GHNeRF: Learning Generalizable Human Features with Efficient Neural Radiance Fields

Arnab Dey[1*] Di Yang[2] Rohith Agaram[3] Antitza Dantcheva[2] Andrew I. Comport[1]
Srinath Sridhar[4] Jean Martinet[1]

[1]I3S-CNRS/Université Côte d'Azur [2]Inria Center at Université Côte d'Azur
[3]IIIT Hyderabad [4]Brown University

Abstract

Recent advances in Neural Radiance Fields (NeRF) have demonstrated promising results in 3D scene representations, including 3D human representations. However, these representations often lack crucial information on the underlying human pose and structure, which is crucial for AR/VR applications and games. In this paper, we introduce a novel approach, termed GHNeRF*, designed to address these limitations by learning 2D/3D joint locations of human subjects with NeRF representation. GHNeRF uses a pre-trained 2D encoder streamlined to extract essential human features from 2D images, which are then incorporated into the NeRF framework in order to encode human biomechanic features. This allows our network to simultaneously learn biomechanic features, such as joint locations, along with human geometry and texture. To assess the effectiveness of our method, we conduct a comprehensive comparison with state-of-the-art human NeRF techniques and joint estimation algorithms. Our results show that GHNeRF can achieve state-of-the-art results in near real-time. The project website: arnabdey.co/ghnerf.github.io.*

1. Introduction

Developing a realistic virtual human model is pivotal for achieving natural experiences in Augmented Reality (AR) / Virtual Reality (VR) applications and interactive games. Moreover, creating custom photo-realistic virtual character from a sparse set of 2D images is one of the core challenges for AR/VR applications. Traditionally, this process has involved the use of elaborate multiview capture systems, which incorporate extensive camera arrays and body markers [15, 34], to create human models with the underlying skeleton structure. These conventional methods predominantly utilize mesh representations, which are inherently constrained in terms of resolution and quality. The underlying structures in these models are typically represented with parametric Skinned Multi-Person Linear (SMPL) models [22] derived from body marker positioning.

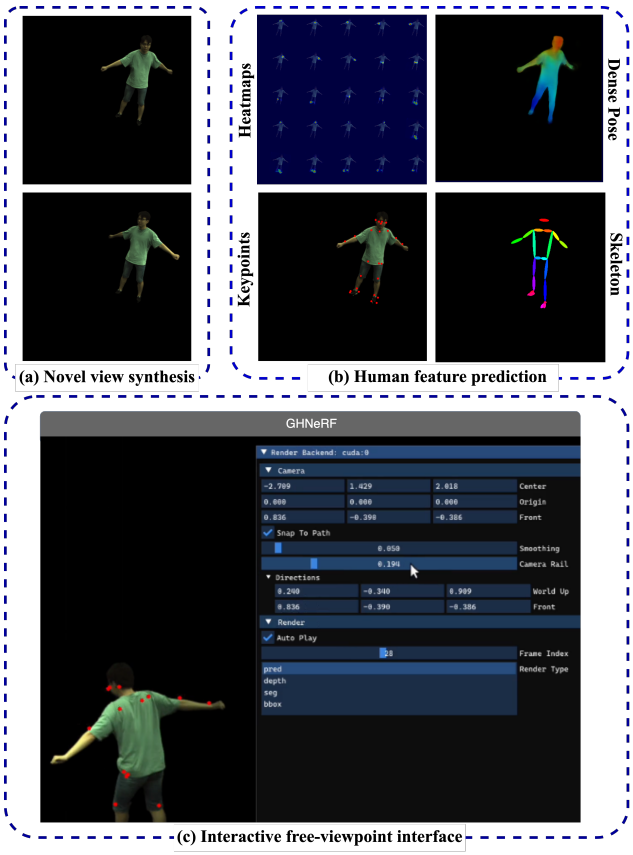

Figure 1. In this work we propose **GHNeRF**, it can simultaneously learns both neural radiance fields and human features from sparse images. (a) shows high quality novel-view renderings. (b) shows generalizable human features (keypoints, dense pose, etc.) estimated by GHNeRF. (c) present interactive tool to render free-viewpoint videos of novel-view and human features.

*Contact email: adey@i3s.unice.fr

Recent advancements in Neural Radiance Fields (NeRF) have demonstrated remarkable potential in generating photorealistic virtual human avatars from mere 2D images [14, 21, 37]. However, existing NeRF-based approaches fall short in providing critical structural biomechanical attributes, crucial for various applications such as AR/VR, 3D animation, human performance analysis, and the medical field. To bridge this gap, we introduce a Generalizable Human feature NeRF (GHNeRF), an end-to-end framework for learning generalizable human NeRF with biomechanic features. Human biomechanics refers to the study of human movement focusing musculoskeletal system, comprising bones, muscles, ligaments, and joints [24]. Within the scope of this paper, the term 'biomechanical features' specifically refers to the skeleton and joints integral to this system. Deviating from previous methods such as PixelNeRF [45], which used a 2D encoder to learn generalizable NeRF for view synthesis, our approach utilizes 2D deep feature extractors to simultaneously learn human features with generalizable NeRF models. Here, we demonstrate that it is possible to learn 3D human features from 2D images using the NeRF architecture. The GHNeRF predicts human features, such as heatmaps, facilitating 2D/3D joint estimation for novel views, which are applied to various downstream applications. We highlight that while we focussed on the *joint prediction*, the architecture can be used to learn other biomechanic properties, such as dense pose and body part segmentation.

Our methodology adopts 2D encoders similar to previous methods [43, 45] aimed at generating pixel-aligned human features from images. For this purpose, we compare two types of encoder inspired by previous state-of-the-art pose estimation algorithms. GHNeRF determines heatmaps corresponding to each joint, along with the color and volume density for each 3D query point. The input for the MLP are pixel-aligned features from encoder, as well as view direction. The heatmaps are generated using volume rendering similar to rendering color in NeRF. We use an efficient and generalizable NeRF architecture as a backbone similar to the one presented by Lin *et al.* [21] that allows for near real-time inference.

To evaluate GHNeRF, we present the result of keypoint estimation tasks using two popular datasets. To our knowledge, our method is the first to provide human biomechanic features from NeRF. Our contributions are summarized as follows.

- We introduce GHNeRF, a novel generalizable NeRF architecture capable of accurately estimating 2D/3D human keypoints.
- GHNeRF demonstrates the ability not only to predict human keypoints but also to estimate complex human features, such as dense poses. This capability can also be achieved through the distillation of SoTA pose estimation algorithms.
- We provide a generalizable approach for predicting human feature, photometric, and geometric representations from 2D sparse images, applicable in interactive, real-time applications.
- We conduct extensive experimental analyses across various types of human images using two distinct datasets to validate the applicability and versatility of GHNeRF .

2. Releted works

The proposed GHNeRF uses sparse multiview images of different humans to learn a generalizable NeRF representation that can also produce a consistent 3D human feature without any prior supervision during inference time. In the following, work related to this research will be discussed.

2.1. NeRF for 3D representation

In recent years, the NeRF-based method has gained significant popularity for the visual quality of 3D scene representations. NeRF [26] represents 3D scenes using MLP by mapping 3D coordinates and 2D view directions to density and color. The original paper [26] and the following research work [1, 2, 25, 30, 35] showed the effectiveness of the neural field compared to other classical methods for representing 3D and 4D scenes. The works [7, 21, 28, 44] address the long training and inference time of the NeRF by using faster sampling techniques, voxel representation, and hash encoding. Another limitation of NeRF-based methods is that they are scene specific, PixelNeRF [45] showed that NeRF models can be generalized by conditioned NeRF on input image. More recently, FeatureNeRF [43] learned deep features using pre-trained vision foundation models for downstream applications such as semantic segmentation and key point transfer. Several methods [19, 38, 50] extended the NeRF's ability by learning scene properties, for example, semantic segmentation of the scene. However, most of the previous work focuses on scene features, such as segmentation. Our work differs from them by learning human biomechanic features with NeRF.

2.2. NeRF for human representation

In recent research, Hu et al. [14] generated genralizable and animatable human NeRF models from a single input image. Although they achieved great results, their method relies on the SMPL parameters as input along with image, which is difficult to obtain in a real-world scenario. Similarly, GM-NeRF [5] used the SMPL model to learn a generalizable human NeRF model. Several works [16, 41, 46] generated NeRF models of human in canonical T-pose (example SMPL [23] T-pose) then map it to a posed space. Similarly, [17, 33, 37, 40] uses pre-existing skeleton data or pose estimator or information from the SMPL model [23]

to reconstruct novel views or novel poses. As an example, A-NeRF [37] employs off-the-shelf pose estimators to initialize their model, while our generalizable method does not require any pose initialization. In this paper, we predict human biomechanic features, such as joint information, directly from 2D images without any supervision.

2.3. Human pose estimation

Human pose estimation has been a long-standing problem in computer vision for decades. Most state-of-the-art approaches for 2D human pose estimation employ 2D CNN architectures for a single image in a strongly supervised setting [3, 6, 9, 13, 18, 29, 39]. For 3D pose estimation, [27, 36] focus on end-to-end reconstruction by directly estimating 3D poses from RGB images without intermediate supervision. [48] applies GCNs for regression tasks, especially 2D to 3D human pose regression. [31] demonstrates that 3D poses in video can be effectively estimated with a fully convolutional model based on dilated TCNs over 2D keypoint sequences. Among these methods, [27, 29, 36, 39] have first incorporated a person detector, followed by the estimation of the joints and then the computation of the pose for each person – however the detection speed is proportional to the number of people in the image. Bottom-up methods such as [3, 6, 18] detect joints via heatmaps and associate body parts, but struggle with occluded or truncated body parts. Our approach integrates an encoder with NeRF to directly estimate heatmaps from 3D NeRF features, enhancing accuracy in predicting non-visible regions in 2D.

3. Method

We present GHNeRF, a unified framework for learning generalizable human features with the efficient NeRF architecture. First, we present an introduction to NeRF and its generalizable variants. Then in Section 3.2, we outline the feature extraction process and explain how to learn human features with NeRF in Section 3.3. Finally, we provide details of keypoint extraction in Section 3.4 .

