# ANR: Articulated Neural Rendering for Virtual Avatars

Amit Raj [1] Julian Tanke[2] James Hays[1]
Minh Vo[3] Carsten Stoll[4] Christoph Lassner[3]

[1]Georgia Tech [2]University of Bonn [3]Facebook Reality Labs [4]Epic Games

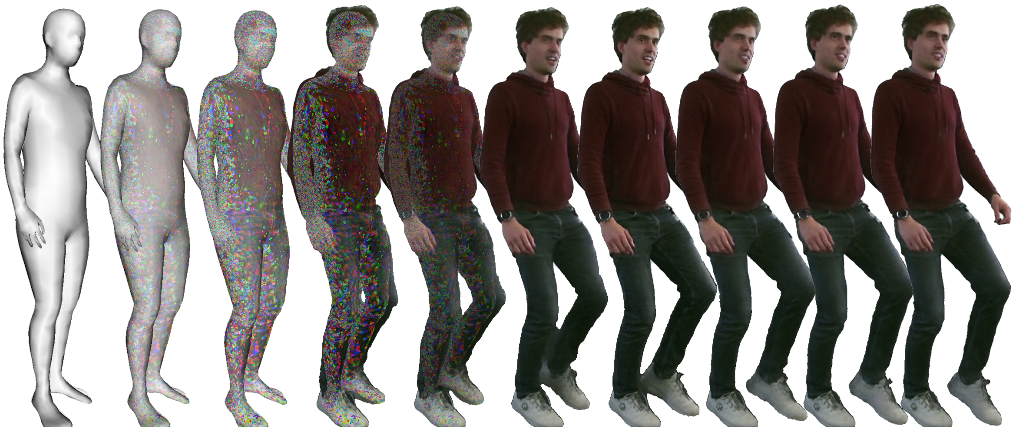


Figure 1: We present Articulated Neural Rendering (ANR), a rendering framework capable of producing highly realistic avatars. Similar to Deferred Neural Rendering (DNR) [38], ANR uses neural networks to convert a latent texture on a coarse mesh (left) into an RGB image (right). Unlike DNR, which is ineffective when the mesh geometry is inaccurate or deforms during motion, ANR explicitly accounts for such geometric misalignment and pose-dependent deformation.

## Abstract

*The combination of traditional rendering with neural networks in Deferred Neural Rendering (DNR) [38] provides a compelling balance between computational complexity and realism of the resulting images. Using skinned meshes for rendering articulating objects is a natural extension for the DNR framework and would open it up to a plethora of applications. However, in this case the neural shading step must account for deformations that are possibly not captured in the mesh, as well as alignment inaccuracies and dynamics—which can confound the DNR pipeline. We present Articulated Neural Rendering (ANR), a novel framework based on DNR which explicitly addresses its limitations for virtual human avatars. We show the superiority of ANR not only with respect to DNR but also with methods specialized for avatar creation and animation. In two user studies, we observe a clear preference for our avatar model and we demonstrate state-of-the-art performance on quantitative evaluation metrics. Perceptually, we observe better temporal stability, level of detail and plausibility. More results are available at our project page:* https://anr-avatars.github.io.

## 1. Introduction

Capturing realistic appearance is one of the important goals of computer vision. Progress in 3D rendering and neural networks has led to approaches with remarkable fidelity [22, 23, 29, 30]. These methods often use expensive and intricate capture setups which prevent easy digitization and transfer of the resulting models [7, 8, 11]. The recent deferred neural rendering paradigm offers an exciting opportunity to work with inaccurate geometry and relatively simple neural shaders while capturing complex scenes with view-dependent effects realistically [1, 27, 38]. In a first step, the geometry is rasterized using a *neural* latent texture which is then translated to an RGB image using a convolutional network. Both, the rendering network as well as the neural texture, are optimized to produce realistic results.

Deferred neural rendering works particularly well for rigid objects. Its pipeline could be extended to deformable objects in a natural way: a *skinned* mesh could be used for capturing the geometry. The rasterized neural texture from the posed mesh could then be translated to an RGB image. While this idea is conceptually simple, the neural network has to learn more complex deformation-dependent effects. Furthermore, the mesh used for rendering is usually not a

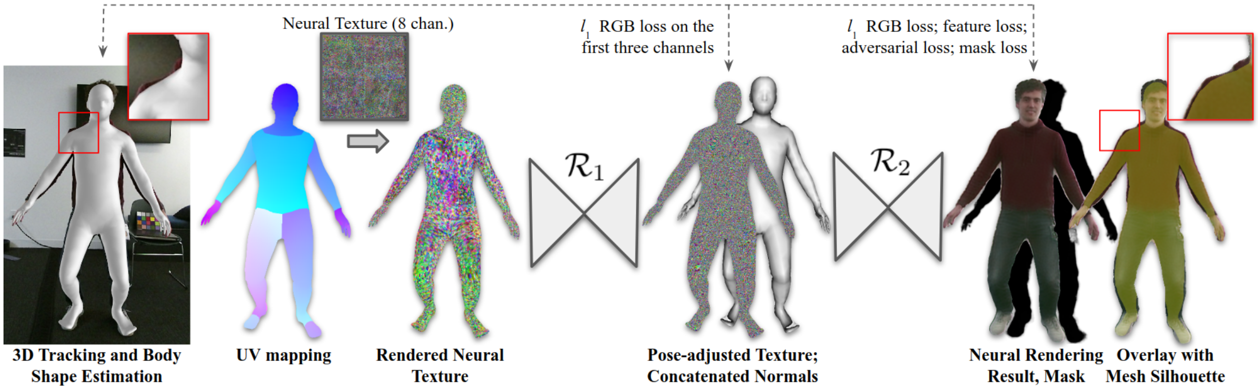

Neural Texture (8 chan.)
l1 RGB loss on the first three channels
l1 RGB loss; feature loss; adversarial loss; mask loss
R1
R2
3D Tracking and Body Shape Estimation
UV mapping
Rendered Neural Texture
Pose-adjusted Texture; Concatenated Normals
Neural Rendering Result, Mask
Overlay with Mesh Silhouette

Figure 2: Schematic overview of the proposed framework. Given a coarse, animated 3D body mesh, ANR produces a detailed avatar. Using rasterized IUV images of the mesh using a weak perspective projection, we render an 8 channel neural texture into image space. A first stage, $\mathcal{R}_1$, transforms the texture into another, refined latent representation, which we combine with the normal information. The second stage $\mathcal{R}_2$ uses this information to create an RGB rendering and a foreground mask. The rendering can extend beyond the coarse mesh, in this case we visualize only slight refinement for painting the shirt.

perfect representation of the real geometry, leading to alignment problems. These problems are currently not taken into account [1, 27, 38], which limits the application of DNR in scenarios with deformable objects.

