DNA-Rendering: A Diverse Neural Actor Repository for High-Fidelity Human-centric Rendering
– Supplementary Material –

Wei Cheng[1] Ruixiang Chen[2*] Siming Fan[1,2*] Wanqi Yin[2*] Keyu Chen[1*]
Zhongang Cai[3] Jingbo Wang[4] Yang Gao[2]
Zhengming Yu[1] Zhengyu Lin[2] Daxuan Ren[3]
Lei Yang[1,2] Ziwei Liu[3] Chen Change Loy[3] Chen Qian[1]
Wayne Wu[1] Dahua Lin[1,4] Bo Dai[1†] Kwan-Yee Lin[1,4†]
[1] Shanghai AI Laboratory [2] SenseTime Research [3] S-Lab, NTU [4] CUHK

In this supplementary material, we provide detailed information about the proposed DNA-Rendering dataset and the attached benchmarks. We first provide dataset statistics, hardware design, and data collection protocol in Sec. A. Then, we discuss the additional information about the annotations, as well as a comparison to other publicly released toolchains in Sec. B. Moreover, we conduct compact discussions on the benchmarks by introducing more detailed settings, additional results, and unfolded comparisons of benchmark methods' conceptual differences in Sec. C. We provide an in-depth discussion on competing datasets and highlight our comparative contributions to society in Sec. D. Finally, we discuss our future work in Sec. E.

A. Dataset Details

A.1. Dataset Statistics

DNA-Rendering has a wide distribution over ethnicity, clothing, actions, and human-object-interaction scenarios. In this section, we present the detailed data distribution in key data aspects, namely ethnicity, age, shape, actions, clothing, and interactive objects.

Ethnicity, Age and Shape. We invite 500 actors with a uniform distribution of gender and a ratio of $4:3:2:1$ for Asian, Caucasian, Black, and Hispanic individuals, respectively. The quota has a wide coverage of age and body shape. We visualize the distribution of actors' age, height, and weight in Fig. S1.

Human Actions. DNA-Renderingcovers both normal actions and professional actions. We maintain a library of 269 human action definitions, including daily-life activities, simple exercises, and social communication. All normal performers are asked to select 9 actions from the action library and perform the picked actions in a free-style manner. There are 153 professional actors among the total 500 performers. These professional actors are asked to dress in their special costumes and perform 6 unique professional actions with skills, including special costume performances, artistic movements, sports activities, *etc.*. Note that different from the intuitive visualization in Fig. 1[1], we visualize fine-grain categories of professional and normal action in Fig. S2a. These labels are classified in terms of a standard human activity subcategory definition[2]. The sunburst chart of distribution is visualized in the middle, and samples of specific categories of labels are visualized in the outer word cloud.

Clothing and Interactive Objects. We create a clothing repository with 527 items, which covers all 50 clothing types in DeepFashion [17] while with a random distribution of color, material, texture, and looseness for each clothing type. We ask each performer to wear three sets of outfits, where one comes from the performer's self-prepared outfit (for both special and normal actors), and the other two are randomly coordinated from our clothing repository. The distribution of cloth statistical distribution on all action sequences and samples of cloth labels is illustrated in Fig. S2b.

A.2. Meta Attributes

We have designed an attribute system for each dimension of the collected data, including basic information about the actors, clothing, and action information for each action sequence. Fig. S3 shows an example of meta attribute information of an action sequence for a professional actor. In terms of actor information, we record the actor's name, gender, ethnicity, age, height, and weight. For clothing information, we describe the upper and lower clothing, and

---

[1] If not specified, the indexes with only Arabic numeral refer to the corresponding sections/figures/tables listed in the main paper.

[2] https://en.wikipedia.org/wiki/Wikipedia:Contents/Human_activities

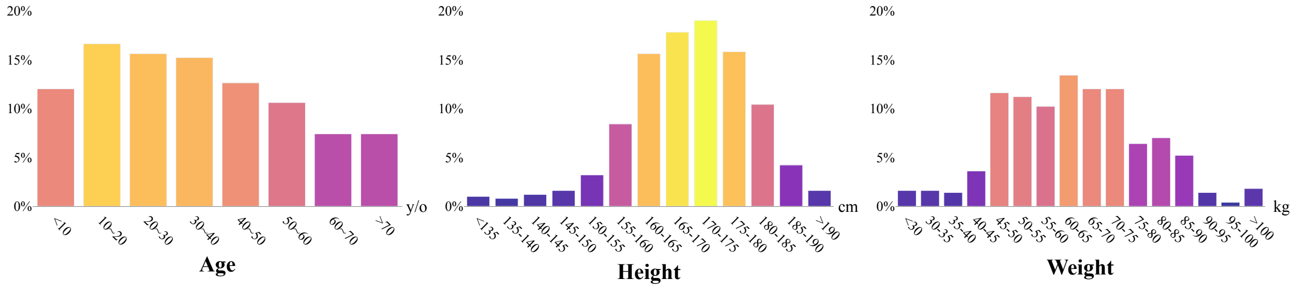
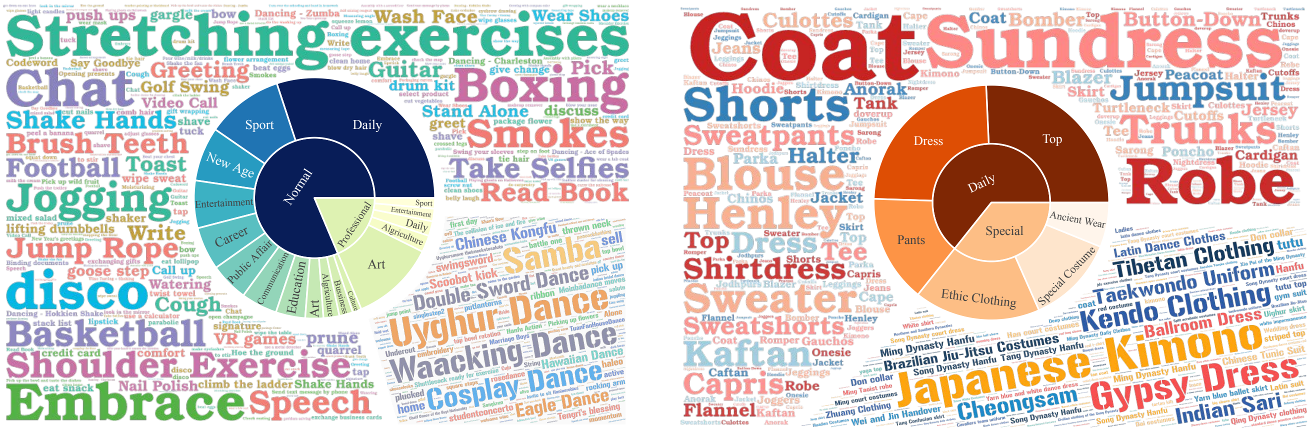


Figure S1: **The distribution of actors' attributes.** We record the age, height, and weight of our invited actors. The statistical results reflect the wide range of the actors' personal attributes.



(a) **Action distribution and labels**

(b) **Cloth distribution and labels**

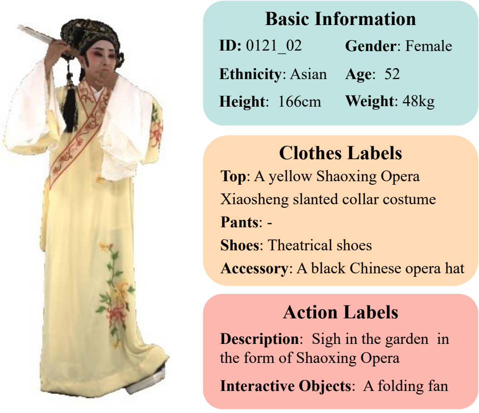
Figure S2: **Illustration of the action and clothes label distribution.** (a). The distribution of action categories and sub-categories is visualized by a sunburst chart in the middle which is surrounded by the word cloud of normal and professional action labels. (b). The distribution of clothes categories and labels are visualized in the same form with (a).

shoe information. These descriptions include information on color, type, and other significant visual features. For action information, we describe the overall content of the action and, if there are any interactive objects, we also describe the type and other significant visual features of those objects.

A.3. Hardware Construction

The main structure of DNA-Rendering's capture system is a dome with a radius of three meters. The camera array built upon the doom consists of multiple types of cameras – ultra-high-resolution 12MP cameras, 5MP industrial cameras, and Azure Kinect cameras. The lighting system provides natural lighting conditions. All cameras are triggered and synchronized by hardware, and synchronized multi-view data are transferred and recorded through our data streaming system.

Camera System. DNA-Rendering has 68 cameras, including 12 ultra-high resolution cameras with 12MP resolution (short for 4096×3000 resolution), 48 industrial cameras at 5MP resolution (*i.e.,* 2448×2048 resolution), and 8 RGB-D Kinect cameras with depth resolution of 576×640. Specifically, the 5MP cameras are mounted on three cycles on the dome skeleton with 1, 2, and 3 meters in height, each circle in height has 16 balanced 5MP cameras with 22.5° angle interval. 12MP cameras are placed uniformly on another two intermediate height level circles, 1.5 and 2.5 meters height respectively. 12MP cameras are installed with a 60° angle



Basic Information

ID: 0121_02 **Gender**: Female

Ethnicity: Asian **Age**: 52

Height: 166cm **Weight**: 48kg

Clothes Labels

Top: A yellow Shaoxing Opera Xiaosheng slanted collar costume

Pants: -

Shoes: Theatrical shoes

Accessory: A black Chinese opera hat

Action Labels

Description: Sigh in the garden in the form of Shaoxing Opera

Interactive Objects: A folding fan

Figure S3: **An example of our meta attribute system.** We record the actors' basic information, costumes, and actions.

interval and interlaced with 5MP cameras. The Kinect cameras are mounted close to the middle level of 5MP cameras, providing the best RGB texture references for depth maps. Such construction of the camera array achieves dense coverage of the human body at multiple heights and angles. 5MP cameras and 12MP cameras are equipped with lenses of 8 mm and 6 mm respectively to achieve the best trade-off between full body proportion-in-view and size of capture volume. Note that, we use data captured from 12MP and 5MP cameras to construct our rendering dataset. The data captured from depth cameras comprise the auxiliary data which provide coarse geometry of human. Noted that we abandon the Kinect RGB cameras during the entire process, due to the bad color consistency.

Lighting System. Our lighting system consists of 16 flat light sources with a color temperature of 5600K ± 300K and an illuminance of 4500 Lux/m. The lighting scale of each light source is 700 × 500 mm. There are eight flat lights on the ground installed with a 45° tilt towards actors in the middle to provide the best lighting on actors. There are extra eight flat lights hung on the roof to strengthen the lighting of upper body parts, especially for human heads. These uniformly distributed flat lights irradiate the whole scene with strong, natural, and balanced illumination.

Data Streaming. To collect, transfer and store the multi-view camera data, we construct a data streaming system that consists of two pieces of equipment for data synchronization – a 10 Giga-byte network, and a high data throughput workstation. The camera system is synchronized by Kinect's trigger signal. First, eight Kinects are configured in a daisy chain and the out-trigger signal is converted to the TTL signal, and the other 60 cameras are triggered by synchronization equipment. The 5MP cameras are connected to six workstations via USB-3.0 ports and four-channel USB cards with PCI-E interfaces. The 12MP cameras are connected to the other three workstations via 10 GigE networks, capture cards, and PCI-E interfaces. To reduce active light interference of Kinect depth cameras, we adopt a 160 μs time delay for each slave device on the chain. The maximum synchronization error of Kinect is 1.12 ms in theory. The maximum synchronization error among all industrial cameras is less than 2 ms, we measure this error by utilizing the image of high-speed flashing LED timer arrays and computing the displayed time differences.

A.4. Data Collection Protocol

We discuss the detailed data collection protocol from five aspects, *i.e.,* data content, system check, core data collection, auxiliary data collection, and post-processing.

Data Content. During everyday data collection, we gather a comprehensive set of data sources, including action data, background data, actors' A-pose data for each outfit, extrinsic calibration data, and the record of performance attributes. We collect intrinsic data and color calibration data only when we apply any modification to the system.

System Check. We conduct a daily system check before formal data collection. The process focuses on the verification of camera parameters and synchronization. Concretely, we will check 1) if the camera parameters remain the same with recorded optimal values (*e.g.,* white balance, gamma, focal length, the valid field of view (FOV), *etc.*). Checking these factors ensures capturing under excellent image quality and valid capture volume. 2) We monitor the network's condition and check the synchronization via a system probe using high-speed flashing LEDs. 3) Finally, we collect extrinsic camera calibration data via a standard data collection process that records the checkerboard rotating as described in Sec. 3.3 in the main paper.

