UVRELIEF: A SURFACE CODEC FOR AVATAR-AS-IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating human avatars as an *Avatar-as-Image* representation has become an active and rapidly growing research direction, where we map an avatar onto a shared UV domain over a parametric body model and turn 3D generation into a 2D problem. Realizing this potential, however, hinges on how to obtain a compact latent space for Avatar-as-Image generation. We present UVRelief, a surface codec for Avatar-as-Image generation that encodes a human as a *UV-native relative shape*: given a textured mesh and its fitted SMPL-X model, UVRelief bakes surface geometry and appearance into *Relief Maps*, where each texel stores a relief displacement relative to the SMPL-X surface and a texture signal. Built on these maps, a VAE encodes them into a compact UV-structured latent space and decodes texel-aligned 3D Gaussian attributes under rendering supervision, and a flow-matching prior is trained in this latent space for image-conditioned 3D avatar generation. Extensive experiments show that UVRelief improves reconstruction fidelity and achieves competitive generation quality compared with existing multi-view and latent avatar generation methods.

1 INTRODUCTION

3D human avatars are central to VR/AR, telepresence, filmmaking, and gaming. Yet creating them still relies largely on manual artistic workflows Blender (2024) or reconstruction pipelines Zhang & Chen (2024); Zhang et al. (2023b); Peng et al. (2021); Qian et al. (2023); Wang et al. (2025a) that require dozens of multi-view cameras, which inflates production cost and limits creative flexibility. Generative models promise to lower this barrier by synthesizing 3D avatars from a single image or text Zhang et al. (2023a); Xiang et al. (2025); Poole et al. (2022).

A compelling and rapidly growing direction is to generate avatars as an *Avatar-as-Image* representation Zhuang et al. (2025); Wang et al. (2025b); Kostyrko et al. (2026), i.e., standardized 2D images over a parametric body model such as SMPL-X. Because all subjects share the same UV layout, geometry and appearance are unrolled into a shared, spatially regular domain in which the same texel always refers to the same anatomical location. This fixed texel-to-anatomy correspondence turns 3D avatar generation into a 2D problem and reduces the difficulty of learning the representation.

Within this paradigm, the standard recipe is to learn a latent space with an autoencoder as codec and then train a generative prior Zhuang et al. (2025); Wang et al. (2025b); Yang et al. (2025); Li et al. (2026b). Most such methods take multi-view images as the autoencoder input, which forces the codec to reconstruct 3D from images; this couples the codec to a large reconstruction model and lets view-dependent errors propagate into the latent.

An appealing alternative is to give the codec a 3D-derived signal in place of images. This has been explored for general objects, such as the geometry image idea that represents shapes as 2D maps that record the absolute 3D position of each texel Yan et al. (2025), but it has not been adapted to human avatars in the shared UV domain. Both routes therefore leave an important choice unresolved: given a human's 3D geometry, how should it be encoded so that the latent space is both faithful and easy to learn?

In this work, we fill this gap with a UVRelief Codec that converts 3D humans to a *UV-native relative shape*, which stores, per texel, the displacement of the surface relative to the underlying SMPL-X mesh—we call it UVRelief, as it resembles a relief floating beyond the UV plane. Given a textured

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

mesh and its fitted SMPL-X model Pavlakos et al. (2019), we sample texels on the shared UV map, emit rays from the corresponding SMPL-X surface points along their normals, and record a per-texel relief displacement and a texture signal; these maps invert back to a colored 3D model with high fidelity. A 2D VAE then learns a compact, UV-structured latent space and decodes texel-aligned 3D Gaussian attributes under rendering supervision, and a flow-matching prior is trained in this latent space for image-conditioned generation. Because UV Relief is a pose-agnostic relative shape, the codec encodes only small residuals over the SMPL-X manifold, and the regular UV layout lets 2D networks replace heavier 3D architectures.

To validate its effectiveness, we compare with two other encoding strategies, varying only the encoding while keeping the decoder, dataset, and supervision fixed: (i) a *cross attention* between learnable UV tokens and multi-view images Zhuang et al. (2025); Wang et al. (2025b); Yang et al. (2025); Li et al. (2026b), and (ii) a *position map* that stores the 3D position of each texel (the geometry-image idea Yan et al. (2025)). Experiments show that cross attention faces difficulty in encoding complex geometries, while position maps appear to place a heavier learning burden on the generative prior, confirming the importance of the *UV-native relative shape*.

The contributions of this work are summarized as follows:

- We introduce **UVRelief Codec** for Avatar-as-Image generation, which encodes a human as a *UV-native relative shape*: a compact, pose-agnostic, and spatially regular Relief Map (per-texel relief displacement plus texture) that is invertible to a 3D model.
- We identify the encoding strategy as the key design choice to transport geometry into Avatar-as-Image latent generation. Our experiments show that a *UV-native relative shape* outperforms an *absolute position map* and an *cross-attention mechanism*.
- We build a complete system—a UV-structured VAE that decodes to texel-aligned 3D Gaussians under rendering supervision, plus a flow-matching prior for image-conditioned generation—and demonstrate improved reconstruction fidelity and competitive generation quality over existing multi-view and latent avatar generation methods Yu et al. (2021); Ho et al. (2023); Shen et al. (2023); Han et al. (2023).

2 Related Work

2.1 Avatar-as-Image: 2D Representations for 3D Humans

Avatar-as-Image represents a 3D human as standardized 2D maps over the UV layout of a parametric body model such as SMPL-X Pavlakos et al. (2019). The idea dates back to early textured-avatar work, which unrolled appearance and geometry into a shared UV domain and predicted a per-texel texture and a displacement map relative to the body template from images Alldieck et al. (2019); Lazova et al. (2019); Lahner et al. (2018); Kostyrko et al. (2026). Because every texel always refers to the same anatomical location, these maps are directly editable in image space, and the underlying shape can be changed by re-posing or re-shaping SMPL-X. The representation later evolved from explicit maps to UV-space neural fields Zhang & Chen (2022) and to UV-aligned 3D Gaussians, which attach Gaussian attributes to texels and jointly learn mesh deformation and Gaussian textures in UV space Jiang et al. (2024). These representations reconstruct high-fidelity humans and support image-based editing, but they are typically obtained by per-subject multi-view reconstruction and optimization, which is expensive and does not scale. We instead ask how to *generate* such a representation from a single image.

2.2 Latent Generation for 3D Avatars

Latent generation has become the dominant recipe for 3D generation since latent diffusion models Rombach et al. (2022) showed that learning in a compact latent space is both easier and readily controllable through cross-attention. Its core is the design of a VAE that encodes a 3D representation into a latent code and decodes it back. For general objects this has produced 3D-native autoencoders that regularize a mesh into a form a transformer can consume—a vecset Zhang et al. (2023a; 2024a); Lai et al. (2025) or structured sparse voxels Xiang et al. (2025)—so that a diffusion prior can be trained on the resulting latents. Omages Yan et al. (2025) represents an object by the per-texel 3D positions of a UV-like map and generates it with a 2D image diffusion model.