We present Articulated Neural Rendering (ANR) to account for these problems. ANR systematically rebuilds DNR from the neural shading model architecture to the optimization scheme. We use ANR to tackle one of the most challenging problems for animation: virtual human avatars. Fig. 1 shows an example of an avatar rendered using ANR.

Concretely, ANR employs a simple statistical human body model fitted to a training video to capture the body shape statistics and 3D pose information for each frame [39]. This body model only represents the coarse body geometry without clothing and hair. Consequently, direct use of the DNR pipeline leads to unrealistic and blurry results. We use keyframes from the video to learn the *static* appearance encoded in the neural texture, and use the other frames to learn the *dynamic* pose-conditioned rendering of the appearance. Our keyframes-based training scheme enables the model to converge 5X faster and produces quantitatively better avatars than DNR. We simultaneously train ANR on multiple identities in a single model, leading to decoupling of the neural texture and the shading model. Owing to the consistent surface parameterization of the statistical body model, our model can leverage such semantic correspondences to modify and mix components from multiple neural textures, enabling virtual try-on by changing regions in the neural texture. While our model works solely in 2D, we experimentally validate that it can render near photorealistic and persistent 3D appearance of people with a very small network (161M parameters). In two user studies, we demonstrate that we not only outperform the DNR pipeline, but also several methods dedicated to creating virtual avatars [36, 41]. Perceptually, the presented method is temporally stable and captures fine appearance details.

Our contributions are threefold. First, we present ANR, a novel neural rendering framework, to generate high-quality virtual avatars from coarse 3D shape and arbitrary skeletal motions. Our key is to account for geometric misalignment of the coarse body mesh and pose-dependent deformation. Second, we showcase ANR as the first neural avatar model that can capture and render multiple identities with only one set of network parameters in addition to an identity specific neural texture map. Third, we demonstrate that ANR allows easy appearance editing or mixing of identities. This is novel in the context of neural rendering for avatars.

## 2. Related Work

Among many methods to create and render articulated models, a majority of them follow the classical pipeline of acquiring an accurate 4D geometry reconstruction with detailed textures painted on this geometry. Using machine learning, several recent methods have set out to perform inference mostly in 2D space, only using rough or no 3D guidance. We will discuss several frameworks from both of these schools as well as some hybrid methods, which are closest to our approach.

**Inference in 3D Space:** The Relightables [11] propose a system that captures accurate geometry and texture using a controlled light stage. This allows for relightable rendering of the captured identity in different environments. Lombardi et al. [22] use a multi camera setup to determine the average texture and deformations on a base mesh and use a neural network to generate view specific texture to render high fidelity images from different viewpoints. Using a similar system, Brualla et al. [28] train a network to perform completion and super resolution of the rendered 3D model. In a single-view regime, Alldiek et al. [2, 3] generate avatars by learning to regress accurate geometry and texture using purely synthetic data. Zhi et al. [45] estimate personalized avatar with fine geometry and texture by finetuning on the test video using self-supervised losses. DeepCap [12] captures accurate geometry from monocular video by predicting a parameterized human configuration and deformation model. Our approach also uses monocular video to capture a digital avatar. However, instead of deforming and refining the avatar geometry, we advocate for avatars with high-capacity texture that compensates for such geometric inaccuracies (such as clothing and hairs) for arbitrary body pose and view-point rendering.

**Inference in 2D Space:** Meanwhile, specific architectures are designed for motion re-targeting, novel view synthesis and identity transfer, which primarily use only pixel and pose information [4, 6, 26, 37, 44]. Neverova et al. [31] use DensePose for novel viewpoint synthesis, which is limited by the DensePose body coverage and accuracy. [32] propose a semi-parametric approach which uses a combination of previously captured RGB(D) images and neural rendering to infer novel views in an approach similar to image based rendering. [25, 19] focus on pose conditioned image generation of people, but with lower resolution. Grigorev et al. [10] solve the novel view synthesis problem by formulating it as texture inpainting in DensePose UV space. SwapNet [33] learns to transfer clothing information by disentangling the notion of pose and clothing without being identity specific. Human appearance transfer [43] learns to generate novel views and transfer identities by performing human parsing and 3D shape and pose fitting. We generate 3D textured avatars, enabling all these tasks with no additional guiding signals or changes in a single framework.

**Hybrid Approaches:** The DNR framework [38] uses a mostly rigid mesh and a neural texture to translate the rendering result into an image. We detail this approach in Sec. 3.1 and reformulate it to account for handling fully articulated objects. Textured neural avatars [36] present a framework to learn neural avatars in an end-to-end manner from multiview data. Unlike this work, we leverage the reconstructed geometry instead of noisy DensePose correspondences to generate the UV coordinates for every pixel, enabling us to maintain better texture consistency across viewpoints. Our work is also related to the Liquid Warping Gan [21] which performs appearance transfer and motion retargeting in a single network. However, our framework provides explicit access to the learned texture allowing for fine grained edits of appearance. Additionally, our framework uses a lower number of parameters, and thus can be trained at a higher resolution. Neural rendering and reenactment [20] trains a network to translate from 3D pose to image. However, their framework involves capturing a rigged template mesh for every individual and requiring additional depth and body part information. Recently, implicit representations with impressive geometry reconstruction of clothed humans reconstruction from a single image are becoming popular [15, 35]. We distinguish ourselves from these methods by generating clothing and body deformations in the rendering stage while using a simple parametric body model to fit the body pose and shape.