Core Data Collection. We invite 4-6 actors per day to perform actions in our studio in different appointed time slots. Once the actors arrive, we will briefly introduce the collection procedure and ask them to sign the authorization agreements first. If the actors agree, they are asked to prepare their outfits, makeup, and actions. Meanwhile, we record the basic information for each actor, including the height, weight, age, ethnicity, and other appearance attributes like the type, color, and material of his/her self-prepared outfit. After the preparation, we ask each actor to perform several actions in his/her self-prepared outfit. Specifically, a normal actor will pick at least three actions from our pre-defined daily activities and perform them in a free-style manner. A professional actor will wear a special outfit and perform at least six unique sets of footage that fit with the professional skill or costume. Then, we ask each actor to change his/her outfits with another two sets that are randomly coordinated from our clothing library. For each new outfit, the actor will perform another three different normal actions. To ensure the performed motion is rational and authentic enough, we will ask each performer to rehearse outside the studio before the formal shooting. After our staff confirms that the action is performed correctly, the actor will perform in the studio again for the formal data collection.

Auxiliary Data Collection. Aside from the core performance data collection, we also record auxiliary data, including the blank background data for the matting process, and A-pose data as a record for the canonical actor model before each round of new outfit recording. To record A-pose data, we require 1) the actor's hands tilt 45°downward the legs with clear distance; 2) the hands should slightly open without clenched fingers or put them together; 3) the farcical expression should keep expressionless with the eyes open and looking straight ahead.

Post-processing. After the action is completed, the center workstation generates fast multi-view preview videos for all cameras, and we check whether the performance content or the filming on each camera view meets the requirement. Actors are asked to re-play the performance if the recorded data is invalid. After collecting all qualified data, we post-process the data in per-day shooting volume Image sequences in RAW format will be converted to the lossless BMP format, and then compressed into a video with a low constant rate factor with the x264 library. The processed data are then uploaded to the cloud server for subsequent dataset processing.

A.5. Limitations on Data Collection

Data Content. To achieve high-fidelity data collection, we set the lighting with invariant and uniform illumination and set the acquisition frame rate to 15 frames per second. We also constrain the field of view of the cameras, to ensure each one can capture the *full-body* movements of a single actor (including the interacted object if there is one) while maintaining the FOV as max as possible. This allows us to capture details such as facial makeup and clothing textures. In future work, we would update our hardware systems and upgrade our capture processes to accommodate different lighting conditions, multiple FOV ranges for multi-person scenes, high-speed capture of subtle movements, and multi-sensory (*e.g.,* auditory, and tactile data) collection.

Failure Cases. During the data collection process, various factors can lead to failure, such as large movements that exceed the field of view of the multi-camera system, or loss of frames due to large volume data transmission fluctuations. Following the standardized capture process, our operators will manually inspect the completeness and effectiveness of all camera data and actor movements after each capture cycle. If any issues are found, the hardware will be instantly checked and the data will be re-captured. Most failure cases are identified and promptly resolved at this stage.

B. Data Annotation Details

B.1. Camera Intrinsic Calibration

As this project targets capturing high-fidelity *whole-body* data, we adopt a long lens that enlarges the human proportion in the camera view. This setup requires a high-quality estimation of camera distortion since some subjects' body parts might appear on the image boundary region. Thus, we use a 3×3 Soduku data collection protocol for intrinsic calibration, as illustrated in Fig. S4. Specifically, to maximize the checkboards' coverage across the whole image space, we separate the image into a 3×3 Soduku and capture the images of the checkerboards' movement in grids. For each grid, we rotate the checkerboard in pitch, row, and yaw direction to enlarge the angle of the boards.

B.2. Camera Color Calibration

To ensure color consistency across multiple cameras, we inject a color calibration process into our data collection.

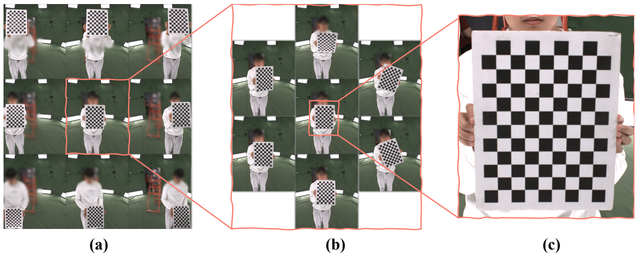


Figure S4: **Intrinsic calibration.** To ensure better distortion coefficient estimation, **(a)** we separate the camera view into 3×3 Soduku and capture the images of a checkerboard (about $1/4$ size of the one used in estimating extrinsic parameters) in every Soduku grid. **(b)** For each grid, we rotate the checkerboard at pitch, row, and yaw angles. This calibration step forces the checker to appear in every corner of the camera view. **(c)** Zoom-in for small-size checkerboard for intrinsic calibration.

A standard color board could be used as the criterion for f color calibration, and the fixed lighting condition in the dome could be treated as a standard condition during calibration. Specifically, the calibration lies in two aspects: 1) Hardware parameter adjustment. We make a rough adjustment on the hardware parameters to make the white balance and color balance of each camera as consistent as possible by human eyes; 2) Fine adjustment. Under a standard light source, we make the standard color board face straightforwardly to the camera to be calibrated at a constant distance, and a single image under this setting is collected; the corner detection algorithm is used to automatically identify the

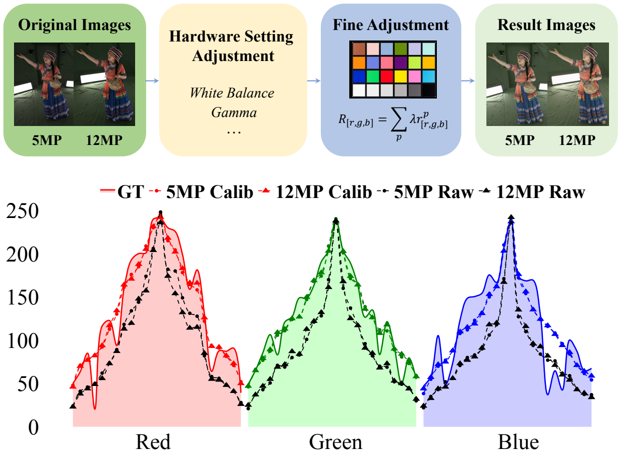


Figure S5: **Color calibration pipeline and calibrated color response.**The color-calibrated images are listed on the right of the flow chart (at the top of the figure). The color responses of two calibrated cameras (Camera 25 is 5 MP, and Camera 51 is 12MP) compared with groundtruth color value of the color checkerboard are plotted below the flow chart, and smoothed spline curve is used. We show the 'Raw responses' after hardware setting adjustment for reference. With the help of the color correction process, the average RGB value consistency between these two cameras ΔE_{00} [21] is improved from 37.79 to 4.15.

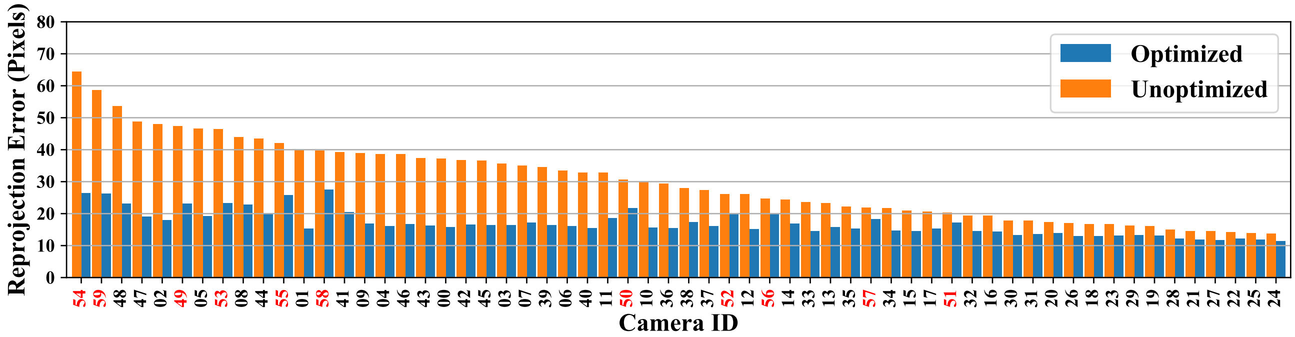


Figure S6: **Evaluation of keypoint quality from every camera view.** We compute the mean reprojection error of 3D keypoints with 2D detection results. Optimization effectively reduces the error to below 30 pixels. Note that camera IDs 0-47 are 5MP cameras, and camera IDs 48-59 are 12MP cameras, high-lighted with red x-ticks.

position of the color board in the image, and the color sampling is performed with the center radius $p = 10$ pixels of each color square. The average color value is taken as the color sampling value. We carry out the polynomial projection of the color sampling value to the standard value via least squares. Note that, we calibrate in RGB form and take $n = 2$ to prevent overfitting. The overall procedure and illustrated results are presented in Fig. S5.

B.3. Keypoints

We highlight that having a large number of camera views allows us to rectify the occasional failures of single-view 2D keypoint detection. For the more natural and stable 3D keypoints, we adopt the following optimization and post-processing strategies: 1) *Keypoint selection.* We dynamically select views for each keypoint in the data sequence, which with the most confident score to the keypoint while ensuring the keypoint can be triangulated. 2) *Bone length constraint.* The bone length is constrained with a fixed length. We use the median bone length after initial triangulation as the target in the optimization. Only the lengths of the main limbs are considered in this step. 3) *Outlier removal.* As a post-processing pipeline, filter modules are designed based on human priors, including a 3D bounding box filter, a movement filter, and a relative position filter. 3D keypoints outliers, which are too far from the body trunk, move too fast between frames, or lead to an inconsistent relative position between frames are removed. An interpolation is applied to recover the missing keypoints. Such a post-processing scheme can assure reliable and consistent face and hand keypoints, even with large-scale occlusions. As shown in Fig. S6, these optimization and post-processing strategies effectively reduce reprojection error compared to triangulation with all available 2D keypoints.

B.4. Parametric Model

In our automatic parametric model annotation pipeline, body shape $\beta \in \mathbb{R}^{n\times 10}$ (or $\beta \in \mathbb{R}^{n\times 11}$ for children [8, 25]) is first estimated based on the bone length calculated from 3D keypoints with the static and less challenging A-pose sequence. We use the estimated body shape parameters as initial values and optimize the full parametric model parameters including pose parameters (body pose, hand pose, and global orientation) $\theta \in \mathbb{R}^{n\times 156}$, and translation parameters $t \in \mathbb{R}^{n\times 3}$ (n is the number of frames) via a modified SMPLify-X for other sequences with dynamic poses. The main energy terms in the optimization are keypoint energy $E_{\mathcal{P}}$, full-body joint angle prior energy E_a, bone length energy $E_{\mathcal{B}}$, and body shape prior energy E_β [26, 20, 1]. The main modification of SMPLify-X in our annotation pipeline is the decoupling body shape optimization and pose optimization, which we empirically find to produce more stable results.

Concretely, we employ bone length energy $E_{\mathcal{B}}$ and body shape prior energy E_β to fine-tune body shape parameters for each sequence of a subject, with the same shape initialization from the A-pose static sequence. Body shape values are kept consistent throughout all the frames in a sequence.

$$E_{shape}(\theta, \beta, t) = \lambda_1 E_{\mathcal{B}} + \lambda_2 E_\beta \tag{S1}$$

We then leverage keypoint energy $E_{\mathcal{P}}$ and full-body joint angle prior energy E_a for pose optimization with body shape fixed.

$$E_{pose}(\theta, \beta, t) = \lambda_3 E_{\mathcal{P}} + \lambda_4 E_a \tag{S2}$$

As shown in Fig. S7, we evaluate the fitting error between 3D keypoints and corresponding regressed SMPLX joints. Body-only keypoints, hand-only keypoints, and all keypoints are evaluated separately. With our multi-view capture system and annotation pipeline, the MPJPE of body-only keypoints is on par with the optical motion capture system in Human3.6M [9, 19], MPJPE of hand-only keypoints is 15.87mm.

B.5. Comparison to Other SMPLX Fitting Methods

Baselines. To analyze the effectiveness of the proposed SMPLX fitting pipeline, we evaluate the accuracy of SMPLX fitting and compare it with three publicly available pipelines, *i.e.,* the baseline MultiviewSMPLifyX [39, 26],

Methods	2D Reprojection Error (pixel)				3D MPJPE (mm)				Run Time (s)
	Body	Hand	Face	Overall	Body	Hand	Face	Overall	
MultiviewSMPLifyX [39]	42.27	33.36	23.91	28.77	55.46	23.25	22.24	27.54	81.33
BodyFitting (GeneBody) [3]	45.68	33.43	34.17	35.67	42.37	32.19	30.67	32.89	29.50
EasyMoCap (ZJU-MoCap) [4]	32.71	33.75	32.72	33.64	36.04	25.37	38.10	33.96	0.69
Ours	29.63	31.41	19.08	24.08	30.20	15.87	16.46	17.52	3.23

Table S1: **Comparison among multi-view SMPLX fitting methods.** Cell color 🟥 🟧 🟨 indicates the best, second best, and third best performance, respectively. Runtime in seconds indicates the average time required for the fitting process for one multi-view frame.

