

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 $\bar{\sigma}, \bar{c}$ 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:

$$C(u) = \sum_{i=1}^M \left[\prod_{j=1}^{i-1} (1 - \alpha_j) \right] \alpha_i \bar{c}(x_i, \Omega), \quad (7)$$

$$\alpha_i = (1 - \exp(\bar{\sigma}(x_i, \Omega) \Delta x_i)), \quad (8)$$

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 [37], we only sample the query points x_i inside a 3D bounding box estimated from the human skeleton in pose Ω .

Unprojection and undeformation. See Fig. 6 for an illustration of the unprojection and undeformation operation defined by Eq. (1).

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

$$\mathcal{L}_{\text{MSE}} = \frac{1}{G \cdot H^2} \sum_{i=1}^G \sum_{u \in P_i} \|C(u) - \hat{C}(u)\|_2^2, \quad (9)$$

where u is a pixel in patch P_i , $C(u)$ is the rendered color of u (as in Eq. (7)) 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 [39], we include

$$\mathcal{L}_{\text{consis}} = \begin{cases} d & \text{if } d \geq \eta, \\ 0 & \text{otherwise,} \end{cases} \quad (10)$$

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 .

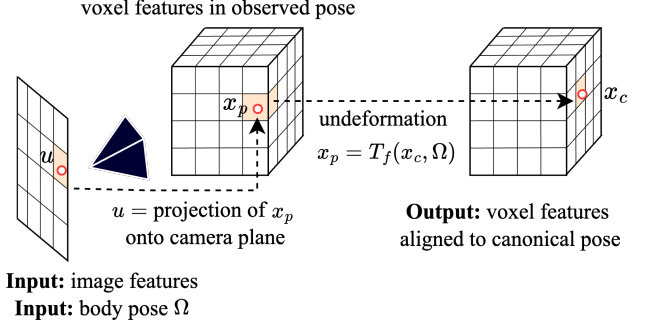


Figure 6. Diagram of the unprojection and undeformation operation defined by Eq. (1).

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 along the X, Y axes and size 16 along the Z axis, which corresponds to the human body shape. 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

$$W(\mathcal{I}) = \text{softmax}(\Delta W(\mathcal{I}) + \log W_0). \quad (11)$$

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 di-

mension 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 2×10^{-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 7.5. We run our optimization for ~ 300 K 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 $c \equiv 1 \pmod 4$ (12 cameras total), while for evaluation we use cameras with index c such that $c \equiv 1 \pmod 8$ (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 [18] 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 [23] pose and shape parameters, we use an off-the-shelf HybrIK [19] model for each frame in the motion sequences independently. We then re-train our models, SHERF (Mo) and GHuNeRF(+) using 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, which 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 from the observed frames. Note that, in this experiment, we use the ground-truth camera poses for both models.

D. Additional Results

Note that a fair comparison to the related GNH [24] is not currently possible since the code and models have not yet been made publicly available.

Method	Accurate body parameters			Estimated body parameters		
	PSNR \uparrow	LPIPS* \downarrow	SSIM \uparrow	PSNR \uparrow	LPIPS* \downarrow	SSIM \uparrow
SHERF (Mo)	26.95	44.12	0.9615	24.23	61.44	0.9450
SHERF (MV)	26.35	43.68	0.9603	-	-	-
GHuNeRF+ (1 obs.)	23.89	44.00	0.9527	23.17	50.24	0.9480
GHuNeRF+ (2 obs.)	23.97	43.72	0.9530	23.27	49.96	0.9483
GHuNeRF+ (4 obs.)	24.00	43.66	0.9531	23.31	49.86	0.9485
GHuNeRF+ (8 obs.)	24.01	43.64	0.9531	23.32	49.85	0.9485
GHuNeRF (1 obs.)	23.87	63.01	0.9474	23.30	68.84	0.9425
GHuNeRF (2 obs.)	23.88	62.98	0.9474	23.34	68.76	0.9427
GHuNeRF (4 obs.)	23.89	63.02	0.9474	23.36	68.76	0.9427
GHuNeRF (8 obs.)	23.88	63.06	0.9474	23.36	68.75	0.9427
Ours (1 observed)	26.70	33.43	0.9638	25.08	42.28	0.9553
Ours (2 observed)	27.38	30.20	0.9670	25.33	40.93	0.9568
Ours (3 observed)	27.64	28.88	0.9683	25.40	40.53	0.9573
Ours (4 observed)	27.66	28.72	0.9685	25.40	40.52	0.9574

