpoints is re-weighted based on how much compatible they are with each skeletal features. The non-local cross-attention mv_att is constructed as:

$$\begin{aligned} mv_att^{\mathbf{x}}_t &= \psi\big(\frac{1}{\sqrt{d_1}} k({s'_{1:C,t}}^{\mathbf{x}}) \cdot k(p^{\mathbf{x}}_{1:C,t})^T\big), \quad mv_att^{\mathbf{x}}_t \in \mathbb{R}^{C\times C} \\ z^{\mathbf{x}}_{1:C,t} &= mv_att^{\mathbf{x}}_t \cdot v(p^{\mathbf{x}}_{1:C,t}) + v({s'_{1:C,t}}^{\mathbf{x}}), \qquad z^{\mathbf{x}}_{1:C,t} \in \mathbb{R}^{C\times d}, \quad z^{\mathbf{x}}_{c,t} \in \mathbb{R}^{1\times d} \end{aligned} \tag{2}$$

where ψ represents the softmax operator along the second axis. Note that k and v are new layers independent from those in the temporal Transformer. Multi-head attention is used in the multi-view Transformer, *i.e.*, $k, v \in \mathbb{R}^{d\to n_{head}\times d_1}$ encode the multi-view features into n_{head} different embedding subspaces. It allows the network to capture the different geometry patterns in multiple views in parallel. The confident observations in each view will have large weights and be highlighted, and vice versa. Finally, we use the view-wise mean of $z^{\mathbf{x}}_t = \frac{1}{C}\sum_c z^{\mathbf{x}}_{c,t} \in \mathbb{R}^d$ as our *meta-time* and *meta-view* representation of the query point x.

The final density $\sigma_t(\mathbf{x})$ and color values $rgb_t(\mathbf{x})$ at time t are computed as:

$$\begin{aligned} \sigma_t(\mathbf{x}) &= MLP_\sigma(z_t^{\mathbf{x}}), \\ rgb_t(\mathbf{X}) &= MLP_{\mathbf{rgb}}(\sum_c (z_{c,t}^{\mathbf{x}}; \gamma_{\mathbf{d}}(\mathbf{d}))/C), \end{aligned} \tag{3}$$

where MLP_σ and $MLP_{\mathbf{rgb}}$ consist of four and two linear layers respectively, and $\gamma_{\mathbf{d}} : \mathbb{R}^{3 \to 6 \times l}$ is a positional encoding of viewing direction $\mathbf{d} \in \mathbb{R}^3$ as in [26] with $2 \times l$ different basis functions. More details on the network architecture can be found in the supplementary material.

3.4 Volume Rendering

The predicted color of a pixel $p \in \mathbb{R}^2$ for a target viewpoint q in the focal plane of the camera and center $\mathbf{r}_0 \in \mathbb{R}^3$ is obtained by marching rays into the scene using the camera-to-world projection matrix, $\mathbf{P}^{-1} = [\mathbf{R}_q|\mathbf{t}_q]^{-1}\mathbf{K}_q^{-1}$ with the direction of the rays given by $\mathbf{d} = \frac{\mathbf{P}^{-1}p - \mathbf{r}_0}{\|\mathbf{P}^{-1}p - \mathbf{r}_0\|}$.

We then accumulate the radiance and opacity along the ray $\mathbf{r}(z) = \mathbf{r}_0 + z\mathbf{d}$ for $z \in [z_{\text{near}}, z_{\text{far}}]$ as defined in NeRF [26] as follows:

$$\mathbf{I}_q(p) = \int_{z_{\text{near}}}^{z_{\text{far}}} \mathbf{T}(z)\sigma(\mathbf{r}(z))\mathbf{c}(\mathbf{r}(z), \mathbf{d})dz, \quad where \quad \mathbf{T}(z) = \exp\left(-\int_{z_{\text{near}}}^{z} \sigma(\mathbf{r}(s))ds\right) \tag{4}$$

In practice, we uniformly sample a set of 64 points $z \sim [z_{near}, z_{far}]$. We set $\mathbf{X} = \mathbf{r}(z)$ and use the quadrature rule to approximate the integral. We compute the 3D bounding box of the SMPL parameters at time t and derive the bounds for ray sampling z_{near}, z_{far}.

