ShareUs: A Unified System for Multi-avatar Reconstruction and Appearance-Motion Editing

Xianyong Fang
Anhui University
Hefei, Anhui Province, P.R. China, 230601
fangxianyong@ahu.edu.cn

Jiarui Li
Anhui University
Hefei, Anhui Province, P.R. China, 230601
e23201123@stu.ahu.edu.cn

Zongxin Shang
Anhui University
Hefei, Anhui Province, P.R. China, 230601
e23201114@stu.ahu.edu.cn

Renlong Dai
Anhui University
Hefei, Anhui Province, P.R. China, 230601
e23301311@stu.ahu.edu.cn

Linbo Wang
Anhui University
Hefei, Anhui Province, P.R. China, 230601
wanglb@ahu.edu.cn

Zhengyi Liu
Anhui University
Hefei, Anhui Province, P.R. China, 230601
liuzywen@ahu.edu.cn

Abstract

Conventional monocular video-based digital human reconstruction and editing methods primarily focus on single individuals, lacking a unified framework for handling multiple avatars. To address this limitation, this paper proposes a novel approach that uses a single model to reconstruct and edit multiple avatars from diverse monocular videos. It consists of three key innovations. First, it adopts normal-guided 3D Gaussians initialization to unify avatars into a canonical space, which helps boost convergence and editing compatibility. Second, it introduces a thin-triplane-based feature space. This space is split into channel-partitioned subspaces with self-attention, which enables pose-independent and detail-rich representation while supporting convenient cross-avatar editing. Third, it employs a hybrid pose encoder that integrates global self-attention and local cross-attention to capture pose-dependent surface details. This encoder is further combined with in-distribution pose interpolation to achieve better generalization to diverse poses. Experimental results demonstrate that the proposed method achieves a balance between reconstruction performance and resource utilization, enhances rendering quality for novel poses, and supports effective cross-avatar editing.

***Keywords:** Multi-avatar, reconstruction, garment editing.*

1. Introduction

Creating high-fidelity clothed digital humans from monocular videos has recently seen significant progress [1, 4, 5, 9, 10, 14, 15, 21, 29]. These methods enable detailed reconstruction of human geometry and appearance, and further support garment transfer [2, 3, 16, 37] and pose retargeting [8, 24, 28, 34, 35]. Such technologies hold transformative potential across a wide range of applications in 3D computer vision and computer graphics, including virtual reality (VR), augmented reality (AR), holographic communication, film production, and game development.

Despite these advances, existing approaches face two major limitations. First, most methods focus on reconstructing a single avatar at a time, requiring a separate model to be trained for each individual. This per-avatar modeling strategy leads to a substantial total parameter count when scaling to multiple avatars, resulting in high storage and computational overhead, and further hinders cross-avatar editing due to the absence of a shared representational space. Second, methods that reconstruct avatars from single images (IDOL [40], LHM [25]) often produce low-fidelity results in unobserved regions, as they lack the geometric constraints provided by multi-view supervision. As a result, the development of a unified framework capable of simultaneously reconstructing multiple high-fidelity avatars and supporting efficient cross-avatar editing remains a critical yet challenging goal.

Addressing this challenge is inherently difficult due to substantial inter-subject variations in body shape and

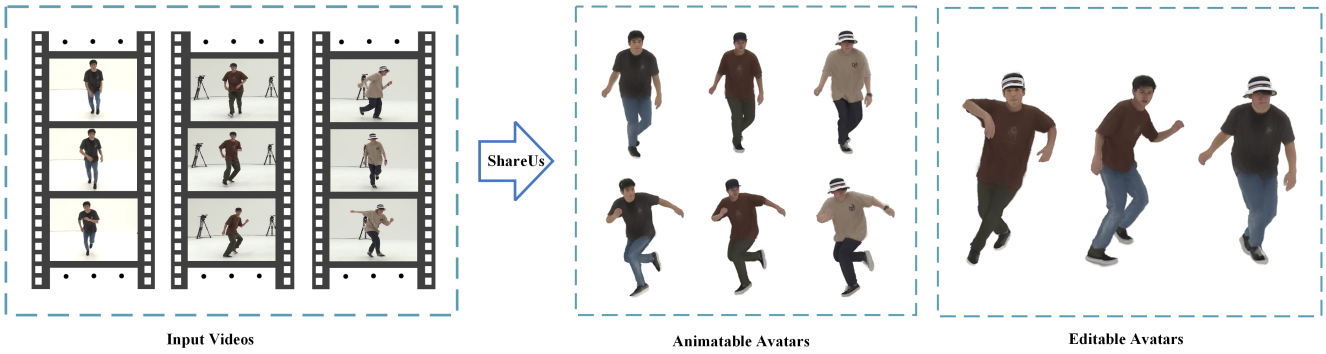


Figure 1: ShareUs accepts videos of multiple subjects (with three examples provided here) and learns their appearances using a single unified model. The system can then freely animate and edit the reconstructed avatars: for example, applying the same poses to all avatars, or modifying their clothing and poses by swapping the characters.

clothing style, as well as strong intra-subject appearance changes caused by pose-dependent deformations such as fabric wrinkles. To achieve robust multi-avatar reconstruction and editing, three key requirements must be satisfied. First, a unified representation is needed to allow a single model to reconstruct and edit multiple avatars within a shared framework. Second, each avatar must be endowed with pose-invariant personalized features that preserve its unique identity across poses. Third, the system must model pose-dependent dynamic features to capture pose-specific appearance details (e.g., fabric wrinkles, muscle contours) that vary with poses.

To meet these requirements, we propose the following design principles:

1. For a unified multi-avatar representation, we adopt a shared neutral SMPL-X template [22], subdivided into dense vertices, as the canonical body proxy. Each vertex serves as the center of a 3D Gaussian Splat (3DGS) [11]. By initializing the Gaussians with orientations aligned to the local surface normals of the SMPL-X mesh, we construct a canonical Gaussian space where all avatars share consistent topology and semantically aligned anatomical structures. This shared space enables joint modeling of multiple avatars and facilitates efficient cross-avatar operations within a single network.

2. Each avatar is represented by a thin-triplane-based feature space, which leverages multiple thin-triplanes to encode high-resolution surface details. These learnable feature spaces encode identity-specific, pose-invariant characteristics, which form a shared "asset pool" across avatars, and thus facilitates convenient cross-avatar editing.

3. To capture pose-dependent dynamics such as clothing wrinkle, we introduce a hybrid pose encoder with in-distribution (ID) pose interpolation. The encoder combines global self-attention to model long-range joint dependencies and local cross-attention to refine segment-specific interactions. For out-of-distribution (OOD) poses, we employ a weighted interpolation strategy over key ID poses, enabling robust generalization to novel poses while preserving temporal coherence.

These designs enable our framework ShareUs to reconstruct multiple high-fidelity avatars from monocular videos with one model, and support cross-avatar editing. Experimental evaluations confirm that ShareUs outperforms state-of-the-art methods on novel pose synthesis and achieves convenient cross-avatar editing.

2. Related Work

2.1. Video-based Single-avatar Reconstruction and Editing

Notable progress has been made in photorealistic reconstruction of human geometry and appearance using neural implicit fields (NeRF) [19] or 3D Gaussian Splatting (3DGS) [11]. These methods [4–6, 9, 12, 20, 24, 28, 34, 35] typically learn a 3D neural representation in a canonical space, which is then deformed into the observation space of different video frames for self-supervised training. The resulting dynamic neural representations enable novel view synthesis and pose-driven animation.

However, existing methods are limited to per-subject reconstruction and lack support for appearance editing. In contrast, our approach enables simultaneous reconstruction of multiple subjects and supports cross-subject garment editing.

Separately, some studies have focused on digital human reconstruction with decoupled body and clothing models [2, 3, 16]. Their core strategy involves using human pars-

ing to obtain semantic categories, which then supervise the prediction of semantic body parts. Notably, these methods still require training separate models for individual subjects to enable garment editing.

2.2. Video-based Multi-avatar Reconstruction and Editing

Learning a unified neural representation for multiple subjects from videos remains a relatively underexplored problem. Existing works [18, 36], focus on generalizable point cloud models but rely heavily on dense 3D ground truths across various poses for supervision, limiting their practicality.