3.1. Preliminaries

Neural Radiance Fields (NeRF) learn 3D scene representations using a multilayer perceptron (MLP). The input to the MLP consists of 3D coordinates $x = (x, y, z)$ and the view direction $\mathbf{d} = (\theta, \phi)$. The outputs are color, $c = (r, g, b)$ and density(σ). It can be represented as: $F(x,\ \mathbf{d}) \rightarrow (c, \sigma)$ then volume rendering is used to generate the final pixel colors from the output. To predict images, first, 3D points are sampled along the rays $r(t) = o + t\mathbf{d}$ passing through each pixel, with o the camera center and $\mathbf{d}$ the direction of the ray. The color and density of the samples are predicted using an MLP as discussed before. The final color of the pixel C of a camera ray $r(t) = o + t\mathbf{d}$ can be calculated as:

$$C(r) = \int_{t_n}^{t_f} T(t)\tau(r(t))c(r(t), \mathbf{d})dt, \tag{1}$$

where $T(t) = exp(-\int_{t_n}^{t} \tau(r(s))ds)$. The function $T(t)$ denotes the accumulated transmittance along the ray from t_n to t, t_n and t_f is near and far bound of the ray. In practice, the color $\widehat{C}(r)$ is estimated by obtaining discrete samples along the ray, and the integral is approximated using numerical quadrature techniques.

In case of generalizable NeRF, the NeRF models are conditioned on the input image I:

$$\begin{aligned} \sigma(x, I) &= g_\sigma(x, f(I)_{\pi(x)}) \\ c(x, \mathbf{d}, I) &= g_c(x, \mathbf{d}, f(I)_{\pi(x)}), \end{aligned} \tag{2}$$

where g_σ and g_c are two MLPs that predict density and color, f is an image encoder and π is a projection function that projects x into the image plane using the known pose and intrinsic. The image passes through an encoder to generate features, then for each query point x, the corresponding pixel-aligned features [45] $f(I)_{\pi(x)}$ are concatenated with the positional encoding of the point before inputting into the NeRF model. Similarly, ENeRF [21] extracts multiscale image features from a CNN- based encoder, and then the encodings are also used as input and to create a cost volume. Given the cost volume, a 3D CNN generates a depth probability volume, which is used to predict the depth probability of a pixel. ENeRF uses depth probabilities to sample points close to the surface, resulting in fewer samples and faster training and inference time.

3.2. Feature extraction

We propose a new architecture to generalize human NeRF with the underlying biomechanic features. The original NeRF model predicts the color c and the density σ for each query point $x = (x, y, z)$, while the most generalizable NeRF models are conditioned on input images. We take inspiration from previous generalizable methods [21, 43, 45], and we use two different encoders: one to generate a human feature and the other for multiscale image features similar to [21]. Each query point is projected on the input images, and then the pixel-aligned image features from each image are combined using a pooling operator [21] that is denoted as $f_{img} = \psi(f_1, ..., f_N)$ where f_N represents the feature of the N^{th} image. Multiscale features are also used to generate voxel-aligned features similar to [21] denoted by f_{voxel}. Subsequently, we introduce a second encoder to encode human features. It has been demonstrated that human features extracted from Transformer-based encoder [8] pre-trained on ImageNet are more effective in generalizing human pose estimation [42, 49] compared to CNN-based features [10, 36]. In this work, we compare both types of

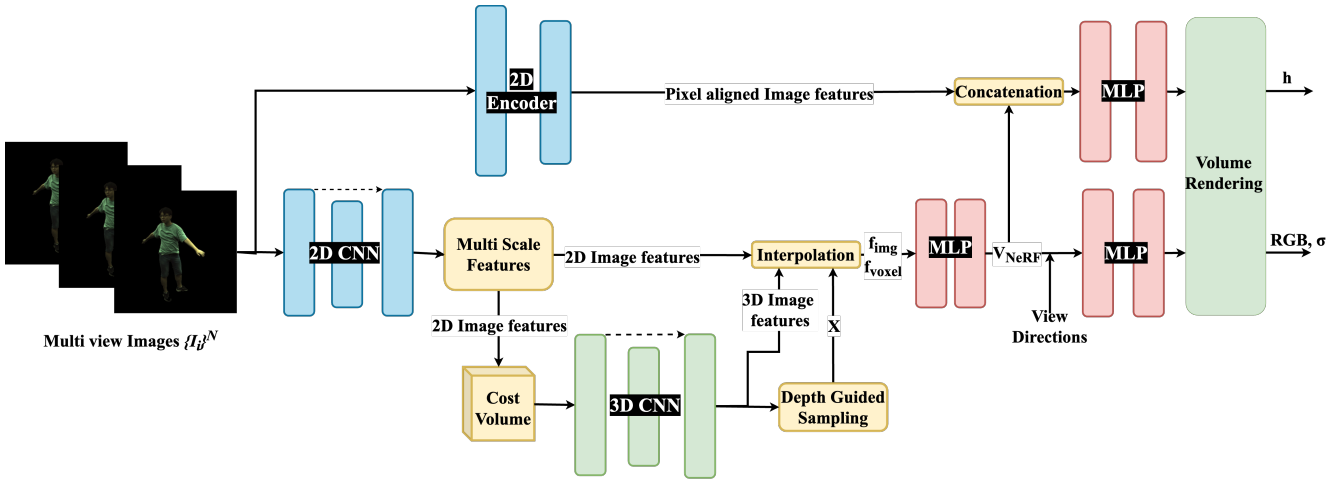

Figure 2. **Overview of the GHNeRF pipeline:** Given an input image I, human features $f_{\mathbf{h}}$ and multi-resolution image features f_{img} can be extracted using a 2D image encoder and a 2D CNN respectively. Subsequently, f_{img} is used to form a cost volume for depth prediction. The predicted depth is used for depth-guided sampling to reduce the number of samples along the ray. For each 3D sample point x along the ray, we combine image and voxel features to input an MLP g_{NeRF}, generating the intermediate NeRF feature V_{NeRF}. Finally, the intermediate NeRF feature V_{NeRF} and the human feature $f_{\mathbf{h}}$ are concatenated and fed into a smaller MLP g_h to produce heatmaps. Furthermore, V_{NeRF} and the view direction $\mathbf{d}$ are combined in another MLP g_c to derive color c. The final pixel color and heatmaps are generated using volume rendering technique.

encoders and select the vision transformer encoder [8] to extract more effective features for human pose estimation. Specifically, we use a pre-trained vision transformer to extract a higher-dimension feature vector $\mathbf{h}$ following [4]. For each query point x, we combine all pixel-aligned human features $f_h = \psi(\mathbf{h}_1, ..., \mathbf{h}_N)$ from input images with a pooling operator.

3.3. Learning human features with NeRF

Gerneralizable NeRF models predict color c and σ for any query points, GHNeRF extends the generalizable NeRF models to predict additional features, in this case human joint locations. Although we have extracted features from images, we still need to incorporate them with NeRF, in order to output 3D consistent human features from NeRF. In this work, we learn intermediate NeRF features $V_{NeRF}(x, I)$ similar to [43]. Then we use a number of small MLPs to predict other outputs from the intermediate NeRF feature:

$$\begin{aligned} V_{NeRF}(x, I) &= g_{NeRF}(f_{img}, f_{voxel}) \\ \sigma(x, I) &= g_{\sigma}(V_{NeRF}(x, I)) \\ c(x, \mathbf{d}, I) &= g_c(V_{NeRF}(x, I), \mathbf{d}) \\ h(x, I) &= g_h(V_{NeRF}(x, I), f_{\mathbf{h}}). \end{aligned} \tag{3}$$

The color is predicted using a smaller MLP g_c that takes the intermediate NeRF input feature V_{NeRF} and the view direction as input. An additional branch predicts human joint locations as heatmaps h from NeRF features. We take intermediate NeRF features before outputting color and density and concatenate with human feature $f_{\mathbf{h}}$ and pass it through a smaller MLP g_h that outputs heatmaps as feature vector $h \in \Re^J$ where J is the number of joints. We can aggregate these feature vectors along the rays similar to color using volume rendering:

$$\widehat{H}(r) = \sum_{i=1}^{N} T_i(1 - exp(-\tau_i \delta_i))h_i, \tag{4}$$

where $h_i = g_h(V_{NeRF}(x, I), f_h)$ and $V_{NeRF}(x, I)$ denotes intermediate NeRF features. The network is optimizing using a set of human images in a random pose and appearance with known camera parameters. The proposed method is optimized using photometric and feature loss. The photometric loss l_{col} is calculated using the mean squared error between the predicted and the ground-truth color. We also add perceptual loss l_{perc} to image patches similar to [21]. Feature loss l_{heat} is the mean square error between the predicted feature and the ground-truth feature in this case heatmaps. The final loss function can be represented as:

$$l = l_{col} + \lambda_p l_{perc} + \lambda_h l_{heat}$$

where λ_p, λ_h weighting coefficients. During training, when ground truth features are not present, our method represents a student network, which can learn heatmaps through distillation of advanced heatmap-based pose estimation algorithms. The pose estimation algorithm [3] acts as a teacher

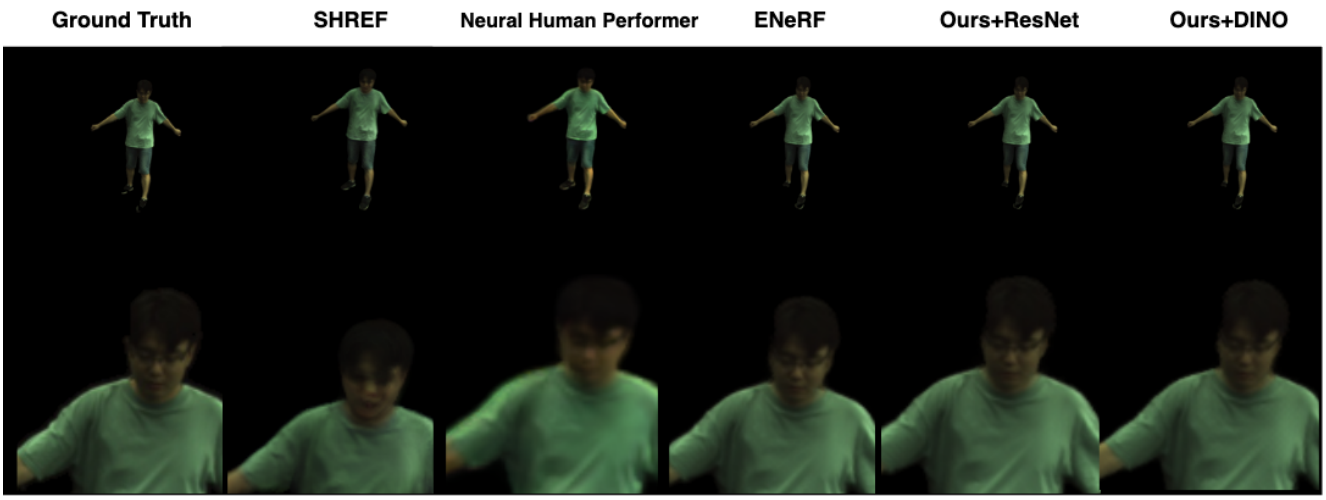

Figure 3. Qualitative comparison of generalization results on ZJU_MoCap unseen test sequence.

network with the ability to predict heatmaps, thus guiding our student network in its heatmap prediction task.