## 3. Approach

Articulated Neural Rendering (ANR) can generate highly detailed representations of articulated objects. Unlike traditional rendering pipelines which use a high resolution mesh and detailed RGB texture for this purpose, we use a low resolution mesh but a high-*dimensional* neural texture to render its detailed RGB appearance from novel views using a neural network. Fig. 2 shows an outline of the proposed framework. In the following, we first present an overview of DNR [38] before presenting our novel ANR framework and its training scheme.

### 3.1. Preliminary: Deferred Neural Rendering

DNR [38] translates high dimensional neural latent textures on traditional meshes into RGB images with a neural translator network. Concretely, let $\mathcal{T}$ be a high-dimensional neural texture (a tensor of shape $W \times H \times C$) and let $\mathcal{R}$ be the neural rendering model converting a neural image $\mathbf{I}^{\text{uv}}$ to RGB color. DNR optimizes

$$\mathcal{T}^*, \mathcal{R}^* = \underset{\mathcal{T},\mathcal{R}}{\text{argmin}} \sum ||\mathbf{I} - \mathcal{R}(\mathcal{T}, \mathbf{I}^{\text{uv}})|| \tag{1}$$

on all training images $\mathbf{I}$ of the same object. The neural image $\mathbf{I}^{\text{uv}}$ is the result of rasterizing the mesh to image space with the appropriate camera parameters and configuration and texturing it with the neural texture $\mathcal{T}$. The model is fully defined with the optimized texture $\mathcal{T}^*$ and the optimized neural rendering model $\mathcal{R}^*$. DNR uses a U-Net architecture [34] for implementing $\mathcal{R}$ and a standard gradient descent based optimization with the ADAM optimizer [18].

### 3.2. Articulated Neural Rendering

While DNR is conceptually powerful, it requires accurate 3D geometry to learn view dependent appearance information. Such an assumption is difficult to make in practice, especially for articulated, clothed human appearance, whose shape is often represented by a coarse statistical body shape model [24] (see Fig. 3). We address this problem in the rendering pipeline, while retaining the ability to work with a coarse, animated mesh to (1) maintain the high rendering speed and (2) be able to optimize the final appearance generation in the neural network. Consequently, we re-visit the neural rendering component, $\mathcal{R}$.

Our first observation is that $\mathcal{R}$ not only has to paint the texture inside regions of the neural image $\mathbf{I}^{\text{uv}}$ but also beyond its boundaries due to the use of the coarse mesh. The network should also be aware of the extent to which it needs

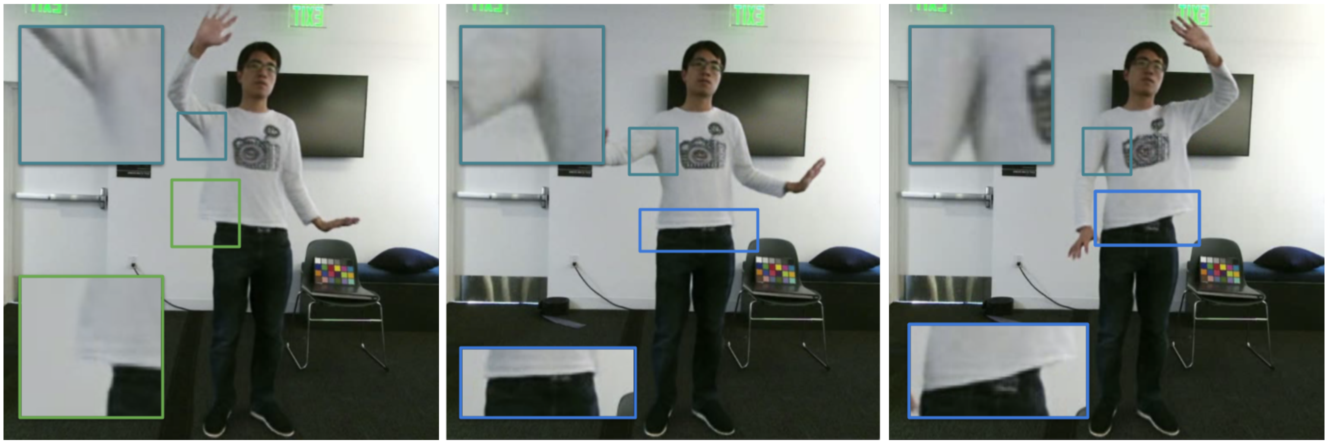


Figure 3: Images generated by ANR on a challenging animation scenario with clothing deformations. We successfully synthesize images of such deformations and regions outside of the body mesh. As highlighted in the figure, region coverage as well as shading are pose dependent. The example frames are unseen poses for this identity and rendering model. Additionally, the model is able to temporally interpolate between this pose and others and adjust the simulated clothing accordingly. We refer to the supplementary video for a demonstration of temporal stability.

to paint outside the boundaries of the rasterized mesh. We address both problems by adding a second prediction: an extra single-channel soft mask $\mathbf{M} \in [0,1]$. The predicted mask is used to blend the generated avatar with the ground-truth background image for training. To prevent the model from predicting a degenerated zero mask (which would minimize the loss to zero), we provide supervision for the mask from an automatic image matting method [9]. Note that while training on the pre-segmented image is another option, this approach is sensitive to erroneous segmentation which prevents the generated images to grow beyond their input coarse body boundaries. Comparing the blended image with the ground truth image allows gradients to flow to the mask which in turn makes it potentially better than the supervised input mask.

While this addresses the immediate problem of generating content outside of the true geometry silhouette, it leaves geometric details and pose-dependent rendering untouched. We notice that naively increasing the capacity of the U-Net does not improve generation quality (see Tab. 2). Furthermore, we observe that the model cannot consistently render local geometry—a problem that increasingly emerges in the articulated setting when geometry is animated. We address problems with the geometry details and the pose-dependent effects at the same time by splitting the neural rendering network in two stages: $\mathcal{R}_1$ and $\mathcal{R}_2$. Both components are shallow U-Nets and produce renderings at original image resolution. We can inject the normal information into the rendering process by concatenating the rendered normal image and output of $\mathcal{R}_1$ to the input of $\mathcal{R}_2$. We enforce an additional RGB loss on the first three output channels of $\mathcal{R}_1$ to aid in convergence. The ANR model is defined as

$$\hat{\mathbf{M}}, \hat{\mathbf{I}}, \hat{\mathbf{J}} = \mathcal{R}_2(\mathcal{R}_1(\mathcal{T}, \mathbf{I}^{\mathrm{uv}}), \mathbf{I}^{\mathrm{norm}}), \tag{2}$$

where $\hat{\mathbf{J}}$ are the first three channels from the result of $\mathcal{R}_1$ and $I^{norm}$ is the rasterized normal image. This model has the necessary capacity and the necessary outputs for handling the articulated neural rendering problem.