3.5 Loss Function

For ground truth target image $\mathbf{I}_{q,t}$, we train both the radiance field and feature extraction network using a simple photo-metric reconstruction loss $\mathcal{L} = \|\hat{\mathbf{I}}_{q,t} - \mathbf{I}_{q,t}\|_2$.

4 Experiments

We present novel view synthesis and 3d reconstruction results of human performances in different generalization scenarios. We compare our method against the current best view-synthesis methods from two classes: body model-based, per-scene optimization methods (Sec. 4.1) and generalizable NeRF methods (Sec. 4.2). We experiment on ZJU-MoCap [31] and AIST datasets [43, 16]. For training and testing of our model as well as the baselines, we remove the background using the foreground mask obtained by an off-the-shelf human parser [13]. Parameters of the human body model (SMPL) [24] is estimated using [7, 31, 9]. Unless otherwise specified, we sample two memory frames $\{t - 20, t + 20\}$ at time t (total three timesteps) and take three canonical input views in all experiments. The details of the dataset splits, training process, additional results and video results are provided in the supplementary material.

4.1 Comparison with body model-based, per-scene optimization methods.

Baselines. For body model-based methods, we compare with the state-of-the-art Neural Body (NB) [31] that combines SMPL and NeRF in a per-scene optimization setting. Neural Textures (NT) [42] renders a coarse mesh with latent texture maps and uses a 2D CNN to render target images. We use the SMPL mesh as the input mesh. NHR [47] extracts 3D features from input point clouds and renders them into 2D images. Since dense point clouds are difficult to obtain from sparse camera views, we take SMPL vertices as input point clouds. These methods have reported that their learnt per-model representations can adapt to new poses of the same performer, *i.e.*, novel pose synthesis.

Setup. We experiment with ZJU-MoCap dataset [31] which provides performance captures of 10 human subjects, captured from 23 synchronized cameras. Each video contains complicated motions such as kicking and Taichi and a length between 1000 to 2000 frames. We consider three different comparison settings as detailed below. We first split the dataset into two parts: source and target videos. In all comparisons, the first 300 frames of either source or target videos are used during

Method	PSNR	SSIM
Trained on **source** models		
NB	23.79	0.887
NHR	22.31	0.871
NT	22.28	0.872
Trained on **source** models		
Ours	**26.94**	**0.929**

a . Test results on **source** models' **unseen** poses

Method	PSNR	SSIM
Trained on **target** models		
NB	22.88	0.880
NHR	22.03	0.875
NT	21.92	0.873
Trained on **source** models		
Ours	**24.75**	**0.906**

b . Test results on **target** models' **unseen** poses

Method	PSNR	SSIM
Trained on **source** models		
NB	28.51	**0.947**
NHR	23.95	0.897
NT	23.86	0.896
Trained on **source** models		
Ours	**28.73**	0.936

c . Test results on **source** models' **seen** poses

Table 1: **Comparison with other body model-based, per-scene optimization methods.**

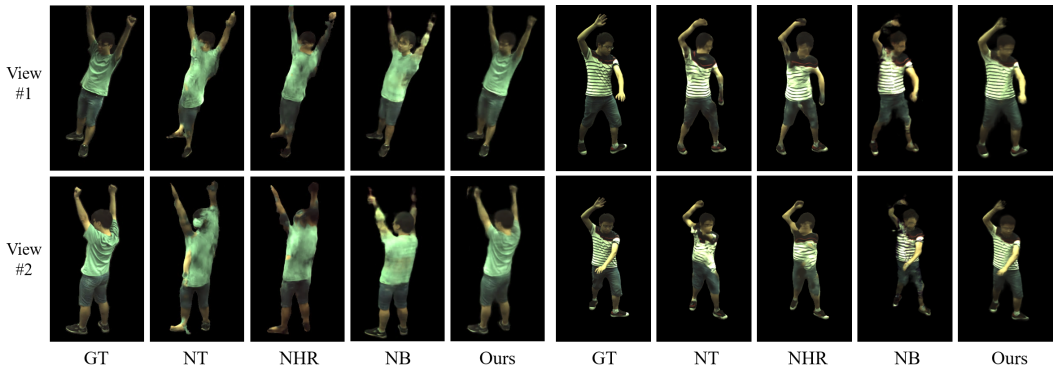

Figure 2: **Pose generalization – comparison with other body model-based, per-person optimization methods.** Results of NT: Neural textures [42], NHR: Neural human rendering [47], NB: Neural body [31] and ours. Novel view synthesis on ZJU-MoCap. Tested on **source** models' **unseen** poses (All methods are trained on **source** models; Competing methods are trained in a per-person manner.)