3.4. Keypoints extraction

The 2D keypoint locations are estimated from the predicted heatmaps generated by NeRF. We calculate the 2D keypoints in a similar way to OpenPose [3]. A Gaussian filter is applied to the heatmaps, and then each channel is converted to a binary map by applying a threshold. Connected regions are created from binary maps, and the peak value within that region is calculated. The pixel with the peak value is then outputted as the 2D keypoint. To extract the 3D keypoints, we query sample points from a 3D volume around the subject and extract a volumetric heatmap. The 3D keypoints are calculated from the volumetric heatmap in a similar way as in the 2D keypoints.

4. Experimental Results

We conducted a thorough evaluation of the ability of our model to learn human features, particularly to estimate human joint locations. We carried out extensive experiments on two distinct datasets and compared our results with those of other leading human NeRF techniques.

4.1. Experimental Setting

Datasets: We trained our model to be applicable to various types of human image using two different datasets, namely ZJU_MoCap [32] and RenderPeople [14]. Both datasets are focused on humans and contain dynamic sequences of different individuals performing various activities. The ZJU_MoCap dataset contains real images, while the RenderPeople dataset contains simulated images. ZJU_MoCap includes 9 dynamic sequences (images, masks, camera parameters, and 2D/3D joint locations) of 9 different individuals performing 9 different actions. We randomly divided 6 sequences for training and 2 for testing and removed one sequence due to missing frame data. For RenderPeople, we randomly chose 440 sequences for training and 60 for testing.

Baseline: We predominantly compare GHNeRF with other methods based on dynamic human NeRF. Although such methods are generalizable, none are capable of generating human features. We have extended ENeRF[21] to output heatmaps by adding an additional output branch and reported its performance as a baseline for the joint estimation task.

Implementation details: We employ ENeRF as the base generalizable NeRF architecture due to its efficiency and generalizability, and proceeded to modify it to generate generalizable human features. We employed two distinct encoders, ResNet [12] and DINO [4], in accordance with the most recent pose estimation techniques. In our experiments, we set the number of input source views to 2. We implemented our generalizable NeRF model using PyTorch. We trained the models with an RTX 3090 GPUs, using the Adam optimizer with an initial learning rate of $5e^{-4}$. We halved the learning rate every 50k iterations, and the model generally converged after about 200k iterations, taking about 18 hours. The weights of different losses are $\lambda_h = 0.5$ and $\lambda_p = 0.01$. For more information on the network architecture and other implementation details, see the Supplementary Material.

Metrics: We employed five different metrics to evaluate the predicted RGB image, heatmaps, and joint estimation quality. Peak Signal-to-Noise Ratio (PSNR in dB): To compare the quality of the RGB reconstruction, the higher is better; Structural Similarity Index (SSIM): To compare im-

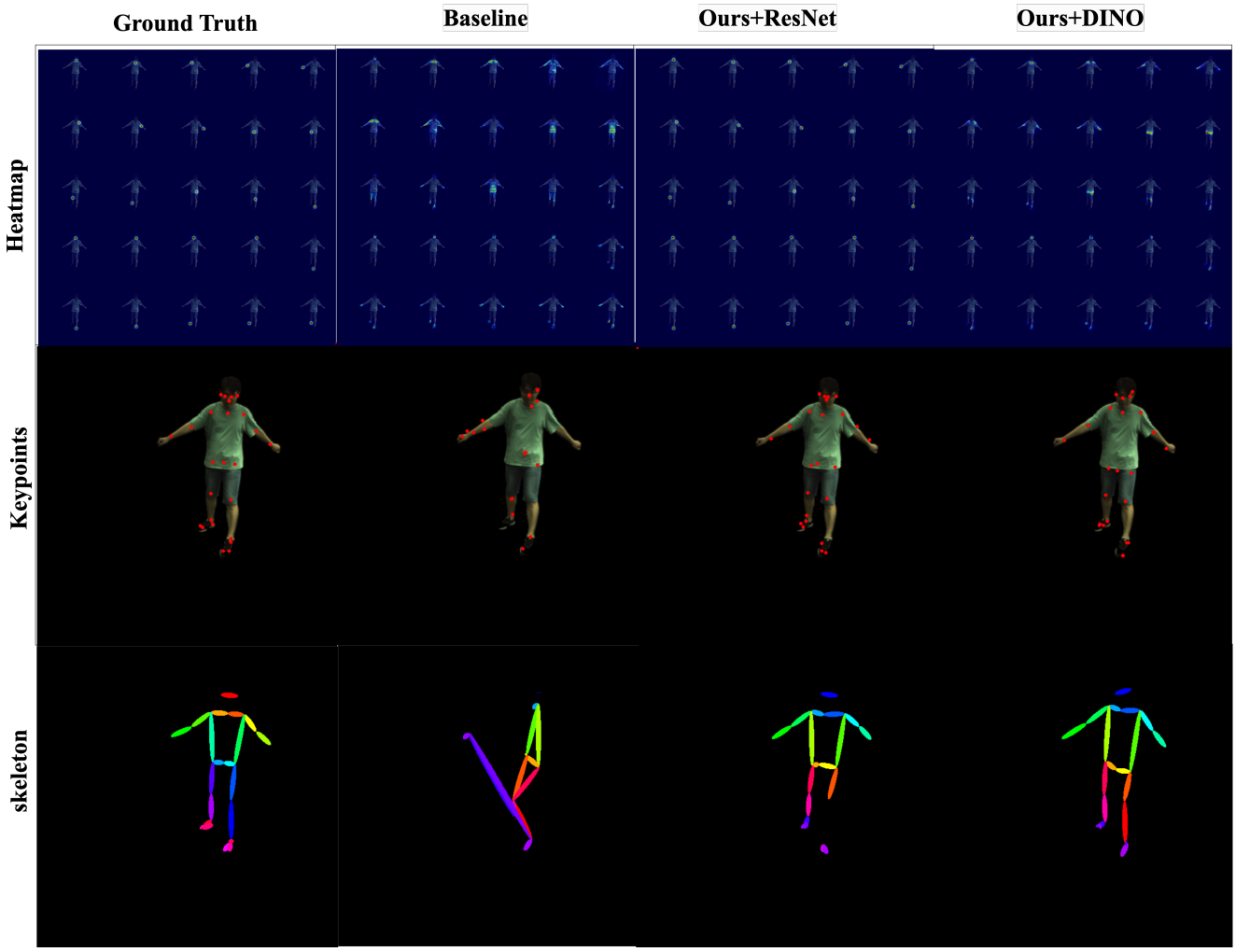

Figure 4. Qualitative result of keypoint estimation on ZJU_MoCap dataset.

age quality in the reconstructed image, the higher is better; Learned Perceptual Image Patch Similarity (LPIPS) [47]: the distance between the patches of the image, the lower means that the patches are more similar; Mean Squared Error (MSE): Mean squared distance between ground truth heatmap and predicted heatmap, lower the better; Percentage of correct keypoints (PCK): Measures whether the predicted keypoints and the true joint are within a certain distance threshold. We use PCK@0.2: Distance between the predicted and true joint $< 0.2\times$ torso diameter.

4.2. Performance on novel view synthesis and joint estimation

We compared our method with recent generalizable NeRF-based methods on dynamic scenes, Table: 1 lists the quantitative result on ZJU_MoCap dataset, which shows our method achieves state-of-the-art performance, while additionally estimating human joints. To establish a baseline, we incorporated an additional heatmap breach into ENeRF. The experiments show that our method maintains the same level of performance in novel-view synthesis compared to state-of-the-art ENeRF [21] but performs significantly better in joint estimation compared to the baseline ENeRF. It also demonstrates that the human feature encoder offers essential information about human features to more accurately estimate heatmaps crucial for better joint estimation. Figure 3 illustrates the qualitative outcomes of various approaches in ZJU_MoCap dataset. Our technique demonstrates highly competitive results in novel-view synthesis and notably outperforms SHREF [14] and the Neural Human Performer [20] in preserving intricate details..

In Figure 4, we have presented qualitative results of human joint estimation task using the ZJU_MoCap dataset. We generated 25 distinct heatmaps representing different keypoints, with each keypoint being highlighted by red markers. We evaluated our approach using two different

Method	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	PCK ↑
SHERF [14]	26.37	0.918	0.1023	-	-
Neural Human Performer [20]	25.76	0.906	0.148	-	-
ENeRF [21]	31.48	0.965	**0.0494**	-	-
ENeRF+Heatmap	31.48	0.965	0.050	0.0005	0.438
Ours+ResNet	31.20	0.963	0.054	0.0004	0.573
Ours+DINO	**31.61**	**0.966**	0.050	**0.0003**	**0.691**

Table 1. Quantitative comparison of generalization (unseen test set) on the ZJU_MoCap dataset, evaluating all methods at 512×512 resolution. For these experiments, we adhered to the default configurations of SHERF, Neural Human Performer, and ENeRF.

Dataset	PSNR↑	SSIM↑	LPIPS↓	MSE↓	PCK↑
ZJU_MoCap+Res	31.20	0.963	0.054	0.0004	0.573
ZJU_MoCap+DINO	31.61	0.966	0.050	0.0003	0.691
RenderPeople+Res	34.44	0.992	0.0131	0.0012	0.521
RenderPeople+DINO	34.75	0.992	0.0131	0.0005	0.502

Table 2. Quantitative results of the proposed method in different datasets. The results represent generalizable performance on unseen scenes from the test set. Both datasets are evaluated on images with resolution 512×512.