### 3.3. Loss Functions and Regularization Scheme

With the higher requirements for stability and level of detail and deformations in the articulated setting, we find that using a simple $\ell_1$ loss is insufficient (see Tab. 2). Furthermore, we observe that it deteriorates performance as the training progresses: once the model learns to reproduce the rough appearance, inaccuracies in tracking and alignment of the mesh have an increasingly negative impact (see Fig. 5). We use adversarial learning and feature loss computation to guide the model to generate realistic and accurate appearance without having to rely on accurate registration. Our loss function is a weighted sum of the photometric loss $\mathcal{L}_p$, feature loss $\mathcal{L}_{feat}$, mask loss $\mathcal{L}_{mask}$, adversarial loss $\mathcal{L}_{adv}$, and total variation loss $\mathcal{L}_{tv_i}$. Note that while the rasterization is non-differentiable, ANR is fully differentiable given the precomputed rasterized UV lookup to paint $I^{\mathrm{uv}}$ from the neural texture $\mathcal{T}$.

**Pixel Loss:** We enforce an $\ell_1$ loss between the generated RGB and ground truth images as

$$\mathcal{L}_p(\hat{\mathbf{M}}, \hat{\mathbf{I}}, \hat{\mathbf{J}}; \mathbf{M}, \mathbf{I}) = \hat{\mathbf{M}}||\hat{\mathbf{J}} - \mathbf{I}|| + \hat{\mathbf{M}}||\hat{\mathbf{I}} - \mathbf{I}||, \tag{3}$$

where $\hat{\mathbf{J}}$ are the first three channels of the result from $\mathcal{R}_1$.

**Mask Loss:** Similarly, we use a Binary Cross Entropy loss for the mask

$$\mathcal{L}_{mask}(\hat{\mathbf{M}}; \mathbf{M}) = \mathrm{BCE}(\hat{\mathbf{M}}, \mathbf{M}) \tag{4}$$

For all following loss definitions, we introduce the shorthand $\hat{\mathbf{I}}'$ for the blended version of the generated output with the scene background $\mathbf{B}$ given the predicted mask: $\hat{\mathbf{I}}' = \hat{\mathbf{M}}\hat{\mathbf{I}} + (1 - \hat{\mathbf{M}})\mathbf{B}$.

**Feature Loss:** To increase sharpness in the rendered outputs, we enforce a feature loss [17]:

$$\mathcal{L}_{feat}(\hat{\mathbf{I}}, \hat{\mathbf{M}}; \mathbf{I}) = \sum_j w_j ||\phi_j(\hat{\mathbf{I}}') - \phi_j(\mathbf{I})||, \tag{5}$$

where $\phi_j$ are features from the $j$-th layer of a pretrained feature extractor and $w_j$ is the weight associated with the $j$-th feature loss term.

**TV Loss:** Since the texture is optimized over multiple frames, slight misalignments can cause the learned texture to have certain high frequency artifacts, especially for small regions such as face and hands. To encourage smooth generated images, we enforce a total-variation loss on both, the mask and the generated image.

$$\mathcal{L}_{tv}(\hat{\mathbf{I}}, \hat{\mathbf{M}}) = \beta_I TV(\hat{\mathbf{I}}') + \beta_m TV(\hat{\mathbf{M}}), \tag{6}$$

where $\beta_I$ and $\beta_m$ are weights associated with Image and mask TV loss respectively (see supp. mat. for a detailed definition of this loss).

**Adversarial Loss:** Adversarial training [14] is well-suited for enforcing realism of the results and encourage the coarse body mask to extend to the true geometry silhouette. To encourage a high level of detail in the results, we use a multi-scale discriminator $\mathcal{D}$ [42] and express the loss as

$$\mathcal{L}_{adv}(\hat{\mathbf{I}}) = \mathcal{D}(\hat{\mathbf{I}}', 1). \tag{7}$$

**Total loss:** The loss used to train $\mathcal{R}$ is then given as

$$\mathcal{L}_{total} = \sum_{i \in \mathfrak{L}} \lambda_i \mathcal{L}_i \tag{8}$$

where $\mathfrak{L} = \{p, feat, mask, adv, tv\}$ is the set of all losses.

### 3.4. Optimization

Despite the extended set of losses and weight balancing, we find that for clothing with large surface deformations, the model starts averaging fine textures in areas of high deformation. To mitigate this problem, we propose a *split optimization* strategy. Specifically, we use a small set of keyframes $\{\mathcal{K}_i\}_{i=1}^n$, capturing *static* salient appearances in the video, to learn the neural texture $\mathcal{T}$ and use the other frames to *dynamically* blend between the appearances in the keyframes in the neural renderer $\mathcal{R}$.