Table 3. Extended quantitative comparison of our method with SHERF [11] and GHuNeRF [18] with various numbers of observed views on the HuMMan [2] dataset. SHERF (Mo) is trained in our monocular framework, and SHERF (MV) is the official model from [11] (multi-view trained). GHuNeRF+ contains the added LPIPS loss. $\text{LPIPS}^* = \text{LPIPS} \times 10^3$.

Method	Accurate body parameters			Estimated body parameters		
	PSNR \uparrow	LPIPS* \downarrow	SSIM \uparrow	PSNR \uparrow	LPIPS* \downarrow	SSIM \uparrow
SHERF (Mo)	28.49	48.22	0.9635	26.93	61.97	0.9536
SHERF (MV)	27.78	49.52	0.9614	-	-	-
GHuNeRF+ (1 obs.)	26.59	53.10	0.9578	26.19	56.46	0.9547
GHuNeRF+ (2 obs.)	26.69	52.92	0.9581	26.28	56.16	0.9550
GHuNeRF+ (4 obs.)	26.70	52.93	0.9582	26.31	56.11	0.9552
GHuNeRF+ (8 obs.)	26.71	52.94	0.9583	26.32	56.09	0.9552
GHuNeRF (1 obs.)	27.59	70.05	0.9562	27.12	74.74	0.9520
GHuNeRF (2 obs.)	27.72	69.76	0.9566	27.24	74.59	0.9524
GHuNeRF (4 obs.)	27.78	69.71	0.9568	27.28	74.54	0.9527
GHuNeRF (8 obs.)	27.81	69.71	0.9569	27.31	74.58	0.9527
Ours (1 observed)	27.86	40.25	0.9630	27.00	47.21	0.9575
Ours (2 observed)	28.35	38.03	0.9651	27.31	45.45	0.9592
Ours (3 observed)	28.63	36.88	0.9663	27.45	44.79	0.9599
Ours (4 observed)	28.65	36.89	0.9664	27.46	44.76	0.9601

Table 4. Extended quantitative comparison of our method with SHERF [11] and GHuNeRF [18] with various numbers of observed views on the DNA-Rendering [5] dataset. SHERF (Mo) is trained in our monocular framework, and SHERF (MV) is trained in the multi-view framework of [11]. GHuNeRF+ contains the added LPIPS loss. $\text{LPIPS}^* = \text{LPIPS} \times 10^3$.

See Tab. 3 and Tab. 4 for an extended quantitative comparison to SHERF [11] (monocular – Mo and multi-view – MV), GHuNeRF [18] and GHuNeRF+ on HuMMan [2] and DNA-Rendering [5]. SHERF (MV) is trained in the original framework of [11], *i.e.* the observed view is in the same pose as the target view but captured from a different viewpoint. Note that SHERF (MV) is still conditioned on a single observed view. For ‘SHERF (MV)’ on the HuMMan dataset we use the official models of [11], while for the DNA-Rendering dataset we retrain it using the multi-view training framework.

Fig. 7 and Fig. 8 show an extended qualitative comparison between our method with $T \in \{1, 2, 3, 4\}$ observed views, SHERF [11] (Mo) and GHuNeRF [18] 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. 7 and top two subjects in Fig. 8). 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 artifacts (which is demonstrated by SHERF results).

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

Fig. 9 and Fig. 10 show an extended qualitative comparison of our method with $T \in \{1, 2, 3, 4\}$ observed views to SHERF (Mo) and GHuNeRF, on the HuMMan and DNA-Rendering 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. 10).

D.2. Video Qualitative Results

We provide video versions of Fig. 7, Fig. 8, Fig. 9 and Fig. 10 in the attached files named `fig_x-video-i.mp4`, where x is the figure number and i is the sequence number (from top to bottom).

E. Broader impact

We acknowledge that our method could potentially have a negative societal impact 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 will release our code publicly to aid with countermeasure analysis.



Figure 7. 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 8. 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 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.



Figure 10. 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.