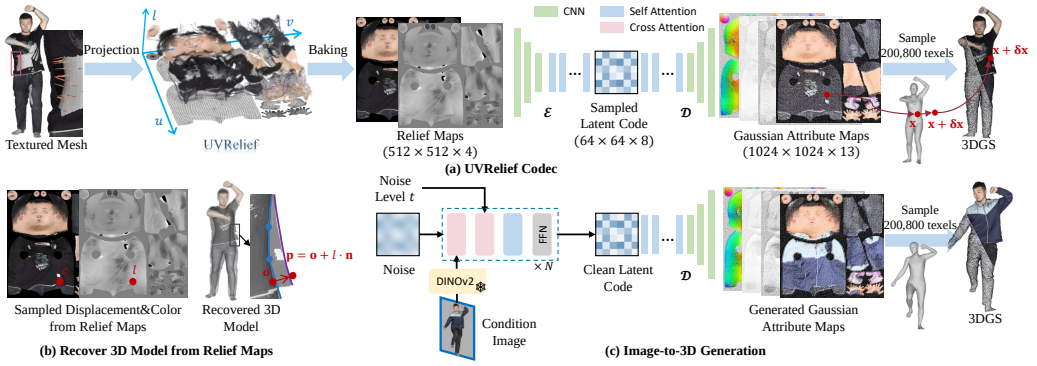


Figure 1: **Overview of the proposed method. (a) UVRelief Codec.** Given a textured mesh and its fitted SMPL-X model, we project texels along the SMPL-X normals to form the UV Relief and bake it into *Relief Maps* ($512 \times 512 \times 4$: texture and relief displacement). A CNN followed by a transformer encoder compresses the maps into a $64 \times 64 \times 8$ latent code, from which a transformer and a CNN decoder reconstruct *Gaussian attribute maps* ($1024 \times 1024 \times 13$). We sample 200,800 texels, animate them with SMPL-X, and render them as 3D Gaussians for supervision. **(b) Recovering 3D models from Relief Maps.** We interpolate the relief displacement l and color $\mathbf{c}$ from the maps and deform SMPL-X points by $\mathbf{p} = \mathbf{o} + l\,\mathbf{n}$, recovering a colored point cloud. **(c) Image-to-3D generation.** A transformer denoiser (cross/self-attention and Feed-forward Network (FFN)) takes a noisy latent code, the noise level t, and a DINOv2-encoded condition image, and predicts a clean latent code, which the codec decoder converts into Gaussian attribute maps for rendering.

Humans are harder: they articulate, and high-quality 3D data is scarce. The dominant avatar methods are therefore multi-view based Zhuang et al. (2025); Wang et al. (2025b); Yang et al. (2025); Li et al. (2026b): they obtain a UV-aligned latent from multi-view images through UV-based queries and cross-attention and decode it into UV-aligned 3D Gaussians. Because their VAE is fed *rendered images*, it is implicitly forced to solve multi-view 3D reconstruction, adding learning complexity from the absence of a direct 3D input. A second line reconstructs each subject with an auto-decoder and learns a generative model over the latents Zhang et al. (2024b; 2025b); this avoids the reconstruction burden but offers neither a cross-sample distribution constraint nor a VAE that scales to large data. In the spirit of TRELLIS Xiang et al. (2025) and CLAY Zhang et al. (2024a), we instead design a human-specific VAE that regularizes a 3D mesh into its input—the Relief Maps that encode relative shape in UV space—and show that this UV-native relative form makes the latent space markedly more learnable than an absolute 3D-position map.

2.3 SDS-based and Multi-view Diffusion Methods

Beyond representing 3D assets in a latent space, another family lifts 2D generative models to 3D. DreamFusion Poole et al. (2022) introduces Score Distillation Sampling, which uses a pretrained 2D diffusion model Rombach et al. (2022) to optimize a 3D object through a differentiable renderer, and later works extend this to avatars Tang et al. (2023); Wu et al. (2025); Zhang et al. (2025a). Multi-view methods instead bypass iterative optimization by generating several views with a 2D model and fusing them into a 3D asset Li et al. (2024); Chen et al. (2025); a common limitation of both is that they struggle to maintain 3D consistency, producing the well-known multi-face (Janus) artifact. Latent generation of 3D avatars instead demands large amounts of high-quality 3D data and is therefore more expensive, but it offers the potential for high-fidelity, 3D-consistent generation. Our goal is to explore whether an effective 3D generative model can be trained from limited data in the human domain, which requires exploiting human priors to remove most of the learning complexity—in particular, an Avatar-as-Image representation that factors out pose—rather than generating directly in 3D as in general object generation.

3 METHOD

Following the Avatar-as-Image paradigm, our goal is to generate a 3D avatar in a compact latent space by encoding a human's 3D geometry—rather than multi-view images—as the codec input. The central design question is *how* to encode that geometry. We encode a human as a *UV-native relative shape*, the *UV Relief*: each texel in the UV grid is deformed to a point beyond the UV plane, forming a floating relief. As shown in Fig. 1, this relative shape could be baked as Relief Maps, where each texel records the displacement and texture; a VAE compresses them into a UV-structured latent and decodes texel-aligned 3D Gaussians under rendering supervision; and an image-conditioned flow-matching prior is trained in this latent space for single-image generation.

3.1 UVRELIEF CODEC

In the spirit of latent generation methods for general objects that convert 3D models to VecSet Zhang et al. (2024a) or sparse voxels Xiang et al. (2025), we represent an avatar as a *Relief Map*, a 2D map over the UV layout of a parametric body model. Each texel stores a scalar *relief displacement*—a signed displacement along the SMPL-X surface normal—together with a texture signal. This is a *UV-native relative shape*: geometry is expressed relative to the SMPL-X template in the UV frame, so it stays smooth inside each UV island. An alternative, directly recording 3D positions per texel, is an *absolute* encoding; its geometry in UV space would be distorted—a spatial misalignment that makes it hard to learn a generative prior.

As illustrated in Fig. 1 (a, left), given a textured mesh and its fitted SMPL-X model, we first sample discretized texels on the UV maps, obtaining their coordinates (u, v), barycentric weights (w_1, w_2, w_3), and face index f. With barycentric interpolation, we determine the sample points on the SMPL-X mesh:

$$\mathbf{o} = w_1\mathbf{v}_1 + w_2\mathbf{v}_2 + w_3\mathbf{v}_3, \tag{1}$$

where $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3) = \mathcal{F}(f)$ are the triangle's vertices. Similarly, we interpolate the surface normal $\mathbf{n}$ and emit a ray from the origin:

$$\mathbf{r}(l) = \mathbf{o} + l\,\mathbf{n}, \tag{2}$$

which intersects the textured mesh to obtain the color $\mathbf{c}$ and displacement l. The UV Relief is then formed as a colored point cloud:

$$\mathcal{P} = \{(\mathbf{u}_i, \mathbf{c}_i) \mid i = 1, ..., H \times W\}, \tag{3}$$

where $\mathbf{u}_i = (u_i, v_i, l_i)$ denotes the point position, $\mathbf{c}_i$ its attributes, and H, W the resolution of the sampled UV maps. To further enhance spatial regularity, we bake the UV Relief $\mathcal{P}$ into UV maps $\mathcal{U}$:

$$\mathcal{U} : (u_i, v_i) \mapsto (l_i, \mathbf{c}_i) \mid (u_i, v_i) \in \Omega, \tag{4}$$

where $\Omega = \{(u_i, v_j) \mid i = 1, ..., H; j = 1, ..., W\}$ denotes the fixed map grid.