We select keyframes by greedily adding a small number of frames in the video sequence such that their cumulative silhouette coverage is maximized. This ensures that the entire pose-space is adequately covered to capture texture details at all locations on the body. Using a smaller number of frames (less than 10% of training frames) reduces the texture averaging. During training, we alternate between training the identity specific neural texture from the keyframes and the rendering network from the remaining frames. Empirically, we observe that this optimization scheme helps the translator network converge up to 5X faster and produces quantitatively better avatars (see Tab. 1). Overall, our optimization alternates between the following two objectives

$$\underset{\mathcal{R}}{\operatorname{argmin}} \sum \mathcal{L}_{total}(\mathbf{I}, \mathbf{M}, \mathcal{R}(\mathcal{T}, \mathbf{I}^{\mathrm{uv}}, \mathbf{I}^{\mathrm{norm}})), \tag{9}$$

$$\underset{\mathcal{T},\mathcal{R}}{\operatorname{argmin}} \sum_{k \in \mathcal{K}} \mathcal{L}_{total}(\mathbf{I}_k, \mathbf{M}_k, \mathcal{R}(\mathcal{T}, \mathbf{I}_k^{\mathrm{uv}}, \mathbf{I}_k^{\mathrm{norm}})). \tag{10}$$

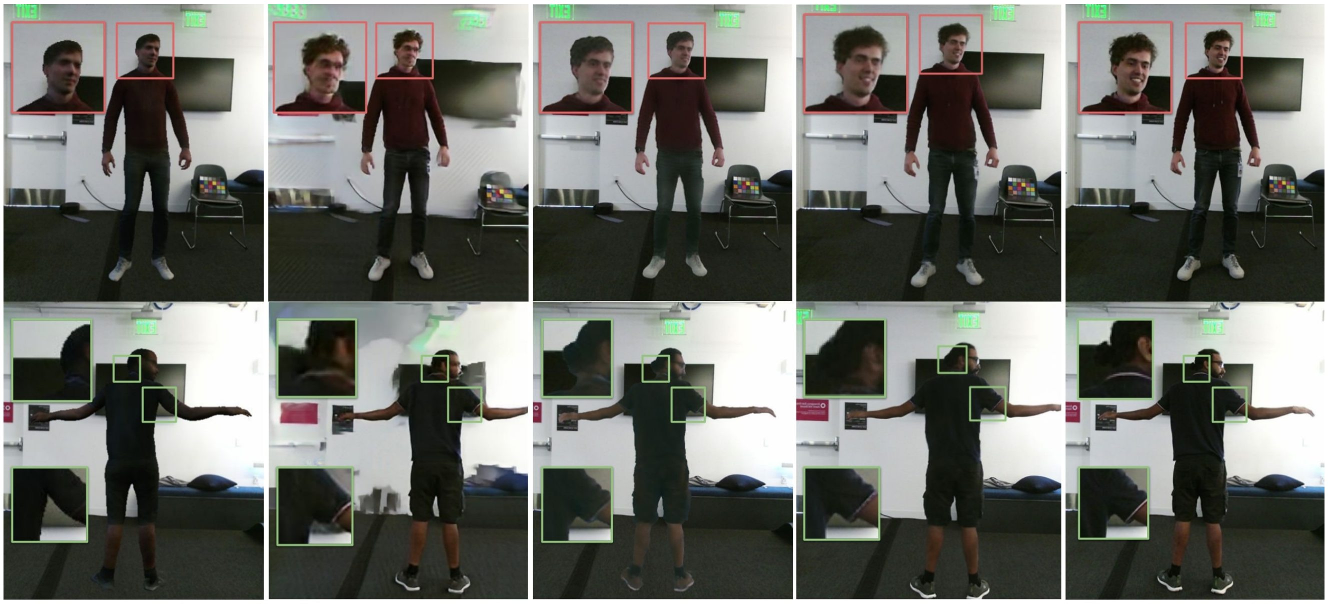
Note that while Eq. 9 is optimized for all the images, Eq. 10 is applied only to the keyframes to mitigate the geometric misalignment of the coarse body mesh.

**Multi-instance Training.** We further extend the training scheme beyond a single capture instance. Since we use the same statistical mesh regardless of identity, allowing us to capture identity information only in the neural texture, our framework can naturally train on multiple identities simultaneously in a single network. During optimization, we select an identity for every step at random and use an identity-specific neural texture $\mathcal{T}_i$ for the respective identity for the update step. The multi-instance training offers the additional benefit that the neural rendering component $\mathcal{R}$ generalizes beyond a single identity and can be used to render new identities by only using a novel neural texture $\mathcal{T}$.

**Regularization.** To improve generalization, we additionally employ two training regularization schemes. First, we use the same initialization of $\mathcal{T}$ for all identities by uniformly sampling in $[-1, 1]$. Since $\mathcal{R}$ has much larger capacity than $\mathcal{T}$, this strategy prevents the model from using the distinct noise patterns in each randomly initialized $\mathcal{T}$ to memorize the identity and thus encourages decoupling of $\mathcal{T}$ and $\mathcal{R}$. Second, we perturb the input sampling grid with a uniform samples from $[-0.02, 0.02]$ and clamp the resulting grid back to $[-1, 1]$. This form of data augmentation prevents the network from relying strictly on the spatial extent of the sampling grid as the ground truth human silhouette can exist outside the rasterized coarse body model.

## 4. Experiments

We use the ANR pipeline to build a realistic virtual human avatar pipeline: we assume a setting where a user performs a recording of themselves with accurate tracking, in which his/her full appearance is visible, to create an avatar model. To ease the tracking, we capture 6 videos using a Kinect V2 where the depth data is only used for tracking. Each video is about 3∼5 min long. We obtain a coarse mesh in real-time by solving an inverse kinematic problem to fit the posed body shape to the 3D point cloud similar to [40], making use of additional detected body keypoints [5]. Our dataset is harder than the previously released iPer dataset



TNA V2V DNR ANR (Ours) Ground Truth

Figure 4: Comparisons for novel pose and view synthesis with Textured Neural Avatar (TNA) [36], vid2vid (V2V) [41] and Deferred Neural Renderer (DNR) [38]. Our method (ANR) preserves the facial details better compared to competing methods. Additionally, our method is able to capture view dependent structures like hairline and clothing overhang more accurately and leads to more realistic and believable shading.

Table 1: Results of novel pose synthesis of avatars learned using different methods. Our model is trained on all identities simultaneously.

| | SSIM ↑ | FLIP ↓ | LPIPS ↓ | rIPFIP ↑ | mFID ↓ | User Study |
|---|---|---|---|---|---|---|
| V2V | 0.9252 | 0.0363 | 0.0703 | - | 140 | 8% |
| TNA | 0.9366 | 0.0323 | 0.1198 | -2.6% | 150 | 3% |
| DNR | 0.9398 | 0.0342 | 0.0918 | 7.7% | 92 | 9% |
| ANR | **0.9738** | **0.0289** | **0.0508** | **18.6%** | 74 | **81.6%** |

[37] as our actors are not centered and are free to move anywhere in the frame. As parametric body model, we use a blendshape-based, SMPL-like [24] human model to provide the coarse mesh structure. The model is coarse and has only 1831 vertices and 3658 faces; the skeletal rig has 74 joints.