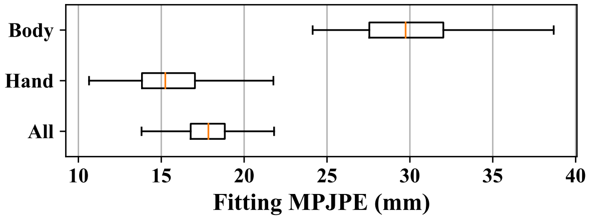


Figure S7: **Evaluation of parametric model registration quality.** We evaluate body-only keypoints, hand-only keypoints, and all keypoints separately. The orange line indicates the median value, the box indicates the lower to the higher quartile, and the whiskers indicate the range of data.

EasyMoCap [4, 30] used in ZJU-MoCap [28, 30] dataset, and BodyFitting used in GeneBody [3] dataset. Specifically, MultiviewSMPLify [39] and BodyFitting [3] directly optimize the error of reprojected 3D SMPLX keypoints to 2D detections. Such a naive strategy is straightforward but lacks outlier robustness (might stuck in absolutely wrong detections or detection flip between left and right), and it is also computationally expensive. On the contrary, both EasyMoCap and the proposed pipeline adopt another strategy that separates the SMPLify process by a triangulation process. This strategy optimizes 3D keypoints from 2D detection and then fits SMPLX from directly on optimized 3D keypoints. As robust designs could be adapted during the triangulation process to eject outliers caused by flipping or occlusion, such a two-step strategy is faster and more robust to outliers. Whereas, one drawback is that the final SMPLX totally rely on the results of triangulation in the first stage by hand-crafted optimization and filtering. Compared to EasyMoCap, incorporate a more sophisticated designed 2D keypoints postprocessing phase, where movement filtering and relative position filtering are used when the given 2D keypoints are not accurate.

Settings. For fairness, we use the same 2D keypoints consisting of human-inspected body labels and hands and faces auto-detection results. We also force all SMPLX models to have 10 facial expression coefficiency and 45 hand PCA components. We run the SMPLX fitting on our benchmark test data with their default SMPLX setting, namely with default penalty energies, their coefficient, and other settings except for the aforementioned modifications. We quantitatively evaluate the MPJPE of 3D keypoints and the reprojected 2D error across all views.

Results. The quantitive results are listed in Tab. S1, we also separately evaluate the accuracy on body, hand, face, and whole body. 3D MPJPE is computed by regressed SMPLX 3D keypoints to human-inspected 3D keypoints. 2D reprojection error is compared by reprojected SMPLX 3D keypoint to input 2D keypoints. The runtime performances are also recorded, indicating the average fitting time usage (excluding other time namely, data IO, *etc.*) for one multi-view frame. Our proposed pipeline outperforms other fitting methods in all categories in terms of both 2D and 3D metrics, and has a more acceptable runtime requirement than EasyMocap [4]. Moreover, MultiviewSMPLify [39, 26] achieves the second-best performance, while its time consumption is exploded by an order of magnitude. In a nutshell, our pipeline ensures the best trade-off between performance and efficiency.

B.6. Matting and Segmentation Refine

As described in the main text, despite the state-of-the-art background matting method [16] achieving impressive matting performance in the majority of our data, there are still several corner cases that fail to extract the foreground correctly, *e.g.*, noisy backgrounds, broken bodies, and missing bodies and objects. We demonstrate these most common corner cases in Fig. S8. To further improve the matting quality, we adopt the traditional computer vision algorithm-GraphCut [6] to refine the predicted masks, and we find such a classical method plays a good fit to the CNN-based method which generates good results on these failure cases.

In order to quantify the improvement, we introduce a manual inspection process to grade the results generated by CNN-based method [16] only and by a subsequent refinement procedure. Noted that due to the large scale of data, generating masks with manual labeling is impractical. More specifically, we ask three annotators to conduct such grading surveys on 500 random multiview sequences. We report the error rates in terms of the type of corner cases in the bar chart in Fig. S8. From the human grading probe, we can conclude that with our proposed hybrid strategy, the error rates in all category decrease by a large margin compared with [16] only, with the overall error rate reduced from 11% to 2%.

B.7. Quality Control of Auto Annotation Results

To ensure the quality of annotation data, we conduct manual quality checks on the auto-annotated results.

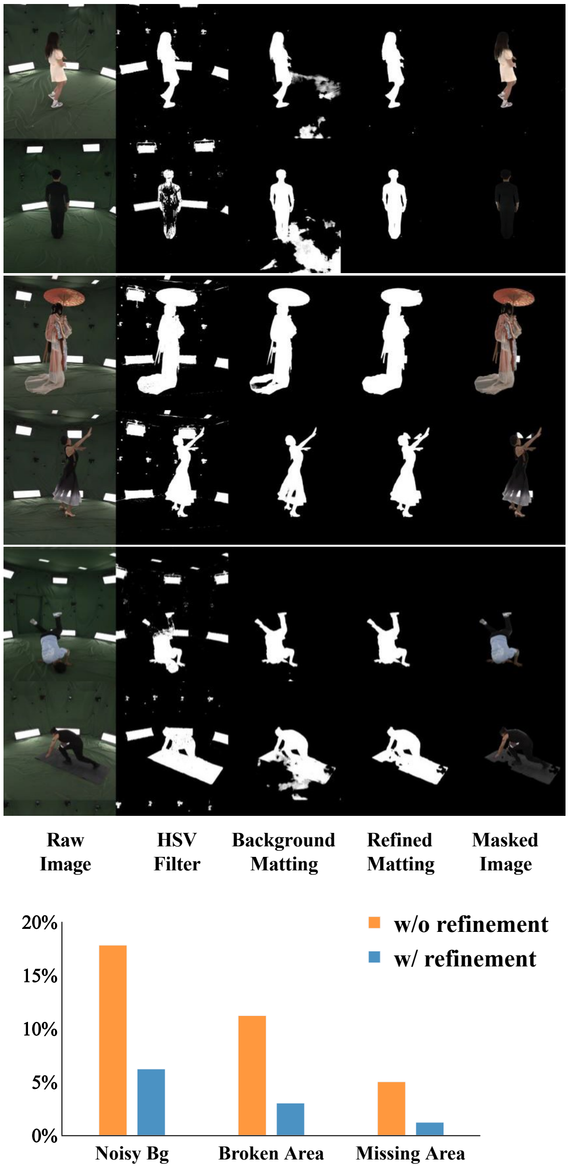


Figure S8: **Examples and statistics on matting refinement.** In the upper image, we show three kinds of common challenging cases and the comparisons among color filtering only, background matting only, and our optimized solution (*Refined Matting*). From top to bottom cases, the problems (before optimization) are noisy backgrounds, broken body areas, and missing areas like body parts and incomplete objects; In the bottom figure, we show the error rates from the sampling survey of the three categories.

Specifically, we perform a human-in-the-loop quality evaluation for the SMPLX and matting results generated by the annotation pipeline.

For SMPLX quality control, we overlay each SMPLX result on the original action data to create a multi-view video and manually verify the quality with the labeling task that requires our human annotator to grade the SMPLX quality. We subdivide the process into three stages. 1) Binary filtering. If the SMPLX human body completely overlaps with the human body in the image or is within the natural shape range of the human body throughout the entire video, it is considered as a qualified SMPLX annotation; otherwise, if there is severe misalignment or distortions on the main body, it is considered as an unqualified one. 2) Quality Grading. For qualified data cases, we further evaluate their subdivision quality, dividing them into five scores based on the unnaturalness of fingers or faces, the alignment of the head and shoulders with the image, *etc*. 3) Keypoints re-annotation. For unqualified cases, we ask the annotators to re-annotate the main skeleton in views with large errors by auto-annotators. The new annotation results are used to rerun the SMPLX results. We repeat the whole process until we achieve valid SMPLX models in all cases.

For matting quality control, we manually evaluate the quality of the annotated video after matting by grading each video's quality. Quality is divided into three levels: A-level, where the entire human body is fully displayed without occlusion, and any interactive objects are fully shown, with no excess areas; B-level, where a small part of the human body or object is missing, or there are a few extra cutouts, with the erroneous pixel area not exceeding one-third of the human body area; and C-level, where there are serious problems and the erroneous pixel area exceeds one-third of the effective human body area. We treat A-level and B-level as acceptable mask annotations, while C-level as failure annotations. Noted that the cases in training and testing data split in our benchmark were manually selected to ensure high-quality annotations. For the sake of rigor, we will release the mask rating as confidence for mask annotation.

C. Benchmark Details

C.1. Methods Overview and Modifications

In this subsection, we review the state-of-the-art methods benchmarked in this paper, and describe the major modification we made to the default implementation for adapting to the proposed dataset.

C.1.1 Static Methods

For static methods, we target to anchor the performances of novel view rendering on static test frames, which could be used as the baseline reference for dynamic methods on certain frozen times.

Instant-NGP [24] as an alter of NeRF [23], which utilizes the multi-resolution hash embedding and smaller network to accelerate the training and evaluation cost without loss of quality. Given the original implementation of Instant-NGP is under the underlying assumption of a moving single camera input or cameras sharing the intrinsic parameter across all camera positions, we modify it to suit multi-camera data with different intrinsic parameters.

NeuS [34] is a hybrid representation that combines neural radiance field with neural SDF, which produces better 3D reconstruction ability than NeRF-based methods [7, 23, 28] on existing datasets, while the rendered images of NeuS are typically not as sharp as NeRF-based methods. When adapting NeuS [34] on the proposed dataset, no special modification is required.

C.1.2 Dynamic Methods

To construct the novel view synthesis and novel pose animation benchmark, we select the most recent state-of-the-art dynamic neural human rendering methods which can learn a neural body avatar from video sequences.

NeuralVolumes [18] formulates a category-agnostic dynamic scene by a canonical voxel-grid decoder, and models the per-frame deformation as a mixture of affine warps that are parameterized by an auto-encoder with image input. Due to this property, we feed the network with 4 balanced views of images from the training views. We also center and scale the camera system by 0.3 to fit the voxel-grid system. During testing on novel pose, novel pose images of the 4 view are input to the auto-encoder.

A-NeRF [31] learns a human NeRF by conditioning the field with coordinates in each bone's local system. Note that its default setting only trains the network in the foreground, which usually leads to artifacts on the floor, we improve this by forcing pixel sampling on non-foreground space which helps to reduce the artifacts.

NeuralBody [28] conditions a dynamic NeRF by time codes as well as structural latent features by sparsely convolving parametric model's vertices in 3D. To run NeuralBody [28], we transform our standard definition of the parametric model to EasyMoCap [4] style, and we train the network using 42 dense views. Note that during novel pose estimation, we fed the network with novel pose SMPLs and linearly extrapolate the time step.

AnimatableNeRF [27] introduces neural blend weights with 3D human skeletons to generate observation-canonical correspondences in dynamic human NeRF. We do the same transformation like NeuralBody [28].

HumanNeRF [36] learns a dynamic neural human model from monocular video. It decouples the motion field by a corrected skeleton movement and non-rigid motion. Different from its original setting, we train the model with dense views by stacking multiview video sequences. It is *important* to point out that HumanNeRF [36] models may collapse on certain data sequences producing meaningless images like the left bottom case in Fig. 6. Such a phenomenon is consistent even with multiple trials of random initialization. Considering to deliver a more straightforward metric meaning in the benchmark, the report numbers of HumanNeRF in Tab. S2 and Tab. 2 only include the valid models.

C.1.3 Generalizable Methods

For novel identity generalization, we evaluate three category-agnostic methods [37, 35, 15] and two methods with human structure priors [13, 22].

PixelNeRF [37] is one of the first generalizable NeRFs that generalize novel objects' color and opacity by pixel-aligned feature-conditioned NeRF. We train it on our dataset with 4 selected views and fuse the multiview image feature with average pooling.

IBRNet [35] predicts the radiance color of novel objects by blending observed color from source views, and inference the opacity from multiview feature fusion.

VisionNeRF [15] upgrades PixelNeRF's [37] image encoder with a global transformer [33] and fuse the multi-level features with 2D CNN features. Like PixelNeRF [37], we fuse the multiview feature with average pooling.

NeuralHumanPerformer [13] combines key components of PixelNeRF [37] and NeuralBody [28], and fuse them with multiview transformer and predict the radiance of human body. Noted there is a slight contradiction between the technical paper and the released implementation on the window size of the temporal transformer. We follow the open-sourced implementation and set the window size to 1 to avoid memory explosion which means the temporal transformer is a dummy module.