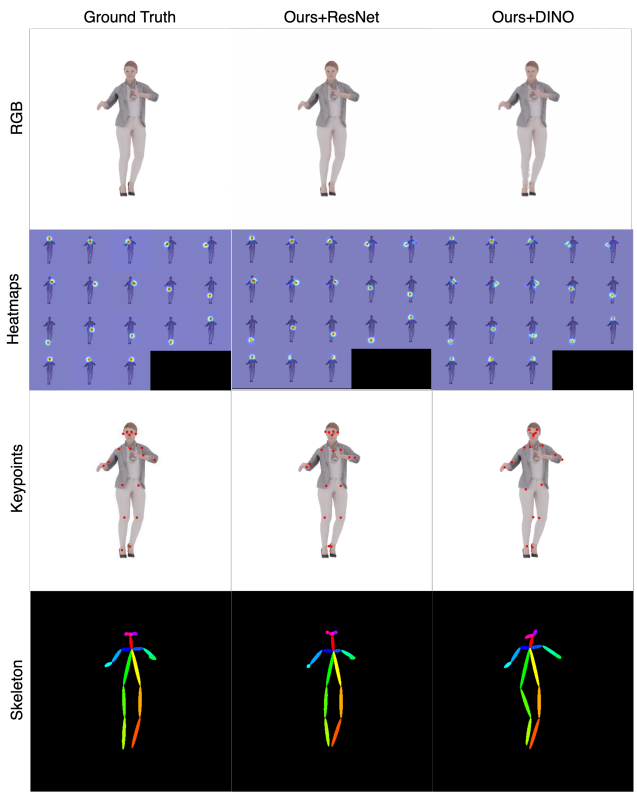

Figure 5. Qualitative result of keypoint estimation on RenderPeople dataset.

types of dataset and reported quantitative results in Table 2. In both datasets, the DINO features showed superior performance in predicting human features as heatmaps. We validate our approach using both real images and simulated images to demonstrate its robustness. Qualitative results of novel-view synthesis and joint estimation on RenderPeople dataset are presented in Figure: 5. To gauge the effectiveness of our proposed method for joint estimation, we compared it with other state-of-the-art pose estimation algorithms and presented the findings in Table 3. Our approach with both ResNet and DINO encoder outperform Alpha Pose and Open Pose, achieving superior PCK and MSE scores. More details, experiments, and results are provided in the Supplementary Material.

	Alpha Pose [10]	Open Pose [3]	GHNeRF +Res	GHNeRF +DINO
PCK ↑	0.647	0.632	0.573	**0.691**
MSE ↓	0.0013	0.0015	0.0004	**0.0003**

Table 3. Quantitative results of keypoint estimation compare to other pose estimation algorithms. We used same ZJU_MoCap test set images of resolution 512×512 to evaluate all three methods.

Dataset	PSNR↑	SSIM↑	LPIPS↓	MSE ↓
ZJU_MoCap+Res	37.22	0.9885	0.0190	0.0039
ZJU_MoCap+DINO	36.51	0.9877	0.0205	0.0019

Table 4. Quantitative results of the dense pose estimation on ZJU_MoCap dataset. Here, MSE is the mean squared error between the predicted and estimated Continuous Surface Embeddings for Dense Pose.

4.3. Performance on dense human pose estimation

In order to showcase GHNeRF 's ability to learn other generalizable human features, we conducted additional experiments to predict dense pose. During training, we use DensePose [11] to generate ground-truth Continuous Surface Embeddings of ZJU_MoCap dataset. We used the same architecture without any modification to learn Continuous Surface Embeddings as human feature from 2D images, which can be used for dense pose estimation. We provide the quan-

titative results in Table 4. The results show that our model can effectively estimate dense pose with different encoders, *e.g.*, ResNet, and DINO, and we find that the DINO encoder performs better compared to ResNet for dense pose estimation similar to joint estimation task. The qualitative results of the estimation of dense pose are presented in Fig. 6. Both qualitative and quantitative findings demonstrate that GHNeRF is capable of learning other generalizable human features beyond just keypoint estimation. This experiment validates our assumption that GHNeRF can learn different human features using the same model architecture.

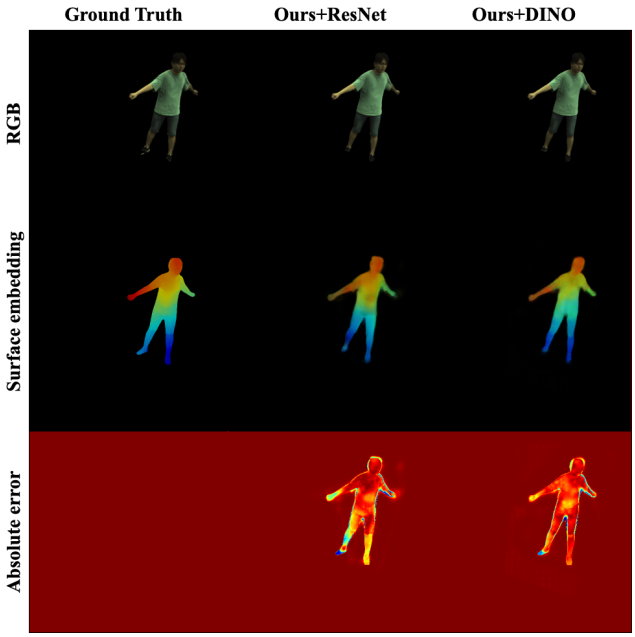

Figure 6. Qualitative result of dense pose estimation on ZJU_MoCap dataset. The absolute error demonstrates the effectiveness of our model with DINO feature in learning dense pose.

4.4. Rendering speed

Inference time of various methods in novel view synthesis and keypoint estimation is illustrated in Table 5. We compared the proposed method with the baseline approach (ENeRF with an extra heatmap branch). While the utilization of the DINO encoder may result in longer inference times, it surpasses other methods by providing superior joint estimation. It may be feasible to attain faster inference time by employing a custom Visual Transformer-based encoder and optimization while maintaining the same level of performance. All experiments were performed on a single RTX 3090 GPU using the PyTorch implementation. We are confident that by optimizing and fine-tuning the code, the rendering time can be improved in the future.

Method	FPS
ENeRF	31.10
ENeRF+Heatmap	27.81
Ours+ResNet18	11.22
Ours+ResNet34	10.49
Ours+DINO	4.08

Table 5. Average rendering speed in FPS(Frame per second). ENeRF+Heatmap represent the baseline method.

4.5. Ablation study

In Table 6, we present the impact of different encoder architectures on the human joint estimation task. We have chosen two different encoder architectures inspired by previous state-of-the-art pose estimation algorithms, namely ResNet [12] and DINO [4]. Both methods produced comparable results in terms of visual quality, but DINO outperformed significantly in the joint estimation task.

Encoder	PSNR↑	SSIM↑	LPIPS↓	MSE↓	PCK↑
ResNet34 *Pre* [12]	31.53	0.965	**0.049**	0.0005	0.454
ResNet34 *Fine* [12]	31.20	0.963	0.054	0.0004	0.573
DINO *Pre* [4]	31.28	0.964	0.051	0.0003	0.682
DINO *Fine* [4]	**31.61**	**0.966**	0.050	**0.0003**	**0.691**

Table 6. Ablation study for keypoint estimation. We show a comparison between different types of encoder for keypoint estimation task. We evaluated both models on ZJU_MoCap dataset. *Pre* represents Pre-trained and *Fine* denotes Finetune during the training.

5. Conclusion

In this paper, we present GHNeRF an end-to-end framework to learn generalizable NeRF to estimate human biomechanic features from 2D images. Through extensive experiments, we have established that our approach can be successfully applied in a variety of settings. We addressed the shortcomings of underlying structure in previous NeRF based methods for humans. The proposed method utilizes an encoder to predict human features using NeRF. In this paper, we focus on estimating human keypoints, and we have also shown how it can be extended to other human features by estimating dense pose. Although our method can estimate human features efficiently, it still has the following shortcomings: 1. It only works in scenes with a single human and it cannot handle multiple humans. 2. The proposed method is limited to humans and does not apply to other animals and articulated objects, which can be a future perspective to learn more general underlying structure.

6. Acknowledgement

This project has received funding from EU's H2020 research and innovation programme under grant agreement No. 847581, the Region Sud and the UCA J.E.D.I.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *CVPR*, pages 5855–5864, 2021. 2

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVRP*, 2022. 2

[3] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE TPAMI*, 2019. 3, 4, 5, 7

[4] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 4, 5, 8, 1

[5] Jianchuan Chen, Wentao Yi, Liqian Ma, Xu Jia, and Huchuan Lu. Gm-nerf: Learning generalizable model-based neural radiance fields from multi-view images. In *CVPR*, 2023. 2

[6] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *CVPR*, 2020. 3

[7] A. Dey, Y. Ahmine, and A.I. Comport. Mip-NeRF RGB-D: Depth Assisted Fast Neural Radiance Fields. *Journal of WSCG*, 30:34–43, 2022. 2

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3, 4

[9] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu. RMPE: Regional multi-person pose estimation. In *ICCV*, 2017. 3

[10] Hao-Shu Fang, Jiefeng Li, Hongyang Tang, Chao Xu, Haoyi Zhu, Yuliang Xiu, Yong-Lu Li, and Cewu Lu. Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. *IEEE TPAMI*, 2022. 3, 7

[11] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7297–7306, 2018. 7, 2

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5, 8

[13] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 3

[14] Shoukang Hu, Fangzhou Hong, Liang Pan, Haiyi Mei, Lei Yang, and Ziwei Liu. Sherf: Generalizable human nerf from a single image. *arXiv preprint arXiv:2303.12791*, 2023. 2, 5, 6, 7

[15] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2014. 1

[16] Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. Instantavatar: Learning avatars from monocular video in 60 seconds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16922–16932, 2023. 2

[17] Wei Jiang, Kwang Moo Yi, Golnoosh Samei, Oncel Tuzel, and Anurag Ranjan. Neuman: Neural human radiance field from a single video. In *ECCV*. Springer, 2022. 2

[18] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *CVPR*, 2019. 3

[19] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12871–12881, 2022. 2

[20] Youngjoong Kwon, Dahun Kim, Duygu Ceylan, and Henry Fuchs. Neural human performer: Learning generalizable radiance fields for human performance rendering. *NeurIPS*, 34:24741–24752, 2021. 6, 7

[21] Haotong Lin, Sida Peng, Zhen Xu, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Efficient neural radiance fields for interactive free-viewpoint video. In *SIGGRAPH Asia Conference Proceedings*, 2022. 2, 3, 4, 5, 6, 7, 1

[22] 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)*, 34:248:1–248:16, 2015. 1

[23] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. 2023. 2

[24] Tung-Wu Lu and Chu-Fen Chang. Biomechanics of human movement and its clinical applications. *The Kaohsiung journal of medical sciences*, 28(2):S13–S25, 2012. 2