### 4.1. Implementation Details

We use a variant of Pix2Pix [16, 42] for both $\mathcal{R}_1$ and $\mathcal{R}_2$ and train the model on $1024 \times 1024$ image resolution. The images are normalized to the range [-1,1]. Each identity is encoded in a $256 \times 256 \times 8$ neural texture. For each recorded sequence, we use the first 1500 frames to train $\mathcal{R}$ and about 150 key frames to train $\mathcal{T}$. The remaining images are used as test set. We augment the data with random cropping and random rescaling by a factor $f \sim [0.5, 1.25]$.

### 4.2. Evaluation

**Baseline and Metric**: We include a comparison with two baselines: Textured Neural Avatar (TNA) [36] and vid2vid (V2V) [41]. These methods are fundamentally different and span the space of 2D (V2V) and 3D (TNA) inference approaches, whereas we aim to find a middle ground. We also present comparisons to baseline DNR [38], trained with additional feature losses for a fairer comparison. Fig. 4 shows these comparisons. Evidently, ANR preserves the facial details better compared to competing methods. Additionally, it is able to capture view dependent structures like hairline and clothing overhang more accurately and leads to more realistic and believable shading. We also quantify these renderings using the standard SSIM, LPIPS, FLIP *supervised metrics* on held out test set frames, and the mFID *unsupervised metric* on human-figure-only avatars in novel poses. Tab. 1 shows these comparisons. Our model outperforms the competing approaches on these benchmarks by a notable margin on all metrics.

Table 2: Loss and model ablation study for the ANR model. The model ablations marked with (-so) are run without the suggested split optimization strategy.

| | SSIM ↑ | LPIPS ↓ | FLIP ↓ |
|---|---|---|---|
| | *Loss ablation* | | |
| Pixel only | 0.968 | 0.086 | 0.029 |
| Pixel+feat | 0.966 | 0.065 | 0.033 |
| Pixel+feat+TV | 0.963 | 0.064 | 0.032 |
| | *Model ablations* | | |
| 1 stage(-so) | 0.962 | 0.070 | 0.036 |
| 1 stage | 0.965 | 0.063 | 0.034 |
| 2 stage(-so) | 0.968 | 0.058 | 0.032 |
| Ours | **0.974** | **0.050** | **0.028** |

**Ablation Study:** To quantify the effectiveness of the pro-

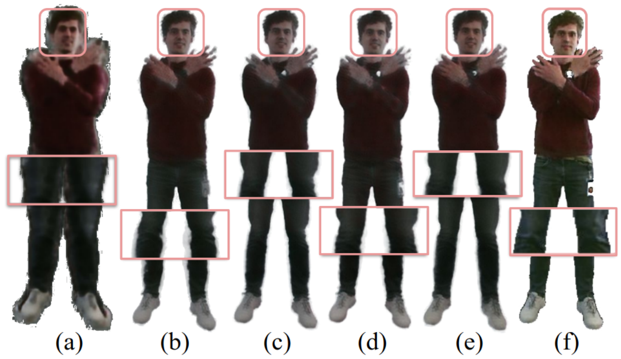


Figure 5: Qualitative ablation study for each term in the loss function and the normal channels on a multi-identity model. The first four columns do not use normal information: **(a)** Only pixel loss; **(b)** Pixel and feature losses; **(c)** Pixel, feature, and mask losses; **(d)** Pixel, feature, mask, and TV losses; **(e)** All losses + normal; **(f)** All loses + normal + split optimization. Notice that the local textures and facial details are better preserved with split optimization.

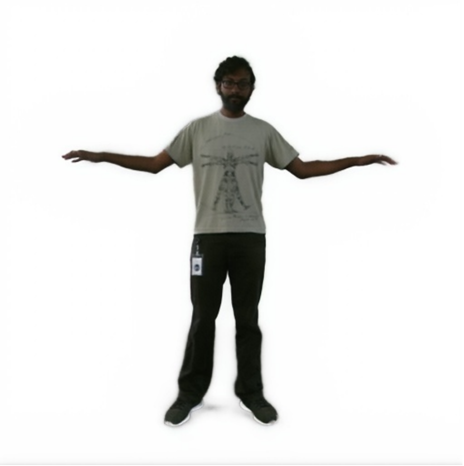


Figure 6: A novel avatar *unseen* during neural renderer training. Only the neural texture is optimized for this identity. **Animated figure is included in the appendix**

posed improvements, we run two ablation studies. Fig. 5 shows the rendered results with different loss terms removed. We notice that without the mask and feature loss (Pixel only), the model produces unrealistically "fat" or "thin" avatars. The feature loss (Pixel+feat) improves the visual quality. Adding the normals improves the level of detail in the reconstruction and aids in reasoning about self occlusion and temporal consistency (shown in the supplementary video), split optimization drastically improves the level of detail. Note the jump in perceptual quality of the rendered face using the split-optimization scheme. Lastly, we show that our two-stage neural render with intermediate normal injection outperforms the single-stage approach with the same capacity, validating our network design choice. This trend is quantitatively confirmed in Tab. 2.

**User Study:** While SSIM, LPIPS, or FLIP are the most widely-used metrics for generative tasks, they are merely proxy metrics which do not pay attention to salient regions (*e.g.*, for faces or patterns on shirts) and do not strictly measure perceptual quality. To demonstrate the efficacy of our method, we conduct a 4-alternative forced-choice perceptual study with 80 participants, where users were given a choice to pick the best avatar out of the results generated from TNA, Vid2Vid, DNR, and our ANR. Each person was presented with 20 stimuli of avatars in novel poses for 5s (see supplementary material for details). ANR was preferred **81.6%** of the time. Furthermore, to test the photorealism of our avatars, we conducted another 2-Alternative forced choice study with 200 participants, where users were presented a real image and an image of our avatar in different poses, and asked to pick the real image. Our model was able to fool users **34%** of the time (50% being random chance) in this test. This shows the realistic rendering performance of our model.

