HumMorph: Generalized Dynamic Human Neural Fields from Few Views

Jakub Zadrożny Hakan Bilen
University of Edinburgh

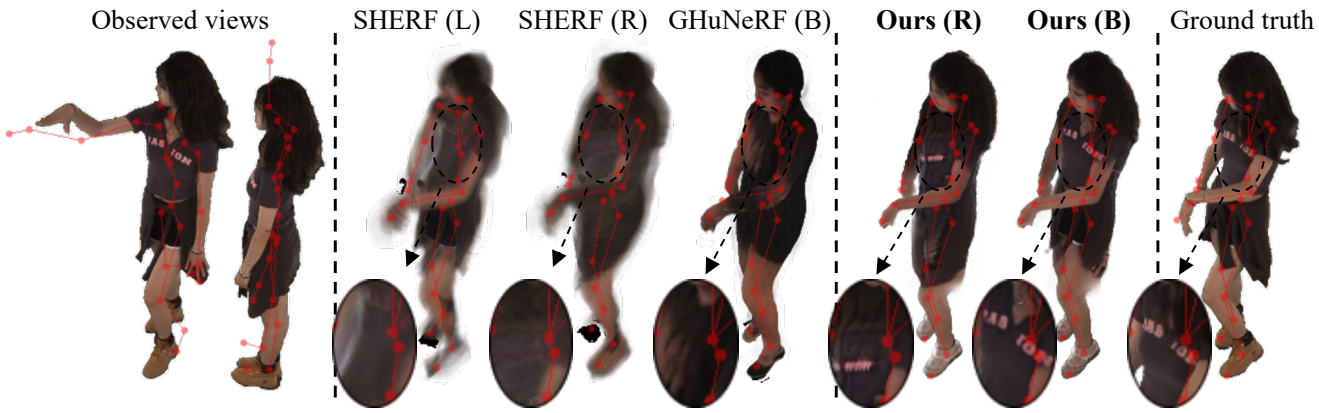

Figure 1. **HumMorph** is a generalized method for free-viewpoint synthesis of humans in novel poses given a few observations. State-of-the-art methods, including SHERF [11] and GHuNeRF [16], require accurate body pose annotations for the observed views. These are typically unavailable in practice and the poses need to be noisily estimated instead, like in the example above (poses shown in red). In this scenario, existing approaches struggle to model details and synthesize oversmoothed renders. In contrast, our approach includes dense 3D processing modules and accounts for pose estimation errors to accurately recover detail. The letter in parentheses indicates which views were supplied to the methods: L – left, R – right or B – both.

Abstract

We propose HumMorph – a novel generalized approach to free-viewpoint rendering of dynamic human bodies with explicit pose control. HumMorph renders a human actor in any specified pose given a few observed views (starting from just one) in arbitrary poses. Our method enables fast inference due to relying only on feed-forward passes through the model. We first construct a coarse representation of the actor in the canonical T-pose, which combines visual features from individual partial observations and fills missing information using learned prior knowledge. The coarse representation is complemented by fine-grained pixel-aligned features extracted directly from the observed views, which provide high-resolution appearance information. We demonstrate that HumMorph is competitive with the state-of-the-art when only a single input view is available, however, we achieve results with significantly better visual quality given just 2 monocular observations. Moreover, previous generalized approaches to this problem were evaluated assuming access to accurate body shape and pose parameters estimated using synchronized multi-camera setups. In contrast, we consider a more practical scenario where these body parameters are noisily estimated directly from the observed views. Our experimental results demonstrate that our architecture is more robust to errors in the noisy parameters and clearly outperforms the state-of-the-art in this setting.

1. Introduction

Synthesizing high-quality and realistic humans for unseen poses and viewpoints is essential for building a realistic and vibrant Metaverse. It has natural applications directly related to augmented/virtual reality (AR/VR), such as 3D immersive communication, but also in wider content creation including movie production. In this work, we focus on learning models that can synthesize humans solely from monocular frames, without requiring costly multi-camera capturing setups. This is a key step towards in-the-wild applicability, where only non-specialized capturing equipments such as mobile devices are available.

Despite the remarkable progress in human modeling from monocular videos, most approaches [8, 18, 26, 31,

34, 36] require training a separate model for each subject, which heavily limits their applicability in practice due to compute and energy requirements. HumanNeRF [34] maps all observations to a canonical Neural Radiance Field (NeRF) [22] and learns a motion field mapping from observation to canonical space. However, subject-specific models such as HumanNeRF [34] require extensive observations for each subject and fail to in-paint details that are not visible in the observations, as they do not learn prior information from multiple subjects.

Recent works [6, 11, 16, 21, 23] learn 'generalized' human models from multiple identities, and generalize to a previously unseen target identity and their unseen poses from a set of observations in a single forward pass significantly speeding up inference and making them more suitable for real-world applications. However, SHERF [11] is designed to learn only from a single observation of a human actor and cannot combine multiple observations to provide high-quality synthesis even when multiple observations are available. In contrast, GHuNeRF [16] allows aggregating information from multiple frames of a monocular video, however, to resolve occlusions it relies on pre-computed visibility masks of a template body mesh. Such masks are not always accurate, especially when body pose is estimated from sparse observed monocular views and GHuNeRF [16] lacks a mechanism to account for that. Both SHERF and GHuNeRF heavily rely on deformations of an appropriate SMPL [20] mesh to spatially register points across different body poses. In the common human datasets [2, 5] used for evaluations, the SMPL parameters are accurately estimated from synchronized multi-view camera setups, which in practice are not available. These parameters should instead be estimated directly from the observed views, resulting in more noisy estimations and significantly worse reconstructions which we report in the experiments.

To tackle this challenge we propose *HumMorph*, a novel efficient generalized model that can synthesize subjects effortlessly from several frames of a monocular video without relying on a template body mesh and using only feed-forward network passes. Our model incorporates prediction of skinning weights, which eliminates the requirement for accurate body shape parameters and leads to more robust human synthesis. At the heart of our architecture lies our *VoluMorph* module that lifts the observed views from 2D to 3D, then aligns them to the canonical pose (T-pose) and finally combines them through 3D convolutions and attention-based aggregation. The combined feature volumes are used to estimate a coarse body model in the canonical pose and a residual correction to initial heuristic skinning weights. Similarly to existing generalized approaches we require the body pose parameters, but we do not require the body shape parameters and instead rely on the 3D skeleton shape in the canonical T-pose. Given accurate poses we demonstrate results of state-of-the-art perceptual visual quality using a single observed view with a significant boost in quality when two input views are available and further improvements with additional observations. Crucially, our experiments show that our architecture, thanks to using dense 3D processing, is significantly more robust than the state of the art when provided with body pose parameters noisily estimated from the observed views.

2. Related Work

Neural scene representations. Our work builds on the recent advancements in neural rendering techniques. NeRF [22] assigns density and color values to points in the 3D space, which enables novel view rendering via alpha-compositing densities and colors along camera rays. The original NeRF [22] requires per-scene optimization and dense set of observed views for supervision. Multiple conditional variants of NeRF were proposed [9, 19, 28, 30, 35] to enable feed-forward inference given sparse observed views. However, these only target *static* scenes.