training, and the remaining next frames (unseen poses) are held out for testing. Note that the baseline methods are always trained in a per-model manner. To validate whether the training is reproducible, we experiment with 5 independent runs with random train/test splits and observe a variance of 0.15 PSNR, showing that the results are quite robust. In each independent run, we used seven models for training and the other three for testing.

Results. We present three different comparison settings to validate our method. First, we evaluate **1) Pose generalization** (Table. 1a and Fig. 2). For all methods, we train on source models, and test on the same source models' unseen poses. Our method significantly outperforms all the baselines and the state-of-the-art Neural Body [31] by **+3.15** PSNR and +0.042 SSIM scores. We also present a very challenging setting: **2) Identity generalization** (Table. 1b). Our method is trained on source models, while other baselines are trained on target models. Then, all methods are tested on the target models' unseen poses. Note that this comparison is *disadvantageous to ours* since the competing methods have *seen* the testing models as they must be trained separately per human model (no identity generalization for baselines). Surprisingly, our unseen-model generalization outperforms all per-scene optimized baselines by a health margin of +1.87 PSNR and +0.026 SSIM scores. The comparison results 1 and 2 indicate that our improvements are not merely from the use of body model prior (SMPL), but that our proposed architecture with the temporal as well as the multi-view transformer can generalize well onto the novel identities and poses, and can produce photo-realistic results. Finally, we show the **3) performance on seen model with seen pose** (Table. 1c), where all the methods are trained on the source models and tested on the seen trained poses. Our method shows comparable results with the state-of-the-art per-scene optimization method [31].

4.2 Comparison with generalizable NeRF methods.

Baselines. Among the recent generalizable NeRF methods [34, 50, 45], we compare with Pixel-NeRF [50] and PVA [34] which focus on very sparse (up to 3 or 4) input views. we reimplement [34] since it is not open-sourced.

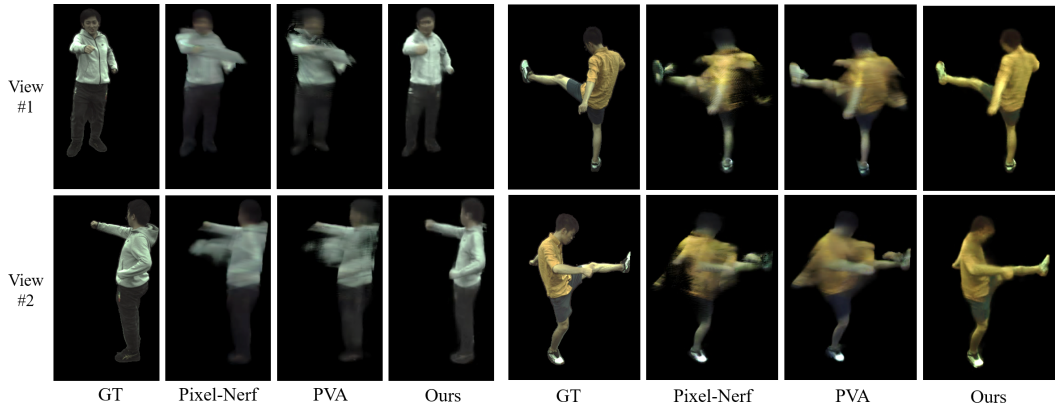

Figure 3: **Identity-and-pose generalization – comparison with generalizable NeRF methods.** Results of Pixel-Nerf [50], PVA: Pixel-aligned volumetric avatar [34] and ours. Novel view synthesis on ZJU-MoCap. Tested on **target** models' **unseen** poses (All methods are trained on **source** models.)