**Model Efficiency:** We calculate the relative improvement in LPIPS of each approach (x) over vid2vid (v2v) scaled by factor of improvement in number of parameters (#p)[13]

$$\mathrm{rIPFIP}(x) = \frac{\mathrm{LPIPS}_{v2v} - \mathrm{LPIPS}_{x}}{\mathrm{LPIPS}_{v2v}} * \frac{log(\#p_{v2v}/\#p_{x})}{log(\#p_{v2v})}$$

Particularly, this metric lies in (-∞,1] and reaches a maximum value for ground truth images. This metrics highlights that we benefit from our *design choices* compared to DNR, and not solely from differences in capacity.

**Generalization:** Fig 6 shows an avatar for which only the neural texture has been optimized on a new subject while keeping the pre-trained neural renderer fixed. We observe details of the T-shirt are also recovered correctly. This example indicates the strong generalization of our neural renderer despite being trained only on a few identities.

## 5. Applications

We use a single ANR model to digitize and render avatars for several applications. Please refer to the supplementary video for more examples.

**Novel View Synthesis:** To render the avatar from novel views, we only need to rasterize the tracked mesh using the scene camera parameters to create the UV lookups. The avatar can be readily generated using the neural renderer $\mathcal{R}$. See Fig. 7 for an illustration and the supplementary video for additional results. The viewpoint stability is unlike most image-based CNN approaches, which often synthesize inconsistent appearance with varying viewpoints [41].

**Animation:** The learned neural identity can be retargeted to any motion from a motion capture database. Fig. 8 shows renderings of the same motion sequence from multiple views. Importantly, our model adds vivid and realistic

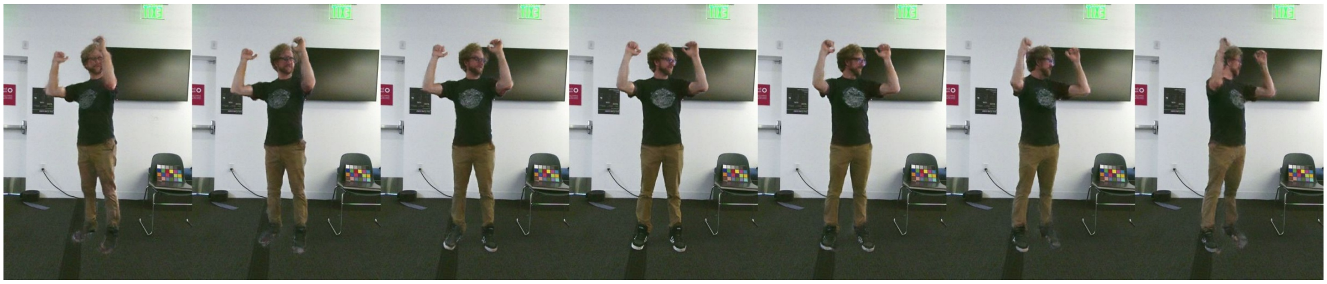


Figure 7: Viewpoint generalization demonstration. The proposed model model is robust to viewpoint variation, even for unseen poses, and shows high level of detail.

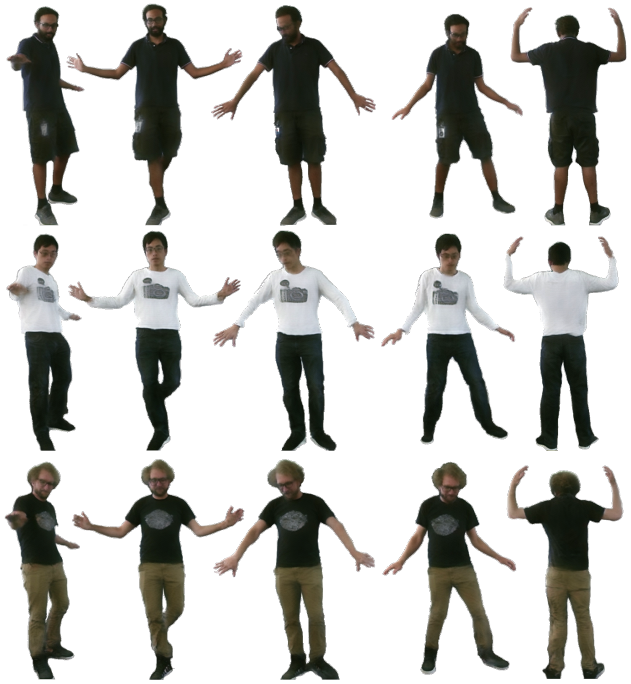


Figure 8: Avatar animation example. Any motion capture data that can be used to animate the base mesh can be used to drive the avatar. All avatars shown here are rendered using a single neural network.

pose-dependent deformation to the rendered avatar, which is not possible for other methods using skinned, but coarse meshes [15]. Fig. 3 provides a detailed view of the pose-dependent deformation appearance generation.

**Replacement of Textures / Virtual Try-On:** The learned neural texture is not directly interpretable. However, for two identities trained on the same neural rendering network, we can swap parts of the neural volume to generate identities with swapped faces/clothing items, as shown in Fig. 9. This is unlike fully 3D based approaches which require detailed captures for each new avatar [11].

## 6. Discussion

We introduce ANR, a novel neural rendering framework, for high-quality virtual avatars with arbitrary skeletal animations and viewpoints. Our key is to account for geometric misalignment and pose-dependent surface deformation.