Neural fields for humans. The use of parametric body models [1, 20, 25] enabled learning dynamic neural fields for humans by providing a prior for the geometry and accurate deformation model. The dominant approach [8, 18, 26, 31, 34, 36] is to model the body in the canonical pose (T-pose) and, during volumetric rendering, deform the query points from observation space to the canonical space using linear blend skinning (LBS). The LBS deformation, however, cannot accurately capture some soft elements like muscles, hair or clothing. To address this issue, several approaches [18, 34, 36] optimize a pose-dependent residual flow field, which acts as a correction to the LBS deformation. SurMo further improves modelling of clothing by optimizing a surface-based 4D motion representation, but requires multi-view videos for supervision. However, these methods are optimized per subject, which is computationally expensive and typically requires extensive multi-view or monocular observations for supervision. Recently developed methods based on 3D Gaussian splatting [10, 12, 14, 24, 27, 32] managed to considerably reduce the optimization time, however, they still do not match the efficiency of feed-forward inference and require more observations for supervision.

Generalized neural fields for humans. To address the challenges of subject-specific human neural field models, several generalized approaches have been explored [4, 6, 7, 11, 15, 16, 21, 23]. They are particularly efficient at inference time as they only require feed-forward passes through the model. Moreover, they reduce the amount of required observations by leaning on prior learned during training. They generally condition the neural field on features extracted from the observed views after deforming query points from the target body pose to the observed poses.

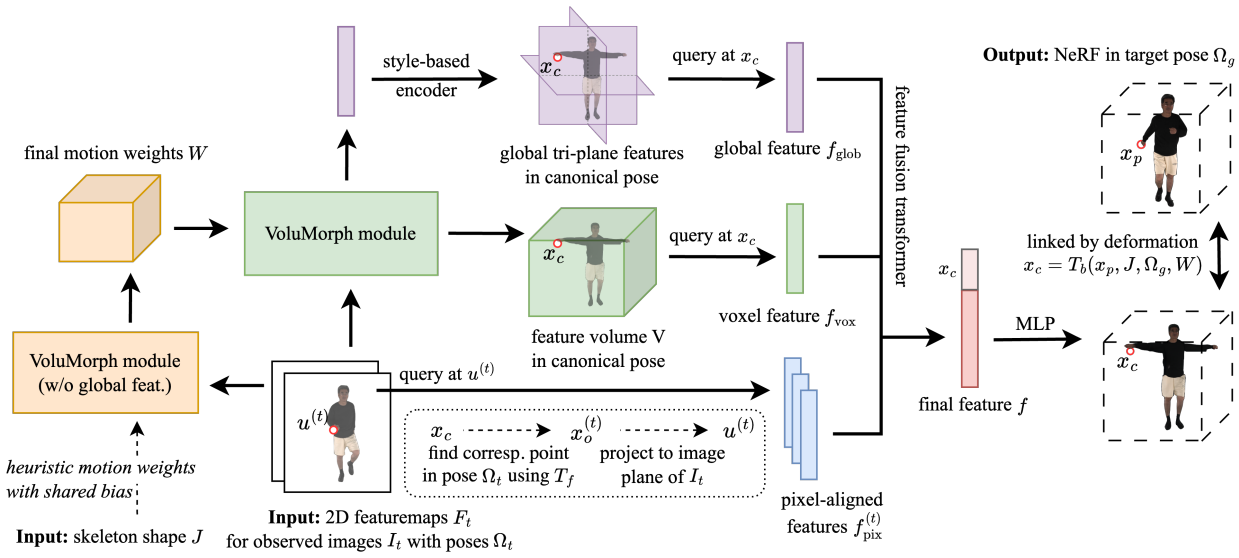

Figure 2. An overview of our approach. First, we extract the 2D featuremaps F_t, which we pass through a VoluMorph module to get the final motion weights W. The features F_t and motion weights W are passed to a second VoluMorph module, which outputs the volume V and a global latent code. Finally, we extract $f_{\text{vox}}, f_{\text{glob}}, f_{\text{pix}}$ and combine them using the feature fusion module to condition the NeRF MLP.

However, most existing approaches [4, 6, 7, 15, 23] assume that multiple views with the same body pose are observed at test time, which is unlikely to be satisfied in practice. The most related to our work are SHERF [11], GHuNeRF [16] and GNH [21], which, similarly to us, focus on the monocular setting, where each observed view captures a different body pose. SHERF [11] reconstructs a human neural field from a single observation by extracting a global feature vector along with local observations aligned using SMPL mesh deformation. GHuNeRF [16] resolves occlusions by projecting the SMPL meshes onto the camera planes. GNH [21] uses a convolutional renderer instead of volumetric rendering, which increases efficiency, but can result in 3D inconsistencies. The existing generalized approaches to human novel view and novel pose rendering rely predominantly on aligning the observed poses to the canonical pose by deforming the SMPL mesh, where the pose and shape parameters are assumed to be known and accurate. We propose an architecture that does not rely on the SMPL mesh but instead processes the observations densely in 3D and can correct slight pose parameter errors.

3. Method

Our goal is to learn a generalized dynamic human neural field which can render a novel subject in a given novel pose Ω_g from given viewpoint $\mathcal{E}_g$ conditioned on T observed views $\mathcal{I} = \{I_t\}_{t=1}^T$. Each view I_t may be capturing the actor in a different pose Ω_t from an arbitrary viewpoint $\mathcal{E}_t$ where we assume that both Ω_t and $\mathcal{E}_t$ are known. The pose parameters Ω_g, Ω_t for the target and observed views contain local joint rotations for $K = 24$ joints relative to the *canonical* pose (T-pose). We also assume that the 3D skeleton shape J in the canonical pose is known, but unlike most previous generalized methods, we do not require the SMPL [20] body shape parameters. Finally, we assume that camera intrinsics $\mathcal{K}$ and extrinsics $\mathcal{E}_g, \mathcal{E}_t$ for the target and observed frames are known.

The overview of our approach is presented in Fig. 2. We represent the subject's body in a given target pose Ω_g with a neural radiance field (NeRF) [22] and use standard volumetric rendering to produce the target images from any given viewpoint $\mathcal{E}_g$. To address the great variety in shapes of posed human bodies, we primarily model the body in the canonical T-pose similar to [11, 16, 34, 36] and represent the observation space neural field (corresponding to the target pose) as a deformation of the canonical field. Specifically, for an observation space query point $x_p \in \mathbb{R}^3$ (corresponding to pose Ω) let $x_c = T_b(x_p, \Omega; \mathcal{I}, J, W)$ be the corresponding point in the canonical space, where T_b is the *backward deformation* detailed in Sec. 3.2. To query the observation space NeRF for color $\bar{\sigma}$ and density $\bar{\mathbf{c}}$ we instead query the respective canonical fields $\sigma, \mathbf{c}$, *i.e.* $\bar{\sigma}(x_p) = \sigma(x_c)$ and $\bar{\mathbf{c}}(x_p) = \mathbf{c}(x_c)$. Naturally, the canonical and observation-space neural fields are conditioned on the observed views $\mathcal{I}$, which we accomplish by extracting a feature vector f for a canonical query point x_c from $\mathcal{I}$ and setting $(\sigma(x_c), \mathbf{c}(x_c)) = \text{MLP}(x_c, f)$.

In the remainder of this section, we describe the key contributions in our approach, namely: (1) the conditioning of the canonical neural field on the observed frames (*i.e.* the

extraction of features f) (Sec. 3.1) and (2) the representation and learning of deformations between the observation and canonical space (Sec. 3.2).