Recently, MIGS [1] has addressed multi-subject generalization from videos by constructing a high-order tensor to model the learnable 3DGS parameters of all training identities. However, this method reconstructs 3D Gaussians for each subject in an unstructured manner. The absence of semantic correspondence between individual Gaussian primitives inherently precludes support for garment editing, which is a critical limitation for interactive multi-avatar applications.

Other lines of research [25, 26, 40] target single-image-based avatar reconstruction for arbitrary individuals. For example, AniGs [26] leverages a reference-image-guided video generation model to produce high-quality multi-view canonical human images (and their corresponding normals) from a single input image. These generated images are then used to optimize the 4DGS parameters of the target avatar, but this approach remains inherently per-subject, as it requires separate optimization for each individual. Methods like IDOL [40] and LHM [25], by contrast, map single images directly to Gaussian attributes in a canonical space, but they overlook the impact of pose variations on surface details, resulting in limited fidelity for pose-dependent rendering.

In contrast, our method unifies all subjects into a shared canonical Gaussian space. It further disentangles pose-independent and pose-dependent features via thin-triplane-based feature spaces and a hybrid pose encoder: pose-independent features capture the personalized surface characteristics of each subject, while pose-dependent features model pose-induced changes in surface details. This design enables three key capabilities: full-body motion retargeting, high-fidelity rendering, and cross-subject editing, effectively addressing the core limitations of prior works.

3. Method

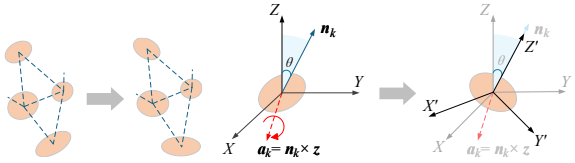
ShareUs is built on a unified framework (Figure 2) to achieve multi-avatar reconstruction and convenient cross-avatar editing from monocular videos with a single model. Its main components are outlined below: (1) It first establish the canonical space using a neutral SMPL-X template, with each vertex as the center of a 3D Gaussian. A normal-guided initialization aligns 3DGS orientations with the template's topology, ensuring consistent body part semantics across all avatars. (2) A thin-triplane-based feature space, which assigns each avatar a learnable thin-triplane set. It effectively captures the personalized pose-independent features of each avatar and thus creates its basis for further editing. (3) A hybrid human pose encoder (combining global and local attention), which models pose-surface interactions, with ID pose interpolation extending generalization to novel poses. These modules synergistically enable high-fidelity multi-avatar reconstruction and convenient cross-editing (e.g., garment transfer via region-specific feature replacement).

3.1. Normal-guided Initialization for 3D Gaussians

Unlike conventional monocular video-based digital human reconstruction methods, which typically target single individuals, our goal is to reconstruct multiple avatars from diverse monocular videos using a single unified model. To implement this unified framework, we use an SMPL-X model configured with zero shape parameters and set to a neutral gender, which serves as a shared template. The SMPL-X model (10,475 vertices and 20,908 faces) is subdivided into 41,853 vertices and 83,632 faces, with each vertex designated as the center of a 3D Gaussian primitive. Under this setup, each reconstructed avatar S is represented as a collection of M 3D Gaussian primitives:

$$S = \{S_k\}_{k=1}^{M}, \quad \text{where } S_k = \{\boldsymbol{\mu}_k, \alpha_k, \boldsymbol{r}_k, \boldsymbol{s}_k, \boldsymbol{c}_k\}. \quad (1)$$

Here, S_k denotes the k-th Gaussian primitive, parameterized by: 3D spatial position $\boldsymbol{\mu}_k \in \mathbb{R}^3$, opacity $\alpha_k \in \mathbb{R}$ (controlling visibility), axis-angle rotation $\boldsymbol{r}_k \in \mathbb{R}^3$ (defining orientation), diagonal scaling vector $\boldsymbol{s}_k \in \mathbb{R}^3$ (regulating shape), and RGB color $\boldsymbol{c}_k \in \mathbb{R}^3$ (modeling appearance).



(a) (b)

Figure 3: Normal-guided initialization for the 3D Gaussian primitives. (a) General view of the normal-guided initialization; and (b) rotation method for the k-th primitive.

In order to align each primitive with the local surface orientation of the SMPL-X mesh, a normal-guided method is proposed (Figure 3). It consists of three steps: (1) Compute the rotation axis $\boldsymbol{a}_k$ for the k-th primitive. Using the vertex normal $\boldsymbol{n}_k$ (derived from the subdivided SMPL-X mesh)

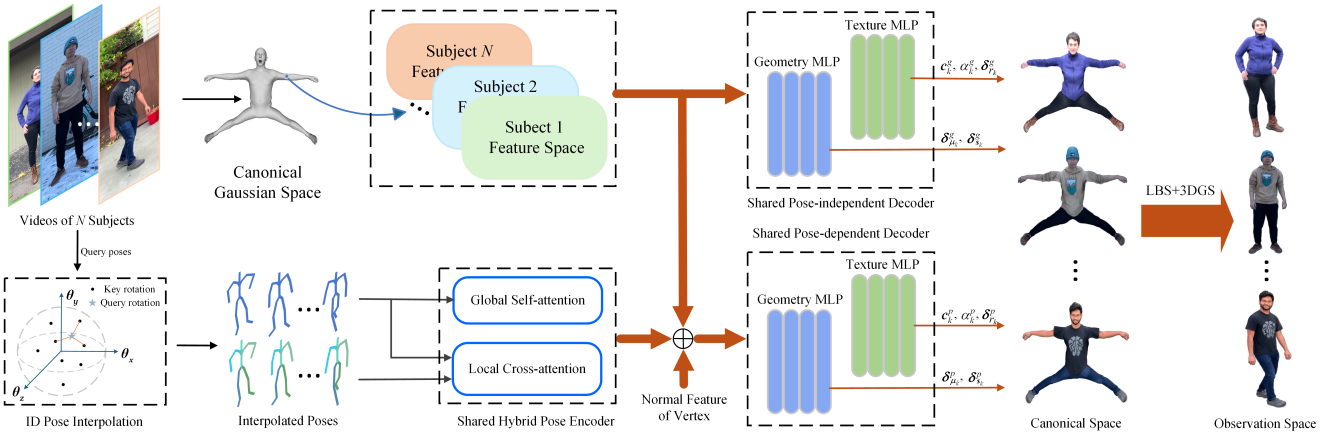


Figure 2: Overview of ShareUs. Dynamically captured subjects are initialized in a canonical Gaussian space and abstracted into a personalized thin-triplane-based feature space. Meanwhile, a hybrid pose encoder models pose-surface interactions, while in-distribution (ID) pose interpolation extends the framework's generalization to novel poses. Finally, two shared decoders predict the pose-independent and pose-dependent attributes of 3D Gaussian primitives, respectively. The aggregation of these attributes yields the deformed 3D Gaussian primitives in the observation space, which are further rendered to enable supervision against the ground truth.

and the unit vector $\boldsymbol{z}$ (representing the Z-axis of the canonical pose's local coordinate system), the axis is determined via the cross product:

$$\boldsymbol{a}_k = \boldsymbol{n}_k \times \boldsymbol{z}. \tag{2}$$

This axis captures the direction of rotation needed to align the primitive with the mesh surface. (2) Compute the rotation angle as the angle between $\boldsymbol{n}_k$ and $\boldsymbol{z}$, quantifying the mismatch between the initial orientation (aligned with $\boldsymbol{z}$) and the target orientation (aligned with $\boldsymbol{n}_k$). (3) Apply Rodrigues' rotation formula to convert the computed axis and angle into the corrected axis-angle rotation $\hat{\boldsymbol{r}}_k$, which serves as the initial rotation for the k-th primitive. For other parameters: The initial position $\hat{\boldsymbol{\mu}}_k$ of each primitive is directly mapped to the corresponding vertex of the subdivided SMPL-X mesh, and the initial scaling $\hat{\boldsymbol{s}}_k$ is computed based on the distances between the current primitive and its neighboring primitives (to ensure full coverage of local mesh geometry).