Method	PSNR	SSIM
Pixel-NeRF	23.17	0.8693
PVA	23.15	0.8663
Ours	**24.75**	**0.9058**

a . Generalization results on ZJU-MoCap.

Method	PSNR	SSIM
Pixel-NeRF	18.06	0.7304
PVA	17.82	0.7211
Ours	**19.03**	**0.8390**

b . Generalization results on AIST.

Table 2: **Comparison with generalizable NeRF methods.**

Setup. In addition to ZJU-MoCap (details are in Sec. 4.1), we experiment on larger AIST dataset [43, 16] to further evaluate different methods' generalization abilities. AIST dataset provides dance videos of 30 human subjects captured from 9 cameras. It contains highly diverse motions, slow to fast, simple to complicated. We split the dataset into 20 and 10 subjects for training and testing respectively, where the testing dataset contains novel models and novel poses.

Novel view synthesis results. Table. 2 shows the comparison. For all datasets and all metrics, our method consistently outperforms the baselines by healthy margins of +1.6 PSNR and +0.037 SSIM scores. Fig. 3 and Fig. 5 present the same tendency in visualizations. Pixel-NeRF and PVA aggregate multi-view observations via average pooling without explicitly considering the correlation between the views. In contrast, our temporal and multi-view transformers learn to model the correlation between input views and integrate different observations to help the NeRF module to produce more accurate results. Another advantage of our method is that the used body model prior provides a robust geometric cue to handle the self-occlusion of human subjects.

3D reconstruction results. We also evaluate 3D reconstruction of generalizable NeRF methods and our method on ZJU-MoCap (Fig. 4) and AIST datasets (Fig. 5) given three input views. The visualization shows that our 3D reconstruction aligns well with the input image, and is more reliable than even the per-person method [31] (*e.g.*, the shape of upper cloth in Fig. 4).

Overall, these results indicate that as human models are complex and occlusion-heavy, more sophisticated designs than simple image-conditioning are required to learn robust and accurate 3D human representations.

4.3 Ablation studies

Table. 3 shows the ablation study on ZJU-MoCap on unseen identities and unseen poses, using three time-steps and three camera views as input. Note that all the items without either temporal or multi-view transformer module use simple average pooling instead, to fuse temporal or multi-view observations respectively.

Complementariness of skeletal and pixel-aligned query features. 'Sk' uses only time-augmented skeletal features (Sec. 3.2) without time-specific pixel-aligned features, while 'Px' uses only the time-specific pixel-aligned features, on the contrary. Both 'only' models show the largest

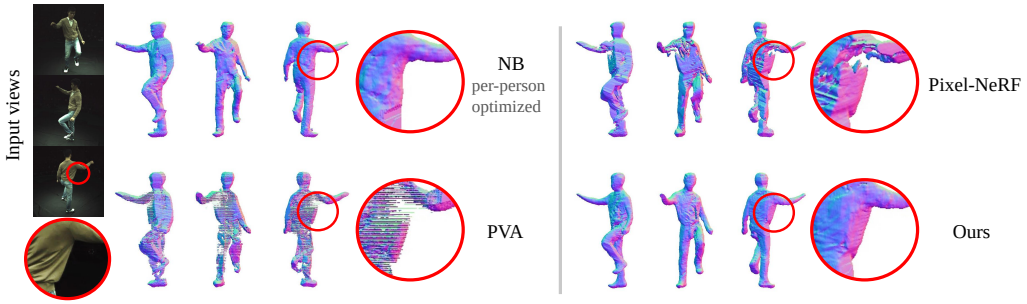

Figure 4: **3D reconstruction on ZJU-MoCap.** Tested on unseen model's unseen pose except Neural Body (per-person optimized). NB: Neural Body [31], PVA: Pixel volumetric avatar [34], Pixel-NeRF [50] and ours.