3.1. The Canonical Body Model

To condition the canonical neural field on the observed views $\mathcal{I}$, we extract three types of features: global f_{glob}, voxel-based f_{vox} and pixel-aligned f_{pix}. The global and voxel-based features are produced by our 3D *VoluMorph* encoding module, which lifts each observed view of the body (in arbitrary poses) into a partial canonical model and combines these into a single, complete canonical representation at a coarse level. The pixel-aligned features complement the coarse model with fine details by extracting observations of a query point directly from the posed observed views. The three features have complementary strengths and tackle different key challenges: f_{vox} can resolve occlusions, inject prior and compensate for slight pose inaccuracies; f_{glob} captures overall characteristics and appearance of the subject through a flat (1D) latent code, which further facilitates prior injection and reconstruction of unobserved regions; f_{pix}, when available, provides direct, high-quality appearance information. Note that the pixel-aligned features are not always relevant (*e.g.* due to occlusions or deformation inaccuracy) or may not be available as some points may have never been observed. We extract a single global and voxel feature for each point as the encoder already aggregates information from all available observations, but use T pixel-aligned features $f_{\text{pix}}^{(t)}$, one for each observed view. The f_{vox}, f_{glob} and $f_{\text{pix}}^{(t)}$ features are combined into a single, final feature vector f for NeRF conditioning by an attention-based feature fusion module.

In this section, let us assume that the backward deformation T_b from the observation space to the canonical space and the forward deformation T_f in the opposite direction are already defined (see Sec. 3.2 for the definitions). We begin the feature extraction process by computing 2D feature maps $F_t = \text{CNN}(I_t)$ extracted for each observed view independently with a U-Net feature extractor similar to [30, 36].

Voxel-based and global features. The objective for our 3D *VoluMorph* encoding submodule is to lift the featuremaps F_t corresponding to observed views in arbitrary poses into a combined 3D feature grid, which is aligned with the canonical body pose. For an overview of the VoluMorph module see Fig. 3. The initial step in VoluMorph is the *unprojection* of the 2D featuremaps F_t into 3 dimensions based on the known camera parameters. The initial feature volumes are aligned to the observed poses instead of the canonical pose, which we correct with a volume *undeformation* operation. Specifically, the combined unprojection and undeformation step is captured as

$$V_t(x_v) = F_t\left[\pi\left(T_f(x_v, \Omega_t), \mathcal{K}, \mathcal{E}_t\right)\right], \qquad (1)$$

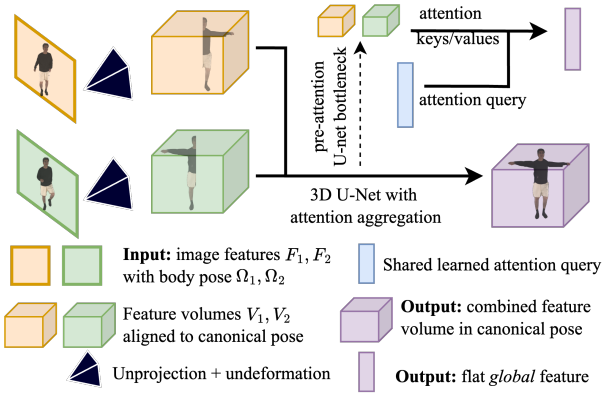

Figure 3. The architecture of our *VoluMorph* module.

where V_t is the undeformed feature grid for view I_t with pose Ω_t, x_v is a 3D grid point, $\pi(\cdot, \mathcal{K}, \mathcal{E}_t)$ is the camera projection operation with instrinsics $\mathcal{K}$ and extrinsics $\mathcal{E}_t$, and $[\cdot]$ is bilinear interpolation. The aligned, partial models (volumes) V_t are combined into a single, complete model V by a 3D U-Net-based convolutional network with attention-based aggregation between views as in [29]. The key feature of this module is that it can learn a semantic understanding of the body and can therefore capture and inject prior knowledge as well as (to some extent) resolve occlusions. We set the voxel feature for a query point x_c in canonical space as $f_{\text{vox}} = V[x_c]$ using tri-linear interpolation.

To further facilitate prior injection, we additionally extract the flat 512-dimensional latent code which underlies f_{glob}. To this end, we attend to the volumes of the bottleneck layer of the 3D U-Net. Let V_t' be the volumes corresponding to V_t downscaled to the minimal resolution by the 3D U-Net (before cross-view aggregation). We then get the global latent z by applying a 4-head attention module where the query is a shared, learned 512-dimensional vector and keys/values are linear projections of the flattened V_t' (see Fig. 3). We choose to extract the latent z from V_t' as they are already processed for high-level information and abstract away the pose diversity, which simplifies cross-view aggregation. Finally, similar to SHERF [11], we pass the latent z through a style-base encoder [13], which returns 3 feature planes defining a tri-plane 3D feature space [3], which is sampled at x_c to get f_{glob}.

Pixel-aligned features. The VoluMorph module operates at a coarse level due to the high memory requirements of voxel-based processing. To effectively increase the rendering resolution, we extract pixel-aligned features $f_{\text{pix}}^{(t)}$ from all observed views for each 3D query point in the canonical space x_c . We utilize the forward deformation T_f to find the corresponding point $x_p^{(t)} = T_f(x_c, \Omega_t)$ in the observation space of view I_t. We then extract $f_{\text{pix}}^{(t)}$ for x_c

from the featuremap F_t using bilinear interpolation at the projected location $u^{(t)} = \pi(x_p^{(t)}, \mathcal{K}, \mathcal{E}_t)$ (see Fig. 2).

Feature fusion. We use an attention-based feature fusion module to determine which (if any) of the pixel-aligned features are relevant and combine them with coarse model defined by the voxel and global features. For a query point x_c we extend its $f_{\text{glob}}, f_{\text{vox}}$ and $f_{\text{pix}}^{(t)}$ with spatial information, *i.e.* x_c coordinates, distance from x_c to nearest joint, and viewing direction at x_c and, for pixel-aligned features, the viewing direction under which the feature was observed in the input views. This information should, intuitively, help the model perform spatial reasoning, *e.g.* rely more on pixel-aligned features when the query and observed viewing directions align and the query point is close to the surface. The extended features are processed with a single transformer encoder layer. The final feature f for x_c is constructed by selecting the most relevant features using an attention layer, where queries are based on the spatial features of x_c (position, viewing direction, etc.) and keys/values are based on the transformer encoder's output.

3.2. Deformation representation and learning

Recall that our definition of the neural field in observation space relies on the *backward deformation* T_b, which for a point x_p in observation space with pose Ω returns a corresponding point $x_c = T_b(x_p, \Omega; \mathcal{I})$ in the canonical pose. Moreover, our feature extraction process (Sec. 3.1) relies on the *forward deformation* T_f. In this section, we explain how these deformations are parameterized and conditioned on the observed views.

Linear Blend Skinning for Humans. Our representation follows [33, 34, 36], but adapts it to the generalized setting with a feed-forward conditioning mechanism. The deformations follow the linear blend skinning (LBS) model

$$T_b(x_p, \Omega; \mathcal{I}) = \sum_{i=1}^{K} w_p^{(i)}(x_p; \mathcal{I})(R_i x_p + t_i), \tag{2}$$

$$T_f(x_c, \Omega; \mathcal{I}) = \sum_{i=1}^{K} w_c^{(i)}(x_c; \mathcal{I}) R_i^{-1}(x_c - t_i), \tag{3}$$

where $w_p^{(i)}, w_c^{(i)}$ are the blend (motion) weights for the i-th bone in the posed and canonical space (respectively), and R_i, t_i are the rotation and translation which transform points on the i-th bone from observation space to canonical. The rotation R_i and translation t_i are explicitly computed from the pose Ω (see [33, 34] for a derivation). Following [34], we only optimize the canonical motion weights $w_c^{(i)}$ and express motion weights in the posed space as

$$w_p^{(i)}(x_p; \mathcal{I}) = \frac{w_c^{(i)}(R_i x_p + t_i; \mathcal{I})}{\sum_{k=1}^{K} w_c^{(k)}(R_k x_p + t_k; \mathcal{I})} \tag{4}$$

in order to improve generalization across poses. Throughout our pipeline the motion weights are represented as a discrete volume $W(\mathcal{I})$ with K channels summing to one at each voxel, where K is the number of joints in J. The continuous motion weights w_c are then computed using trilinear interpolation from the discrete volume.