3.2. Thin-triplane-based Feature Space

Now the common structure among multiple subjects has been created. Next, an efficient and expressive feature space is expected to assign to each avatar, which should be able to capture the differences between avatars. Specifically, each avatar should get an independent feature representation for their unique personalized appearances, so that further editing is possible. The feature will be used to regress the attributes of 3D Gaussian primitives.

Triplanes [23] effectively encode multi-directional geometric cues via orthogonal projections while compressing 3D data into 2D planes. However, a single triplane exhibits three critical limitations: (1) It suffers from feature entanglement: diverse semantics (e.g., texture details, geometric edges, color gradients) are mixed within its channels, causing mutual interference between semantic signals and yielding blurred representations; (2) Its high-dimensional unstructured features lead to optimization inefficiency: the downstream MLP faces conflicting parameter updates (e.g., tuning weights for texture may disrupt geometric fitting), easily trapping the model in local optima.

To address these issues while preserving the total parameter count, we split the single triplane into K channel-partitioned thin-triplanes. Each thin-triplane forms an independent subspace, enabling implicit semantic disentanglement and reducing optimization complexity. A self-attention module is further introduced: leveraging the Q/K/V mechanism [32], it dynamically models cross-subspace correlations, adaptively weighting complementary features (e.g., texture-wrinkle interactions) for context-aware integration (Figure 4).

Technically, a 3D query point $\boldsymbol{v}_q = (x_q, y_q, z_q)^T$ is first normalized to the range $[-1, 1]$ according to the global bounding box of the point cloud and then orthogonally projected onto the XY-, YZ-, and ZX-planes. Corresponding features are extracted via bilinear interpolation ($\mathcal{B}$), and concatenated to form the k-th thin-triplane feature $\boldsymbol{f}_k$ for the 3D point:

$$\boldsymbol{f}_k(\boldsymbol{v}_q) = \bigoplus_{c \in \{xy, yz, zx\}} \mathcal{B}(\mathbf{F}_{k,c}, \mathrm{proj}_c(\boldsymbol{v}_q)), \tag{3}$$

where: (1) $\mathbf{F}_{k,c}$ denote the three orthogonal 2D feature

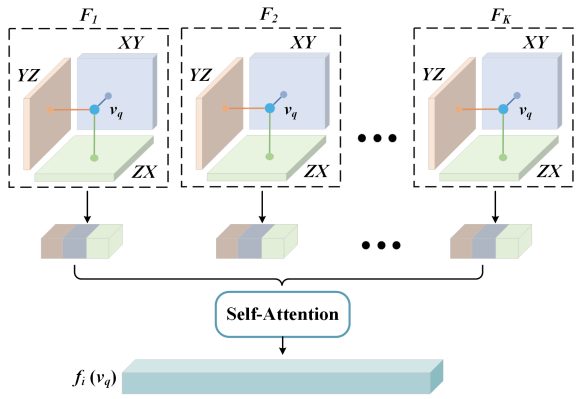


Figure 4: Principle of the thin-triplane-based feature space.

planes of the k-th thin-triplane corresponding to XY-, YZ-, ZX-directions, respectively; (2) $\text{proj}_c(\boldsymbol{v}_q)$ is the orthogonal projection of $\boldsymbol{v}_q$ onto plane c; (3) $\oplus$ represents the channel-wise concatenation operation.

For the K thin-triplane features $\{\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_K\}$ (each $\boldsymbol{f}_k \in \mathbb{R}^D$, with D as the feature dimension per thin-triplane), we apply a self-attention module to model cross-subspace correlations. The operation is compactly formulated as:

$$\text{Attn}\left(\{\boldsymbol{f}_k\}_{k=1}^K\right) = \text{SelfAtten}\left(\mathbf{F}\right) \in \mathbb{R}^{K\times D}, \qquad (4)$$

where: (1) $\mathbf{F} = [\boldsymbol{f}_1; \boldsymbol{f}_2; \ldots; \boldsymbol{f}_K] \in \mathbb{R}^{K\times D}$ is the stacked matrix of K thin-triplane features; (2) SelfAtten$(\cdot)$ denotes the standard self-attention function (implementing query/key/value projections and softmax-weighted aggregation). The attention output is reshaped to $\mathbb{R}^{KD}$, which serves as the query point's final feature $\boldsymbol{f}_i$ for downstream processing.

In practice, we set $K = 4$, where each thin-triplane has a size of $3 \times 128 \times 128 \times 8$. Here, "3" corresponds to the three orthogonal projection axes of the thin-triplane, and "8" denotes the number of feature channels per axis.

3.3. Hybrid Pose Encoder with ID Pose Interpolation

Thin-triplane-based feature space exhibits limitations in regressing the diverse properties of 3D Gaussians, particularly, they cannot model pose-dependent surface wrinkles and subtle color variations which remain invariant across different human poses. Therefore, this space cannot adapt to dynamic geometric and appearance changes induced by pose shifts. To address this, we propose a hybrid attention-based pose encoder, according to two critical observations about pose-pattern interactions. (1) Local specificity: The pose of a localized body segment (e.g., arms) exerts a more direct influence on garment wrinkles around that segment than full-body pose. (2) Global correlation: Each joint's pose is inherently coupled with its neighboring joints (e.g., shoulder pose constrains elbow movement).

The vertices of the shared template are first partitioned into five non-overlapping semantic groups, corresponding to the five core segments of the human kinematic chain: upper body ($\boldsymbol{\mathcal{V}}_{\text{ub}}$), left arm ($\boldsymbol{\mathcal{V}}_{\text{la}}$), right arm ($\boldsymbol{\mathcal{V}}_{\text{ra}}$), left leg ($\boldsymbol{\mathcal{V}}_{\text{ll}}$), and right leg ($\boldsymbol{\mathcal{V}}_{\text{rl}}$). Correspondingly, we define the global body pose parameters as $\boldsymbol{\theta}_b = [\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T, \ldots, \boldsymbol{\theta}_J^T]^T$, where J denotes the total number of kinematic joints. Specifically, the root joint, head joint, and hand joints are excluded from $\boldsymbol{\theta}_b$, as they exert minimal influence on garment wrinkles. We then split $\boldsymbol{\theta}_b$ into five segment-specific pose subsets, denoted as $\boldsymbol{\theta}_{\text{ub}}, \boldsymbol{\theta}_{\text{la}}, \boldsymbol{\theta}_{\text{ra}}, \boldsymbol{\theta}_{\text{ll}}, \boldsymbol{\theta}_{\text{rl}}$. Each subset contains pose parameters of four joints, which pertain exclusively to its corresponding segment. A one-to-one mapping is established between vertex groups and segment-specific pose subsets: All vertices in $\boldsymbol{\mathcal{V}}_{\text{ub}}$ are assigned $\boldsymbol{\theta}_{\text{ub}}$, vertices in $\boldsymbol{\mathcal{V}}_{\text{la}}$ are assigned $\boldsymbol{\theta}_{\text{la}}$, and this rule applies similarly to $\boldsymbol{\mathcal{V}}_{\text{ra}}$, $\boldsymbol{\mathcal{V}}_{\text{ll}}$, and $\boldsymbol{\mathcal{V}}_{\text{rl}}$. This mapping (Figure 5) constrains pose-driven feature learning for each vertex to its affiliated kinematic segment, effectively avoiding interference from irrelevant joint parameters (e.g., leg joint parameters do not affect arm vertex features).

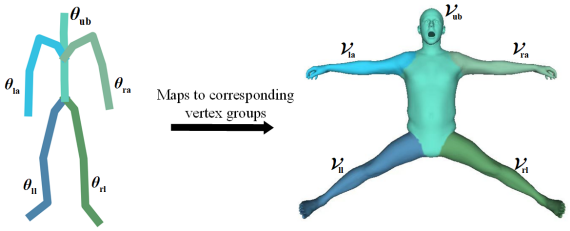


Figure 5: Mapping between pose subsets and vertex groups.