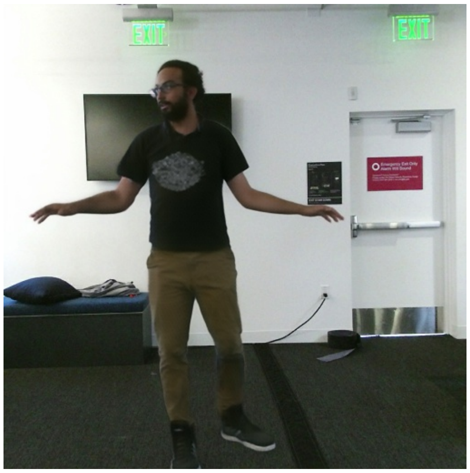


Figure 9: Virtual Try-On example. ANR enables texture mixing by swapping the regions of the neural texture. This example validates the disentanglement of appearance and neural shading network when ANR is trained on multiple identities. **Animated figure is in the appendix.**

Our solutions are carefully integrated into an end-to-end learning framework with a novel neural rendering architecture and adjusted optimization scheme. ANR can render multiple avatars using a single neural rendering model. By decoupling of texture and geometry ANR enables mixing and editing of appearance. For higher quality results, fine-tuning the model on a specific identity is an option. This makes the resulting avatars directly applicable in use cases where the range of motion is known or can be estimated well, for example for virtual assistants or game characters.

We notice that large and consistent tracking errors often leads to blurry appearance synthesis. This is where the split-optimization is not effective. One potential solution toward resiliency to large pose tracking errors is explicit pose and shape refinement via inverse rendering. Additionally, ANR currently bakes the scene lighting to the neural appearance. Incorporating intrinsic decomposition to decouple lighting and surface reflectance is a prominent future direction.

**Acknowledgement** This work was done while AR, JT, and CS were at Facebook. We thank Tiancheng Zhi and Tony Tung for help with data processing and Michael Zollhöefer for fruitful discussions.

## References

[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. *arXiv preprint arXiv:1906.08240*, 2019. 1, 2

[2] Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. Learning to reconstruct people in clothing from a single rgb camera. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[3] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[4] Guha Balakrishnan, Amy Zhao, Adrian V Dalca, Fredo Durand, and John Guttag. Synthesizing images of humans in unseen poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8340–8348, 2018. 3

[5] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018. 5

[6] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5933–5942, 2019. 3

[7] 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. 1

[8] 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. 1

[9] 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. 4

[10] Artur Grigorev, Artem Sevastopolsky, Alexander Vakhitov, and Victor Lempitsky. Coordinate-based texture inpainting for pose-guided image generation. *arXiv preprint arXiv:1811.11459*, 2018. 3

[11] 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. 1, 2, 8

[12] Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Deepcap: Monocular human performance capture using weak supervision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020. 3

[13] Steven D Hickson. *Encoding 3D Contextual Information For Dynamic Scene Understanding*. PhD thesis, Georgia Institute of Technology, 2020. 7

[14] Jingwei Huang, Justus Thies, Angela Dai, Abhijit Kundu, Chiyu Jiang, Leonidas J Guibas, Matthias Nießner, and Thomas Funkhouser. Adversarial texture optimization from rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1559–1568, 2020. 5

[15] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. Arch: Animatable reconstruction of clothed humans, 2020. 3, 8

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 6

[17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 5

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[19] Christoph Lassner, Gerard Pons-Moll, and Peter V Gehler. A generative model of people in clothing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 853–862, 2017. 3

[20] 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, 2018. 3

[21] Wen Liu, Zhixin Piao, Min Jie, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 3

[22] Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. Deep appearance models for face rendering. *ACM Transactions on Graphics (TOG)*, 37(4):1–13, 2018. 1, 2

[23] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Transactions on Graphics (TOG)*, 38(4):65, 2019. 1

[24] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 2015. 3, 6

[25] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems*, pages 406–416, 2017. 3

[26] Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. Disentangled person image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 99–108, 2018. 3

[27] Ricardo Martin-Brualla, Rohit Pandey, Sofien Bouaziz, Matthew Brown, and Dan B Goldman. Gelato: Generative latent textured objects. In *ECCV*, 2020. 1, 2

[28] Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, et al. Lookingood: enhancing performance capture with real-time neural re-rendering. *arXiv preprint arXiv:1811.05029*, 2018. 2

[29] 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, 2020. 1

[30] Koki Nagano, Huiwen Luo, Zejian Wang, Jaewoo Seo, Jun Xing, Liwen Hu, Lingyu Wei, and Hao Li. Deep face normalization. *ACM Transactions on Graphics (TOG)*, 38(6):1–16, 2019. 1

[31] Natalia Neverova, Riza Alp Guler, and Iasonas Kokkinos. Dense pose transfer. In *Proceedings of the European conference on computer vision (ECCV)*, pages 123–138, 2018. 3

[32] Rohit Pandey, Anastasia Tkach, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Ricardo Martin-Brualla, Andrea Tagliasacchi, George Papandreou, Philip Davidson, Cem Keskin, et al. Volumetric capture of humans with a single rgbd camera via semi-parametric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9709–9718, 2019. 3

[33] Amit Raj, Patsorn Sangkloy, Huiwen Chang, James Hays, Duygu Ceylan, and Jingwan Lu. Swapnet: Image based garment transfer. In *European Conference on Computer Vision*, pages 679–695. Springer, 2018. 3

[34] 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. 3

[35] 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 Conference on Computer Vision and Pattern Recognition*, June 2020. 3

[36] 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. 2, 3, 6

[37] Chenyang Si, Wei Wang, Liang Wang, and Tieniu Tan. Multistage adversarial losses for pose-based human image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 118–126, 2018. 3, 6

[38] 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. 1, 2, 3, 6

[39] Aaron Walsman, Weilin Wan, Tanner Schmidt, and Dieter Fox. Dynamic high resolution deformable articulated tracking. In *3DV*, 2017. 2

[40] Aaron Walsman, Weilin Wan, Tanner Schmidt, and Dieter Fox. Dynamic high resolution deformable articulated tracking. In *3D Vision*, 2017. 5

[41] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 2, 6, 7

[42] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 5, 6

[43] Mihai Zanfir, Alin-Ionut Popa, Andrei Zanfir, and Cristian Sminchisescu. Human appearance transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5391–5399, 2018. 3

[44] Bo Zhao, Xiao Wu, Zhi-Qi Cheng, Hao Liu, Zequn Jie, and Jiashi Feng. Multi-view image generation from a single-view. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 383–391, 2018. 3

[45] Tiancheng Zhi, Christoph Lassner, Tony Tung, Carsten Stoll, Srinivasa G Narasimhan, and Minh Vo. Texmesh: Reconstructing detailed human texture and geometry from rgb-d video. In *ECCV*, 2020. 2