[25] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, pages 7210–7219, 2021. 2

[26] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Comm. of the ACM*, 65(1):99–106, 2021. 2

[27] Gyeongsik Moon, Juyong Chang, and Kyoung Mu Lee. Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image. In *ICCV*, 2019. 3

[28] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2

[29] G. Ning, Z. Zhang, and Z. He. Knowledge-guided deep fractal neural networks for human pose estimation. *IEEE TMM*, 2017. 3

[30] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo

Martin-Brualla. Nerfies: Deformable neural radiance fields. In *CVPR*, pages 5865–5874, 2021. 2

[31] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *CVPR*, 2019. 3

[32] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *CVPR*, 2021. 5

[33] Sida Peng, Chen Geng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Implicit neural representations with structured latent codes for human body modeling. *IEEE TPAMI*, 2023. 2

[34] Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J. Black. Dyna: A model of dynamic human shape in motion. *ACM Transactions on Graphics, (Proc. SIGGRAPH)*, 34(4):120:1–120:14, 2015. 1

[35] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, pages 10318–10327, 2021. 2

[36] Grégory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. LCR-Net++: Multi-person 2D and 3D Pose Detection in Natural Images. *IEEE TPAMI*, 2019. 3

[37] Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose. *NeurIPS*, 2021. 2, 3

[38] Bing Wang, Lu Chen, and Bo Yang. Dm-nerf: 3d scene geometry decomposition and manipulation from 2d images. *arXiv preprint arXiv:2208.07227*, 2022. 2

[39] Haoqian Wang, W. P. An, Xingzheng Wang, Lu Fang, and Jiahui Yuan. Magnify-net for multi-person 2d pose estimation. *ICME*, 2018. 3

[40] 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 *CVPR*, pages 16210–16220, 2022. 2

[41] Hongyi Xu, Thiemo Alldieck, and Cristian Sminchisescu. H-nerf: Neural radiance fields for rendering and temporal reconstruction of humans in motion. *Advances in Neural Information Processing Systems*, 34:14955–14966, 2021. 2

[42] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. ViTPose: Simple vision transformer baselines for human pose estimation. In *NeurIPS*, 2022. 3

[43] Jianglong Ye, Naiyan Wang, and Xiaolong Wang. Featurenerf: Learning generalizable nerfs by distilling foundation models. *arXiv preprint arXiv:2303.12786*, 2023. 2, 3, 4

[44] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *CVPR*, pages 5752–5761, 2021. 2

[45] 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, 3, 1

[46] Zhengming Yu, Wei Cheng, Xian Liu, Wayne Wu, and Kwan-Yee Lin. Monohuman: Animatable human neural field from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16943–16953, 2023. 2

[47] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 6

[48] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N. Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *CVPR*, 2019. 3

[49] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. In *ICCV*, 2021. 3

[50] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021. 2

Supplementary Material: GHNeRF: Learning Generalizable Human Features with Efficient Neural Radiance Fields

1. Implementation Details

In the following section, we present details regarding implementation to ensure reproducibility. It is important to note that we did not extensively optimize the architecture or the training procedure due to the significant computational time and resource needed. Thus, there is the possibility that different variations of the hyperparameter can result in a better model.

1.1. Human Feature Encoder

This section presents a comprehensive explanation of our human feature encoder, as introduced in the main paper. Our approach integrates two encoder architectures: DINO and ResNet with a focus on the ResNet34 and ResNet18 variants. For an image with dimensions $H \times W$, the feature map obtained from the ResNet encoder has dimensions $512 \times H \times W$. Following the approach of PixelNeRF [45], we utilized a pre-trained ResNet model on ImageNet. We extracted a feature pyramid similar to PixelNeRF and concatenated them to generate a feature volume of size $512 \times H/2 \times W/2$. Finally, the feature volume was upsampled to generate a human feature with dimensions $512 \times H \times W$.

Additionally, our framework incorporates a pre-trained DINO [4] ViT-Small model with a patch size of 8, which was obtained from the official GitHub repository. To generate the final feature, we extracted features from the 9th and 11th layers of the DINO model and concatenated them. The resulting feature has a shape of $384 \times H/8 \times W/8$. Subsequently, we up-sampled the features using bilinear interpolation to obtain a feature shape of $384 \times H \times W$.

1.2. NeRF Architecture

We utilized an MLP named g_{NeRF} to produce intermediate NeRF features from images. Subsequently, smaller MLPs were used to generate the outputs. In our experiments, we used 2 fully connected layers for g_{NeRF}, which takes f_{img} and f_{voxel} as inputs, following a similar approach as described in ENeRF [21]. To generate the density σ from the intermediate NeRF feature, we used an MLP g_{σ}, which consists of a single linear layer followed by a softmax layer. To estimate the pixel color, we used an MLP g_c with 2 linear layers and 2 ReLU activation functions. The heatmap generation was performed using the g_h MLP, which takes V_{NeRF} and f_h as input. g_h utilizes 2 linear layers with ReLU and Sigmoid activation functions.

1.3. Experimental Setup

We assessed the performance of our method using two datasets: ZJU_MoCap and RenderPeople. For the ZJU_MoCap dataset, we utilized 6 dynamic sequences, namely *CoreView_315, CoreView_377, CoreView_387, CoreView_390, CoreView_394,* and *CoreView_393* for training, and *CoreView_313 and CoreView_386* for testing. We did not include the *CoreView_392* sequence in our evaluation, as it is missing frame data. We used the 2D joint locations provided by ZJU_MoCap to generate heatmaps during training. For training and testing, we limited the frames to an initial 600 frames and divided the total number of cameras equally for training and testing purposes. During training and testing, we generate a 3D bounding box around the dynamic entity using the SMPL model provided in ZJU_MoCap dataset, we project it to obtain a bounding mask and make the colors of pixels outside the mask as zero. For RenderPeople, we used the foreground mask of the dataset. Rays are sampled only inside the mask regions. We calculate PSNR, SSIM, and LPIPS within the masked region. For the RenderPeople dataset, we randomly chose 440 sequences for the training set and 60 sequences for the test set. As RenderPeople does not provide any keypoint information, we used OpenPose as a teacher network to learn the heatmap feature. We utilized 8 samples and 32 volume planes for the course network in all of our experiments, while for the fine network, we used 4 samples and 8 volume planes. We implemented our method and baseline with PyTorch. We report the evaluation metrics and the rendering speed using a single RTX 3090 GPU. We plan to incorporate more datasets for the purpose of benchmarking in future.

2. Additional Experiments and Results

In this section, we present additional results and experiments.

2.1. Coordinate Loss Function

To enhance the spatial perception of our NeRF representation, we introduce a coordinate loss, l_{coord}, aimed at minimizing the Mean Squared Error (MSE) between the input 3D coordinates and the 3D points regressed by the network. This is achieved by incorporating an additional branch in the output to approximate the input query point x. The MLP g_{co}, responsible for this task, processes intermediate NeRF features through a linear layer followed by a ReLU activation function. The composite loss function is formulated as follows:

$$l = l_{col} + \lambda_p l_{perc} + \lambda_h l_{heat} + \lambda_c l_{coord} \qquad \text{(S1)}$$

where the weighting coefficients λ_p, λ_h, and λ_c are set to 0.01, 0.5, and 0.01/0.05, respectively. Table 1 shows the

quantitative results of our experiments with coordinate loss.

2.2. Additional results

In this section, we further explore the qualitative and quantitative results obtained from the ZJU_MoCap and RenderPeople datasets.

2.2.1 ZJU_MoCap dataset:

Additional qualitative insights for the novel view synthesis on ZJU_MoCap are illustrated in Figures S1 and S2. Our proposed method, GHNeRF , uses heatmaps for keypoint estimation. The estimated heatmaps generated by our method are shown and compared in Figures S3 and S4. We have observed missing data in the ground-truth heatmaps. To ensure accurate evaluation metrics, we have excluded the keypoints associated with these missing data. We have compared our 2D keypoint estimate with the baseline in Figures S5 and S6.

2.2.2 RenderPeople dataset:

In order to demonstrate the effectiveness of our approach on various human images, we evaluated its performance using the RenderPeople dataset, which is a simulated dataset. The RenderPeople dataset does not include any ground-truth keypoints, therefore, we train our model for the keypoint estimation task by distilling a state-of-the-art pose estimation algorithm. We provided qualitative results of the novel view synthesis on the RenderPeople dataset in Figure S7. In Figure S8, we present the performance of our model in heatmap estimation and keypoint prediction. We used an image resolution of 512×512 for all experiments conducted on the RenderPeople dataset.

2.2.3 Dense Pose estimation:

We conducted additional experiments to demonstrate that GHNeRF can be utilized to estimate various human features beyond just keypoints. Our model was trained on ZJU_MoCap dataset to predict dense human pose as Continuous Surface Embedding. We trained our model by distilling the SoTA DensePose[11] algorithm. We have presented the qualitative results of dense pose estimation with the ResNet and DINO encoder in Figure S9 and Figure S10, respectively.

Encoder	PSNR	SSIM	LPIPS	MSE	PCK
ResNet34	31.20	0.963	0.054	0.0004	0.573
DINO	31.61	0.966	0.050	0.0003	0.687
ResNet34+co+0.01	31.57	0.964	0.057	0.0010	0.292
ResNet34+co+0.05	31.33	0.958	0.072	0.0008	0.427

Table 1. Quantitative results of coordinate loss experiments compare to other methods.