The local pose features $\boldsymbol{f}_l$ are computed through cross-attention, where $[\boldsymbol{\theta}_{\text{ub}}, \boldsymbol{\theta}_{\text{la}}, \boldsymbol{\theta}_{\text{ra}}, \boldsymbol{\theta}_{\text{ll}}, \boldsymbol{\theta}_{\text{rl}}]$ serves as queries ($\boldsymbol{Q}$), with $\boldsymbol{\theta}_b$ as both keys ($\boldsymbol{K}$) and values ($\boldsymbol{V}$). The global pose features $\boldsymbol{f}_g$ are computed through self-attention, where queries $\boldsymbol{Q}$, keys $\boldsymbol{K}$, and values $\boldsymbol{V}$ are all $\boldsymbol{\theta}_b$. Combining both $\boldsymbol{f}_g$ and $\boldsymbol{f}_l$ through concatenation leads to the pose-dependent feature $\boldsymbol{f}_d$.

In monocular videos, the number of available poses is often limited, while the poses required for animation-driven rendering are diverse. If each pose is encoded into a distinct pose feature, erroneous rendering results may occur during the driving phase. This is because the model only encounters a limited set of poses during training, and thus fails to generalize to out-of-distribution (OOD) poses. A natural solution is to represent these OOD poses using in-distribution (ID) poses, which is achieved through interpolation.

Given the training poses (rotations) $\{\boldsymbol{\theta}_j^t \mid \boldsymbol{\theta}_j^t \in \mathfrak{so}(3), 1 \le t \le T\}$ of the j-th joint, we first sample M key rotations via farthest point sampling. The distance metric

between two rotations is computed as [7, 30]:

$$d(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = 1 - \left|\mathrm{q}(\boldsymbol{\theta}_1)^\top \mathrm{q}(\boldsymbol{\theta}_2)\right| \in [0, 1], \tag{5}$$

where $\mathrm{q}(\cdot)$ is a function that maps an axis-angle vector to a unit quaternion, and $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are two axis-angle vectors. The sampled key rotations $\{\hat{\boldsymbol{\theta}}_j^m \mid 1 \leq m \leq M\}$ cover most of the seen poses in the training dataset.

For a query pose vector $\boldsymbol{\theta}_b = [\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T, \ldots, \boldsymbol{\theta}_J^T]^T$, we first split it into rotations of each joint $\{\boldsymbol{\theta}_j \mid 1 \leq j \leq J\}$. For the query rotation of the j-th joint, $\boldsymbol{\theta}_j$, we search for the H nearest key rotations $\{\hat{\boldsymbol{\theta}}_j^h\}_{h=1}^H$ using Eq. (5) as the distance metric, and interpolate corresponding rotations as a weighted sum:

$$\tilde{\boldsymbol{\theta}}_j = \frac{\sum_{h=1}^H w(\boldsymbol{\theta}_j, \hat{\boldsymbol{\theta}}_j^h)\hat{\boldsymbol{\theta}}_j^h}{\sum_{h=1}^H w(\boldsymbol{\theta}_j, \hat{\boldsymbol{\theta}}_j^h)}, \tag{6}$$

where $w(\boldsymbol{\theta}_j, \hat{\boldsymbol{\theta}}_j^h) = 1 - d(\boldsymbol{\theta}_j, \hat{\boldsymbol{\theta}}_j^h)$ is the blending weight. The query pose $\boldsymbol{\theta}_b$ is thus transformed into the interpolated pose $\tilde{\boldsymbol{\theta}}_b = [\tilde{\boldsymbol{\theta}}_1^T, \tilde{\boldsymbol{\theta}}_2^T, \ldots, \tilde{\boldsymbol{\theta}}_J^T]^T$.

3.4. Predicting the Final Gaussian Attributes

The pose-independent decoder employs two shared MLPs to decode the thin-triplane feature $\boldsymbol{f}_i$ for the k-th Gaussian primitive: (1) A shared geometry MLP ($\mathcal{M}_{\text{geo}}^{\text{inde}}$) that predicts the pose-independent position offset $\boldsymbol{\delta}_{\boldsymbol{\mu}_k}^g$ and scaling ratio $\boldsymbol{\delta}_{\boldsymbol{s}_k}^g$ of the primitive; (2) a shared appearance MLP ($\mathcal{M}_{\text{tex}}^{\text{inde}}$) that predicts the pose-independent color $\boldsymbol{c}_k^g$, opacity α_k^g, and axis-angle variation $\boldsymbol{\delta}_{\boldsymbol{r}_k}^g$ of the primitive:

$$\begin{cases} \boldsymbol{\delta}_{\boldsymbol{\mu}_k}^g,\ \boldsymbol{\delta}_{\boldsymbol{s}_k}^g = \mathcal{M}_{\text{geo}}^{\text{inde}}(\boldsymbol{f}_i), \\ \boldsymbol{c}_k^g,\ \alpha_k^g,\ \boldsymbol{\delta}_{\boldsymbol{r}_k}^g = \mathcal{M}_{\text{tex}}^{\text{inde}}(\boldsymbol{f}_i). \end{cases} \tag{7}$$

To model pose-dependent surface details, we first introduce $\boldsymbol{f}_n$, the normal feature of SMPL-X vertices which is derived from vertex normals after deformation from the canonical to the observed space. The pose-dependent decoder mirrors the structure of the pose-independent decoder, with two shared refined MLPs: (1) A shared refined geometry MLP ($\mathcal{M}_{\text{geo}}^{\text{de}}$) that decodes $\boldsymbol{f}_i$ and $\boldsymbol{f}_d$ to predict the pose-dependent position offset $\boldsymbol{\delta}_{\boldsymbol{\mu}_k}^p$ and scaling ratio variation $\boldsymbol{\delta}_{\boldsymbol{s}_k}^p$ of the k-th primitive; (2) shared refined appearance MLP ($\mathcal{M}_{\text{tex}}^{\text{de}}$) that takes $\boldsymbol{f}_i$, $\boldsymbol{f}_d$ and $\boldsymbol{f}_n$ as inputs to predict the pose-dependent color $\boldsymbol{c}_k^p$, opacity α_k^p, and axis-angle variation $\boldsymbol{\delta}_{\boldsymbol{r}_k}^p$ of the k-th primitive:

$$\begin{cases} \boldsymbol{\delta}_{\boldsymbol{\mu}_k}^p,\ \boldsymbol{\delta}_{\boldsymbol{s}_k}^p = \mathcal{M}_{\text{geo}}^{\text{de}}(\boldsymbol{f}_i, \boldsymbol{f}_d), \\ \boldsymbol{c}_k^p,\ \alpha_k^p,\ \delta_{\boldsymbol{r}_k}^p = \mathcal{M}_{\text{tex}}^{\text{de}}(\boldsymbol{f}_i, \boldsymbol{f}_d, \boldsymbol{f}_n). \end{cases} \tag{8}$$

Finally, the Gaussian primitives of each subject (in the canonical space) are deformed to the posed space using linear blend skinning (LBS) [17]. Their scaling coefficients, colors, and opacities remain unchanged during this process. For the k-th Gaussian primitive in the posed space, its position $\boldsymbol{\mu}_k$, scaling coefficients $\boldsymbol{s}_k$, axis-angle $\boldsymbol{r}_k$, color $\boldsymbol{c}_k$, and opacity α_k are formulated as the combination of pose-dependent and pose-independent attributes.