KeypointNeRF [22] use IBRNet [35] as the backbone, and tailors 3D keypoints as human prior into the framework. It conditions the radiance field with relative depth to every 3D keypoint in each source camera coordinate. We train the network with 24 SMPL main skeleton keypoints. Noted that different from other generalizable methods which allow arbitrary resolution rendering, KeypointNeRF [22] is suited to render square-sized images with 2^n width and height. We render out a minimum squared image that can cover the desired resolution then crop out the valid part.

C.2. Benchmark Details

As a supplement to the benchmark part in the main paper, we describe the detailed benchmark settings and additional analysis of results.

C.2.1 Novel View Synthesis

Detailed Settings. As we described the task in Sec. 4.2 and reviewed the methods in Sec. C.1, we evaluate the dynamic methods' novel view synthesis ability on the benchmark test set, which consists of 13 splits with 39 performance sequences. During training, we train models on each sequence separately, with the 42 multiview (training views) image sequences of the first 80% frames. Evaluations are performed on the same seen human poses but with every 45 frame skip and only calculated on 18 unseen camera poses. For static methods, we train separate models on multi-view images of each single frame. To evaluate the high-fidelity rendering of these benchmark methods, we train and test the models on *half of the origin resolution*, namely 1024×1224 and 1500×2048 for 5MP and 12MP images respectively.

Detailed Results. As a supplement to the result analysis in the main text, we present more detailed results and analysis in this subsection. Detailed quantitative results across our testing splits are listed in Tab. S2, which correspond to the bubble charts in Fig. 4. We also illustrate the qualitative results in Fig. S9. From the ranking in Tab. S2, we can observe that A-NeRF [31], NeuralBody [28] and HumanNeRF [36] achieve the best numbers in most of the test splits in terms of PSNR, SSIM and LPIPS [38]. As the module designs of these methods might play vital roles in such distinct results, we further analyze the phenomenon by unfolding their conceptual differences as follows: *NeuralVolumes* [18] adopts a VAE [12]-style neural rendering framework that encodes and decodes both the affinity transformation field and rendering volume from sparse view references. Such a paradigm absorbs the strength of VAE that compresses input multi-view features into one compact latent representation space, which follows the Gaussian assumption. Thus it could generalize well on novel views (achieving top-three performance in PSNR) with acceptable rendering quality. Whereas, such a framework also inherits the smooth problem of VAE that leads to not sharp enough qualitative results. *A-NeRF* [31] is a conditioned NeRF that utilizes local joint ordinate information of query points. It samples a small box center at a random point in the foreground and adds a proportion of background points to regularize the empty space, using only mean square error loss (MSE). This strategy enforces the network to encode the dynamic NeRF with local bone coordinates, which is efficient in the human foreground region. The overall novel view synthesis ability of A-NeRF [31] is appealing, especially in PSNR. However, due to the sparsity characteristic of skeleton representation, A-NeRF tends to generate dilated artifacts (see *Motion-Medium* and *Motion-Hard* in Fig. S9), more obvious with novel pose in off-body parts of *Interaction-Hard* case in Fig. S10. *NeuralBody* [28] and *AnimatableNeRF* [27] first compute a 3D bounding box from SMPL, then train/infer on the reprojected 3D box region and fill the outer region with the background color. Thus, their SSIM scores are typically greater than other methods that infer the whole image. Whereas, the bounding box only helps the network consider the main body and ignore the object, which leads to both methods can not reconstruct large interacting objects (as illustrated in the sword part of *Interaction-Medium* case in Fig. S9) *HumanNeRF* [36] combines the strength of previous methods' design, it samples squared boxes on reprojected 3D bounding box, and trained model with a perceptual [10] loss and MSE loss. This results in best LPIPS performance, as well as the best visual performance with sharp texture in qualitative results. However, as HumanNeRF [36] designs a human motion prior that is Gaussian distribution along body parts or bones. This prior may lead to training failure on loose clothing and interactive objects, as illustrated in the *Deformation-Hard* and *Interaction-Medium* cases in Fig. S10.

C.2.2 Novel Pose Animation

Detailed Settings. We conduct novel pose experiments on dynamic methods on the same models in novel view synthesis. Specifically, by training on the first 80% frames of each case, we test the animatable model with input of the pose sequences extracted from the last 20% frames with a 15-frame skip. Depending on the pose-condition scheme of different methods, the test input can be divided into two categories, the SMPL parameters and the image features. We use the same testing view and rendering resolution as the novel view synthesis experiment.

Detailed Results. We present the detailed novel pose ani-

Splits	**PSNR↑**							**SSIM↑**							**LPIPS*↓**						
	NGP	NS	NV	AN	NB	AnN	HN	NGP	NS	NV	AN	NB	AnN	HN	NGP	NS	NV	AN	NB	AnN	HN
Motion-Simple	30.97	27.49	27.85	29.15	27.84	25.89	25.49	0.979	0.973	0.966	0.974	0.978	0.974	0.955	31.52	44.18	57.74	52.75	53.32	56.10	62.08
Motion-Medium	31.40	30.04	28.16	29.07	27.47	24.93	24.80	0.980	0.980	0.970	0.975	0.981	0.971	0.966	25.12	31.33	50.03	45.28	48.12	56.43	33.66
Motion-Hard	29.05	28.49	26.10	27.55	25.16	24.54	22.93	0.972	0.976	0.959	0.967	0.976	0.976	0.964	41.35	40.13	77.54	69.75	71.25	63.35	53.42
Deformation-Simple	31.63	28.01	28.09	29.63	28.18	27.42	28.30	0.981	0.972	0.968	0.975	0.976	0.975	0.974	29.02	42.62	48.17	41.68	48.18	45.44	23.70
Deformation-Medium	30.01	29.65	29.77	30.52	29.22	26.29	26.60	0.972	0.975	0.971	0.974	0.979	0.972	0.963	41.14	37.18	39.36	43.51	46.95	52.69	29.12
Deformation-Hard	29.79	30.80	27.19	28.11	24.77	21.93	21.48	0.967	0.973	0.954	0.957	0.969	0.958	0.934	46.09	44.83	70.04	70.16	81.14	84.59	83.36
Texture-Simple	30.53	31.39	27.85	30.45	29.13	25.36	27.39	0.978	0.988	0.974	0.984	0.988	0.979	0.979	36.02	23.53	58.78	43.09	41.53	53.69	24.72
Texture-Medium	30.85	31.33	28.50	30.53	29.62	22.46	27.40	0.978	0.982	0.968	0.977	0.984	0.959	0.971	29.32	27.04	47.33	37.99	41.46	75.08	25.01
Texture-Hard	29.16	28.23	26.73	27.36	25.69	19.98	24.78	0.966	0.956	0.942	0.947	0.963	0.946	0.950	36.48	58.55	79.68	79.17	77.36	94.60	34.66
Interaction-No	31.31	31.90	29.05	28.71	27.77	23.82	26.77	0.978	0.985	0.972	0.975	0.984	0.971	0.971	34.00	23.65	50.36	56.07	49.79	67.40	30.00
Interaction-Simple	31.55	32.13	29.09	30.12	29.56	25.93	28.59	0.982	0.987	0.976	0.981	0.987	0.977	0.978	27.58	21.79	45.55	46.48	41.69	57.18	22.00
Interaction-Medium	28.82	27.15	25.59	25.72	25.65	22.02	23.51	0.967	0.968	0.955	0.956	0.975	0.997	0.953	43.33	52.52	73.05	85.50	68.03	92.90	54.39
Interaction-Hard	30.03	29.29	28.09	28.25	25.00	22.92	23.87	0.972	0.977	0.962	0.964	0.972	0.961	0.951	43.64	39.71	57.97	63.42	71.00	81.49	57.38
Overall	30.39	29.68	27.85	28.86	27.31	24.11	25.53	0.975	0.976	0.964	0.970	0.978	0.970	0.962	35.74	37.47	58.12	56.53	56.91	67.76	41.04

Table S2: **Benchmark results on novel view synthesis task.** State-of-the-art methods' performance of novel test views on seen poses in each benchmark split. We abbreviate Instant-NGP [24] as 'NGP', NeuS [34] as 'NS', A-NeRF [31] as 'AN', NeuralVolumes [18] as 'NV', NeuralBodyas 'NB' [28], AnimatableNeRF [27] as 'AnN', and HumanNeRF [36] as 'HN'. Cell color 🟥 🟧 🟨 indicate the best, second best, and third best performance in the same split respectively. We exclude the static methods NGP and NS during ranking and separate them with dash lines.

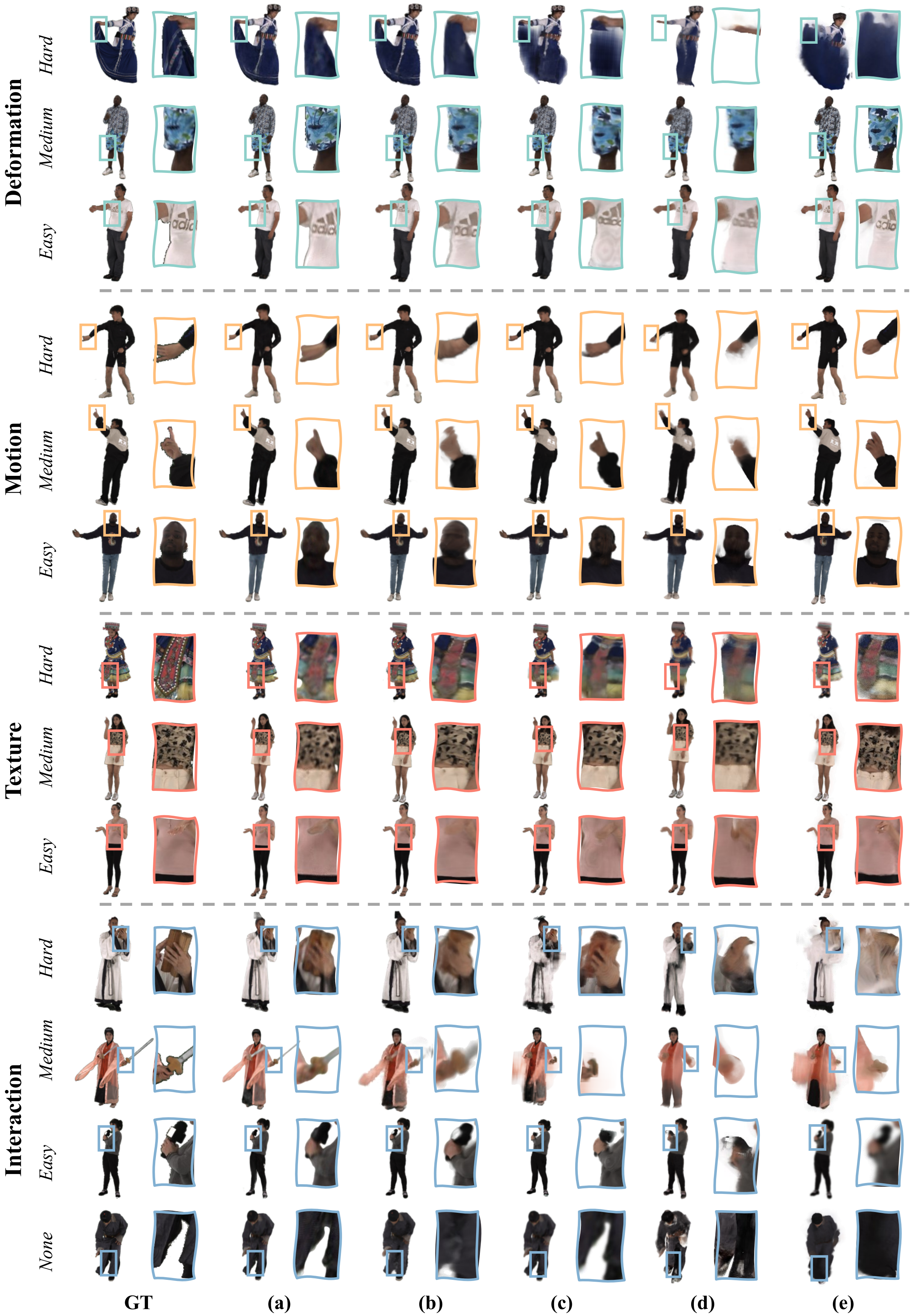


Figure S9: **Novel view synthesis results on each data split**. From top to bottom, we illustrate the rendering results generated by **(a-e)**. NeuralVolumes [18], A-NeRF [31], NeuralBody [28], AnimatableNeRF [27], HumanNeRF [36]. Please zoom in for better visualization.