Conditioning deformations on observed views. The deformations T_b, T_f are conditioned on the observed views $\mathcal{I}$ through motion weights w_c. Following [33, 34] we begin with a heuristic initial estimate of the motion weights W_0 constructed by placing ellipsoidal Gaussians around the i-th bone for the i-th channel of W_0. We additionally refine the initial weights with a learned bias represented as a 3D CNN output from a random fixed latent code (shared between all subjects). The initial motion weights define through Eq. (2) an initial guesses $\tilde{T}_f$ for the forward deformation. We then estimate an observation-conditioned correction $\Delta W(\mathcal{I})$ to the initial W_0 using our VoluMorph module introduced in Sec. 3.1 (w/o the global latent) with $\tilde{T}_f$ as the forward deformation (in Eq. (1)). The intuition behind this choice is that $\tilde{T}_f$ will provide a rough alignment of observed views to the canonical pose from which the convolutional architecture of VoluMorph will recover the body shape (and through that the motion weights) despite some remaining misalignment. Note that, as shown in Fig. 2, we use two separate VoluMorph modules, one for the neural field features V and one for the motion weights ΔW.

3.3. Network training

We optimize all components of our model end-to-end to minimize the loss function $\mathcal{L} = \mathcal{L}_{\text{LPIPS}} + \lambda_1 \mathcal{L}_{\text{MSE}} + \lambda_2 \mathcal{L}_{\text{consis}}$, where $\mathcal{L}_{\text{MSE}}$ is the pixel-wise mean squared error from the ground-truth image, $\mathcal{L}_{\text{LPIPS}}$ is the LPIPS [37] perceptual loss. We include the $\mathcal{L}_{\text{consis}}$ term proposed by [36], which regularizes the motion weights by minimizing $\|x_p - T_f(T_b(x_p, \Omega), \Omega)\|^2$. We use $\lambda_1 = 0.3, \lambda_2 = 0.5$ in our experiments.

Furthermore, we found that the prediction of motion weights becomes more challenging when the pose parameters are noisily estimated from the observed images. To mitigate this we introduce additional regularization guidance

$$\mathcal{L}_{\text{near}} = \sum_{x \in W} \sum_{k=1}^{K} w_c^{(k)} d(x, B_k), \tag{5}$$

where $x \in W$ are voxel position in the motion weights volume W and $d(x, B_k)$ is the distance from x to the k-th bone (line segment), which encourages assigning points to their nearest bone. We use $\mathcal{L} + \lambda_3 \mathcal{L}_{\text{near}}$ with $\lambda_3 = 0.1$ as the loss function and increase λ_2 to 2 for experiments with estimated poses.

Method	HuMMan [2]				DNA-Rendering [5]			
	Accurate body params.		*Estim. body params.*		*Accurate body params.*		*Estim. body params.*	
	PSNR ↑	LPIPS* ↓	PSNR ↑	LPIPS* ↓	PSNR ↑	LPIPS* ↓	PSNR ↑	LPIPS* ↓
SHERF (Mo)	26.95	44.12	24.23	61.44	28.49	48.22	26.93	61.97
GHuNeRF+ (1 obs.)	23.89	44.00	23.17	50.24	26.59	53.10	26.19	56.46
GHuNeRF+ (2 obs.)	23.97	43.72	23.27	49.96	26.69	52.92	26.28	56.16
GHuNeRF+ (4 obs.)	24.00	43.66	23.31	49.86	26.70	52.93	26.31	56.11
GHuNeRF (4 obs.)	23.89	63.02	23.36	68.76	27.78	69.71	**27.28**	74.54
Ours (1 observed)	26.73	33.41	25.08	42.28	27.82	40.36	26.81	47.59
Ours (2 observed)	27.43	30.37	25.33	40.93	28.30	38.13	27.10	45.88
Ours (3 observed)	27.69	29.30	**25.40**	40.53	28.59	36.91	27.24	45.22
Ours (4 observed)	**27.73**	**29.28**	**25.40**	**40.52**	**28.60**	**36.85**	27.25	**45.20**

Table 1. Quantitative comparison of our method with SHERF [11] and GHuNeRF [16] with various numbers of observed views. SHERF (Mo) is trained in our monocular framework, while GHuNeRF+ contains the added LPIPS loss. LPIPS* = LPIPS $\times 10^3$.

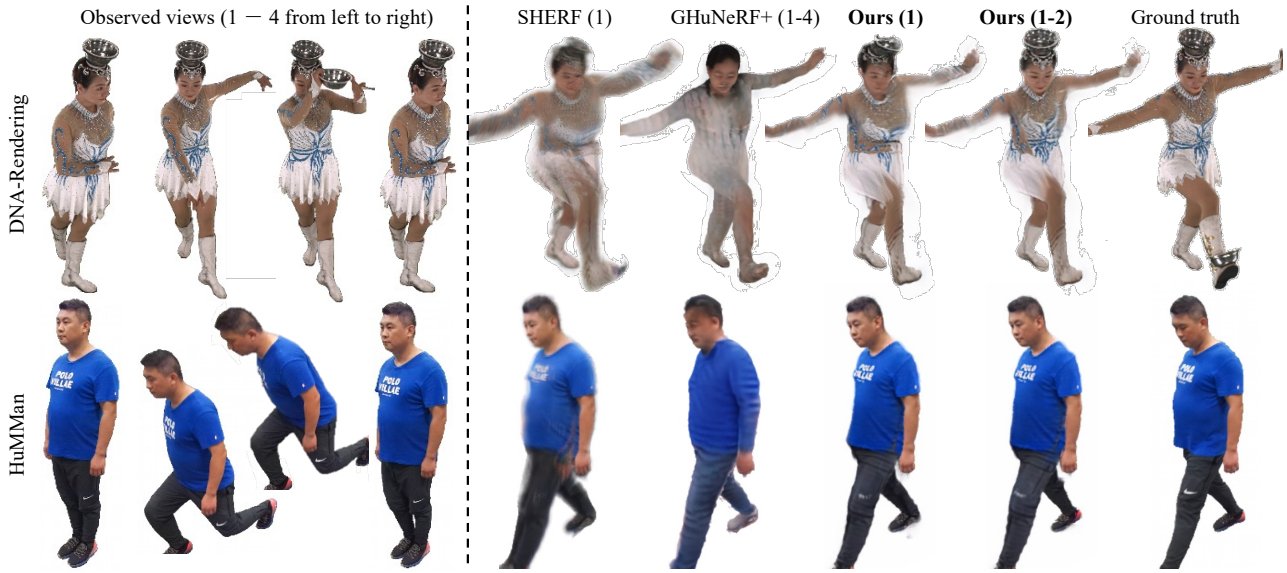

Figure 4. Qualitative comparison with SHERF (Mo) [11] and GHuNeRF+ [16] given accurate shape and pose parameters. Numbers in parentheses indicate the range of observed views supplied to the respective models.

4. Experiments

We evaluate our approach on two large-scale human motion datasets: HuMMan [2] and DNA-Rendering [5]. The HuMMan dataset consists of 339 motion sequences performed by 153 subjects. We use 317 sequences for training and 22 sequences for hold-out evaluation according to the official data split. We use parts 1 and 2 from the DNA-Rendering dataset, which together consist of 436 sequences performed by 136 subjects. We split the sequences into 372 for the training set and 64 for the test set. We assign all sequences performed by the same actor to either the train or test set, which results in 118 training subjects and 18 test subjects. To reduce the computational requirements, we subsample the camera sets from 10 to 8 on HuMMan and from 48 to 6 on DNA-Rendering (see the suppl. material for details).