$$\begin{cases} \boldsymbol{\mu}_k = \sum_{i=1}^{n_b} w_i \boldsymbol{B}_i \cdot (\hat{\boldsymbol{\mu}}_k + B_S(\boldsymbol{\beta}, \mathcal{S}, \hat{\boldsymbol{\mu}}_k) + \boldsymbol{\delta}_{\boldsymbol{\mu}_k}^g + \boldsymbol{\delta}_{\boldsymbol{\mu}_k}^p), \\ \boldsymbol{s}_k = \hat{\boldsymbol{s}}_k \cdot (\boldsymbol{\delta}_{\boldsymbol{s}_k}^g + \boldsymbol{\delta}_{\boldsymbol{s}_k}^p), \\ \boldsymbol{r}_k = \mathcal{A}(\sum_{i=1}^{n_b} w_i \boldsymbol{R}_i) \cdot \boldsymbol{\delta}_{\boldsymbol{r}_k}^g \cdot \boldsymbol{\delta}_{\boldsymbol{r}_k}^p \cdot \hat{\boldsymbol{r}}_k, \\ \boldsymbol{c}_k = \boldsymbol{c}_k^g + \boldsymbol{c}_k^p, \\ \alpha_k = \alpha_k^g + \alpha_k^p, \end{cases} \tag{9}$$

where: (1) $\hat{\boldsymbol{\mu}}_k$ is the vertex coordinate of the query template; (2) $B_S(\boldsymbol{\beta}, \mathcal{S}, \hat{\boldsymbol{\mu}}_k)$ is the vertex offset at $\hat{\boldsymbol{\mu}}_k$ caused by the shape parameters $\boldsymbol{\beta}$ on the SMPL-X base template $\mathcal{S}$; (3) $\hat{\boldsymbol{s}}_k$ is the initial size of each 3D Gaussian primitive computed in the posed space according to its distances to neighboring 3D Gaussian primitives; (4) $\hat{\boldsymbol{r}}_k$ is the initial axis-angle of the Gaussian sphere computed from the normal vector; (5) w_i is the skinning weight for each joint; (6) $\mathbf{B}_i$ is the transformation matrix for each joint; (7) n_b is the number of joints; (8) $\boldsymbol{R}_i$ is the rotational component of the transformation matrix $\boldsymbol{B}_i$; and (9) $\mathcal{A}(\cdot)$ is the function that converts a rotation matrix to an axis-angle representation.

(a) (b) (c) (d)

Figure 6: Comparison of the rendering results between pose-independent and pose-dependent features. (a) GT; (b) pose-independent prediction; (c) pose-dependent prediction; (d) prediction difference shown in red.

To achieve disentanglement between pose-independent and pose-dependent features, we apply the total loss function (Section 3.6) to two sets of predictions during training: (1) the outputs of the pose-independent MLPs (i.e.,

$\boldsymbol{\delta}^g_{\boldsymbol{\mu}_k}$, $\boldsymbol{\delta}^g_{\boldsymbol{s}_k}$, $\boldsymbol{c}^g_k$, α^g_k, $\boldsymbol{\delta}^g_{\boldsymbol{r}_k}$) after being deformed to the posed space via linear blend skinning (LBS) (consistent with Eq. (9)) and (2) the aggregated results of pose-independent and pose-dependent predictions (i.e., combined position, scaling, color, opacity, and rotation attributes as formulated in Eq. (9)). This dual supervision strategy enforces that pose-independent features exclusively encode personalized static characteristics (e.g., inherent body shape, base garment texture) while pose-dependent features capture dynamic details induced by pose variations (e.g., fabric wrinkles, view-dependent color shifts). The separate constraint on pose-independent outputs and the joint constraint on fused results effectively prevent feature entanglement, ensuring clear disentanglement and complementary collaboration between the two types of features.

Figure 6 compares the performance of using only pose-independent attributes versus combining pose-dependent and pose-independent attributes. It can be observed that pose-dependent attributes primarily capture surface variations, including fine details such as fabric wrinkles and material textures, as well as view-dependent color shifts induced by varying lighting angles.

3.5. Garment Transfer Among Avatars

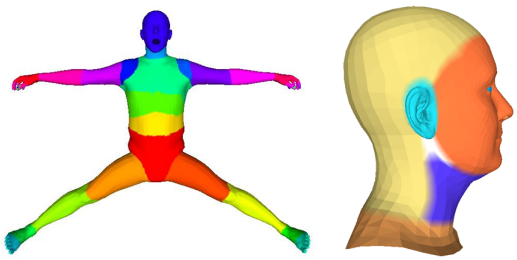
The canonical Gaussian space guarantees that all avatars have aligned semantic body parts and possess the same structural configuration as SMPL-X.

We can determine the masks for different body parts based on the pre-defined skinning weights in SMPL-X (Figure 7a). For instance, the joint indices corresponding to the left foot are "7" and "10". In this case, the vertices whose 7th value in their skinning weights is the largest and those whose 10th value is the largest are identified as the vertices belonging to the left foot. By following this approach, we can determine the masks for different body regions. Additionally, FLAME [13] provides masks for several different head regions (Figure 7b).

For transferring the shoes from a source avatar to a target avatar, we first obtain the vertex features F_s and F_t of the source and target avatars, respectively, in their respective thin-triplane-based feature spaces. The vertex features in the target avatar's shoe region (denoted as F_t^{shoe}) are replaced with those of the source avatar's corresponding region (denoted as F_s^{shoe}). The updated vertex features of the target avatar F_t^{new} can be expressed as:

$$F_t^{new} = F_t - F_t^{shoe} + F_s^{shoe}. \tag{10}$$

F_t^{new} are then input into the subsequent decoders to decode the Gaussian attributes of the target avatar after the shoe replacement.



(a) (b)

Figure 7: Masks for different body regions. (a) SMPL-X masks; (b) Flame masks.

3.6. Training

The learnable parameters of our network include the elements of the thin-triplane-based feature spaces for all subjects, the SMPL-X parameters for all video frames, and the parameters of the shared MLPs and shared refined MLPs. The overall training loss consists of color loss, perceptual loss, Laplacian loss and smooth loss.

The color loss $\mathcal{L}_{color}$ is the $L1$ loss between ground truth $\mathcal{I}(\theta)$ and rendered image $\hat{\mathcal{I}}(\theta)$ for each estimated pose θ,

$$\mathcal{L}_{color} = \left\| \mathcal{I}(\theta) - \hat{\mathcal{I}}(\theta) \right\|_1. \tag{11}$$

The perceptual loss $\mathcal{L}_{\text{perceptual}}$ ensures that the images rendered by 3DGS and the ground truth images are close at the feature map level. We use VGG [27] as the backbone network to compute LPIPS (Learned Perceptual Image Patch Similarity) [38],

$$\mathcal{L}_{\text{perceptual}} = \left\| \text{VGG}(\mathcal{I}(\theta)) - \text{VGG}\left(\hat{\mathcal{I}}(\theta)\right) \right\|_2^2. \tag{12}$$

The Laplacian loss $\mathcal{L}_{\text{lap}}$ improves the smoothness of attribute values between each Gaussian primitive and its neighboring 3D Gaussian primitives,

$$\mathcal{L}_{\text{lap}} = \sum_k \| S_k - \vartheta(S_k) \|_2^2. \tag{13}$$

where ϑ means the neighbors which are found through the shared faces of Gaussian primitives and limited to maximally 10.

The smooth loss $\mathcal{L}_{\text{smooth}}$ penalizes the vertex offsets predicted by the model:

$$\mathcal{L}_{\text{smooth}} = \sum_k \left\| \boldsymbol{\delta}^g_{\boldsymbol{\mu}_k} + \boldsymbol{\delta}^p_{\boldsymbol{\mu}_k} \right\|_2^2. \tag{14}$$

To address potential error accumulation in distal regions (e.g., hands) along the kinematic chain, we introduce a hand-specific normal constraint loss ($\mathcal{L}_{hand-reg}$) to enforce anatomically consistent deformations.

For Gaussian primitives in hand regions, we constrain their position offsets to align with the surface normals of the canonical SMPL-X template, a strategy employed in ExAvatar [20]. This prevents unreasonable offset directions that could lead to hand deformation. The loss is formulated as:

$$\mathcal{L}_{\text{hand-reg}} = \frac{1}{N_{\text{hand}}} \sum_{k \in \mathcal{K}_{\text{hand}}} \max\big(0, 1 - \big\langle \frac{\delta_{\mu_k}}{\|\delta_{\mu_k}\|_2 + \epsilon}, n_k \big\rangle\big), \tag{15}$$

where: (1) $\mathcal{K}_{hand}$ denotes the set of Gaussian primitives in left/right hand regions ; (2) $N_{hand} = |\mathcal{K}_{hand}|$ is the number of hand-region primitives; (3) $\delta_{\mu_k} = \delta^g_{\mu_k} + \delta^p_{\mu_k}$ is the total position offset of the k-th primitive; (4) n_k is the surface normal of the corresponding subdivided SMPL-X vertex (in canonical space); (5) $\epsilon = 1e-6$ avoids division by zero; (6) $\langle \cdot, \cdot \rangle$ denotes the dot product.