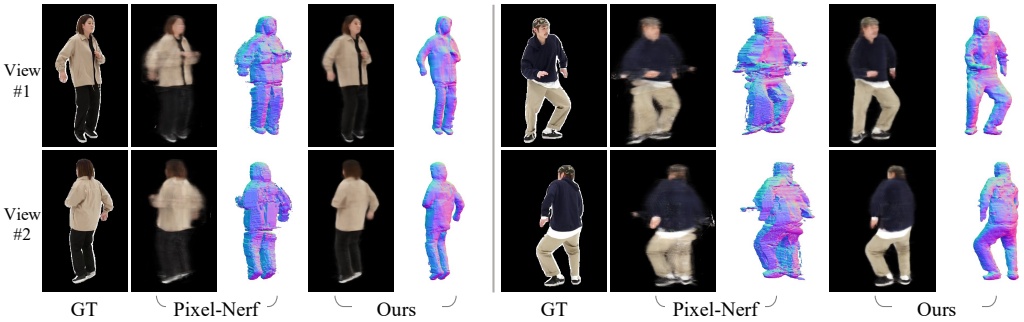

Figure 5: **Generalization results on AIST.** Novel view synthesis and 3D reconstruction results on unseen models' unseen poses.

drops compared to our full model, and 'Sk + Px' model improves them by +1.2 PSNR and +0.9 PSNR respectively. This validates the complementariness between the skeletal and pixel-aligned features in that one is time-augmented but involves slight geometric deviations, while another is time-specific and represents exact query location, as discussed in Sec. 3.3.

Impact of temporal and multi-view transformers. 'Sk + Px' uses no transformers so far, falling behind our full model by -1.3 PSNR score. Then 'Sk + Px + T' adds the temporal transformer and improves +0.7 PSNR score, showing its effectiveness in aggregating information over video frames. 'Sk + Px + MV' uses the multi-view transformer module and shows the largest gain of +1.0 PSNR, indicating the efficacy of learnt cross-attention between the skeletal features and pixel-aligned features, as well as the importance of learnt inter-view correlations. Our full model 'Sk + Px + T + MV' shows the best use of all the proposed components and achieves 24.75 PSNR and 0.9058 SSIM.

Impact of temporal memory length. The length of the temporal memory describes the maximum number of skeletal features from all the saved time steps that each SMPL vertex (and temporal transformer) has access to. We experiment with varying lengths of the memory, fixing the number of views as one. If the timestep length is set to one (no memory frame is used), the method practically degenerates to a frame-by-frame method. Table. 4-left shows that incorporating longer-term information consistently improves the results.

Impact of number of camera views. Table. 4-right shows our model's testing results with the different number of input views, fixing the number of timesteps as one. Our method degrades reasonably as the input views become very sparse (as few as one).

4.4 Running time for inference

The Neural Human Performer takes $1.373s$ to render one 512×512 target image, given 3 input views and 3 timesteps on Intel i7 3.7GHz CPU with one GTX 1080 Ti GPU, which is comparable to other per-scene optimization methods that cannot generalize. Specifically, it takes $0.019s$ for the construction of time-augmented skeletal features (Sec. 3.2), $1.343s$ for Sparse convolution-based

Variant	Skeletal	Pixel-aligned	T-transformer	MV-transformer	PSNR	SSIM
Sk	✓				22.31	0.8865
Px		✓			22.58	0.8780
Sk + Px	✓	✓			23.47	0.8906
Sk + Px + T	✓	✓	✓		24.21	0.9016
Sk + Px + MV	✓	✓		✓	24.44	0.9034
Sk + Px + T + MV	✓	✓	✓	✓	**24.75**	**0.9058**

Table 3: **Ablation study.** Generalization results on ZJU-MoCap. Sk: sleletal features, Px: pixel-aligned features, T: temporal transformer, MV: multi-view transformer.

# Timesteps	# Views	PSNR	SSIM
1	1	20.13	0.835
2		20.84	0.857
3		21.76	0.870
5		22.04	0.891

# Timesteps	# Views	PSNR	SSIM
1	1	20.13	0.835
	2	21.82	0.871
	3	23.33	0.906
	4	23.51	0.913

Table 4: **Impact of different length of memory (left) and number of camera views (right).** Generalization results on ZJU-MoCap.

trilinear sampling (98% of the runtime), $0.008s$ for cross-attention and multi-view aggregation (Sec. 3.3) and $0.001s$ for final color and density prediction (Sec. 3.4).