4.1. Implementation Details

To simulate practical conditions where multi-view synchronized videos are unlikely to be available, we train our models in the fully monocular framework. Specifically, during training the observed frames are selected from the same camera as the target frame. At training time we provide our model with 2 randomly sampled observed frames at each step and render 32x32 patches of the target frames downscaled to 0.25 of the original resolution.

Architecture. We use the same U-Net feature extractor as [36]. We use a voxel grid with resolution 32 in our VoluMorph modules. The architecture of the 3D U-Net convolutional network inside our VoluMorph modules follows [29]. We adopt the style-based encoder from [11] to decode the global feature tri-planes. The 2D featuremaps F_t

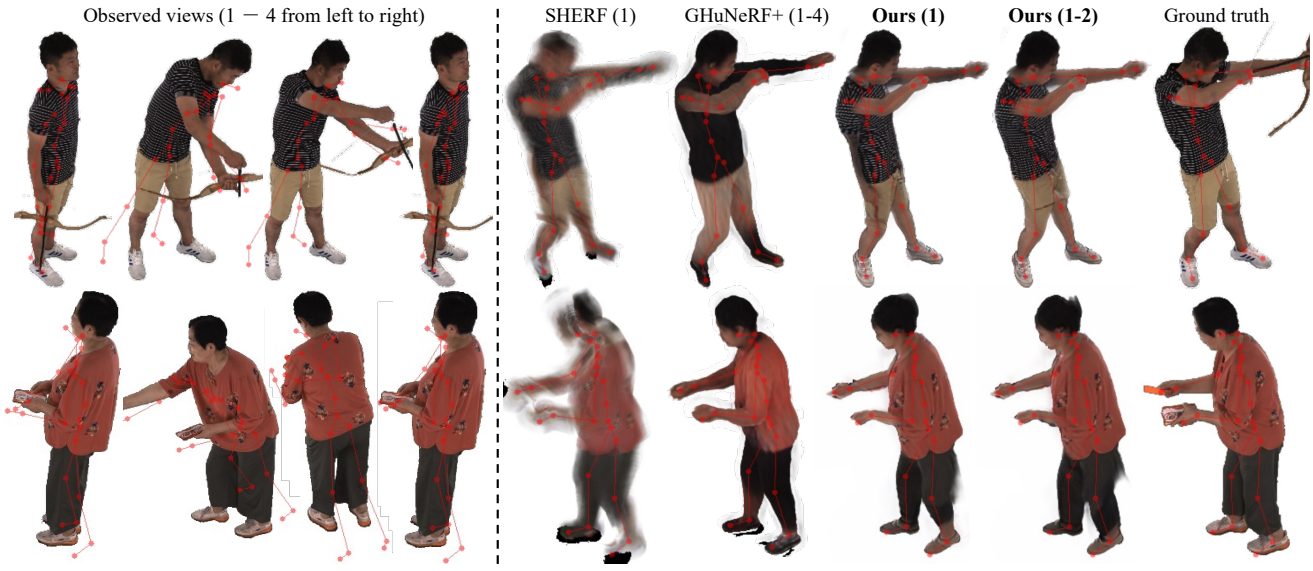

Figure 5. Qualitative comparison with SHERF (Mo) [11] and GHuNeRF+ [16] on DNA-Rendering [5] with shape and pose parameters estimated from observed views. Numbers in parentheses indicate the range of observed views supplied to the respective models.

and 3D features $f_{\text{glob}}, f_{\text{vox}}, f_{\text{pix}}$ are 32-dimensional. All attention layers in the feature fusion module use 4 heads and have internal dimension 64. We use an 8-layer MLP NeRF decoder with 256-dimensional hidden layers. See the suppl. material for more details on architecture and optimization. We will release our code publicly upon acceptance.

4.2. Evaluation Protocol

We compare our approach to two state-of-the-art methods: SHERF [11] (which can only be conditioned on a single view) and GHuNeRF [16]. Note that a fair comparison to related GNH [21] is not currently possible since the code and models have not yet been made publicly available. In the evaluations, we follow our monocular protocol, i.e. we provide observed frames from the same camera as the target frame. Hu et al. train SHERF [11] in a multi-view framework, hence, to ensure a fair comparison, we re-train SHERF in our monocular framework and we refer to it as SHERF (Mo). We also re-train the GHuNeRF model since it was not originally evaluated on the datasets used in this work. Moreover, the original GHuNeRF does not include a perceptual LPIPS [37] loss term, which causes it to produce oversmoothed renders. For a fair comparison, we also report results for GHuNeRF+, which is our version of GHuNeRF with an LPIPS term added to the loss. For each baseline training, we match our hyperparameter setups as closely as possible to the ones used by respective authors. We train GHuNeRF using 4 observed views as it was originally designed to work with many frames.

To measure the rendering quality given *novel poses* we split the motion sequences approximately in half at frame $\lfloor \frac{T+1}{2} \rfloor$, where T is the sequence length, and provide observations from the first half, while we render the frames from the second half (see the suppl. material for details on the choice of observed frames). We evaluate the rendering quality using two common metrics: pixel-wise PSNR and the perceptual LPIPS [37] metric.

In our experiments we consider two scenarios depending on whether accurate body parameters are available for the observed frames. If they are not, we use an off-the-shelf HybrIK [17] model to estimate the SMPL [20] shape and pose parameters for each observed frame independently. For this experiment, we re-train our models and the baselines using a mixture of accurate and estimated parameters (see the supplement for details). At test time we provide the models with the estimated parameters for the observed frames, but use accurate poses for the target frame (the target pose *cannot* be estimated). The shape parameters for the target frame are not provided and must be estimated from the observed views. We use the ground-truth camera poses and leave the integration of camera pose estimation as future work.

4.3. Quantitative comparison

The results of our quantitative evaluation are presented in Tab. 1. Our method achieves a significantly better perceptual score (LPIPS) compared to SHERF and GHuNeRF on both datasets given even a single observed view. However, with 1 observation our PSNR score is below that of SHERF, which we attribute to the fact that PSNR favours oversmoothed results over slight misalignments. Many subjects in the datasets are dressed in complex clothing with intricate details and failing to properly model its folding dy-

Method	*Accurate body parameters*				*Estimated body parameters*			
	1 observed		2 observed		1 observed		2 observed	
	PSNR ↑	LPIPS* ↓	PSNR ↑	LPIPS* ↓	PSNR ↑	LPIPS* ↓	PSNR ↑	LPIPS* ↓
Baseline	27.52	53.16	27.92	49.47	22.21	61.42	22.35	59.52
+ ΔW	27.58	43.61	28.16	40.46	26.45	51.62	26.85	49.50
+ f_{vox}	27.69	40.73	28.15	38.49	26.21	54.72	26.50	51.59
+ $\Delta W + f_{\text{vox}}$	**27.83**	**40.27**	**28.31**	**37.80**	26.48	51.06	26.87	48.26
+ $\Delta W + f_{\text{vox}} + f_{\text{glob}}$	27.82	40.37	28.30	38.13	26.62	48.83	27.00	47.01
+ $\Delta W + f_{\text{vox}} + f_{\text{glob}} + \mathcal{L}_{\text{reg}}$	-	-	-	-	**26.81**	**47.59**	**27.10**	**45.88**

Table 2. Ablation study results on DNA-Rendering [5] using accurate and estimated body parameters. LPIPS* = LPIPS $\times 10^3$. See the main text for a description of the components. Note that we do not use $\mathcal{L}_{\text{near}}$ with accurate body parameters.

namic, which none of the methods are targeting, can significantly lower the PSNR score. Our method outperforms SHERF in both PSNR and especially LPIPS given 2 views on HuMMan and 3 views on DNA-Rendering. Moreover, Tab. 1 shows that, while the quality of all models degrades when using estimated parameters, our method achieves a better perceptual score than the baselines when they are provided with accurate parameters. Finally, the performance of our method on all metrics generally improves as it is given additional observed views, but it saturates at 3 observations. Note that in our experiments additional views beyond the 3rd often do not considerably increase pose diversity. While the performance of GHuNeRF+ also increases given additional views, this effect is less pronounced.