Our overall objective function is as follows:

$$\mathcal{L}_{total} = \lambda_{\text{color}}\mathcal{L}_{\text{color}} + \lambda_{\text{perceptual}}\mathcal{L}_{\text{perceptual}} + \mathcal{L}_{\text{reg}}.$$
$$\mathcal{L}_{\text{reg}} = \lambda_{\text{lap}}\mathcal{L}_{\text{lap}} + \lambda_{\mathcal{L}_{\text{smooth}}}\mathcal{L}_{\text{smooth}} + \lambda_{\mathcal{L}_{\text{hand-reg}}}\mathcal{L}_{\text{hand-reg}}. \tag{16}$$
$$\tag{17}$$

In our experiments, we adopt the following weight settings: $\lambda_{\text{color}} = 0.8$, $\lambda_{\text{perceptual}} = 0.2$, $\lambda_{\text{lap}} = 10000$, $\lambda_{\text{hand-reg}} = 10$ and $\lambda_{\text{smooth}} = 10000$. Specifically, the large values of λ_{lap} and λ_{smooth} are designed to guarantee the structural validity of 3D Gaussian primitives.

4. Experiments

4.1. Setup

The system is implemented using PyTorch, trained on a single NVIDIA 4090 GPU, and optimized with the Adam optimizer. The learning rates of all MLPs (Multi-Layer Perceptrons) and thin-triplane parameters start at $1e-3$ and decay to 10% of their initial values over the training iterations; in contrast, the learning rates of all learnable SMPL-X parameters start at $1e-4$ and similarly decay to 10% of their initial values during training. Subjects from the same dataset are jointly trained using a single model.

4.2. Evaluation Datasets, Metrics and Comparison Methods

We evaluate the system on the NeuMan [10] and AIST++ [31] datasets. NeuMan provides outdoor monocular videos (each lasting approximately 10–20 seconds and featuring a moving person); we used its bike, citron, lab, jogging, and seattle videos, with the training and test splits following NeuMan's official setup. AIST++ includes multi-view videos of dozens of dancers; we selected videos of 5 dancers, extracted clips from these videos to construct our dataset, and subsequently split the dataset into training and test sets. All evaluation videos are preprocessed via ExAvatar [20] to extract data (e.g., human body model parameters, masks).

Performance is evaluated using three standard metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) [33], and Learned Perceptual Image Patch Similarity (LPIPS) [38]. For other editing operations supported by our method (e.g., texture editing), only visual results are provided (no ground truth available).

We compare our method with state-of-the-art approaches: LHM [25], ExAvatar [20], GaussianAvatar [5], 3DGS-Avatar [24], and GauHuman [6]. For fairness, all their results are computed via official codes without test-time optimization. LHM [25] only supports single-image input and, therefore, one clear front-facing human frame per monocular video is extracted for testing.

4.3. Qualitative Results

Figures 8 and 9 illustrate that our method achieves excellent rendering results in novel pose synthesis on both NeuMan and AIST++. For example, our method delivers the best performance in terms of facial details, hand details, garments continuity, and wrinkles.

Figure 10 illustrates that our method enables high-fidelity garment transfer among reconstructed avatars, with exchangeable regions including hair, upper garments, pants, shoes, and more.

4.4. Quantitative Results

Tables 1 and 2 demonstrate that our method achieves the best novel pose synthesis performance on both NeuMan and AIST++.

4.5. Ablation Studies

Two critical hyperparameters are first tuned: triplane partition count (K) and length (L). Figure 11a shows $K = 4$ optimizes rendering quality, while Figure 11b demonstrates increasing L beyond 128 gives diminishing returns (rendering quality plateaus here). Thus, $K = 4$ and $L = 128$ are selected to balance performance and resource use.

Next, an ablation study on normal-guided initialization, thin-triplane-based feature space, and hybrid pose encoder with ID interpolation is conducted on the AIST++ dataset, focusing on three aspects: (1) Initialization Choice: 3D Gaussians are initialized either via normal-guided initialization or with all rotation parameters set to the zero axis-angle; (2) Pose Encoding: human poses are represented either via our hybrid pose encoder with ID interpolation (HPE-II) or via the baseline 6D pose encoding method [39]; (3) Triplane Architecture: the feature triplane is either kept as a monolithic structure or partitioned into four thin-triplanes along the channel dimension.

Results (Table 3 and Figure 12) indicate that ShareUs' key components, namely normal-guided initialization, the

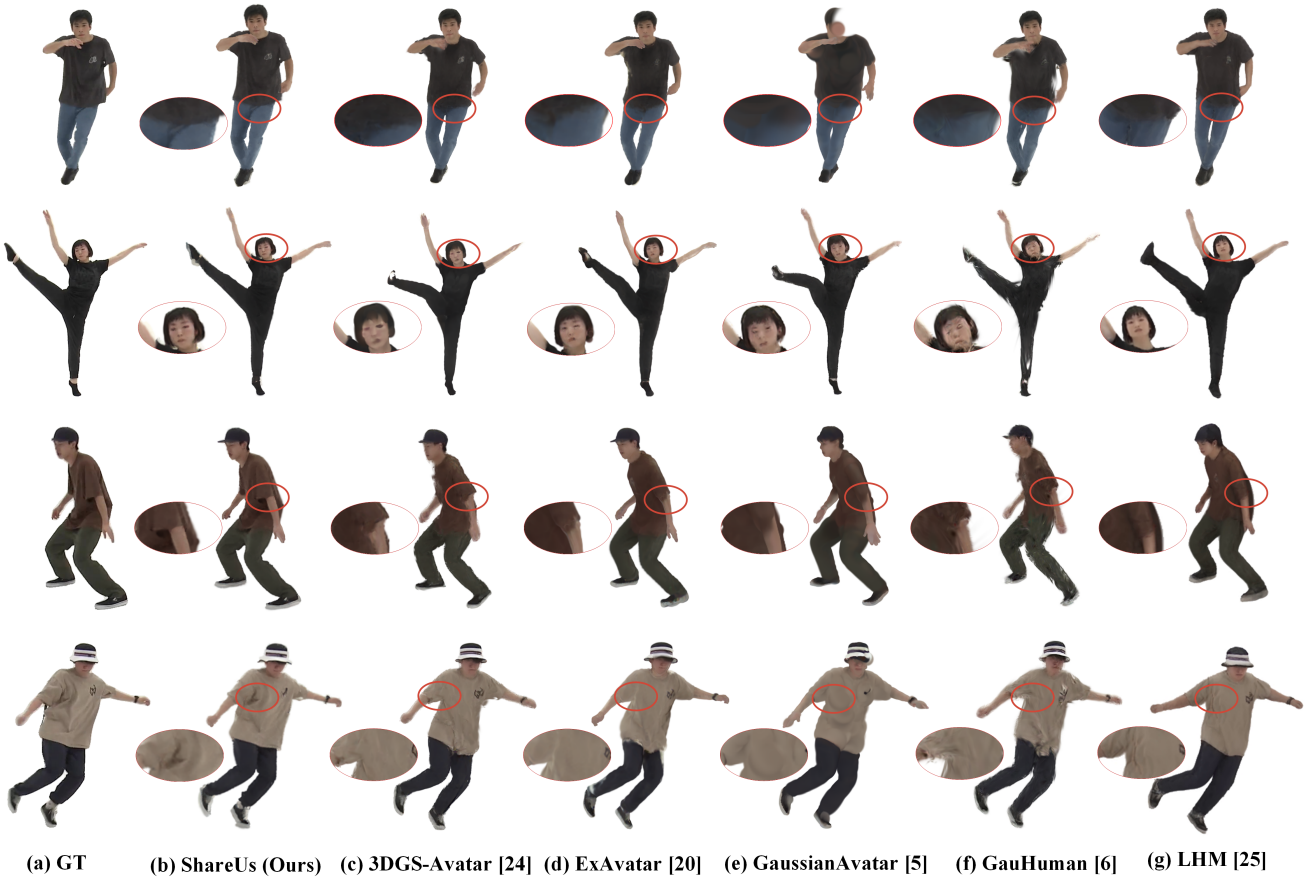


Figure 8: Visual comparison among different methods on the test set of AIST++.

Table 1: Statistical comparison on NeuMan test set.