Recovering 3D Models from Relief Maps. Relief Maps can be inverted back to 3D with minimal information loss. As shown in Fig. 1 (b), we sample points $\mathbf{o}$ on the SMPL-X mesh and interpolate the displacement l and color $\mathbf{c}$ from $\mathcal{U}$; the reconstructed point cloud is: $\mathbf{p} = \mathbf{o} + l \cdot \mathbf{n}$, with $\mathbf{c}$ attached as its color.

Compared to raw 3D representations, Relief Maps are pose-agnostic, spatially regular, and encode only small residuals over the SMPL-X manifold, so they can be processed by 2D transformer networks rather than denser 3D architectures Xiang et al. (2025). Because each texel stores a single displacement along the SMPL-X normal, UV Relief is a height field and cannot represent non-height-field topology; this is the codec's main modeling assumption.

We train a VAE that compresses the maps into a compact latent. As in Fig. 1 (a), we concatenate texture and relief displacement into a 4-channel input $\mathcal{U} \in \mathbb{R}^{H \times W \times 4}$ and process it with a CNN that reduces the spatial resolution by eight:

$$\mathcal{U}_{low} = \mathrm{CNN}_{\mathrm{encoder}}(\mathcal{U}). \tag{5}$$

The output $\mathcal{U}_{low} \in \mathbb{R}^{h \times w \times c}$ is then flattened, concatenated with position encoding PE, and fed as input to a transformer-based encoder consisting of several self-attention blocks:

$$(\boldsymbol{m}, \boldsymbol{\sigma}) = \mathcal{E}(\mathcal{U}_{low}; \mathrm{PE}), \tag{6}$$

where the mean and variance $(\boldsymbol{m}, \boldsymbol{\sigma})$ define the distribution for sampling the low-dimensional latent code $\mathbf{z} \in \mathbb{R}^{(h \cdot w) \times d}$, which is then processed by a transformer-based decoder:

$$\mathcal{U}'_{low} = \mathcal{D}(\mathbf{z}). \tag{7}$$

Next we convert $\mathcal{U}'_{low} \in \mathbb{R}^{h \times w \times c}$ to 3D, selecting 3DGS for its efficient rendering. A CNN-based decoder up-samples to a 1024×1024 Gaussian-attribute map:

$$\{\delta\mathbf{x}, \alpha, \mathbf{R}, \mathbf{S}, \mathbf{c}\}_{1024 \times 1024} = \mathrm{CNN}_{\mathrm{decoder}}(\mathcal{U}'_{low}). \tag{8}$$

From this map we sample n texels following IDOL Zhuang et al. (2025) and lift them, guided by the posed SMPL-X, into 3D Gaussians:

$$\mathcal{G} = \{\mathbf{x} + \delta\mathbf{x}, \alpha, \mathbf{R}, \mathbf{S}, \mathbf{c}\}_n, \tag{9}$$

with $n = 200{,}800$. The Gaussians are rendered and compared against images of the original textured mesh. The codec is supervised only through these rendered images:

$$\mathcal{L} = \mathcal{L}_{\mathrm{L1}} + 0.2\,(1 - \mathrm{SSIM}) + 0.5\,\mathcal{L}_{\mathrm{LPIPS}} + 10^{-6}\mathcal{L}_{\mathrm{KL}} + \mathcal{L}_{\mathrm{reg}}, \tag{10}$$

where $\mathcal{L}_{\mathrm{reg}}$ combines volume, opacity, and Gaussian-offset regularizers. We use *no* UV-space reconstruction loss; the latent is learned purely from rendering.

3.2 Image-Conditioned Latent Diffusion

To model the latent distribution we employ rectified flow Lipman et al. (2022), whose forward process linearly interpolates between a data sample $\mathbf{z}_0$ and noise ϵ, $\mathbf{z}(t) = (1 - t)\,\mathbf{z}_0 + t\,\epsilon$. The backward process is a time-dependent vector field $\mathbf{v}(\mathbf{z}, t)$ that moves noisy samples toward the data, approximated by a network $\mathbf{v}_\theta$ trained with the conditional flow matching (CFM) objective Lipman et al. (2022):

$$L_{CFM}(\theta) = \mathbb{E}_{t,\,\mathbf{z}_0,\,\epsilon}\,\|\mathbf{v}_\theta(\mathbf{z}, t) - (\epsilon - \mathbf{z}_0)\|_2^2\,. \tag{11}$$

The architecture of $\mathbf{v}_\theta$ (Fig. 1 (c)) consists of cross/self-attention and feed-forward layers, with timestep information injected via adaptive layer normalization (AdaLN) and a gating mechanism Peebles & Xie (2023). Conditioning enters the cross-attention layers as keys and values: the image is encoded by DINOv2 Oquab et al. (2023) (vitl14_reg) and its layer-normalized patch tokens serve as keys and values. We train with classifier-free guidance (10% condition dropout) and a logit-normal timestep schedule, and sample with 50 steps at guidance scale 3.

4 Experiments

4.1 Experiment Settings

Dataset. Following previous image-to-avatar generation works Chen et al. (2025); Li et al. (2024), we collect 2,445 meshes from THuman2.1 Yu et al. (2021), 647 from CustomHuman Ho et al. (2023), and 1,096 from 2K2K Han et al. (2023). To further enlarge the training set, we reconstruct 4,000 meshes from MVHumanNet++ Li et al. (2026a) via a multi-view reconstruction method Zhang et al. (2026), giving about 8,000 meshes in total. We estimate SMPL-X parameters for every mesh with OmniFit Cai et al. (2026), which unifies the shape and pose distributions across sources instead of using each dataset's own fits; the full preprocessing pipeline and the resulting shape and pose distributions are detailed in Appendix B. For the 2K2K dataset, we remove certain subjects due to age-related differences that prevent reliable fitting. For evaluation, we select 100 scans from the X-Humans Shen et al. (2023) dataset, which contains 233 high-quality textured scan sequences from 20 participants. To ensure diversity in our test set, we select 5 scans for each participant, covering a range of ages, body shapes, poses, and appearances. We also download images from Unsplash website unsplash (2026) for qualitative in-the-wild evaluation.

Compared Methods and metrics. We evaluate on VAE reconstruction and image-to-avatar generation. For VAE reconstruction we compare with TRELLIS Xiang et al. (2025), a 3D-native object generation method using sparse voxels, and its version finetuned on the same dataset; we use its second-stage VAE, which decodes voxel-attached 3DGS from aggregated multi-view DINOv2 Oquab et al. (2023) features with a low-resolution voxel grid computed from the GT mesh as input.