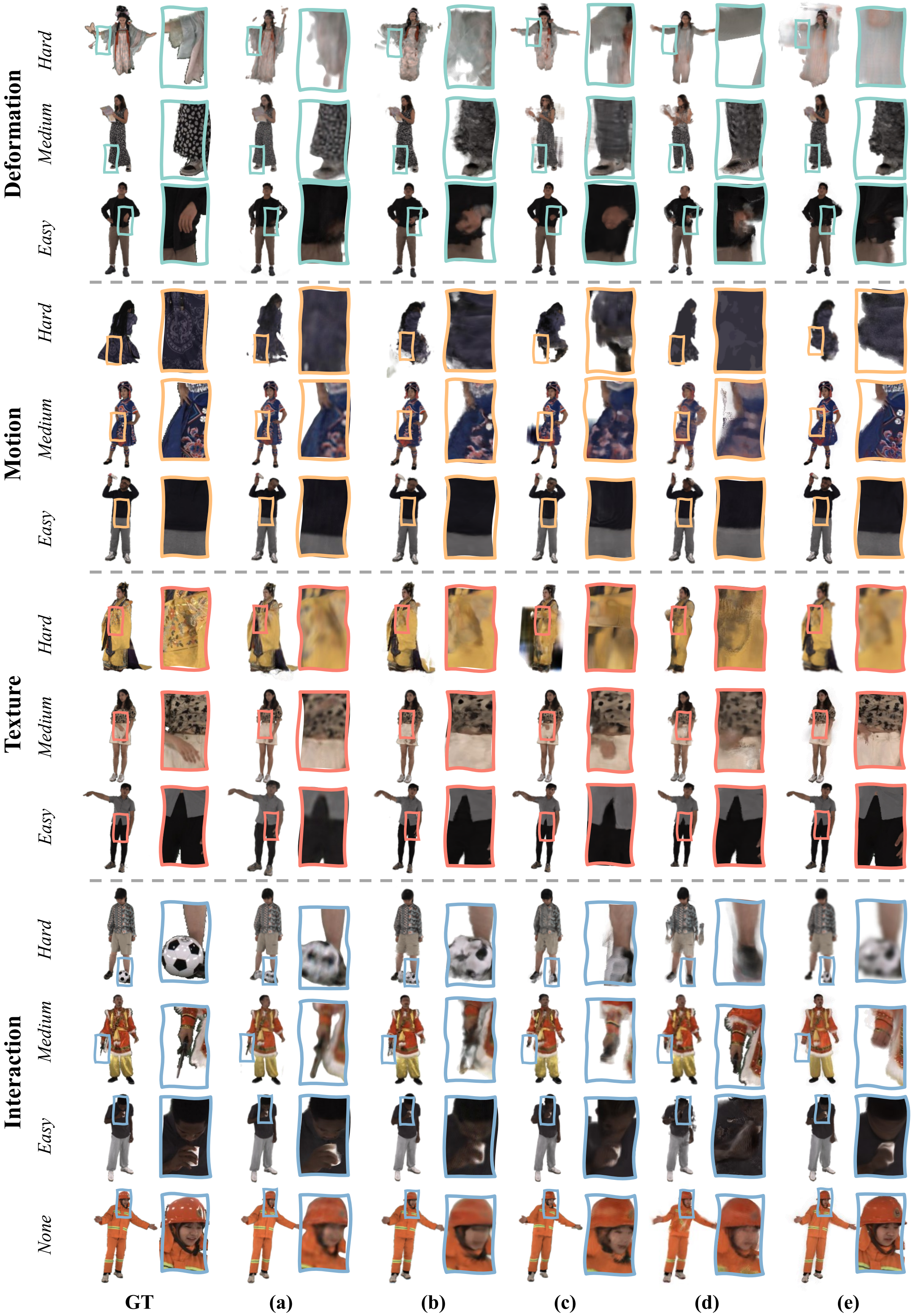


Figure S10: **Novel pose animation results on each data split**. From top to bottom, we illustrate the reposing results generated by **(a-e)**: NeuralVolumes [18], A-NeRF [31], NeuralBody [28] AnimatableNeRF [27], and HumanNeRF [36]. Please zoom in for better visualization.

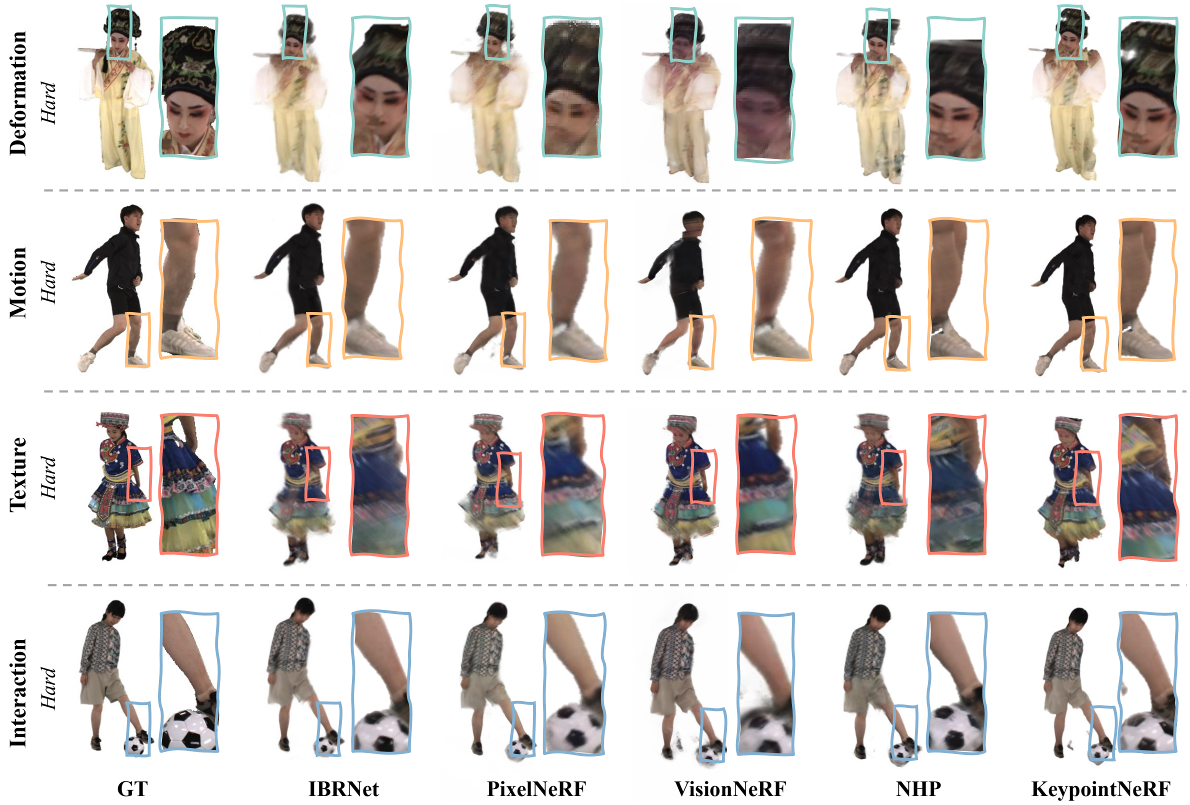


Figure S11: **Novel ID synthesis results on each data split**. Splits with hard difficulty are visualized to illustrate the robustness of generalizable methods on hard cases.

mation results on 13 testing splits in Tab. 2 of the main paper, and show the qualitative samples of results in Fig. S10 in this subsection. Similar to novel view synthesis task, A-NeRF [31] achieves the best PSNR performance, Neural-Body [28] has the best SSIM score, and HumanNeRF [36] gets the best LPIPS [38]. Differently, NeuralVolumes [18]'s performance decrease by a large margin, especially in *Motion* splits. One of the underlying reasons is that the affinity field learning in the NeuralVolumes [18] only relies on the latent code that is learned from multiview images and regularized by KL-divergence. Such a methodology is tied in a global warping manner, which might be relatively less affected by factors like global deformation in distance within a short movement change, but is vulnerable to unseen local motion (*e.g.,* wrong head pose of *Interaction-Medium* case in Fig. S10, and strained border of actor's shirt in *Texture-Easy* case in Fig. S10). Moreover, the design cannot preserve global scale in unseen poses, due to the wrong prediction of global affine transform sometimes (*e.g.,* the zoom scale of the actor in *Texture-Easy* case in Fig. S10). The methods [31, 28, 27, 36] with the explicit human pose information as input can typically generate reasonable animation results in terms of the local first motion, as shown in Fig. S10. Whereas, we draw the other major conclusion that current methods fail to model *Deformation* and *Interaction* properly. The typical examples shown in Fig. S10 are the loose cloth case of *Deformation-Hard*, and the interactive objects in *Interaction-Medium* and *Interaction-Hard* cases. How to properly model non-rig or out-of-body motions while preserving the advantages from explicit body representations is worth great pondering.

C.2.3 Novel Identity Rendering

Detailed Settings. For novel identity synthesis, we review five state-of-the-art methods [37, 35, 22, 5, 13], and described their modification in Sec. C.1.3. The testing set is the same 39 testing sequences in the novel view and the novel pose benchmark. The training set consists of 400 sequences with full coverage of all categories and difficulties. We select four balanced views as source views. These source view images are cropped and resized into 512×512 resolution (same with the official implementation in [22, 3, 13]). For category-agnostic methods [37, 35, 15], we only provide segmentation and camera parameters during training and testing. For methods with human prior, we also input the fitted SMPLX or 3D keypoints. We train models on the full 60 views in the training identity sequences. For inference, we evaluate the unseen identities on the same 18 test views used in novel view and novel pose tasks, but on full sequences with frame skip at 45. All methods are trained under the same 8-V100 machine environment, and evaluated on a single V100.

Detailed Results. In the main text, we draw the conclusion that generalization methods use human prior [22, 13] is more robust than category-agnostic methods [37, 35, 15] according to the results reported in Tab. 3. To further illustrate this conclusion, we compare the qualitative results in *Hard* level of each data factor in Fig. S11. Human prior methods generally render better images with more precise human shape and texture compared to category-agnostic methods, especially in the *Texture-Hard* and *Deformation-Hard* cases. Moreover, in addition to the influence of conceptual difference between image blending strategies to direct radiance prediction in the main text, we also compare the generalization of image feature extractors between PixelNeRF [37] and VisionNeRF [15]. VisionNeRF [15] uses a similar structure of PixelNeRF [37], but mainly incorporates the local CNN-based image encoder with a global vision transformer. Such a design achieves better visual quality on average with sharper texture details and higher scores in both Fig. S11 and Tab. 3, since the transformer is capable to learn more global coherence features across source views.

C.2.4 Benchmarks with Sparse View Training

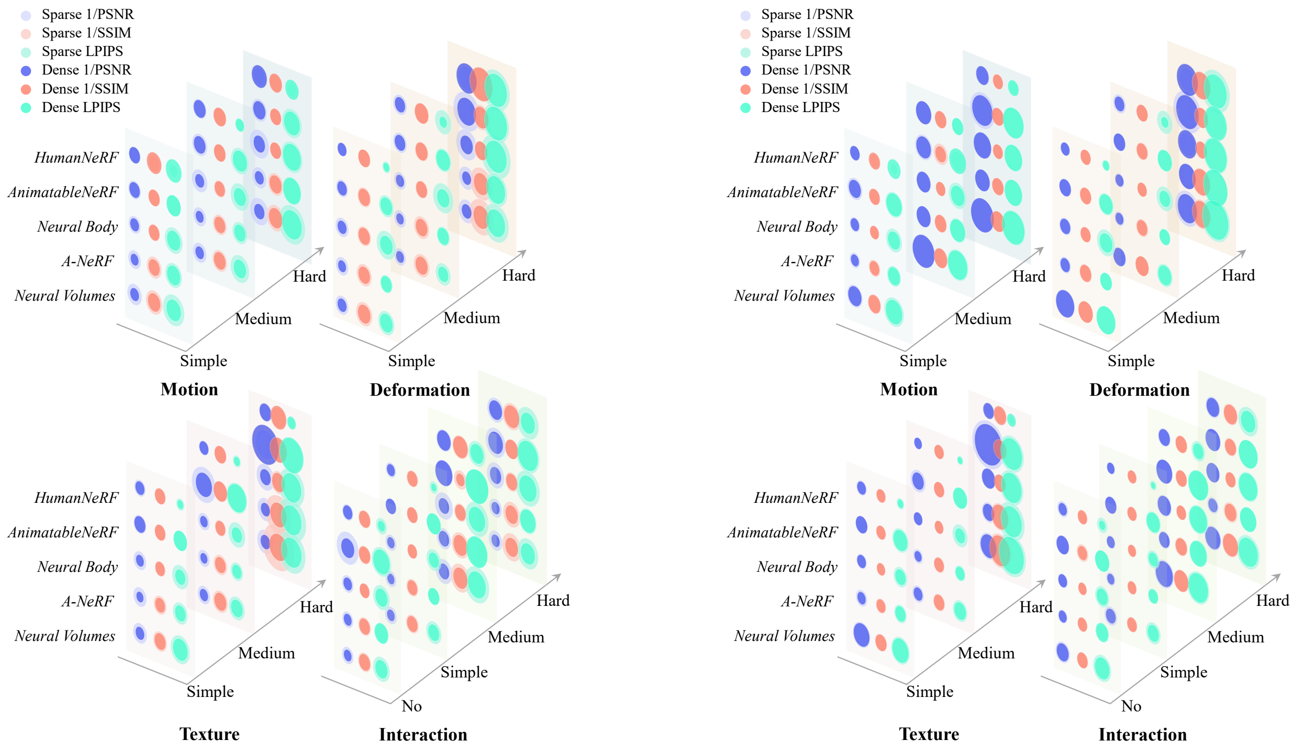
Compare with dense view human synthesis, rendering humans from sparse views or even with a monocular image setting, take a step further in narrowing the domain gap between structured in-door data capture and unstructured open-world data collection. Relaxing the requirement on structured data could help towards portable human avatar generation and may further enable applications like bullet-time rendering from online videos. Thus, aside from the main benchmarks, we also evaluate the state-of-the-art methods' performances under a spare-view training setting.