Model	bike			citron			jogging			lab			seattle		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
3DGS-Avatar [24]	24.15	0.950	0.030	24.01	0.958	0.022	24.51	0.953	0.030	24.23	0.960	0.028	25.32	0.966	0.019
ExAvatar [20]	29.18	0.967	0.025	31.04	0.976	0.017	28.71	0.966	0.025	28.25	0.973	0.033	29.21	0.981	0.017
GaussianAvatar [5]	23.19	0.953	0.043	25.36	0.965	0.032	24.20	0.958	0.033	23.12	0.952	0.045	25.97	0.973	0.020
GauHuman [6]	21.48	0.946	0.034	23.86	0.965	0.020	22.51	0.960	0.036	22.57	0.949	0.036	24.17	0.970	0.031
LHM [25]	22.64	0.947	0.033	23.14	0.962	0.025	23.45	0.963	0.034	24.15	0.952	0.033	23.75	0.967	0.032
ShareUs(Ours)	**30.02**	**0.970**	**0.021**	**32.47**	**0.981**	**0.011**	**30.20**	**0.972**	**0.021**	**30.19**	**0.980**	**0.024**	**30.12**	**0.983**	**0.016**

Table 2: Statistical comparison on AIST++ test set.

Model	dance04			dance06			dancer08			dance20			dance21		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
3DGS-Avatar [24]	24.97	0.976	0.025	22.11	0.967	0.043	27.10	0.987	0.012	27.62	0.988	0.012	28.58	0.988	0.011
ExAvatar [20]	29.10	0.986	0.021	25.49	0.979	0.033	31.35	0.992	0.010	31.09	0.991	0.011	31.37	0.991	0.010
GaussianAvatar [5]	23.34	0.975	0.034	22.31	0.971	0.040	27.07	0.988	0.013	27.81	0.988	0.013	28.81	0.989	0.012
GauHuman [6]	25.66	0.971	0.026	22.77	0.958	0.041	29.18	0.987	0.012	28.03	0.986	0.012	30.28	0.988	0.010
LHM [25]	23.76	0.976	0.028	23.52	0.973	0.038	29.36	0.988	0.013	27.51	0.986	0.014	24.85	0.968	0.025
ShareUs(Ours)	**30.45**	**0.988**	**0.018**	**27.05**	**0.982**	**0.021**	**32.42**	**0.993**	**0.009**	**32.75**	**0.992**	**0.010**	**32.88**	**0.993**	**0.009**

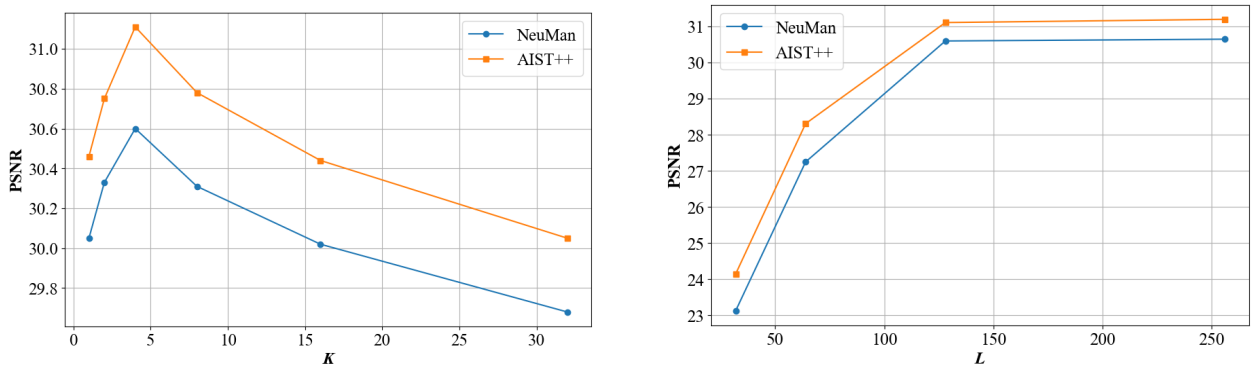
Figure 9: Visual comparison of NeuMan avatar animation methods under novel out-of-distribution poses.

Figure 10: Demonstration of garment transfer among reconstructed avatars.

thin-triplane-based feature space, and the hybrid pose encoder with ID interpolation, deliver demonstrably superior rendering quality for novel poses. These improvements are reflected in better preservation of fine details (e.g., clothing

Table 3: Ablation study for the effectiveness of our normal-guided initialization, thin-triplane-based feature space, and hybrid pose encoder with ID interpolation on the test set of AIST++.

w/o Normal-guided	w/ Normal-guided	6D Pose	HPE-II	One Triplane	4 Thin-Triplanes	PSNR	SSIM	LPIPS
✓	-	✓	-	✓	-	29.22	0.974	0.027
✓	-	✓	-	-	✓	29.78	0.980	0.023
✓	-	-	✓	✓	-	29.92	0.984	0.020
✓	-	-	✓	-	✓	30.56	0.989	0.015
-	✓	✓	-	✓	-	29.70	0.978	0.024
-	✓	✓	-	-	✓	30.42	0.983	0.020
-	✓	-	✓	✓	-	30.55	0.985	0.017
-	✓	-	✓	-	✓	**31.11**	**0.990**	**0.014**



(a) The number of triplane partitions K

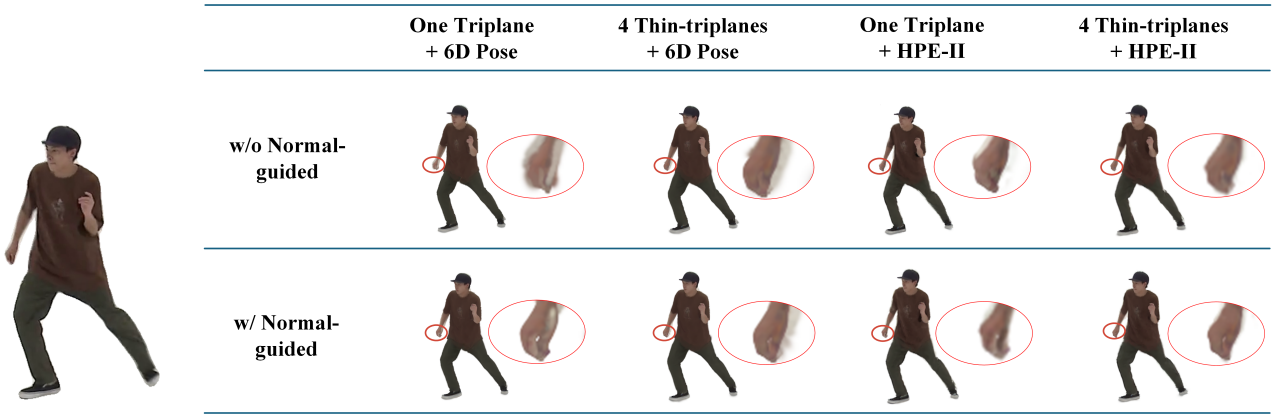
(b) The triplane length L

Figure 11: Ablation study on the two critical hyperparameters.

logos) and smoother delineation of body edges (Figure 12).

5. CONCLUSION

This work aims at the limitation of conventional monocular video-based digital human reconstruction methods, which focus solely on single individuals and proposes a unified framework for reconstructing and editing multiple avatars from diverse monocular videos using a single model. Key innovations include: (1) A canonical Gaussian space is established, where 3D Gaussians are initialized via a normal-guided method to ensure consistent topology and semantic alignment of body parts across all avatars; (2)



(a) GT

(b) Rendering results by combining different settings.

Figure 12: Example results from ablation study on the main components of ShareUs.

a thin-triplane-based feature space is designed to disentangle pose-independent features and enable garment transfer through the replacement of region-specific features; (3) a hybrid pose encoder captures pose-dependent surface details, while out-of-distribution pose interpolation extends generalization to diverse poses beyond training data. Experimental results confirm that the proposed framework balances reconstruction performance and resource utilization, enhances rendering quality for novel poses, and supports effective cross-avatar editing. It thus provides a practical solution for multi-avatar reconstruction and asset sharing in digital human research.