For generation we further include SiTH Ho et al. (2024), PSHuman Li et al. (2024), DreamGaussian Tang et al. (2023), IDOL Zhuang et al. (2025), and SyncHuman Chen et al. (2025). IDOL is a single-view reconstruction method that learns an alignment between an image and a learnable UV-space latent. PSHuman and SyncHuman are multi-view diffusion methods that generate several views simultaneously; PSHuman fuses them via inverse rendering, while SyncHuman adopts a multi-view conditioned TRELLIS decoder. DreamGaussian is an SDS-based method requiring iterative optimization at a slower speed, and SiTH reconstructs shapes from single views with a ControlNet and implicit fields.

For evaluation metrics, we adopt the reference-based metrics PSNR, SSIM, and LPIPS, as well as the non-reference metric FID Mathiasen & Hvilshøj (2021). PSNR and SSIM measure pixel-level differences, LPIPS denotes perceptual differences, and FID is a distribution-level metric.

Implementations. In our experiments, the UV map resolution (H, W) is set to $(512, 512)$ and downsampled to $(h = 64, w = 64)$ using equation (5). The channel dimension c is 512. The latent code $\mathbf{z}$ has channel size $d = 8$. The number of sampled 3DGS points n is 200,800, following IDOL Zhuang et al. (2025). For the transformer-based encoder $\mathcal{E}$, decoder $\mathcal{D}$, and denoiser $\mathbf{v}_\theta$, we use the TRELLIS implementation with 12 blocks each and a middle channel size of 768. The VAE is trained on 4 A100 GPUs with a batch size of 4 per card for 2 days. The denoiser is trained on 4 A100 GPUs with a batch size of 8 per card for 2 days.

Table 1: VAE reconstruction on the 100-scan X-Humans test set. Best results in gold. Method references are given in Sec. 4.1.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
TRELLIS	20.55	0.9126	0.0941	25.21
TRELLIS (tuned)	22.87	0.9272	0.0846	19.36
Ours	23.46	0.9346	0.0780	10.22

4.2 Experimental Results of VAE-based Reconstruction

For VAE-based reconstruction we compare with TRELLIS and its finetuned version on the same human-specific dataset. Table 1 shows that our approach achieves the best performance on all metrics, indicating superior image-level accuracy and reduced distributional divergence; qualitative comparisons in Fig. 2 support this, as our method shows the highest faithfulness to the ground truth, accurately reconstructing regions such as the ponytail, head, and clothing where TRELLIS produces blurrier results. Even after finetuning, TRELLIS (tuned) produces an inaccurate face for the second row and blurry clothing for the third row. These findings highlight the superior ability of our VAE to compress 3D models into low-dimensional latent codes with limited human-specific data, which we attribute to the human prior introduced by expressing geometry as a UV Relief in UV space, whereas TRELLIS treats a human as a general object in 3D space.

Table 2: Image-conditioned generation on the 100-scan X-Humans test set. Best results are highlighted as 1st and 2nd. Method references are given in Sec. 4.1.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
DreamGaussian	16.19	0.8622	0.1499	37.08
SiTH	17.11	0.8849	0.1208	20.51
PSHuman	18.29	0.8916	0.1076	16.31
TRELLIS	17.26	0.8836	0.1211	25.57
IDOL	18.34	0.8750	0.1356	34.32
SyncHuman	18.75	0.8980	0.0999	14.48
Ours	20.79	0.9107	0.1009	15.20

4.3 Experimental Results of Generation

For the image-conditioned generation task, the quantitative comparison in Table 2 shows that our approach achieves the best PSNR and SSIM while remaining competitive on the perceptual metrics. In Fig. 3, our method generates correct arm anatomy and finer hand structures, while the compared methods produce blurrier results or distorted arms. This shows the advantage of the strong human

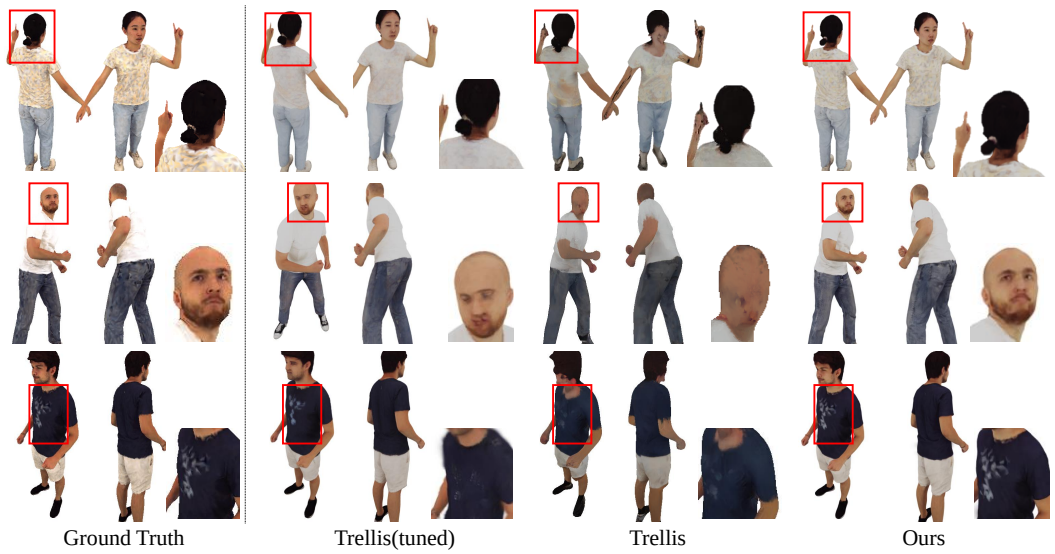


Figure 2: **Qualitative Comparison of VAE Reconstruction.**

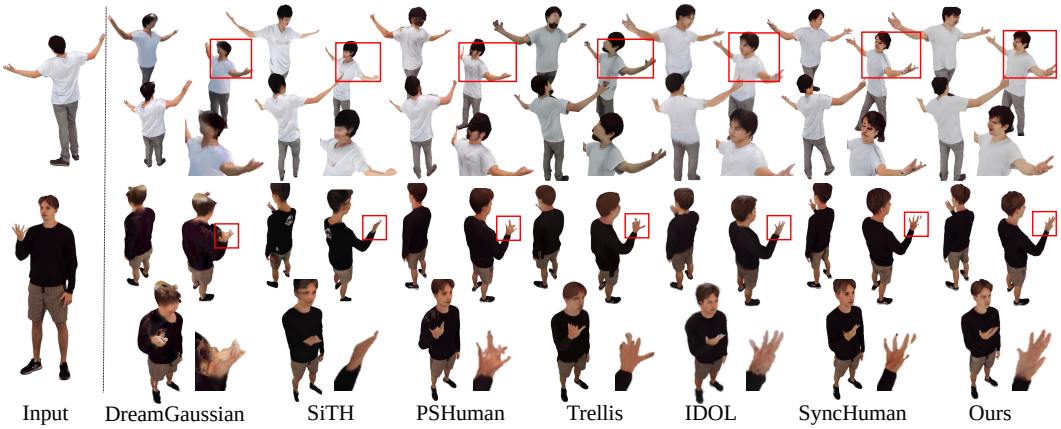


Figure 3: **Qualitative Comparison of Generation.**

prior provided by the SMPL-X model and the effective alignment of the UV-structured latent space. IDOL, which likewise adopts a human prior through cross attention between learnable UV latents and images, nevertheless produces blurred clothing and faces.