Setting. Similar to the dense novel view and novel pose experiments, we retrain separate models for each dynamic human rendering method. The key difference of the sparse view settings from the dense ones is that, each model is trained with only four balanced views, namely Camera 1, 13, 25, and 37. The other setting details are the same as the dense novel view and novel pose benchmarks.

Results. We list the quantitative results of both sparse novel view synthesis and sparse novel pose animation in Tab S3, and compare the difference between dense and sparse view training in Fig S12. For the sparse novel view synthesis task, we can observe that HumanNeRF [36] achieves the best PSNR and LPIPS metric in most of the splits, and NeuralBody [28] gets the best SSIM performance. Originally sparse-designed methods, NeuralBody [28], AnimatableNeRF [27], and HumanNeRF [36] ranked top-3 in all evaluation metrics, this phenomenon is totally different from dense results in Tab. S2. The performance gap between sparse view settings and the dense ones can be easily observed from the inflating bubbles in Fig. S12a. In the figure, the bubbles with darker colors refer to the performances under dense view settings, and the ones with lighter

	Splits	**PSNR↑**					**SSIM↑**					**LPIPS*↓**				
		NV	AN	NB	AnN	HN	NV	AN	NB	AnN	HN	NV	AN	NB	AnN	HN
Novel View Synthesis	Motion Simple	23.31	24.32	24.77	24.40	26.04	0.948	0.954	0.972	0.975	0.973	84.02	66.33	70.53	57.75	32.26
	Motion Medium	23.79	24.18	22.89	23.01	23.84	0.955	0.959	0.972	0.973	0.963	73.66	59.10	72.82	70.43	37.83
	Motion Hard	21.69	23.31	21.45	22.84	25.02	0.943	0.952	0.971	0.970	0.972	104.23	75.17	83.73	78.06	35.29
	Deformation Simple	24.56	24.85	24.45	25.27	27.59	0.951	0.956	0.964	0.964	0.971	61.39	46.81	72.32	56.19	30.46
	Deformation Medium	25.40	25.36	25.29	24.07	24.06	0.953	0.955	0.972	0.965	0.959	57.19	49.92	76.15	60.26	61.82
	Deformation Hard	22.89	23.08	21.34	20.18	22.80	0.931	0.935	0.964	0.948	0.937	96.88	84.66	110.75	98.65	104.05
	Texture Simple	23.58	24.00	24.28	24.31	25.24	0.959	0.964	0.980	0.974	0.974	76.06	58.66	64.69	54.79	32.99
	Texture Medium	24.41	25.08	25.01	20.37	26.41	0.953	0.959	0.975	0.962	0.967	63.33	49.54	67.54	82.32	32.62
	Texture Hard	22.39	22.69	21.89	21.45	25.95	0.920	0.926	0.951	0.950	0.958	110.36	115.52	103.13	86.19	26.35
	Interaction No	24.83	25.23	24.71	20.70	25.28	0.961	0.964	0.980	0.964	0.966	68.06	49.58	67.65	79.91	45.11
	Interaction Simple	24.67	25.38	25.39	25.47	25.95	0.963	0.968	0.982	0.977	0.971	61.90	43.24	59.19	57.79	34.74
	Interaction Medium	22.11	23.23	21.67	21.84	22.84	0.938	0.943	0.965	0.961	0.948	99.50	80.25	94.61	91.78	69.95
	Interaction Hard	23.55	24.06	21.93	21.15	22.18	0.939	0.944	0.961	0.959	0.941	84.15	70.00	96.80	94.30	87.81
	Overall	23.63	24.21	23.47	22.70	24.86	0.947	0.952	0.970	0.965	0.962	80.06	65.29	79.99	74.49	48.56
Novel Pose Animation	Motion Simple	21.17	24.05	24.31	21.34	25.62	0.941	0.952	0.972	0.953	0.972	93.60	66.80	76.33	82.37	32.45
	Motion Medium	19.40	21.64	21.50	20.53	21.04	0.941	0.947	0.968	0.936	0.951	100.36	78.29	83.66	61.69	54.61
	Motion Hard	19.09	21.39	20.64	18.85	23.58	0.938	0.948	0.967	0.950	0.967	112.72	80.41	87.99	95.97	40.50
	Deformation Simple	20.73	23.89	23.43	23.00	26.17	0.938	0.950	0.962	0.960	0.966	87.29	55.29	81.29	67.70	35.39
	Deformation Medium	22.43	24.89	25.19	23.12	22.76	0.943	0.951	0.971	0.952	0.955	70.79	52.05	75.68	61.72	70.44
	Deformation Hard	19.13	20.83	21.34	18.52	21.41	0.920	0.923	0.964	0.941	0.929	131.86	108.27	110.74	91.70	129.75
	Texture Simple	20.44	23.42	24.27	23.07	24.20	0.950	0.962	0.980	0.972	0.972	87.66	59.22	65.08	77.59	36.64
	Texture Medium	23.16	24.56	25.14	21.23	26.63	0.950	0.957	0.977	0.957	0.967	69.64	51.30	72.68	69.96	30.81
	Texture Hard	20.23	21.90	21.65	17.93	25.66	0.911	0.921	0.951	0.932	0.956	136.26	124.49	111.66	116.84	24.88
	Interaction No	21.45	24.90	24.37	22.08	23.52	0.953	0.962	0.979	0.948	0.962	88.23	53.08	69.11	70.76	50.70
	Interaction Simple	22.18	25.05	25.32	22.77	25.99	0.958	0.967	0.982	0.970	0.971	71.61	43.02	59.29	72.53	32.95
	Interaction Medium	19.83	22.83	21.33	20.60	22.40	0.931	0.943	0.966	0.959	0.947	107.04	76.08	94.86	92.98	66.90
	Interaction Hard	20.55	23.06	21.93	21.02	21.42	0.927	0.937	0.961	0.945	0.934	110.91	81.78	96.79	92.63	95.35
	Overall	20.75	23.26	23.11	21.08	23.88	0.939	0.948	0.969	0.952	0.958	97.54	71.54	83.47	81.11	53.95

Table S3: **Benchmarks with sparse view training.** We abbreviate NeuralVolumes [18] as 'NV', A-NeRF [31] as 'AN', NeuralBody [28] as 'NB', AnimatableNeRF [27] as 'AnN' and HumanNeRF [36] as 'HN'. Cell color 🟥 🟧 🟨 indicate the best, second best, and third best performance in the same split respectively.



(a) Novel view synthesis with different training view numbers.

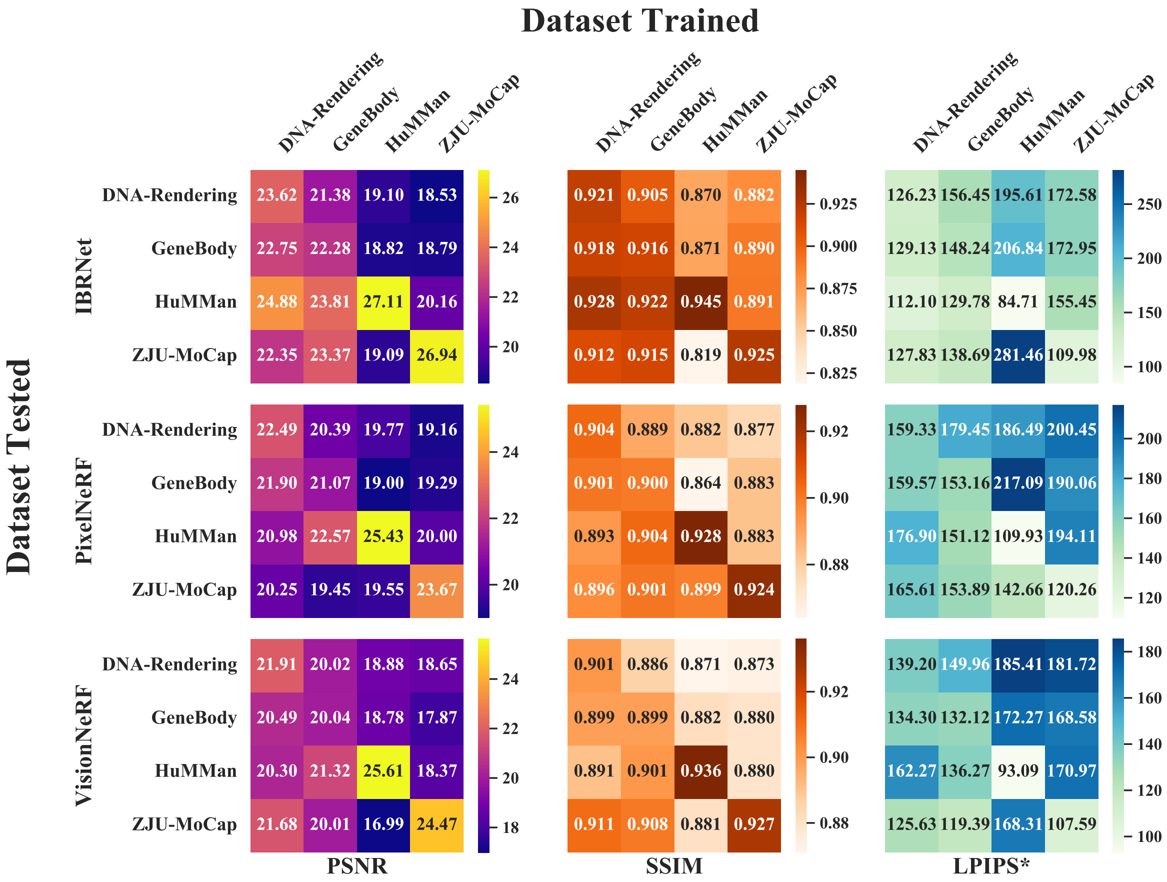
(b) Novel pose animation with different training view numbers.

Figure S12: **Visualization of quantitative comparison between dense view training and sparse view training.** For (a) novel view and (b) novel pose tasks, we compare the dense view training (42 views) and sparse view (4 views) training on different data splits. Bubbles in dark tones are with dense view training, and other ones in light tones are with sparse view training. Noted that the scale used here is different from Fig. 4 for better comparison visualization.

colors refer to the results under sparse view settings. The underlying reason for the phenomenon lies in the natural difficulty in sparse neural field supervision – the fewer training observations require a network to have the more powerful capability on learning multi-view relationships (both interpolation and extrapolation) and proper hallucinating, to approximate precise geometry. NeuralBody [28], AnimatableNeRF [27], and HumanNeRF [36] all adopt strong human priors with SMPL mesh, blend weights, and motion priors. Thus, they are more robust to sparse observations. In contrast, A-NeRF [31] integrates only skeleton prior that is sparse in human shape representation, and the category-agnostic method NeuralVolumes [18] relies on dense observations to overfit to a particular distribution. These two methods' performances drop significantly when given fewer views during training. Similar to the observation from dense novel view benchmarks, the current dynamic human method performs unsatisfactorily when *Deformation* and *Texture* difficulty increase. Due to the complex texture and off-body non-rigidity, the results in the sparse setting further enlarge the gap with the dense one. In contrast with the phenomena in sparse novel view synthesis, the quantitative results on the sparse novel pose animation task (Tab. S3) show similar trends compared to the dense view setting. Specifically, HumanNeRF [36] and NeuralBody [28] perform the best among the three metrics. A-NeRF [31] shows better pose generalization ability compared to AnimatableNeRF [27], considering the fact that AnimatableNeRF [28] performs better in novel view synthesis tasks given seen human poses. When comparing the differences between dense and sparse settings, quantitative results display a relatively smaller performance drop. The phenomenon reveals that, the training observations from both dense and sparse view settings are not adequate enough for the benchmark methods to learn a compact dynamic field for unseen poses.

D. Cross-dataset Details

In this section, we provide more details as a supplement to the cross-dataset evaluation mentioned in the main paper. Specifically, we first describe the criteria for selecting the compared datasets, and review the key attributes of these datasets in Sec. D.1. Then, we introduce the implementation and setting details in Sec. D.2. We discuss the results in Sec. D.3. Finally, we analyze the impact of color consistency on multi-camera datasets in Sec. D.4.