Our method may produce artifacts due to inaccurate foreground segmentation in videos. It may also yield distorted results when subjects exhibit excessive motion, as the adopted human pose estimation method may output erroneous poses in such scenarios and cause reconstruction distortions. Addressing these limitations will further enhance the robustness of ShareUs for broader applications.

References

[1] A. Chatziagapi, G. G. Chrysos, and D. Samaras. MIGS: Multi-Identity Gaussian Splatting via Tensor Decomposition. In *ECCV*, pages 388–408. Springer, 2024. 1, 3

[2] H. Chen, B. Peng, Y. Tao, and J. Zhang. D^3-Human: Dynamic Disentangled Digital Human from Monocular Video. pages 10836–10846, 2025. 1, 2

[3] J. Chen. GGAvatar: Reconstructing Garment-Separated 3D Gaussian Splatting Avatars from Monocular Video. In *ACM MMAsia*, pages 1–7, 2024. 1, 2

[4] C. Guo, T. Jiang, X. Chen, J. Song, and O. Hilliges. Vid2Avatar: 3D Avatar Reconstruction from Videos in the Wild via Self-Supervised Scene Decomposition. In *CVPR*, pages 12858–12868, 2023. 1, 2

[5] L. Hu, H. Zhang, Y. Zhang, B. Zhou, B. Liu, S. Zhang, and L. Nie. GaussianAvatar: Towards Realistic Human Avatar Modeling from a Single Video via Animatable 3D Gaussians. In *CVPR*, pages 634–644, 2024. 1, 2, 8, 9

[6] S. Hu, T. Hu, and Z. Liu. GauHuman: Articulated Gaussian Splatting from Monocular Human Videos. In *CVPR*, pages 20418–20431, 2024. 2, 8, 9

[7] D. Q. Huynh. Metrics for 3D Rotations: Comparison and Analysis. *Journal of Mathematical Imaging Vision*, 35(2):155–164, 2009. 6

[8] B. Jiang, Y. Hong, H. Bao, and J. Zhang. SelfRecon: Self Reconstruction Your Digital Avatar from Monocular Video. In *CVPR*, pages 5605–5615, 2022. 1

[9] T. Jiang, X. Chen, J. Song, and O. Hilliges. InstantAvatar: Learning Avatars from Monocular Video in 60 Seconds. In *CVPR*, pages 16922–16932, 2023. 1, 2

[10] W. Jiang, K. M. Yi, G. Samei, O. Tuzel, and A. Ranjan. NeuMan: Neural Human Radiance Field from a Single Video. In *ECCV*, pages 402–418. Springer, 2022. 1, 8

[11] B. Kerbl, G. Kopanas, T. Leimk"uhler, and G. Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM TOG*, 42(4):139–1, 2023. 2

[12] M. Kocabas, J.-H. R. Chang, J. Gabriel, O. Tuzel, and A. Ranjan. HUGS: Human Gaussian Splats. In *CVPR*, pages 505–515, 2024. 2

[13] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero. Learning a Model of Facial Shape and Expression from 4D Scans. *ACM TOG*, 36(6):194:1–194:17, 2017. 7

[14] Z. Li, Z. Zheng, Y. Liu, B. Zhou, and Y. Liu. PoseVocab: Learning Joint-structured Pose Embeddings for Human Avatar Modeling. In *ACM SIGGRAPH*, 2023. 1

[15] Z. Li, Z. Zheng, L. Wang, and Y. Liu. Animatable Gaussians: Learning Pose-dependent Gaussian Maps for High-fidelity Human Avatar Modeling. In *CVPR*, 2024. 1

[16] S. Lin, Z. Li, Z. Su, Z. Zheng, H. Zhang, and Y. Liu. LayGA: Layered Gaussian Avatars for Animatable Clothing Transfer. In *ACM SIGGRAPH*, pages 1–11, 2024. 1, 2

[17] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A Skinned Multi-Person Linear Model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. 2023. 6

[18] Q. Ma, J. Yang, S. Tang, and M. J. Black. The Power of Points for Modeling Humans in Clothing. In *ICCV*, pages 10974–10984, 2021. 3

[19] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[20] G. Moon, T. Shiratori, and S. Saito. Expressive Whole-Body 3D Gaussian Avatar. In *ECCV*, pages 19–35. Springer, 2024. 2, 8, 9

[21] J. Mu, S. Sang, N. Vasconcelos, and X. Wang. ActorsNeRF: Animatable Few-Shot Human Rendering with Generalizable NeRFs. In *ICCV*, pages 18391–18401, 2023. 1

[22] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black. Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. In *CVPR*, pages 10975–10985, 2019. 2

[23] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger. Convolutional Occupancy Networks. 2020. 4

[24] Z. Qian, S. Wang, M. Mihajlovic, A. Geiger, and S. Tang. 3DGS-Avatar: Animatable Avatars via Deformable 3D Gaussian Splatting. In *CVPR*, pages 5020–5030, 2024. 1, 2, 8, 9

[25] L. Qiu, X. Gu, P. Li, Q. Zuo, W. Shen, J. Zhang, K. Qiu, W. Yuan, G. Chen, and Z. Dong. LHM: Large Animatable Human Reconstruction Model from a Single Image in Seconds. *ICCV*, pages –, 2025. 1, 3, 8, 9

[26] L. Qiu, S. Zhu, Q. Zuo, X. Gu, Y. Dong, J. Zhang, C. Xu, Z. Li, W. Yuan, L. Bo, et al. AniGS: Animatable Gaussian Avatar from a Single Image with Inconsistent Gaussian Reconstruction. In *CVPR*, pages 21148–21158, 2025. 3

[27] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[28] S.-Y. Su, T. Bagautdinov, and H. Rhodin. NPC: Neural Point Characters from Video. In *ICCV*, pages 14795–14805, 2023. 1, 2

[29] S.-Y. Su, F. Yu, M. Zollhöfer, and H. Rhodin. A-NeRF: Articulated Neural Radiance Fields for Learning Human Shape, Appearance, and Pose. *NeuIPS*, 34:12278–12291, 2021. 1

[30] G. Tiwari, D. Anti, J. E. Lenssen, N. Sarafianos, T. Tung, and G. Pons-Moll. Pose-NDF: Modeling Human Pose Manifolds with Neural Distance Fields. 2022. 6

[31] S. Tsuchida, S. Fukayama, M. Hamasaki, and M. Goto. AIST Dance Video Database: Multi-Genre, Multi-Dancer, and Multi-Camera Database for Dance Information Processing. In *ISMIR*, volume 1, page 6, 2019. 8

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention Is All You Need. *NeuIPS*, 30, 2017. 4

[33] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE T-IP*, 13(4):600–612, 2004. 8

[34] C.-Y. Weng, B. Curless, P. P. Srinivasan, J. T. Barron, and I. Kemelmacher-Shlizerman. HumanNeRF: Free-Viewpoint Rendering of Moving People from Monocular Video. In *CVPR*, pages 16210–16220, 2022. 1, 2

[35] Z. Yu, W. Cheng, X. Liu, W. Wu, and K.-Y. Lin. MonoHuman: Animatable Human Neural Field from Monocular Video. In *CVPR*, pages 16943–16953, 2023. 1, 2

[36] H. Zhang, S. Lin, R. Shao, Y. Zhang, Z. Zheng, H. Huang, Y. Guo, and Y. Liu. CloSET: Modeling Clothed Humans on Continuous Surface with Explicit Template Decomposition. In *CVPR*, pages 501–511, 2023. 3

[37] J. Zhang, X. Li, Q. Zhang, Y. Cao, Y. Shan, and J. Liao. HumanRef: Single Image to 3D Human Generation via Reference-Guided Diffusion. In *CVPR*, pages 1844–1854, 2024. 1

[38] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*, pages 586–595, 2018. 7, 8

[39] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the Continuity of Rotation Representations in Neural Networks. In *CVPR*, pages 5745–5753, 2019. 8

[40] Y. Zhuang, J. Lv, H. Wen, Q. Shuai, A. Zeng, H. Zhu, S. Chen, Y. Yang, X. Cao, and W. Liu. IDOL: Instant Photorealistic 3D Human Creation from a Single Image. In *CVPR*, pages 26308–26319, 2025. 1, 3