SyncHuman attains the best LPIPS and FID while our method leads on PSNR and SSIM; we further compare with the **multi-view images** it generated in Fig. 4. SyncHuman produces incorrect body structure such as the legs of the first row and the arm and head of the second row. This issue also occurs for other multi-view based methods such as PSHuman, showing an inherent limitation. In contrast, our approach consistently predicts accurate structures, even for challenging poses, due to its inherently 3D-native generation process.

To further validate the generation capability, we conduct **in-the-wild generation** in Fig. 5, using MultiHMR Baradel et al. (2024) for SMPL-X estimation. SyncHuman produces severe seam artifacts caused by inconsistency in the generated multi-view images, and Hunyuan3D Lai et al. (2025) tends to produce oversmoothed, cartoon-like appearance (second row), while TRELLIS produces further artifacts such as the broken legs. Our method still cannot fully preserve input details such as the red lining of the first row and facial details, which we attribute to the relatively small training data and imperfect SMPL-X fitting; larger-scale data, face-image conditioning, and more accurate fitting, as in Yang et al. (2025); Chen et al. (2025); Kostyrko et al. (2026), could improve on this. Nev-

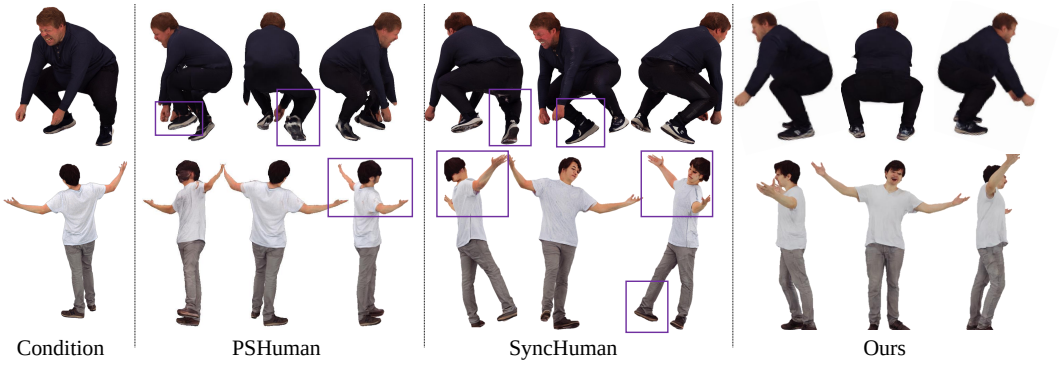


Figure 4: **Examples of both the input condition maps and the corresponding generated multi-view images.**

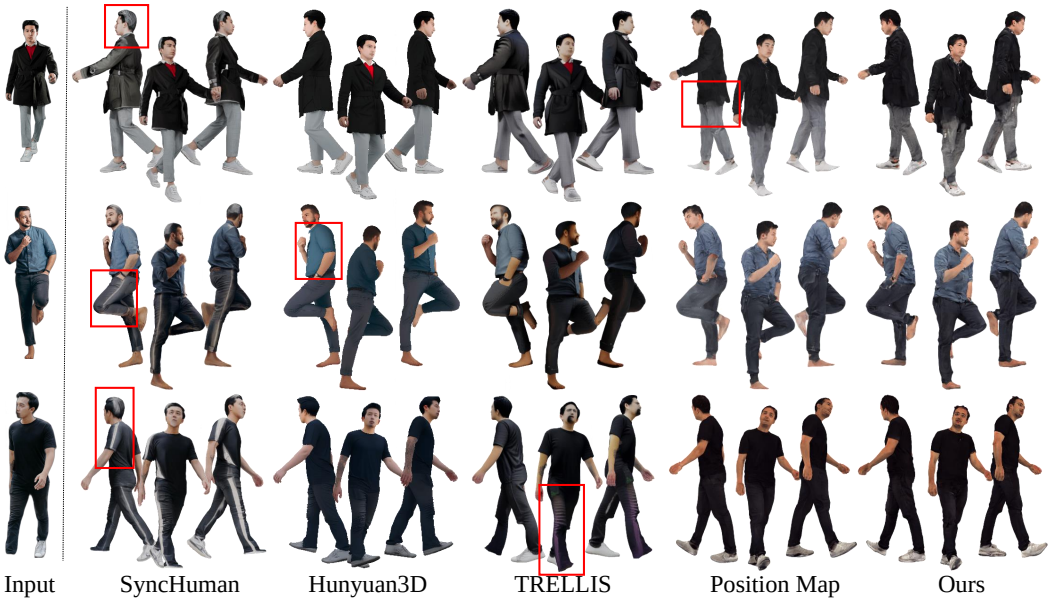


Figure 5: **In-the-wild results.** Each row is one input photograph; the columns show the results of each method (ours, position map, SyncHuman, TRELLIS, Hunyuan3D) from three viewpoints.

ertheless, it produces correct clothing geometry, such as the overcoat for the first row, and faithful clothing appearance for the second and third rows.

4.4 Ablation Study

We evaluate the encoder input for VAE reconstruction, which determines the quality of the latent space used for generation. Table 3 reports these ablations: our full model uses the UVRelief map (relief displacement plus texture); the *position map* variant replaces the relief displacement with the absolute 3D position of each texel; and the *cross attention* variant lets a learnable UV latent attend to multi-view images, following multi-view image-based VAEs such as SIGMAN.

Table 3 shows that the full model clearly outperforms the 'cross attention' entry by a large margin, while the 'position map' entry stays close on VAE reconstruction; in Fig. 6, 'cross attention' struggles to reconstruct complete limbs and faithful texture. While 'position map' is close to ours on the VAE metrics, the in-the-wild evaluation reveals a substantially larger gap: as shown in Fig. 5, it fails to reconstruct faithful clothing geometry such as the overcoat in the red box, which we attribute to



Ground Truth (a) Ours (b) Cross Attention (c) Position Map

Figure 6: **Qualitative results from the ablation study.**

Table 3: Encoder-side ablation. All variants share the decoder, data, and supervision; only the encoder input changes. "position map": absolute 3D-position encoding; "cross attention": a learnable UV latent attends to multi-view images (SIGMAN-like). Best results in gold.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
Ours	23.46	0.9346	0.0780	10.22
position map	23.35	0.9327	0.0815	10.76
cross attention	21.22	0.9182	0.0916	18.11

the distorted geometry of absolute positions in UV space as opposed to the relative shape of our full model. A supplementary video provides the in-the-wild comparison.

5 Limitations and Future Directions

The proposed UVRelief Codec has several limitations. It does not focus on facial regions, and the small face area in our training images leads to blurred facial details. Our training is limited to the dataset of Sec. 4.1, so scaling it, as in LCM and SIGMAN, may further improve generalization. UV Relief is a height field and cannot represent non-height-field topology such as multi-layer structure, a common limitation of avatar-as-image representations that a multi-layer field Zhang et al. (2025b) could address. Because the codec is defined relative to a fitted SMPL-X model, it inherits the accuracy of that fit, and baking appearance as per-texel RGB without separating albedo from lighting makes relighting difficult. Our results also correspond to single-pose transfer; extending them to temporally coherent dynamic sequences is an important direction for future work.