5 Limitations

We tackle some shortcomings of existing body model-based and generalizable NeRF methods with a focus on generalizable human performance rendering, but there are challenges yet to be explored. 1) The skeleton-based body model [24] cannot generalize to generic humans with loose clothing and garments with large deformations. Recovering or adapting on the fly to such 3D shapes of dressed humans is an interesting research question. 2) While we demonstrate our method on real data from the ZJU-MoCap and AIST datasets, our training currently focuses only on the foreground-masked region, and the training corpus does not capture the wide variation of backgrounds and illumination of the in-the-wild scenes, which could be tackled in the future work. 3) Selection of the memory frames is left as a hyper-parameter (we use an interval of 20 frames), and the benefit of time information will decrease when the human performance is very slow or even stationary. The learnable selection of important memory frames is an important research direction to explore.

6 Societal impact

We discuss the potential societal impact of our work. The positive side is that the human performance synthesis is the key component of realizing telepresence, which has become more important especially in this pandemic era. In the future, people physically apart can feel like they are in the same space and feel connected with a few inexpensive webcams and AR/VR headsets thanks to the development of the telepresence. The negative aspect is that it can help organizations easier to identify people by reconstructing them from an only small number of surveillance cameras. We strongly hope that our research could be used in positive directions.

7 Conclusion

We present Neural Human Performer, a generalizable radiance field network based on a parametric body model that can synthesize free-viewpoint videos for arbitrary human performers from sparse camera views. Leveraging the trackable visual features from the input body motion prior, we propose a combination of a temporal Transformer and a multi-view Transformer that integrates multi-time and multi-view observations in a feed-forward manner. Our method can produce photo-realistic view synthesis of new unseen poses and identities at test time. In various generalization settings on ZJU-MoCap and AIST datasets, our method achieves state-of-the-art performance outperforming the body model-based per-scene optimization methods as well as the generalizable NeRF methods.

References

[1] Kara-Ali Aliev, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. *arXiv preprint arXiv:1906.08240*, 2(3):4, 2019.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[3] Joel Carranza, Christian Theobalt, Marcus A Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. *ACM transactions on graphics (TOG)*, 22(3):569–577, 2003.

[4] 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):1–13, 2015.

[5] Edilson De Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. In *ACM SIGGRAPH 2008 papers*, pages 1–10. 2008.

[6] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 145–156, 2000.

[7] Junting Dong, Qing Shuai, Yuanqing Zhang, Xian Liu, Xiaowei Zhou, and Hujun Bao. Motion capture from internet videos. In *European Conference on Computer Vision*, pages 210–227. Springer, 2020.

[8] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (TOG)*, 35(4):1–13, 2016.

[9] Qi Fang, Qing Shuai, Junting Dong, Hujun Bao, and Xiaowei Zhou. Reconstructing 3d human pose by watching humans in the mirror. In *CVPR*, 2021.

[10] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2367–2376, 2019.

[11] Juergen Gall, Carsten Stoll, Edilson De Aguiar, Christian Theobalt, Bodo Rosenhahn, and Hans-Peter Seidel. Motion capture using joint skeleton tracking and surface estimation. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1746–1753. IEEE, 2009.

[12] Chen Gao, Yichang Shih, Wei-Sheng Lai, Chia-Kai Liang, and Jia-Bin Huang. Portrait neural radiance fields from a single image. *arXiv preprint arXiv:2012.05903*, 2020.

[13] Ke Gong, Xiaodan Liang, Yicheng Li, Yimin Chen, Ming Yang, and Liang Lin. Instance-level human parsing via part grouping network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 770–785, 2018.

[14] Kaiwen Guo, Peter Lincoln, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Dourgarian, et al. The relightables: Volumetric performance capture of humans with realistic relighting. *ACM Transactions on Graphics (TOG)*, 38(6):1–19, 2019.

[15] Youngjoong Kwon, Stefano Petrangeli, Dahun Kim, Haoliang Wang, Eunbyung Park, Viswanathan Swaminathan, and Henry Fuchs. Rotationally-temporally consistent novel view synthesis of human performance video. In *European Conference on Computer Vision*, pages 387–402. Springer, 2020.