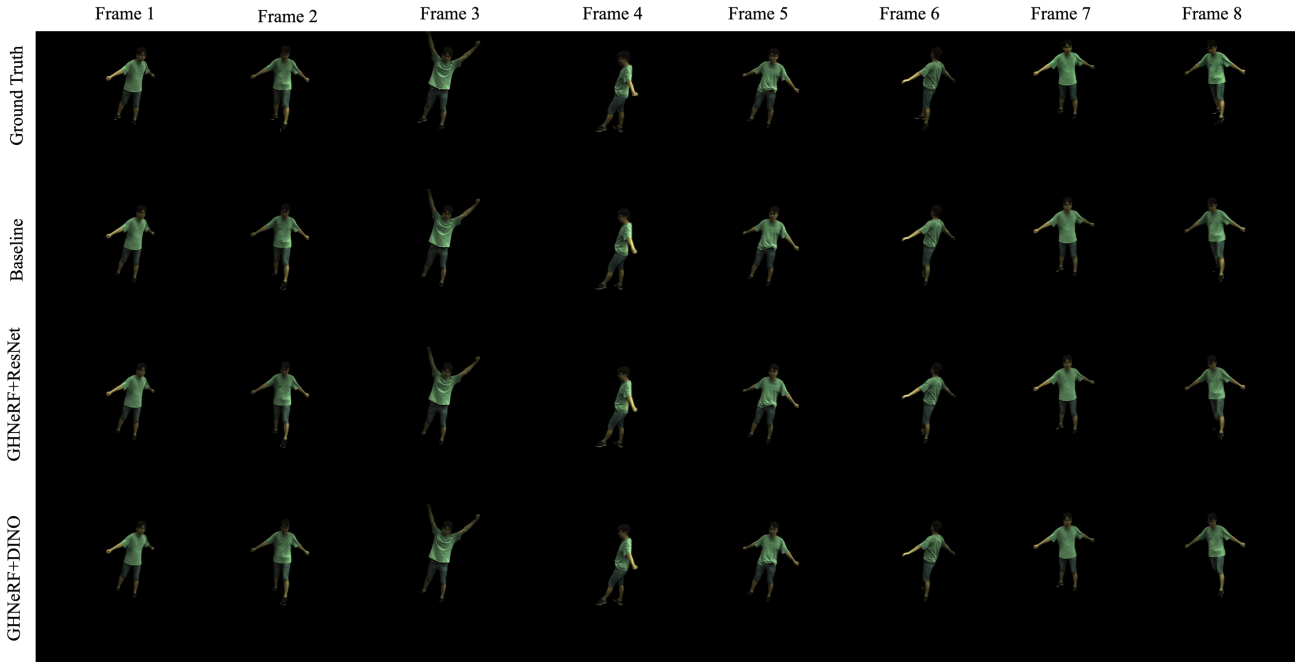

Figure S1. Qualitative results on *CoreView_313* sequence of ZJU_MoCap dataset.

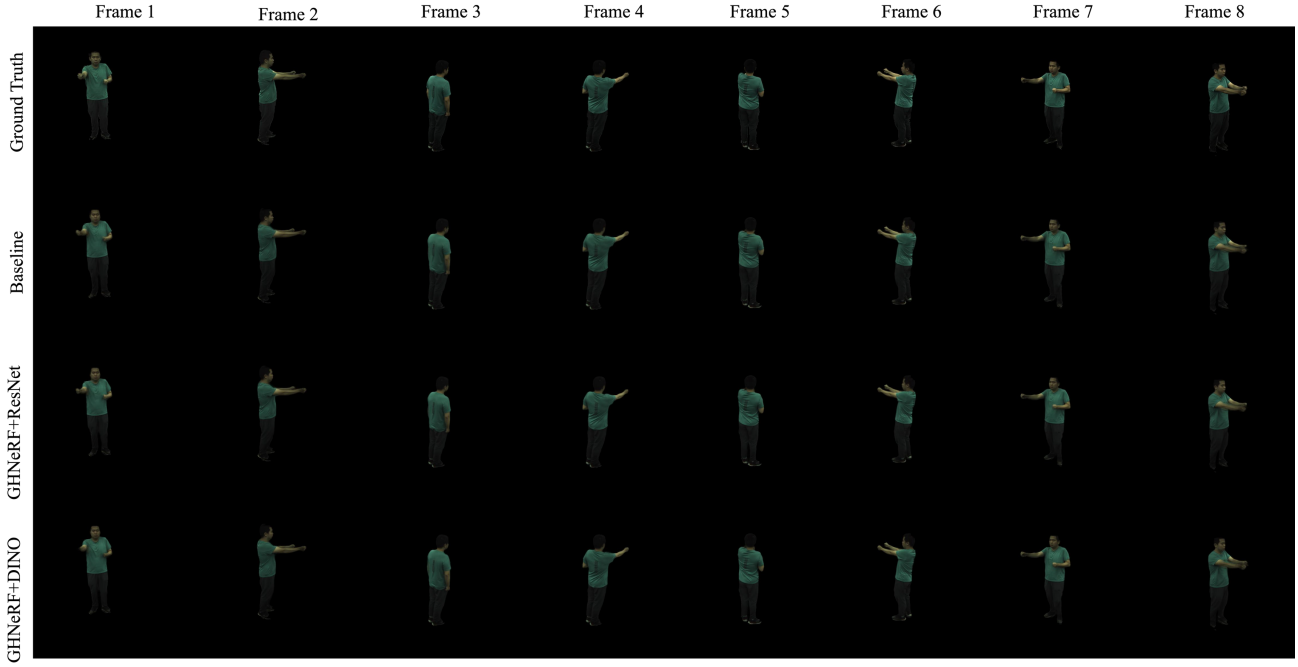

Figure S2. Qualitative results on *CoreView_386* sequence of ZJU_MoCap dataset.

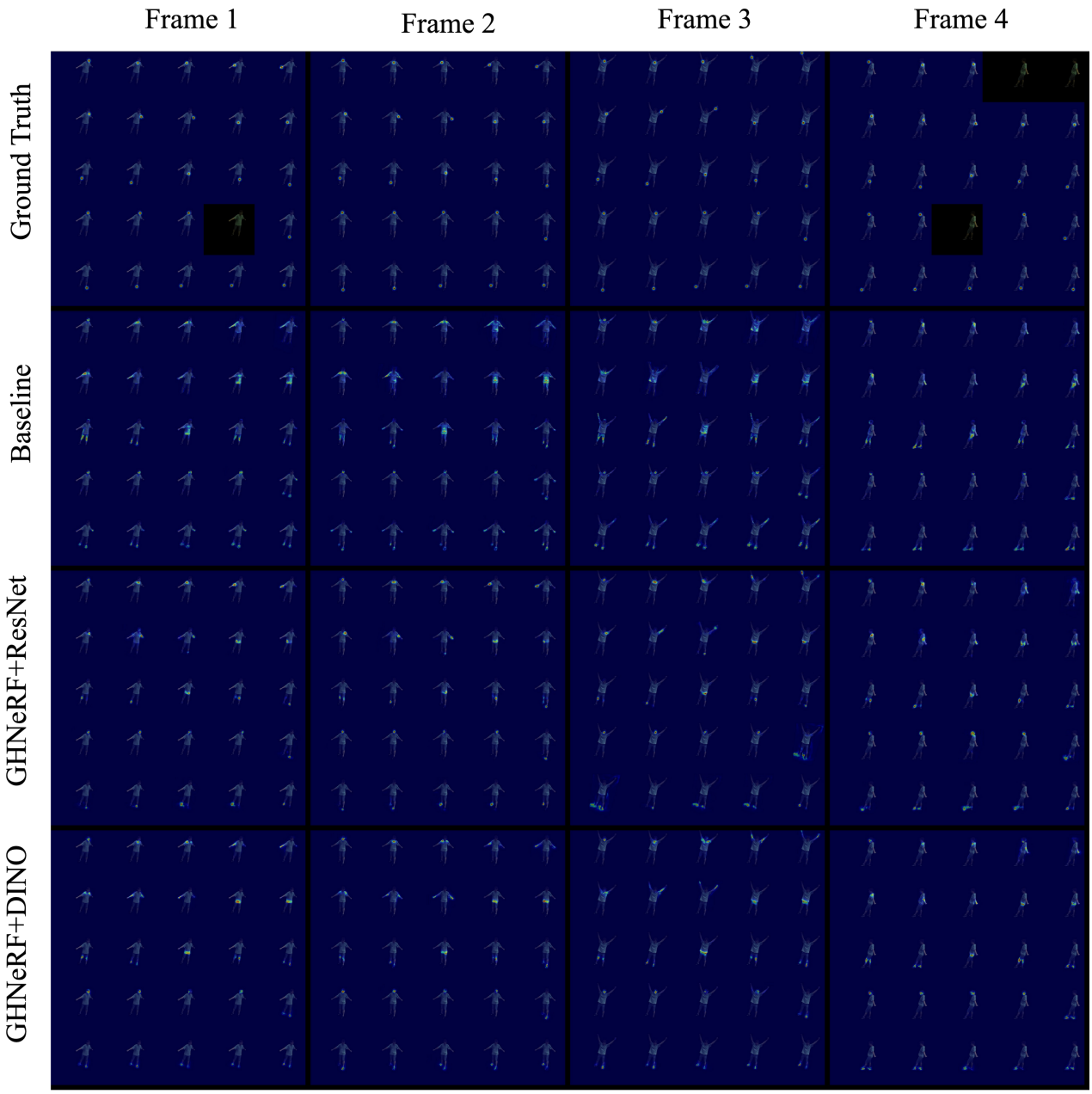

Figure S3. Qualitative results of heatmap prediction on *CoreView_313* sequence of ZJU_MoCap dataset. We estimated 25 keypoints and visualized each channel separately in 5×5 grids.