6 Conclusion

In this paper, we introduced UVRelief, a surface codec for high-fidelity 3D human avatar generation as an avatar-as-image representation. Unlike existing methods that rely on multi-view input or proceed in the original 3D space, UVRelief builds a *UV-native relative shape* from 3D models, represented as per-texel relief displacements and per-texel texture signals in UV Relief Maps. This combination of direct transport of geometry and human prior results in a compact, pose-agnostic, and spatially regular latent space that enables high-fidelity image-to-avatar generation.

REFERENCES

Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2shape: Detailed full human body geometry from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2293–2303, 2019.

Fabien Baradel, Matthieu Armando, Salma Galaaoui, Romain Brégier, Philippe Weinzaepfel, Grégory Rogez, and Thomas Lucas. Multi-hmr: Multi-person whole-body human mesh recovery in a single shot. In *European Conference on Computer Vision*, pp. 202–218. Springer, 2024.

Blender. Blender. `https://www.blender.org/`, 2024.

Zeyu Cai, Yuliang Xiu, Renke Wang, Zhijing Shao, Xiaoben Li, Siyuan Yu, Chao Xu, Yang Liu, Baigui Sun, Jian Yang, and Zhenyu Zhang. Omnifit: Multi-modal 3d body fitting via scale-agnostic dense landmark prediction. *arXiv preprint arXiv:2604.21575*, 2026.

Wenyue Chen, Peng Li, Wangguandong Zheng, Chengfeng Zhao, Mengfei Li, Yaolong Zhu, Zhiyang Dou, Ronggang Wang, and Yuan Liu. Synchuman: Synchronizing 2d and 3d generative models for single-view human reconstruction. *arXiv preprint arXiv:2510.07723*, 2025.

Daniel Gatis. rembg, February 2025. URL `https://github.com/danielgatis/rembg`.

Sang-Hun Han, Min-Gyu Park, Ju Hong Yoon, Ju-Mi Kang, Young-Jae Park, and Hae-Gon Jeon. High-fidelity 3d human digitization from single 2k resolution images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12869–12879, 2023.

Hsuan-I Ho, Lixin Xue, Jie Song, and Otmar Hilliges. Learning locally editable virtual humans. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pp. 21024–21035, 2023.

Hsuan-I Ho, Jie Song, and Otmar Hilliges. Sith: Single-view textured human reconstruction with image-conditioned diffusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Yujiao Jiang, Qingmin Liao, Xiaoyu Li, Li Ma, Qi Zhang, Chaopeng Zhang, Zongqing Lu, and Ying Shan. Uv gaussians: Joint learning of mesh deformation and gaussian textures for human avatar modeling. *arXiv preprint arXiv:2403.11589*, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL `https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/`.

Margaret Kostyrko, Yuxuan Xue, Garvita Tiwari, and Gerard Pons-Moll. Revisiting avatar-as-image: High-fidelity registration is all you need. In *European Conference on Computer Vision (ECCV)*, 2026.

Zorah Lahner, Daniel Cremers, and Tony Tung. Deepwrinkles: Accurate and realistic clothing modeling. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 667–684, 2018.

Zeqiang Lai, Yunfei Zhao, Haolin Liu, Zibo Zhao, Qingxiang Lin, Huiwen Shi, Xianghui Yang, Mingxin Yang, Shuhui Yang, Yifei Feng, et al. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details. *arXiv preprint arXiv:2506.16504*, 2025.

Verica Lazova, Eldar Insafutdinov, and Gerard Pons-Moll. 360-degree textures of people in clothing from a single image. In *International Conference on 3D Vision (3DV)*, 2019.

Chenghong Li, Hongjie Liao, Yihao Zhi, Xihe Yang, Zhengwentai Sun, Jiahao Chang, et al. Mvhumannet++: A large-scale dataset of multi-view daily dressing human captures with richer annotations for 3d human digitization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026a.

Junxuan Li, Rawal Khirodkar, Chengan He, Zhongshi Jiang, Giljoo Nam, Lingchen Yang, Jihyun Lee, Egor Zakharov, Zhaoen Su, Rinat Abdrashitov, et al. Large-scale codec avatars: The unreasonable effectiveness of large-scale avatar pretraining. *arXiv preprint arXiv:2604.02320*, 2026b.

Peng Li, Wangguandong Zheng, Yuan Liu, Tao Yu, Yangguang Li, Xingqun Qi, Xiaowei Chi, Siyu Xia, Yan-Pei Cao, Wei Xue, et al. Pshuman: Photorealistic single-view human reconstruction using cross-scale diffusion. In *European Conference on Computer Vision (ECCV)*, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Alexander Mathiasen and Frederik Hvilshøj. Backpropagating through fréchet inception distance, 2021. URL https://arxiv.org/abs/2009.14075.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10975–10985, 2019.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

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 *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9054–9063, 2021.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting. *arXiv preprint arXiv:2312.09228*, 2023.

Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Kaiyue Shen, Chen Guo, Manuel Kaufmann, Juan Jose Zarate, Julien Valentin, Jie Song, and Otmar Hilliges. X-avatar: Expressive human avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16911–16921, 2023.

Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023.

unsplash. unsplash. https://unsplash.com/, 2026.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025a.

Yanwen Wang, Yiyu Zhuang, Jiawei Zhang, Li Wang, Yifei Zeng, Xun Cao, Xinxin Zuo, and Hao Zhu. Tera: Rethinking text-guided realistic 3d avatar generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10686–10697, 2025b.

Yiqian Wu, Malte Prinzler, Xiaogang Jin, and Siyu Tang. Text-based animatable 3d avatars with morphable model alignment. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–11, 2025.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21469–21480, 2025.

Xingguang Yan, Han-Hung Lee, Ziyu Wan, and Angel X. Chang. An object is worth 64x64 pixels: Generating 3d object via image diffusion. In *International Conference on 3D Vision (3DV)*, 2025.

Yuhang Yang, Fengqi Liu, Yixing Lu, Qin Zhao, Pingyu Wu, Wei Zhai, Ran Yi, Yang Cao, Lizhuang Ma, Zheng-Jun Zha, et al. Sigman: Scaling 3d human gaussian generation with millions of assets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5122–5133, 2025.

Tao Yu, Zerong Zheng, Kaiwen Guo, Pengpeng Liu, Qionghai Dai, and Yebin Liu. Function4d: Real-time human volumetric capture from very sparse consumer rgbd sensors. *arXiv preprint arXiv: 2105.01859*, 2021.

Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)*, 42(4):1–16, 2023a.

Jingbo Zhang, Xiaoyu Li, Hongliang Zhong, Qi Zhang, Yanpei Cao, Ying Shan, and Jing Liao. Humanref-gs: Image-to-3d human generation with reference-guided diffusion and 3d gaussian splatting. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025a.

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)*, 43(4):1–20, 2024a.

Rui Zhang and Jie Chen. Ndf: Neural deformable fields for dynamic human modelling. In *European Conference on Computer Vision (ECCV)*, 2022.