4.4. Qualitative results

Fig. 4 and Fig. 5 show qualitative comparisons with the baselines given accurate and estimated body parameters (respectively), with varying numbers of observed frames. Note that SHERF [11] only accepts a single observation. As seen in Fig. 4 our renders are overall more sharp and detailed compared to both SHERF and GHuNeRF+. Given the single observation SHERF struggles to resolve occlusions or impose a smoothness prior, which results in 'phantom' limbs imprinted on the torso. Although in some cases our method exhibits a similar same issue, it can combine two observations to better match the body geometry and remove the artifacts. In general, GHuNeRF+ produces oversmoothed results, which do not reproduce details and only loosely reconstruct the original appearance. As shown in Fig. 5 our method successfully reconstructs clothing details even when provided with estimated body poses. Moreover, despite considerable pose estimation errors, it can combine the two input views to refine the body model (see top row Fig. 5). See the suppl. material for more qualitative results.

4.5. Ablation study

Tab. 2 presents results of an ablation study, which validates the design and usage of the main components of our model. The baseline model does not include the motion weights corrrection module ΔW and relies only on the pixel-aligned features. We then subsequently add further components, namely motion weights correction ΔW, the voxel-features f_{vox}, the global feature f_{glob} and the additional regularization $\mathcal{L}_{\text{near}}$ (see Eq. (5)). As show in Tab. 2, including the voxel features module grants the largest boost in rendering quality when using accurate body parameters. Without them the model will struggle to resolve occlusions and inject prior, which can result in rendering errors. Combining that with the motion weights correction improves results further, although by a smaller margin. Note that with accurate body parameters the use of global feature is not required, however we include it in our final model for consistency. In contrast, both the motion weights correction and the global feature have a significant impact when using estimated body parameters. In that case, applying additional regularization $\mathcal{L}_{\text{near}}$ refines the results further due to encouraging more natural deformations. Finally, Tab. 2 measures the positive impact of supplying the 2nd observation.

5. Conclusion

We introduced *HumMorph*, a novel generalized approach for free-viewpoint synthesis of human bodies with explicit pose control conditioned on a variable number of views. We demonstrated results of state-of-the-art perceptual visual quality given a single observed view and a significant boost in quality when two conditioning views are available. We also demonstrated that our approach is significantly more robust to inaccuracies when body pose parameters are noisily estimated compared to prior methods.

Limitations and future work. Despite the increased robustness of our approach to pose estimation errors, the resulting renders still show considerable room for improvement. Moreover, in this work we use ground truth camera poses. Investigating the use of estimated camera poses and adjusting the model accordingly is an interesting direction for future work. HumMorph also does not explicitly model deformations of clothing and usually cannot reconstruct interactions with objects, which requires further research.

References

[1] Thiemo Alldieck, Hongyi Xu, and Cristian Sminchisescu. imGHUM: Implicit Generative Models of 3D Human Shape and Articulated Pose. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5441–5450, Montreal, QC, Canada, 2021. IEEE. 2

[2] Zhongang Cai, Daxuan Ren, Ailing Zeng, Zhengyu Lin, Tao Yu, Wenjia Wang, Xiangyu Fan, Yang Gao, Yifan Yu, Liang Pan, and others. Humman: Multi-modal 4d human dataset for versatile sensing and modeling. In *European Conference on Computer Vision*, pages 557–577. Springer, 2022. 2, 6

[3] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, and others. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16123–16133, 2022. 4

[4] Mingfei Chen, Jianfeng Zhang, Xiangyu Xu, Lijuan Liu, Yujun Cai, Jiashi Feng, and Shuicheng Yan. Geometry-guided progressive nerf for generalizable and efficient neural human rendering. In *European Conference on Computer Vision*, pages 222–239. Springer, 2022. 2, 3

[5] Wei Cheng, Ruixiang Chen, Siming Fan, Wanqi Yin, Keyu Chen, Zhongang Cai, Jingbo Wang, Yang Gao, Zhengming Yu, Zhengyu Lin, and others. Dna-rendering: A diverse neural actor repository for high-fidelity human-centric rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19982–19993, 2023. 2, 6, 7, 8

[6] Arnab Dey, Di Yang, Rohith Agaram, Antitza Dantcheva, Andrew I. Comport, Srinath Sridhar, and Jean Martinet. GH-NeRF: Learning Generalizable Human Features with Efficient Neural Radiance Fields. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2812–2821, Seattle, WA, USA, 2024. IEEE. 2, 3

[7] Xiangjun Gao, Jiaolong Yang, Jongyoo Kim, Sida Peng, Zicheng Liu, and Xin Tong. MPS-NeRF: Generalizable 3D Human Rendering From Multiview Images. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1–12, 2024. 2, 3

[8] Chen Guo, Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. Vid2Avatar: 3D Avatar Reconstruction from Videos in the Wild via Self-supervised Scene Decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2

[9] Philipp Henzler, Jeremy Reizenstein, Patrick Labatut, Roman Shapovalov, Tobias Ritschel, Andrea Vedaldi, and David Novotny. Unsupervised Learning of 3D Object Categories from Videos in the Wild. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4698–4707, Nashville, TN, USA, 2021. IEEE. 2

[10] Liangxiao Hu, Hongwen Zhang, Yuxiang Zhang, Boyao Zhou, Boning Liu, Shengping Zhang, and Liqiang Nie. GaussianAvatar: Towards Realistic Human Avatar Modeling from a Single Video via Animatable 3D Gaussians. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[11] Shoukang Hu, Fangzhou Hong, Liang Pan, Haiyi Mei, Lei Yang, and Ziwei Liu. SHERF: Generalizable Human NeRF from a Single Image. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9318–9330, Paris, France, 2023. IEEE. 1, 2, 3, 4, 6, 7, 8

[12] Shoukang Hu, Tao Hu, and Ziwei Liu. Gauhuman: Articulated gaussian splatting from monocular human videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20418–20431, 2024. 2

[13] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 4

[14] Muhammed Kocabas, Jen-Hao Rick Chang, James Gabriel, Oncel Tuzel, and Anurag Ranjan. Hugs: Human gaussian splats. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 505–515, 2024. 2

[15] Youngjoong Kwon, Dahun Kim, Duygu Ceylan, and Henry Fuchs. Neural Human Performer: Learning Generalizable Radiance Fields for Human Performance Rendering. In *Advances in Neural Information Processing Systems*, pages 24741–24752. Curran Associates, Inc., 2021. 2, 3

[16] Chen Li, Jiahao Lin, and Gim Hee Lee. GHuNeRF: Generalizable Human NeRF from a Monocular Video. In *2024 International Conference on 3D Vision (3DV)*, pages 923–932. IEEE, 2024. 1, 2, 3, 6, 7

[17] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3383–3393, 2021. 7, 2

[18] Ruilong Li, Julian Tanke, Minh Vo, Michael Zollhöfer, Jürgen Gall, Angjoo Kanazawa, and Christoph Lassner. TAVA: Template-free Animatable Volumetric Actors. In *Computer Vision – ECCV 2022*, pages 419–436, Cham, 2022. Springer Nature Switzerland. 1, 2