[16] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Learn to dance with aist++: Music conditioned 3d dance generation. *arXiv preprint arXiv:2101.08779*, 2021.

[17] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, and Zhaoyang Lv. Neural 3d video synthesis. *arXiv preprint arXiv:2103.02597*, 2021.

[18] Yiyi Liao, Katja Schwarz, Lars Mescheder, and Andreas Geiger. Towards unsupervised learning of generative models for 3d controllable image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5871–5880, 2020.

[19] Baoyuan Liu, Min Wang, Hassan Foroosh, Marshall Tappen, and Marianna Pensky. Sparse convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 806–814, 2015.

[20] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *arXiv preprint arXiv:2007.11571*, 2020.

[21] Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Hyeongwoo Kim, Florian Bernard, Marc Habermann, Wenping Wang, and Christian Theobalt. Neural rendering and reenactment of human actor videos. *ACM Transactions on Graphics (TOG)*, 38(5):1–14, 2019.

[22] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2019–2028, 2020.

[23] 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.

[24] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.

[25] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.

[26] 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 *European Conference on Computer Vision*, pages 405–421. Springer, 2020.

[27] Ryota Natsume, Shunsuke Saito, Zeng Huang, Weikai Chen, Chongyang Ma, Hao Li, and Shigeo Morishima. Siclope: Silhouette-based clothed people. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4480–4490, 2019.

[28] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020.

[29] Kyle Olszewski, Sergey Tulyakov, Oliver Woodford, Hao Li, and Linjie Luo. Transformable bottleneck networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7648–7657, 2019.

[30] 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.

[31] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *CVPR*, 2021.

[32] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. *arXiv preprint arXiv:2003.04618*, 2, 2020.

[33] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. *arXiv preprint arXiv:2011.13961*, 2020.

[34] Amit Raj, Michael Zollhoefer, Tomas Simon, Jason Saragih, Shunsuke Saito, James Hays, and Stephen Lombardi. Pva: Pixel-aligned volumetric avatars. *arXiv preprint arXiv:2101.02697*, 2021.

[35] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2304–2314, 2019.

[36] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 84–93, 2020.

[37] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020.

[38] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2437–2446, 2019.

[39] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *arXiv preprint arXiv:1906.01618*, 2019.

[40] Carsten Stoll, Juergen Gall, Edilson De Aguiar, Sebastian Thrun, and Christian Theobalt. Video-based reconstruction of animatable human characters. *ACM Transactions on Graphics (TOG)*, 29(6):1–10, 2010.

[41] Zhuo Su, Lan Xu, Zerong Zheng, Tao Yu, Yebin Liu, et al. Robustfusion: Human volumetric capture with data-driven visual cues using a rgbd camera. Springer, 2020.

[42] 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.

[43] Shuhei Tsuchida, Satoru Fukayama, Masahiro Hamasaki, and Masataka Goto. Aist dance video database: Multi-genre, multi-dancer, and multi-camera database for dance information processing. In *ISMIR*, pages 501–510, 2019.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[45] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. *arXiv preprint arXiv:2102.13090*, 2021.

[46] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[47] 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*, pages 1682–1691, 2020.

[48] 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.

[49] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

[50] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. *arXiv preprint arXiv:2012.02190*, 2020.

[51] Wentao Yuan, Zhaoyang Lv, Tanner Schmidt, and Steven Lovegrove. Star: Self-supervised tracking and reconstruction of rigid objects in motion with neural rendering. *arXiv preprint arXiv:2101.01602*, 2020.

[52] Zerong Zheng, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. Deephuman: 3d human reconstruction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7739–7749, 2019.

[53] Noah Snavely Oliver Wang Zhengqi Li, Simon Niklaus. Neural scene flow fields for space-time view synthesis of dynamic scenes. *CVPR*, 2021.

[54] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See the abstract and Section 1
   (b) Did you describe the limitations of your work? [Yes] See Section 5
   (c) Did you discuss any potential negative societal impacts of your work? See Section 6
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See the supplementary material. Code will be released upon publication.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See the supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See the supplementary material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] See the supplementary material.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] The datasets used are publicly available dataset for research purpose.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See the supplementary material.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]