Ruiqi Zhang and Jie Chen. Mesh-centric gaussian splatting for human avatar modelling with real-time dynamic mesh reconstruction. In *ACM Multimedia 2024*, 2024.

Ruiqi Zhang, Jie Chen, and Qiang Wang. Explicifying neural implicit fields for efficient dynamic human avatar modeling via a neural explicit surface. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 1955–1963, 2023b.

Ruiqi Zhang, Jiacheng Wu, and Jie Chen. Mesh splatting for end-to-end multiview surface reconstruction. *arXiv preprint arXiv:2601.21400*, 2026.

Weitian Zhang, Yichao Yan, Yunhui Liu, Xingdong Sheng, and Xiaokang Yang. E3gen: Efficient, expressive and editable avatars generation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, pp. 68606869, 2024b.

Weitian Zhang, Yichao Yan, Sijing Wu, Manwen Liao, and Xiaokang Yang. Disentangled clothed avatar generation with layered representation, 2025b. URL https://arxiv.org/abs/2501.04631.

Peng Zheng, Dehong Gao, Deng-Ping Fan, Li Liu, Jorma Laaksonen, Wanli Ouyang, and Nicu Sebe. Bilateral reference for high-resolution dichotomous image segmentation. *CAAI Artificial Intelligence Research*, 2024.

Yiyu Zhuang, Jiaxi Lv, Hao Wen, Qing Shuai, Ailing Zeng, Hao Zhu, Shifeng Chen, Yujiu Yang, Xun Cao, and Wei Liu. Idol: Instant photorealistic 3d human creation from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26308–26319, 2025.

AI USE STATEMENT

We used generative AI tools only for language polishing—improving grammar, wording, and readability—of the manuscript. We have not used generative AI tools to conceive research ideas, design experiments, implement methods, generate results or figures, or interpret findings. All AI-assisted text was reviewed and verified by the authors, who take responsibility for the final content of this work.

In this appendix, we will introduce the following contents:

- UV Relief Codec Visualization in Appendix A
- Data Preprocessing in Appendix B.
- Decoding-Target Ablation in Appendix C.
- Geometric Evaluation in Appendix D.
- Method Selection and Evaluation Protocols in Appendix E.

Supplementary videos of 360° orbiting renderings of the generated avatars accompany this paper.

A UVRELIEF CODEC VISUALIZATION

This section visualizes the UVRelief Codec process, as described in Sec. 3, demonstrating the transformation from the ground truth mesh to the reconstructed colored point cloud. Specifically, we illustrate each stage: beginning with the ground truth mesh, progressing through the UV relief, relief displacement, and RGB map representations, and culminating in the reconstructed colored point cloud. To showcase the robustness and versatility of our approach, we provide samples from diverse datasets encompassing various human types and genders. As highlighted by the red boxes, our method faithfully recovers fine textures in regions such as pants and loose clothing, demonstrating its effectiveness in preserving intricate garment details. These results confirm that UVRelief can efficiently adapt 3D models into compact 2D representations, accurately restore details from the original 3D data, and establish a solid foundation for neural network workflows—including powerful VAE training and future scalability.

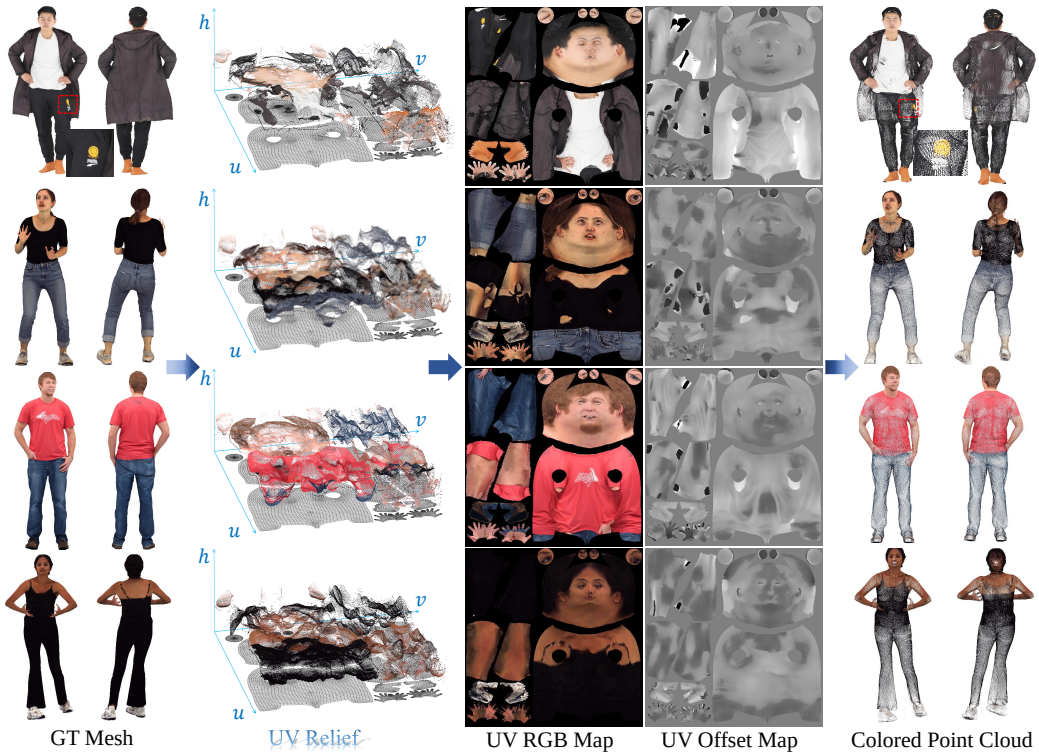


Figure 7: **UVRelief Codec Visualization.** The stages of the UVRelief Codec process are shown, starting from the ground truth mesh and progressing through UV relief, relief displacement, and RGB map, to the reconstructed colored point cloud.

B Data Preprocessing

Our training set combines three public scan datasets (THuman2.1, CustomHuman, and 2K2K) with additional data from MVHumanNet++, from which we reconstruct 4,000 meshes via multi-view reconstruction, giving about 8,000 meshes in total. Because the parametric fits shipped with these sources differ substantially in shape and pose distribution, we do not use them directly. Instead, we fit SMPL-X to every mesh with OmniFit Cai et al. (2026), a scale-agnostic multi-modal body-fitting method, and align all fits to a common neutral, real-world-scale frame. From the fitted model we then bake the UV Relief Maps: we sample texels on the shared UV layout, emit rays from the SMPL-X surface along its normals, and record the per-texel relief displacement and texture. We visualize the SMPL-X shape and pose parameters with t-SNE before and after OmniFit (Figs. 8 and 9): the dataset-provided fits form clearly separated clusters, whereas OmniFit maps all sources into a single, well-overlapping distribution, which we found to be important for training a shared latent space.

Viewpoints. We use three disjoint sets of cameras. During training we render 150 views around each mesh and supervise the decoded 3D Gaussians through the differentiable rasterizer. The condition image for the diffusion prior is drawn from a separate set of 24 views that also surround the mesh, with pitch limited to $(-45°, 45°)$; a random one is selected per instance during training. At evaluation we render 100 golden-angle views with pitch in $[-60°, 60°]$, compare them against the ground-truth mesh rendered with the same cameras, and use ICP to align methods that do not predict metric-scale output.