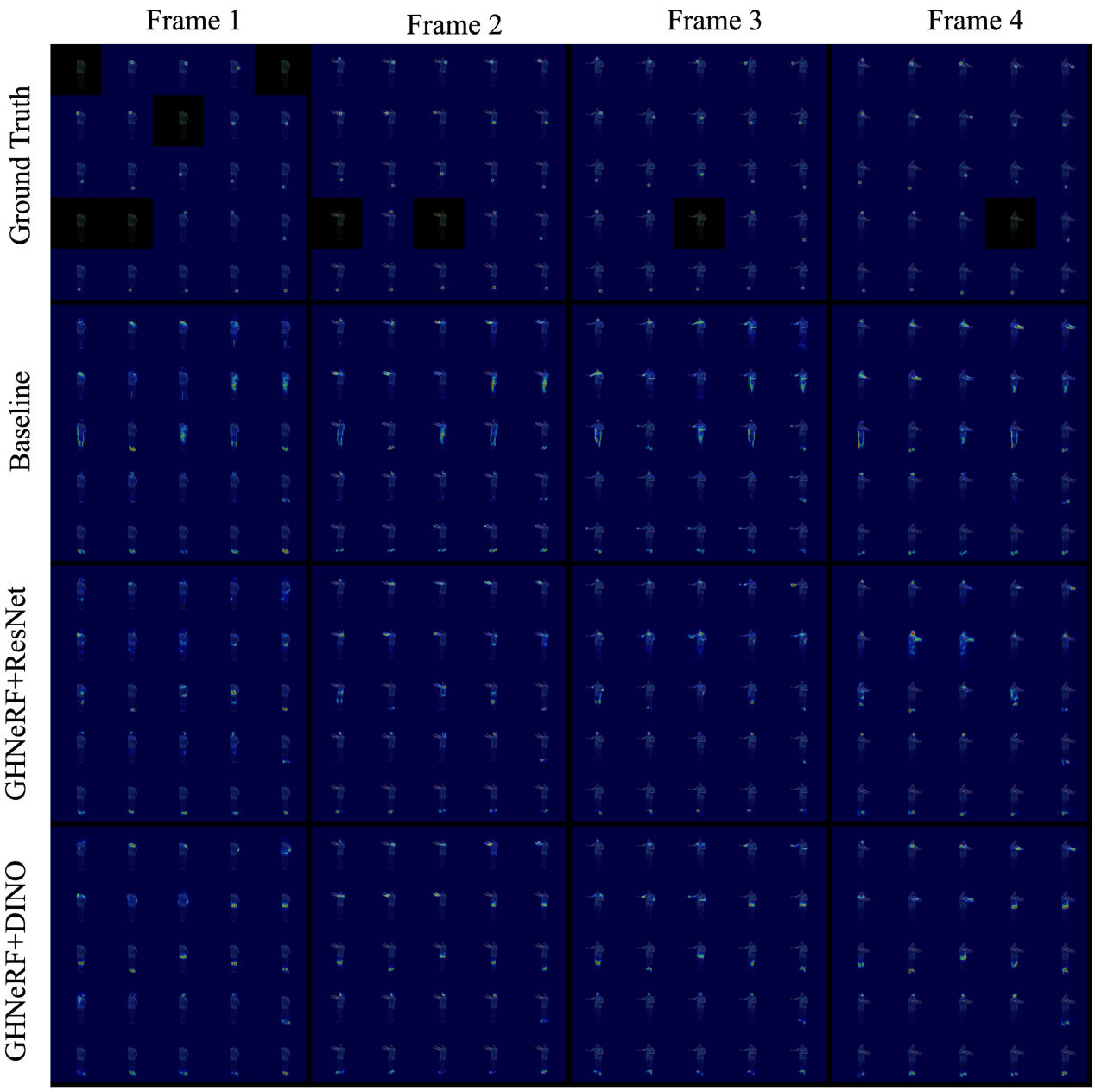

Figure S4. Qualitative results of heatmap prediction on *CoreView_386* sequence of ZJU_MoCap dataset. We estimated 25 keypoints and visualized each channel separately in the 5×5 grids.

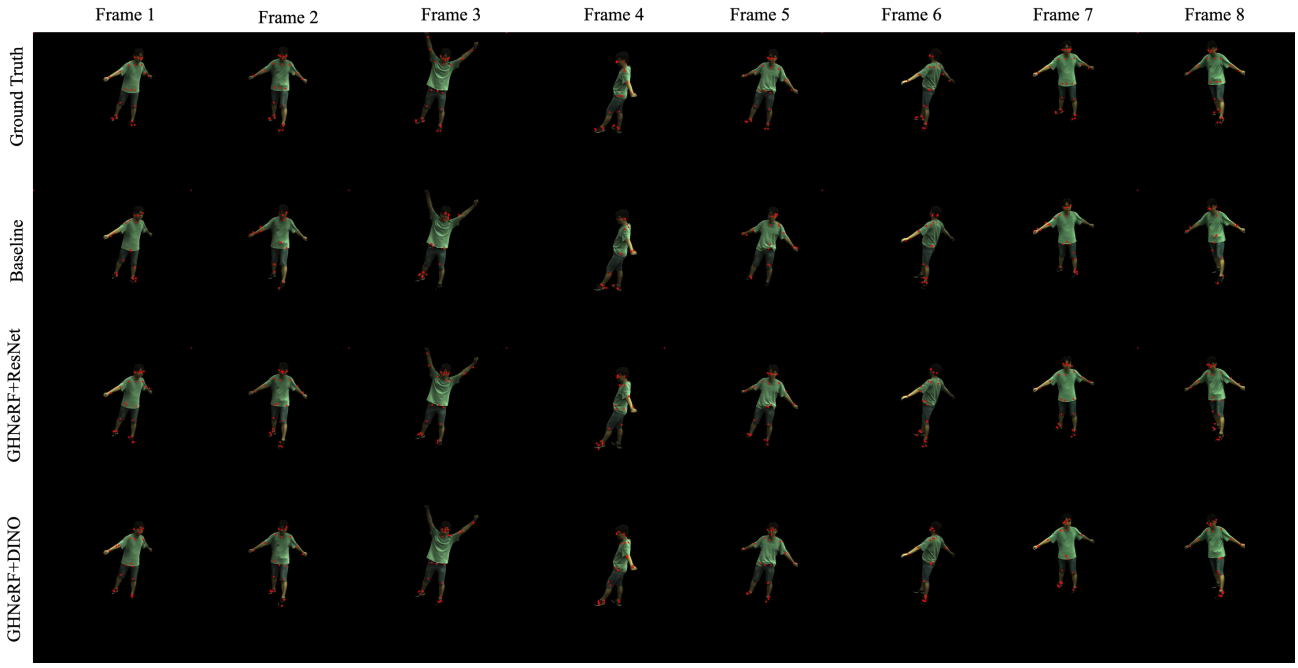

Figure S5. Qualitative results of keypoint estimation on *CoreView_313* sequence of ZJU_MoCap dataset.

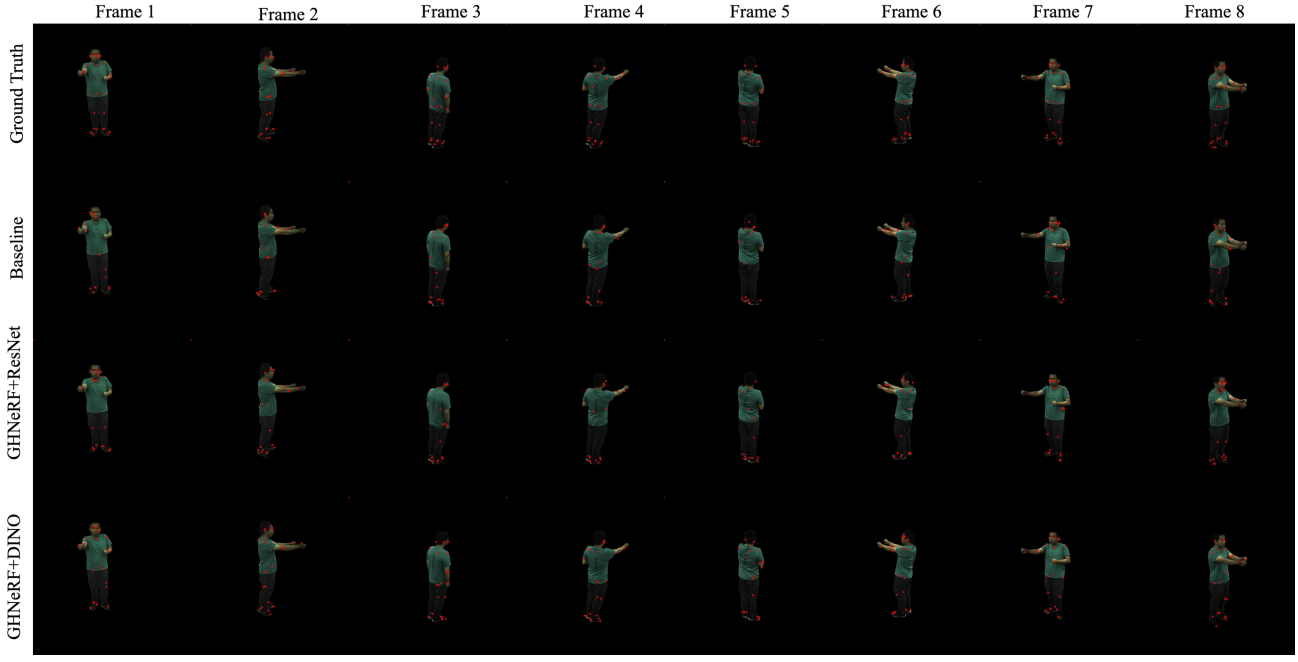

Figure S6. Qualitative results of keypoint estimation on *CoreView_386* sequence of ZJU_MoCap dataset.

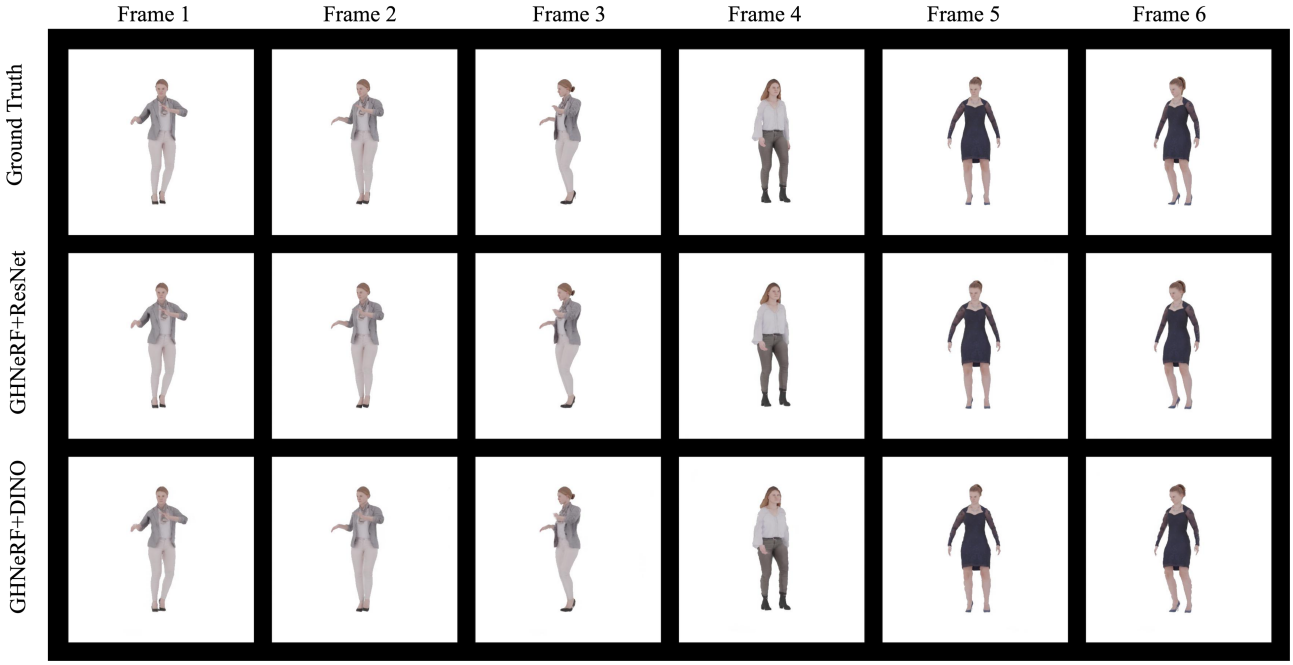

Figure S7. Qualitative results of novel view synthesis on RenderPeople dataset.