IBRNet

Dataset Tested	PSNR				SSIM				LPIPS*			
	DNA-Rendering	GeneBody	HuMMan	ZJU-MoCap	DNA-Rendering	GeneBody	HuMMan	ZJU-MoCap	DNA-Rendering	GeneBody	HuMMan	ZJU-MoCap
DNA-Rendering	23.62	21.38	19.10	18.53	0.921	0.905	0.870	0.882	126.23	156.45	195.61	172.58
GeneBody	22.75	22.28	18.82	18.79	0.918	0.916	0.871	0.890	129.13	148.24	206.84	172.95
HuMMan	24.88	23.81	27.11	20.16	0.928	0.922	0.945	0.891	112.10	129.78	84.71	155.45
ZJU-MoCap	22.35	23.37	19.09	26.94	0.912	0.915	0.819	0.925	127.83	138.69	281.46	109.98

PixelNeRF

Dataset Tested	PSNR				SSIM				LPIPS*			
	DNA-Rendering	GeneBody	HuMMan	ZJU-MoCap	DNA-Rendering	GeneBody	HuMMan	ZJU-MoCap	DNA-Rendering	GeneBody	HuMMan	ZJU-MoCap
DNA-Rendering	22.49	20.39	19.77	19.16	0.904	0.889	0.882	0.877	159.33	179.45	186.49	200.45
GeneBody	21.90	21.07	19.00	19.29	0.901	0.900	0.864	0.883	159.57	153.16	217.09	190.06
HuMMan	20.98	22.57	25.43	20.00	0.893	0.904	0.928	0.883	176.90	151.12	109.93	194.11
ZJU-MoCap	20.25	19.45	19.55	23.67	0.896	0.901	0.899	0.924	165.61	153.89	142.66	120.26

VisionNeRF

Dataset Tested	PSNR				SSIM				LPIPS*			
	DNA-Rendering	GeneBody	HuMMan	ZJU-MoCap	DNA-Rendering	GeneBody	HuMMan	ZJU-MoCap	DNA-Rendering	GeneBody	HuMMan	ZJU-MoCap
DNA-Rendering	21.91	20.02	18.88	18.65	0.901	0.886	0.871	0.873	139.20	149.96	185.41	181.72
GeneBody	20.49	20.04	18.78	17.87	0.899	0.899	0.882	0.880	134.30	132.12	172.27	168.58
HuMMan	20.30	21.32	25.61	18.37	0.891	0.901	0.936	0.880	162.27	136.27	93.09	170.97
ZJU-MoCap	21.68	20.01	16.99	24.47	0.911	0.908	0.881	0.927	125.63	119.39	168.31	107.59

Figure S13: **Cross-dataset affinity map for category-agnostic generalization methods**. We crossly evaluate models trained on each dataset and plot their performance on testing splits on each dataset. The PSNR, SSIM, and LPIPS* are plotted in separate matrices.

D.1. Compared Datasets

To evaluate the potential of the proposed dataset on boosting algorithms' generalizability from the data engineering aspect, we compare the proposed dataset with the most commonly used human-centric multiview datasets on the generalizable neural rendering task. For a fair comparison, we select datasets with foreground segmentation annotations and dense camera views. Thus, several well-known datasets are not suitable for this evaluation. For example, Human3.6M [9] only contains four RGB cameras, CMU Panoptic [11] and AIST++ [32, 14] lack official segmentation annotation[3]. We select ZJU-MoCap [28], HuMMan [1] and GeneBody [3] for comparison. In this subsection, we discuss their main features and their adaption to generalizable methods.

ZJU-MoCap [28] is currently the most widely used dataset in human neural rendering domain. It contains 10 multiview performance sequences, with accurate camera calibration, human segmentation as well as SMPL annotation. The main drawback of this dataset is the lack of diversity in clothing and motion and without human objection interaction. Besides, color consistency design might be ignored in ZJU-MoCap [28], where obvious color differences can be observed between neighboring cameras, as shown in Fig. S16a. When training generalizable models on this dataset, we adopt the official splits and follow the implementation in KeypointNeRF [22].

HuMMan [1] is a human action dataset with data captured under 10 synchronized Kinect RGB-D cameras. It contains 400k sequences and 500 human actions which emphasize muscle-related movements. The clothing diversity is marginal where most subjects wear sports and daily costumes, and there is no human-object interaction either. As the source images come from the Kinect sensor, they might be stuck in low-quality, and obvious color differences can be found in HuMMan [1] dataset. Note that the full dataset is still unreachable, we train models on the released version, with its official list that contains a training split with 317 sequences and a testing split with 22 sequences. Noted that different from other datasets, where cameras are organized in a world coordinate near the origin that axis alignment with the real world, HuMMan [1] uses a coordinate system relative to the first camera. Thus, we make a rigid transfor-

---

[3] Some human rendering methods [13, 2] use their own tools to generate mask, we exclude these mask sources for fairness.

mation to eliminate the coordinate system difference.

GeneBody [3] is a recent multi-view human performance capture dataset, which contains relatively wide diversity coverage among clothing, motion, and interactions. It captures human performance with 48 synchronized 5MP cameras with a low proportion-in-view of the main subject, where the average height of the human bounding box is around 600 pixels. We use its official splits with 40 training sequences and 10 testing sequences.

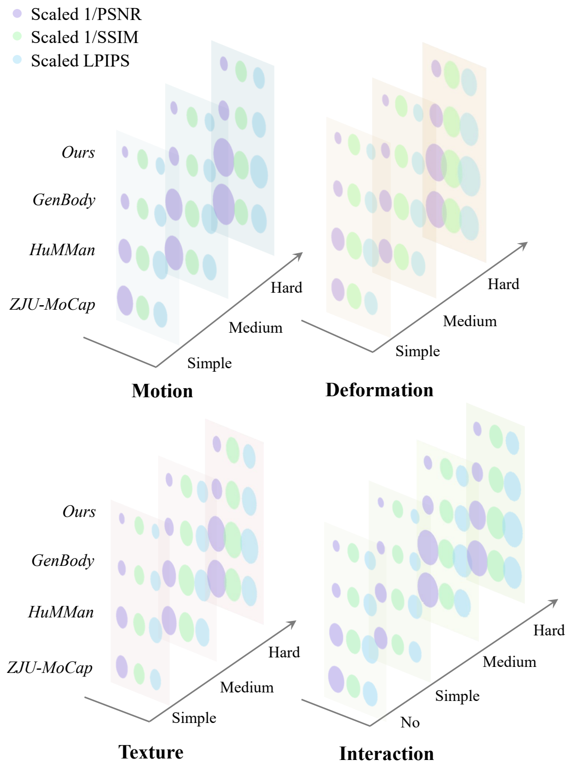


Figure S14: **Cross-dataset evaluation on our DNA-Rendering data splits**. We visualize the performance of models trained on different datasets on the proposed dataset's splits.

D.2. Detailed Settings

Multiple factors might affect the impact assessment across different datasets on neural rendering tasks, *eg.,* the proportion of subjects in camera views, data scale, annotation accuracy, source view selection, training status, *etc.* Our main goal is to investigate where the *diversity* of the proposed dataset can benefit the generalization of human rendering. We conduct the experiments in the following settings. 1) In order to ensure the fairness of dataset comparison, we need to unify several input conditions, *e.g.,* number and resolution of source views, *etc.* Meanwhile, we only investigate the category-agnostic generalizable methods, namely PixelNeRF [37], IBRNet [35] and VisionNeRF [15], to avoid difference of input and accuracy of human prior in multiple datasets. 2) Evaluating the whole object-centric images with background removal produces a large rendering metric gap if the center subjects' proportions in camera views differ a lot. Thus, different from the novel pose benchmark in Sec. 4 and Sec. C.2.3 where a half resolution is used, we crop the subjects out from the original image in all datasets with square bounding boxes and resize them to 512 × 512 resolution for both source views and target views[4]. 3) As the discussed datasets are all captured in dense circling camera arrays, we manually select four balanced views as the reference views at the same height that have a roughly 90-degree interval. 4) During training, we use the same learning rate over all datasets and stop the training process with the same global step. Each model is trained on one 8-V100 machine with distributed data-parallel stopping at $200k$ iterations. 5) Finally, we train and evaluate all the models based on the official splits of each dataset. For the comparable data volume magnitude of test samples, we size the data volume of test frames or test views on each dataset with $\langle$ 45 frame-skip, 18 test views $\rangle$ on GeneBody and DNA-Rendering , $\langle$ 45 frame-skip, 12 uniform sampled test views $\rangle$ on ZJU-MoCap , and $\langle$ 8 frame-skip, 6 test views $\rangle$ on HuMMan, respectively.

D.3. Additional Results

In the main paper, we present the average PSNR performance over all three general scene methods, *i.e.,* PixelNeRF [37], IBRNet [35], and VisionNeRF [15]. Here, we present the performances of these methods individually (Fig. S13), and illustrate the qualitative results (Fig. S15). We unfold the result analysis in terms of in-domain, and cross-domain in Sec. D.3.1 and Sec. D.3.2 respectively. A discussion on the impact of color consistency is discussed in Sec. D.4.

D.3.1 In-domain

In-domain refers to the problem of evaluating models with the trainset and testset sharing the same underlying data distribution. We observe two key phenomena: 1) models trained on datasets with low data diversity achieve better in-domain results. As shown in the diagonal elements of the matrices in Fig. S13, for in-domain generalization performance, methods trained on HuMMan [1] and ZJU-MoCap [28] achieve the best and second performances with relatively appealing metric values. In contrast, their in-domain performances on GeneBody and DNA-Rendering are worse than the other two datasets. Noting that test sets in ZJU-MoCap and HuMMan only contain cases with textureless clothing and easy motion illustrated in Fig. S15, which are easy cases in terms of data

---

[4] Note that this setting leads to the absolute values in cross-dataset evaluation worse than the ones in novel identity task, due to the larger proportion of human foreground.

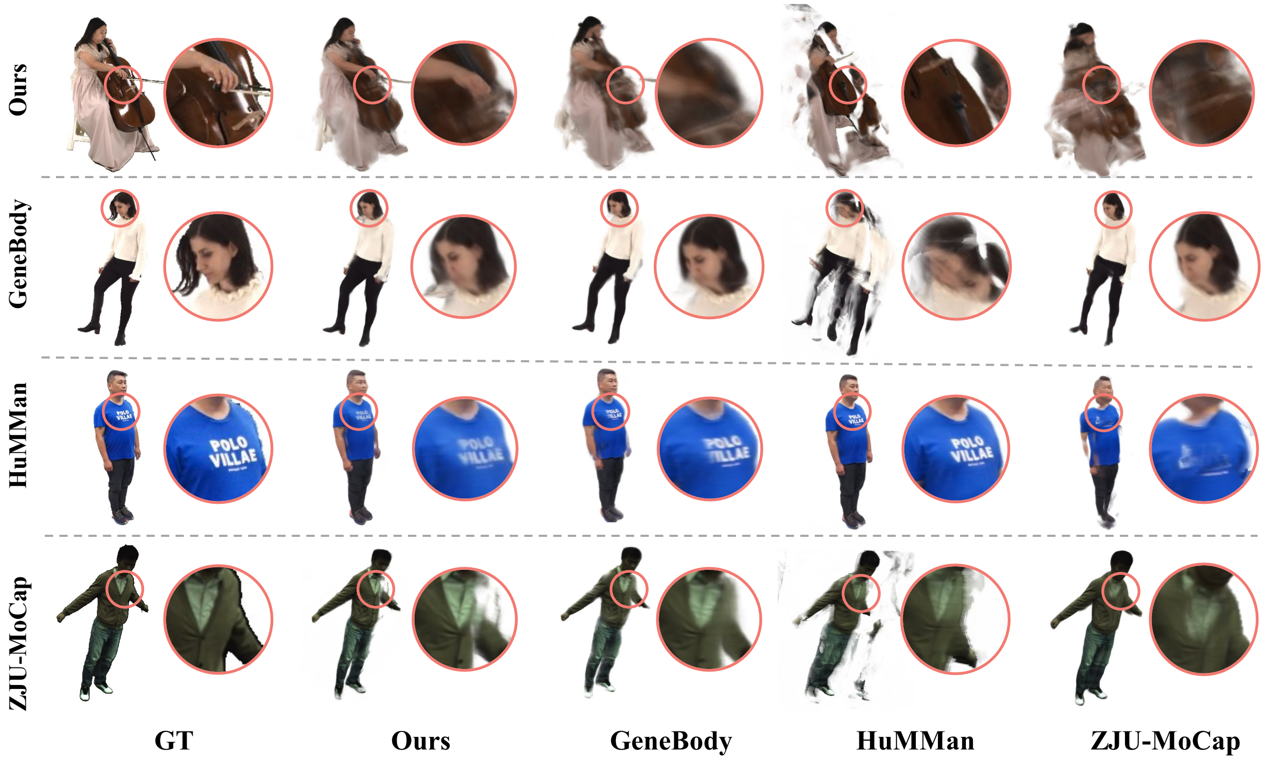


Figure S15: **Qualitative cross-dataset results**. We demonstrate samples from different datasets (left labels) generated by IBRNet [35] models trained on different datasets (bottom labels).

difficulty. Besides, the nature of textureless data and easy human geometry is relatively friendly to category-agnostic generalization methods that conduct multiview image feature aggregation in a common manner without geometry prior. 2) Larger data volume and diversity boosts in-domain performance. Concretely, like the proposed dataset, GeneBody [3] contains a train and test split with a wide distribution of clothing, accessories, and motion, while with far less data volume and diversity compared to DNA-Rendering (about 10% data volume of our dataset). Despite both test sets of GeneBody and proposed dataset containing cases with even distribution in multiple difficulties, all three methods demonstrate our boost on in-domain performance.