[19] Kai-En Lin, Lin Yen-Chen, Wei-Sheng Lai, Tsung-Yi Lin, Yi-Chang Shih, and Ravi Ramamoorthi. Vision Transformer for NeRF-Based View Synthesis from a Single Input Image. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 806–815, Waikoloa, HI, USA, 2023. IEEE. 2

[20] 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)*, 34(6):248:1–248:16, 2015. Publisher: ACM. 2, 3, 7

[21] Mana Masuda, Jinhyung Park, Shun Iwase, Rawal Khirodkar, and Kris Kitani. Generalizable Neural Human Renderer, 2024. arXiv:2404.14199. 2, 3, 7

[22] 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. *Commun. ACM*, 65(1):99–106, 2022. 2, 3

[23] Xiao Pan, Zongxin Yang, Jianxin Ma, Chang Zhou, and Yi Yang. TransHuman: A Transformer-based Human Representation for Generalizable Neural Human Rendering. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3521–3532, Paris, France, 2023. IEEE. 2, 3

[24] Haokai Pang, Heming Zhu, Adam Kortylewski, Christian Theobalt, and Marc Habermann. ASH: Animatable Gaussian Splats for Efficient and Photoreal Human Rendering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1165–1175, Seattle, WA, USA, 2024. IEEE. 2

[25] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019. 2

[26] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable Neural Radiance Fields for Modeling Dynamic Human Bodies. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14314–14323, 2021. 1, 2

[27] Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3DGS-Avatar: Animatable Avatars via Deformable 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5020–5030, 2024. 2

[28] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common Objects in 3D: Large-Scale Learning and Evaluation of Real-life 3D Category Reconstruction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10881–10891, Montreal, QC, Canada, 2021. IEEE. 2

[29] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Viewset Diffusion: (0-)Image-Conditioned 3D Generative Models from 2D Data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8863–8873, 2023. 4, 6

[30] 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. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4688–4697, Nashville, TN, USA, 2021. IEEE. 2, 4

[31] Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. ARAH: Animatable Volume Rendering of Articulated Human SDFs. In *European Conference on Computer Vision*, 2022. 1, 2

[32] Jing Wen, Xiaoming Zhao, Zhongzheng Ren, Alexander G. Schwing, and Shenlong Wang. GoMAvatar: Efficient Animatable Human Modeling from Monocular Video Using Gaussians-on-Mesh. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2059–2069, Seattle, WA, USA, 2024. IEEE. 2

[33] Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. Vid2actor: Free-viewpoint animatable person synthesis from video in the wild. *arXiv preprint arXiv:2012.12884*, 2020. 5

[34] Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. HumanNeRF: Free-viewpoint Rendering of Moving People from Monocular Video. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16189–16199, New Orleans, LA, USA, 2022. IEEE. 2, 3, 5, 1

[35] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 2

[36] Zhengming Yu, Wei Cheng, Xian Liu, Wayne Wu, and Kwan-Yee Lin. MonoHuman: Animatable Human Neural Field from Monocular Video. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16943–16953, Vancouver, BC, Canada, 2023. IEEE. 2, 3, 4, 5, 6, 1

[37] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, Salt Lake City, UT, 2018. IEEE. 5, 7

HumMorph: Generalized Dynamic Human Neural Fields from Few Views

Supplementary Material

A. Additional Details Regarding the Method

Rendering equations. In Sec. 3 we define the density and color functions $\bar{\sigma}, \bar{\mathbf{c}}$ of a NeRF in observation space corresponding to pose Ω. We use volumetric rendering to synthesize the target image. Specifically, the color of each pixel u in the rendering is computed as follows:

$$C(u) = \sum_{i=1}^{M} \left[\prod_{j=1}^{i-1} (1 - \alpha_j) \right] \alpha_i \bar{\mathbf{c}}(x_i, \Omega), \tag{6}$$

$$\alpha_i = \left(1 - \exp\left(\bar{\sigma}(x_i, \Omega) \Delta x_i\right)\right), \tag{7}$$

where $x_i \in \mathbb{R}^3$ for $1 \leq i \leq M$ are points along ray r_u passing through pixel u in the image plane and $\Delta x_i = \|x_{i+1} - x_i\|_2$. Following HumanNeRF [34] we only sample the query points x_i inside a 3D bounding box estimated from the human skeleton in pose Ω.

Network training and loss functions. In a single training step, we render $G = 6$ patches P_i of size $H \times H$ with $H = 32$ which are used to compute $\mathcal{L}_{\text{LPIPS}}$ with a VGG backbone. We also have

$$\mathcal{L}_{\text{MSE}} = \frac{1}{G \cdot H^2} \sum_{i=1}^{G} \sum_{u \in P_i} \|C(u) - \hat{C}(u)\|_2^2, \tag{8}$$

where u is a pixel in patch P_i, $C(u)$ is the rendered color of u (as in Eq. (6)) and $\hat{C}(u)$ is the ground truth color of u. The deformation consistency component $\mathcal{L}_{\text{consis}}$ encourages consistency between the forward and backward deformations T_f, T_b (respectively; see Sec. 3.2). Recall that, intuitively, we should have $T_f(T_b(x_c, \Omega), \Omega) = x_c$ for a point x_c in canonical space and pose Ω. However, with the LBS deformation model, this condition is rarely satisfied and it depends on the motion weights W. Following MonoHuman [36], we include

$$\mathcal{L}_{\text{consis}} = \begin{cases} d & \text{if} \quad d \geq \eta, \\ 0 & \text{otherwise}, \end{cases} \tag{9}$$

where $d = \|x_p - T_f(T_b(x_p, \Omega), \Omega)\|_2^2$ for a point x_p in the observation space with pose Ω, in the loss function to regularize the motion weights. We compute $\mathcal{L}_{\text{consis}}$ on all query points used in volumetric rendering and use $\eta = 0.05$.

B. Additional Implementation Details

To better preserve the low-level information we concatenate the feature maps F_t with resized input images I_t. Hence, the pixel aligned features f_{pix} have dimensionality $32 + 3$. Both the motion weights and the voxel features VoluMorph submodules output a 32-dimensional voxel grid of size $32 \times 32 \times 32$. The output of the voxel features VoluMorph submodule is directly sampled to create f_{vox} features, which are also 32-dimensional. The output feature volume for motion weights correction is additionally projected (coordinate-wise) into $K = 24$ channels (one per joint) using a 1×1 convolution. The output of the convolution is the observation-conditioned correction $\Delta W(\mathcal{I})$ in log-space, which is combined with the initial estimate W_0 as follows

$$W(\mathcal{I}) = \text{softmax}(\Delta W(\mathcal{I}) + \log W_0). \tag{10}$$

Feature fusion module. Here we provide additional details on the implementation of the feature fusion module introduced in Sec. 3.1. Let x_c be a query point in canonical space. We describe how our feature fusion module computes the final feature f for a single x_c, which in practice is applied independently to each query point.

The feature vectors are first extended with positional encodings of spatial information on the query point x_c: its coordinates, the viewing direction on x_c in the target render transformed to canonical space and the vector from x_c to the nearest joint in the skeleton. We additionally append the motion weights W sampled at x_c, which serve as proxy information on the body shape. For the pixel-aligned features we also append the viewing direction (transformed to canonical space) under which the features were observed. The extended features are then aligned using two separate 2-layer MLPs with hidden dimensions 128 and output dimensions 64. The aligned features are processed by a transformer encoder layer with 4 attention heads and internal dimension 64.