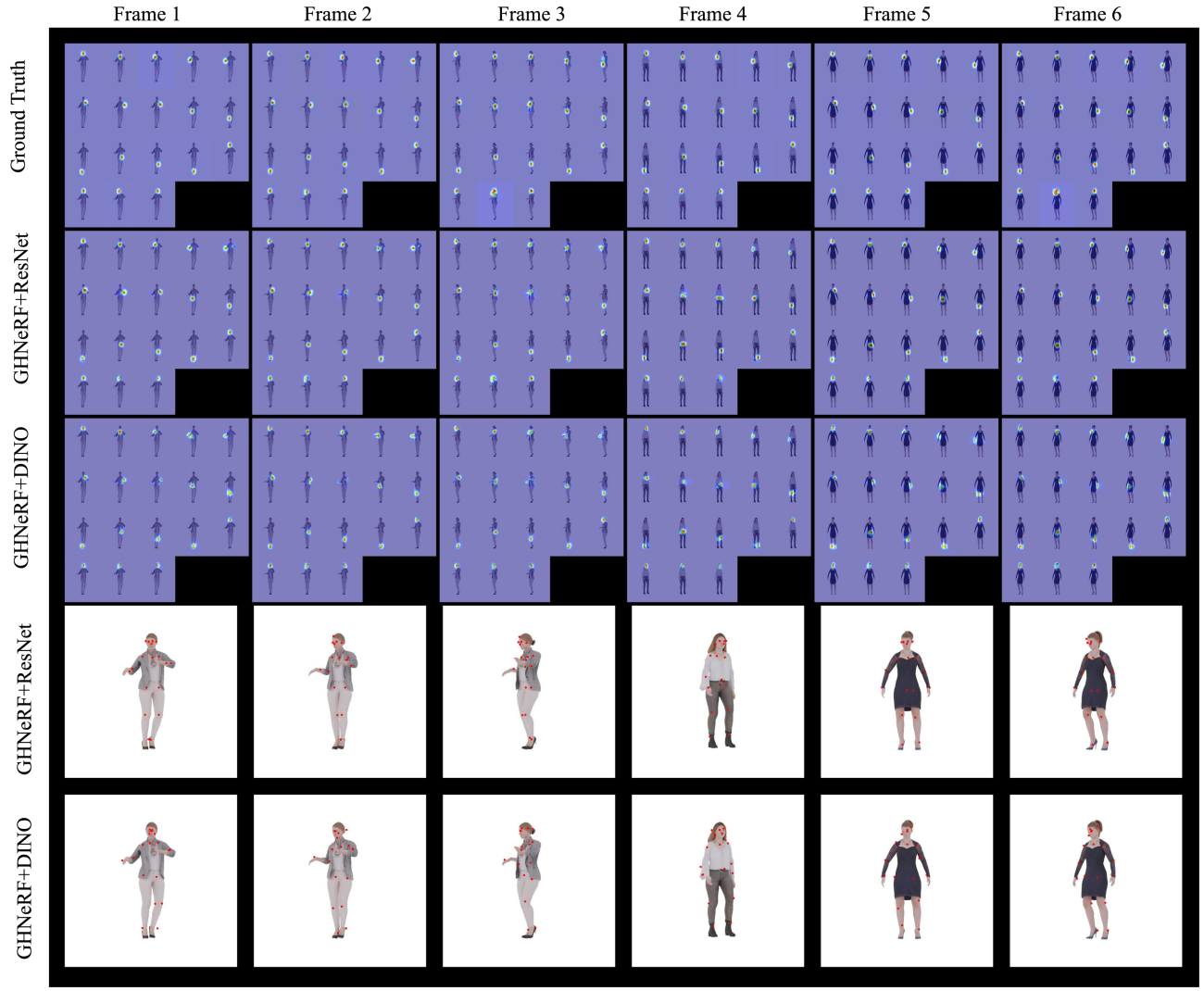

Figure S8. Qualitative results on RenderPeople dataset. The illustration shows predicted heatmaps along with estimated keypoints from heatmaps.

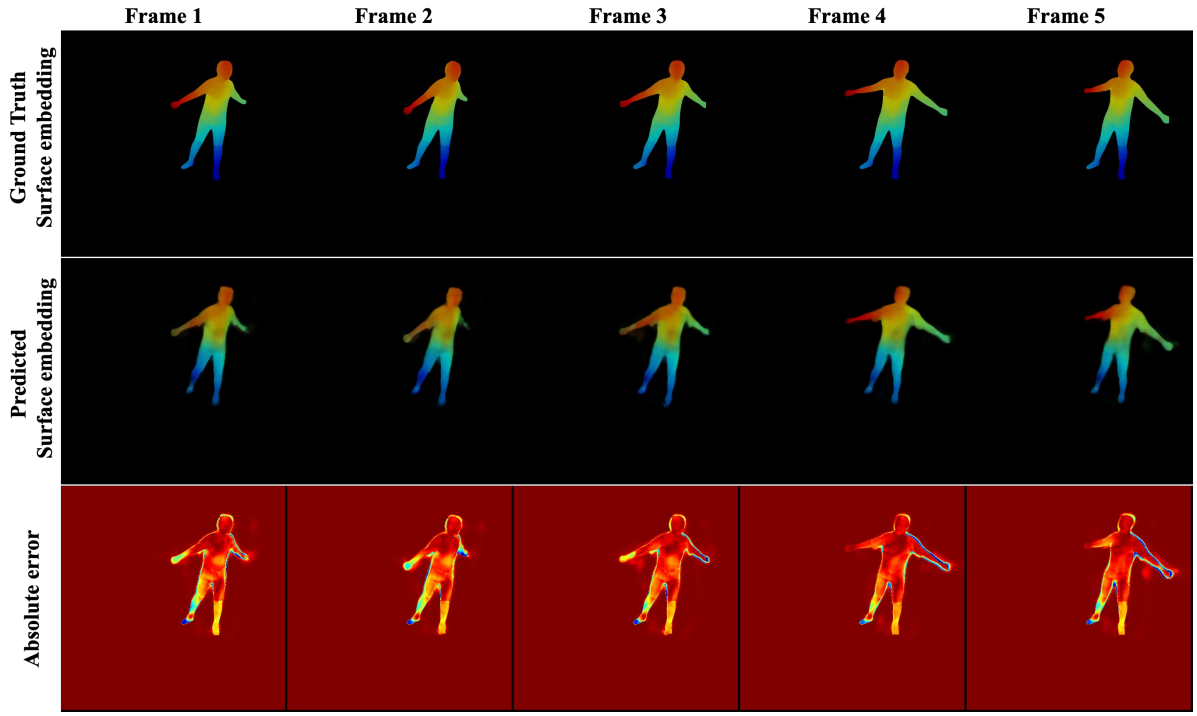

Figure S9. Qualitative results of dense pose estimation with ResNet encoder. We have compared ground truth and predicted Continuous Surface Embeddings.

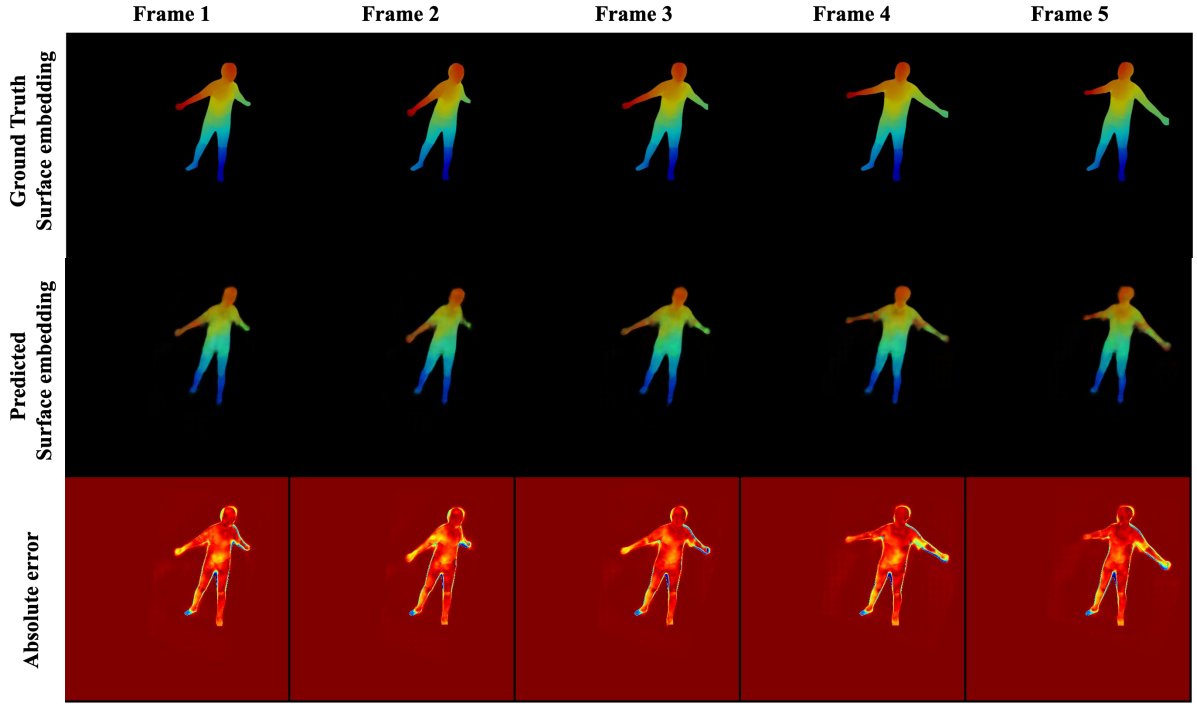

Figure S10. Qualitative results of dense pose estimation with DINO encoder. We have compared ground truth and predicted Continuous Surface Embeddings.