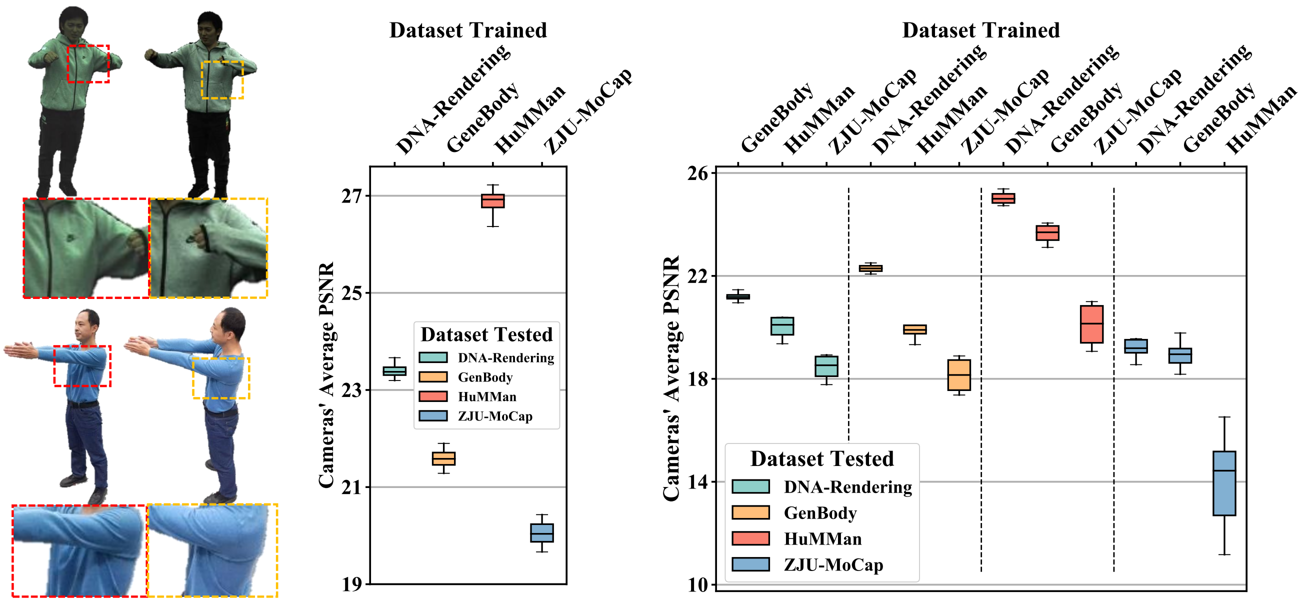
D.3.2 Cross-domain

Cross-domain refers to directly evaluating the pre-trained model on one dataset to the test split of another dataset, which is represented by the off-diagonal elements in Fig. S13. We expand the result analysis in two folds: 1) datasets with large variations of data attributes and difficulties boost the cross-domain generalization. Higher performance degradation can be observed in off-diagonal elements in each row in ZJU-MoCap [28] and HuMMan [1] in Fig. S13. Besides, even when ZJU-MoCap [28] trained model test on cross-domain case with easy clothing and motion, the cross-domain performance is still far from acceptable, refer to left most blue T-shirt case in Fig. S15. On the contrary, GeneBody [3] and DNA-Rendering experience a very marginal degradation when evaluating other test sets. 2) Larger data volume and diversity can also boost cross-domain robustness. As mentioned in the in-domain part, DNA-Rendering is 10 times than GeneBody in terms of data volumes, such improvement helps increase the model's generalization ability in considering margin. In the overwhelming majority of first-row elements, models with the proposed dataset get the best cross-domain results and even perform better than in-domain results of GeneBody.

Cross-domain on DNA-Rendering Splits. To further investigate the performance of different dataset-trained models' performance on different human performance dimensions and difficulties, we visualize their performance on DNA-Rendering splits in Fig. S14. We plot the in-domain result of our dataset-trained model as a reference bar. GeneBody [3] has a relatively wide range of dimensions across our dimension and it achieves the best rendering quality among all splits. On the other hand HuMMan [1] and ZJU-MoCap [28], only contain easy clothing, motion, and interaction, the performance on more difficult splits drop significantly espec compared to GeneBody [3].

D.4. Impact of Color Consistency

To further analyze the impact of color consistency of training dataset in generalizable rendering, we unfold the



(a) (b) (c)

Figure S16: **Cross-dataset evaluation across views.** (a) Examples of the color differences between neighboring cameras in ZJU-MoCap [28] (top) and HuMMan [1] (bottom). (b) **In-domain** statistics on view average PSNR when trained and tested on the same dataset. (c) **Cross-domain** statistics on view average PSNR when trained and tested on different datasets. In both (b) and (c), the horizontal axis means different datasets trained, and the color of the box separates the datasets tested. The line in the middle indicates the median value, the box indicates the lower to the higher quartile, and the whiskers indicate the range of average PSNR across views.

stats across views on the cross-dataset results. Due to the different groundtruth in different views, it is hard to draw any conclusion from any single frame. Thus, we expand the average PSNR across camera views and analyze the statistics. Noted that we only select the test views which have very close angle distances from the nearest source view, to erase the performance gap from the viewpoints. The average PSNR across testing cameras is plotted in Fig. S16. More concretely, we visualize the in-domain statistics of cameras' average PSNR in Fig. S16b. When training and testing a model on the same dataset, the camera color distinction will remain constant. Models trained on the datasets that cannot ensure color consistency across views (illustrated in Fig. S16a) might treat the color difference of different views as the view-depend effect and memorize it. The variance of cameras' average performance in such datasets is slightly higher than GeneBody [3] and the proposed dataset. Cross-domain generalization span on views is also plotted in Fig. S16c. Different from in-domain statistics, when generalizing on other datasets, models trained on datasets with color inconsistency all suffer a major average performance dropping, and the variance of camera performance becomes even larger. This phenomenon is very noticeable on cross-evaluation between ZJU-MoCap [28] and HuMMan [1]. While models trained on the proposed dataset as well as GeneBody [3], have very small view performance variations between each other. The increased view variation on ZJU-MoCap [28] and HuMMan [1] is due to the nature color difference on their groundtruth. In a nutshell, with the best color consistency, the proposed dataset can benefit the community by providing high-quality data with faithful probing capability across views.

E. Future Work

Leaderboard. In the current human-centric rendering community, researchers from different institutions use different datasets and experimental settings to evaluate the performances of their algorithms. There is no agreement across institutions to benchmark human rendering methods under the same criterion yet. To reduce such divergence and align the standards, we proposed a large-scale diverse dataset DNA-Rendering, and construct a complete benchmark in three human-centric rendering tasks. Benchmarks are evaluated on different data splits on different factors and difficulties. Additionally, we conduct a cross-dataset evaluation which demonstrates the proposed dataset can benefit the community from its diversity and coverage. In DNA-Rendering, all actors are signed with agreements before data collection. Thus, all of the data is with a Creative Commons license and free for use under certain usage agreements. In the future, we will host a web-based leaderboard, and release the easy-to-run tools to the community to better reduce the divergence.

Robust Human-centric Matting Refinement. In the annotation pipeline, we use Grab-cut [29] to refine bad results of CNN-based methods, yet it is still not perfect. Since per-frame human labeling is impractical due to the volume of captured data, we involve human checks over segmentation results, and only cases without major artifacts in the whole sequence across views will be released. We tried 3D methods to further refine results, *e.g.*, using Instant-NGP [24] to train with valid views and infer the bad views. However, the results are not appealing enough (blurry edges). We will further investigate more robust tools. These challenges could also benefit matting research. We believe with the development of relevant techniques, matting robustness will be improved in the near future.

New Benchmarks. In this paper, we set up three task benchmarks as a kick-off of the DNA-Rendering dataset. Our datasets could potentially be used in many other tasks related to human-centric rendering, such as garment modeling/animation and human shape completion. We encourage and welcome the community to join us to unlock more downstream tasks.

References

[1] Zhongang Cai, Daxuan Ren, Ailing Zeng, Zhengyu Lin, Tao Yu, Wenjia Wang, Xiangyu Fan, Yang Gao, Yifan Yu, Liang Pan, et al. Humman: Multi-modal 4d human dataset for versatile sensing and modeling. In *ECCV*, 2022.

[2] Yue Chen, Xuan Wang, Xingyu Chen, Qi Zhang, Xiaoyu Li, Yu Guo, Jue Wang, and Fei Wang. Uv volumes for real-time rendering of editable free-view human performance. In *CVPR*, 2023.

[3] Wei Cheng, Su Xu, Jingtan Piao, Chen Qian, Wayne Wu, Kwan-Yee Lin, and Hongsheng Li. Generalizable neural performer: Learning robust radiance fields for human novel view synthesis. *arXiv preprint*, 2022.

[4] EasyMocap Contributors. Easymocap - make human motion capture easier. `https://github.com/zju3dv/EasyMocap`, 2021.

[5] T. Derose, M. Kass, and T. Truong. Subdivision surfaces in character animation. In *SIGGRAPH*, 1998.

[6] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *SIGGRAPH*, 1996.

[7] Peter Hedman and Johannes Kopf. Instant 3d photography. *TOG*, 2018.

[8] Nikolas Hesse, Sergi Pujades, Javier Romero, Michael J Black, Christoph Bodensteiner, Michael Arens, Ulrich G Hofmann, Uta Tacke, Mijna Hadders-Algra, Raphael Weinberger, et al. Learning an infant body model from rgb-d data for accurate full body motion analysis. In *MICCAI*, 2018.

[9] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *TPAMI*, 2013.

[10] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[11] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *ICCV*, 2015.

[12] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint*, 2013.

[13] Youngjoong Kwon, Dahun Kim, Duygu Ceylan, and Henry Fuchs. Neural human performer: Learning generalizable radiance fields for human performance rendering. *NeurIPS*, 2021.

[14] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *ICCV*, 2021.

[15] 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 *WACV*, 2023.

[16] Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian L Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Real-time high-resolution background matting. In *CVPR*, 2021.

[17] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, 2016.

[18] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: learning dynamic renderable volumes from images. *TOG*, 2019.

[19] Matthew Loper, Naureen Mahmood, and Michael J Black. Mosh: Motion and shape capture from sparse markers. *TOG*, 2014.

[20] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *TOG*, 2015.

[21] MR Luo, G Cui, and B Rigg. The development of the cie 2000 colour-difference formula: Ciede2000. *Color Research and Application*, 2001.

[22] Marko Mihajlovic, Aayush Bansal, Michael Zollhoefer, Siyu Tang, and Shunsuke Saito. Keypointnerf: Generalizing image-based volumetric avatars using relative spatial encoding of keypoints. In *ECCV*, pages 179–197, 2022.

[23] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[24] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *TOG*, 2022.

[25] Priyanka Patel, Chun-Hao P Huang, Joachim Tesch, David T Hoffmann, Shashank Tripathi, and Michael J Black. Agora: Avatars in geography optimized for regression analysis. In *CVPR*, 2021.

[26] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and

Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, 2019.

[27] 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 *ICCV*, 2021.

[28] 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.

[29] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. " grabcut" interactive foreground extraction using iterated graph cuts. *TOG*, 2004.

[30] Qing Shuai, Chen Geng, Qi Fang, Sida Peng, Wenhao Shen, Xiaowei Zhou, and Hujun Bao. Novel view synthesis of human interactions from sparse multi-view videos. In *SIGGRAPH*, 2022.

[31] 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.

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

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

[34] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *NeurIPS*, 2021.

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

[36] 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*, 2022.

[37] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021.

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

[39] Zerong Zheng, Tao Yu, Yebin Liu, and Qionghai Dai. Pamir: Parametric model-conditioned implicit representation for image-based human reconstruction. *TPAMI*, 2021.