The standalone spatial information on the query point (*i.e.* coordinates, viewing direction, vector to the nearest joint and sampled motion weights) is aligned with the features using a 2-layer MLP with hidden dimension 128 and output dimension 64. The final feature f is computed with a 4-head attention layer with internal dimension 64, where the (aligned) standalone spatial information on x_c is used as a query and the transformer encoder's outputs are used as keys/values.

Optimization. We optimize the parameters of our model using the Adam optimizer with learning rate 3×10^{-5} for the motion weights correction submodule and 2×10^{-4} for the rest. We additionally delay the optimization of the motion weights module until iteration 5K. We found that optimizing the motion weights end-to-end with the rest of the pipeline can in some cases introduce training instabilities,

which we contain by clipping the loss gradients to L2 norm of 10. We run our optimization for 300K iterations on 4 NVIDIA RTX 6000 GPUs, which takes about 5 days.

C. More Details on the Experiments

Selection of cameras. To reduce the computational cost of running our experiments, we subsample the camera sets of both datasets. For training and evaluation on the HuMMan dataset [2] we drop the cameras with indices 2 and 7 (the ones with the highest vertical position). For training on the DNA-Rendering dataset [5] we keep cameras with index c such that $c \equiv 1 \mod 4$ (12 cameras total), while for evaluation we use cameras with index c such that $c \equiv 1 \mod 8$ (6 cameras total). We use the same camera subset for training and evaluation of all models, including baselines.

Image resolution. During training of our method on both datasets we render the frames (patches) at $\frac{1}{4}$ of the original resolution, *i.e.* 480×270 for the HuMMan dataset and 512×612 for the DNA-Rendering dataset. We train SHERF [11] and GHuNeRF [16] on the HuMMan dataset using $\frac{1}{3}$ of the original resolution, *i.e.* 634×356 and using $\frac{1}{4}$ of the original resolution on the DNA-Rendering dataset, *i.e.* 512×612. We evaluate our method and the baselines using $\frac{1}{3}$ of the original resolution on the HuMMan dataset and using $\frac{1}{4}$ of the original resolution on the DNA-Rendering dataset.

Subsampling frames. We subsample the frames of all motion sequences in the DNA-Rendering dataset [5] to a maximum of 30 frames per sequence. We perform the subsampling at constant intervals across the full length of each sequence. We use the full sequences in the HuMMan dataset [5].

Selection of observed frames. During training our models are provided with $T = 2$ observed frames, which are uniformly sampled from the full motion sequence (without the target frame). The observed frames are sampled from the same camera as the target frame. During monocular training, SHERF [11] (Mo) is provided with a random frame (except the target frame) from the same camera as the target frame. GHuNeRF during training is supplied with 4 randomly sampled observed frames.

For evaluation, we split the motion sequences approximately in half at frame $\lfloor \frac{T+1}{2} \rfloor$, where T is the sequence length, and provide observations from the first half, while we measure the quality of reconstruction on the frames from the second half. Specifically, when T is the motion sequence length (in frames) the observed frames are selected based on the table below:

Num. observ.	Indices of observed frames
1	0
2	0, $\lfloor T \cdot \frac{1}{4} \rfloor$
3	0, $\lfloor T \cdot \frac{1}{4} \rfloor$, $\lfloor T \cdot \frac{3}{8} \rfloor$
4	0, $\lfloor T \cdot \frac{1}{4} \rfloor$, $\lfloor T \cdot \frac{3}{8} \rfloor$, $\lfloor T \cdot \frac{1}{8} \rfloor$

Note that, as SHERF [11] only accepts a single observed frame, in the quantitative experiments it is provided with the first frame of each sequence. We provide qualitative results of SHERF given other observed frames. In the qualitative results, the index of the observed frame number i is the last entry of row i in the table above.

C.1. Estimated Body Shape and Pose Parameters

To obtain the estimated SMPL [20] pose and shape parameters we use an off-the-shelf HybrIK [17] model for each frame in the motion sequences independently. We then retrain both our models, SHERF (Mo) and GHuNeRF with a mixture of accurate and estimated parameters. At each training step, we use the estimated parameters with probability p or the accurate parameters with probability $1 - p$, where p increases linearly throughout the training from 0 at the beginning to 0.75 at roughly half of the training process.

When using estimated body parameters, during both training and evaluation, we provide the models with the estimated body shape and pose parameters for the observed frames. However, we always provide accurate pose parameters for the target frames, is motivated by the practical scenario, where pose parameters are either transferred from a different motion or generated with a separate model. Furthermore, since the target frame is not known in practice, estimating the target pose is not meaningful. In contrast, the body shape is always assumed to be unknown and therefore has to be estimated. Note that, in this experiment, we use the ground-truth camera poses for both models.

D. Additional Results

Fig. 6 and Fig. 7 show an extended qualitative comparison between our method with $T \in \{1, 2, 3, 4\}$ observed views, SHERF [11] (Mo) and GHuNeRF+ [16] on the HuMMan [2] and DNA-Rendering [5] datasets, respectively. As discussed in Sec. 4.4, SHERF frequently struggles to match the observed view to the underlying geometry, which results in incorrect renders in novel poses with 'phantom' limbs (typically arms) imprinted on the torso (see the top 2 subjects in Fig. 6 and top two subjects in Fig. 7). In most cases, this problem is observed regardless of which view SHERF observes – as long as the arms of the subject overlap with their body in the observed view, they are usually imprinted *somewhere* on the torso. While our method sometimes displays a similar pattern when it observes a single view, it matches the geometry correctly and resolves this issue when provided with 2 (or more) observations. To achieve that it has to combine information from available observations while resolving occlusions and/or making use of the prior (*e.g.* smoothness) as information from any of the observations alone is not enough to eliminate the artefacts (which is demonstrated by SHERF results).

D.1. Extended Results with Estimated Body Shape and Pose Parameters

Fig. 8 and Fig. 9 show an extended qualitative comparison of our method with $T \in \{1, 2, 3\}$ observed views to SHERF (Mo) and GHuNeRF+, on the DNA-Rendering and HuMMan datasets (respectively) when using estimated body shape and pose parameters. The renders produced by our method are significantly sharper compared to SHERF and, in contrast to the baselines, our method correctly replicates most of the details found in the observed views. Moreover, our method generates fewer artifacts compared to SHERF when filling in missing information using prior (see *e.g*. the legs and shoes of all subjects in Fig. 8).

D.2. Video Qualitative Results

We provide video versions of Fig. 6, Fig. 7, Fig. 8 and Fig. 9 in the attached files named `fig_x_video.mp4`, where x is the figure number.

E. Broader impact

We acknowledge that our method could potentially have negative societal if misused to create fake images or videos of real people. Any public deployments of this technology should be done with great care to ensure that ethical guidelines are met and with safeguards in place. We plan to release our code publicly to aid with counter-measure analysis.

Figure 6. Extended qualitative comparison between our method SHERF (Mo) and GHuNeRF+ on the HuMMan dataset. Numbers in parentheses indicate the range of observed views supplied to the respective models. Best viewed in color and zoomed in for details.

Figure 7. Extended qualitative comparison between our method, SHERF (Mo) and GHuNeRF+ on the DNA-Rendering dataset. Numbers in parentheses indicate the range of observed views supplied to the respective models. Best viewed in color and zoomed in for details.

Figure 8. Extended qualitative comparison between our method, SHERF (Mo) and GHuNeRF+ on the DNA-Rendering dataset when using estimated body shape and pose parameters. Numbers in parentheses indicate the range of observed views supplied to the respective models. Best viewed in color and zoomed in for details.

Figure 9. Extended qualitative comparison between our method, SHERF (Mo) and GHuNeRF+ on the HuMMan dataset when using estimated body shape and pose parameters. Numbers in parentheses indicate the range of observed views supplied to the respective models. Best viewed in color and zoomed in for details.