Codec training objective. The UVRelief VAE is trained with a rendering loss, a KL term, and geometry regularizers:

$$\mathcal{L} = \mathcal{L}_{\mathrm{L1}} + 0.2\,(1 - \mathrm{SSIM}) + 0.5\,\mathcal{L}_{\mathrm{LPIPS}} + 10^{-6}\mathcal{L}_{\mathrm{KL}} + 10^{4}\mathcal{L}_{\mathrm{vol}} + 10^{-3}\mathcal{L}_{\mathrm{opac}} + \mathcal{L}_{\mathrm{off}}. \quad (12)$$

Here $\mathcal{L}_{\mathrm{L1}}$, SSIM and $\mathcal{L}_{\mathrm{LPIPS}}$ are computed on the rendered images; the regularizers act on the decoded Gaussians: $\mathcal{L}_{\mathrm{vol}} = \mathrm{mean}(\prod_k \sigma_k)$ is their mean volume, $\mathcal{L}_{\mathrm{opac}} = \mathrm{mean}((\alpha - 1)^2)$ encourages opacity close to one, and $\mathcal{L}_{\mathrm{off}} = \mathrm{mean}(\|\delta\mathbf{x}\|_2^2)$ shrinks the per-Gaussian position residual; $\mathcal{L}_{\mathrm{KL}} = \frac{1}{2}\,\mathrm{mean}(\boldsymbol{m}^2 + \boldsymbol{\sigma}^2 - \log\boldsymbol{\sigma}^2 - 1)$ is the KL divergence from the encoder posterior $(\boldsymbol{m}, \boldsymbol{\sigma})$ to a standard normal. There is *no* UV-space reconstruction loss; the latent is learned purely from rendering.

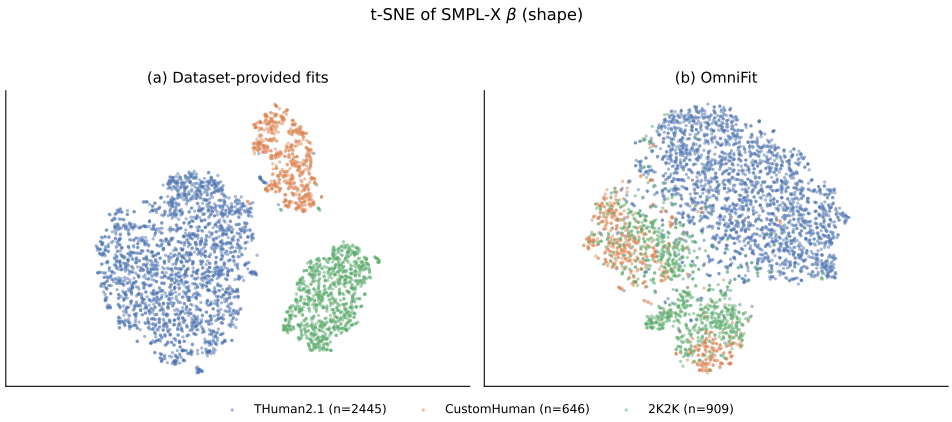


Figure 8: **t-SNE of SMPL-X shape parameters (β).** Left: dataset-provided fits; right: OmniFit refits. OmniFit maps all sources onto a single, well-overlapping distribution.

C Decoding-Target Ablation

We ablate the decoder while keeping the encoder input (UVRelief maps) fixed. D0 decodes directly to texel-aligned 3D Gaussians (our default); D1 auto-encodes the UV map and converts it to 3DGS with a fixed hand-crafted rule; D2 adds a learned CNN for the conversion; D3 predicts UV attributes and 3D Gaussian attributes jointly. All four variants use the same 3DGS rasterizer, so the comparison isolates the decoder design from the rendering backend. Table 4 reports the results.

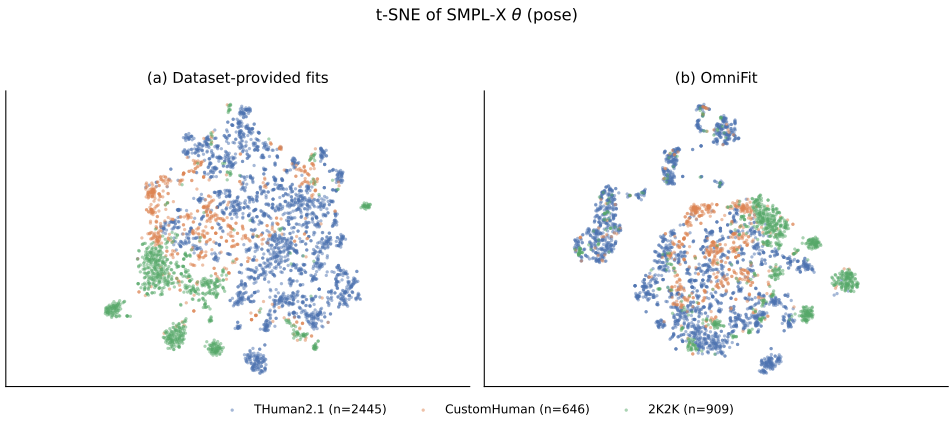


Figure 9: **t-SNE of SMPL-X pose parameters (θ).** Left: dataset-provided fits; right: OmniFit refits.

Table 4: Decoding-target ablation.

#	Variant	PSNR ↑	SSIM ↑	LPIPS ↓
D0	UV→3DGS (default)	**23.46**	**0.9346**	**0.0780**
D1	UV→UV → manual 3DGS	19.77	0.8855	0.1181
D2	UV→UV → CNN → 3DGS	23.51	0.9334	0.0788
D3	UV→UV + 3DGS (joint)	23.42	0.9329	0.0790

Two conclusions follow. *Decoding must be learned*: replacing the learned decoder head with a fixed hand-crafted conversion drops PSNR by 3.7 dB (D1 vs. D0). *The specific decoder design matters little*: direct decoding (D0, 23.46 dB) is matched within 0.05 dB by the two-stage and joint variants (D2: 23.51 dB, D3: 23.42 dB). The encoder input is thus the decisive factor, which is what our controlled comparison studies.

D Geometric Evaluation

While our main results are rendering-space metrics, we also measure geometry directly with the bidirectional Chamfer Distance (CD, in cm) under one shared protocol: predictions are ICP-aligned to the ground-truth mesh, and CD is computed between 100K randomly sampled points. Table 5 groups methods by output format (3DGS or mesh). Unlike IDOL Zhuang et al. (2025), which also outputs UV-aligned 3DGS but takes a single image as input, our VAE with UVRelief maps achieves substantially better geometry (1.57 cm vs. 4.28 cm), confirming that the geometric advantage comes from the UVRelief input rather than from the decoder alone.

Protocol nuance. CD is more favorable to mesh outputs: the ground truth is a mesh surface, so mesh-based baselines are sampled on the surface, whereas 3DGS outputs are measured on Gaussian centers that need not lie on it. This explains why Trellis_VAE reports a lower CD (1.15 vs. 1.57 cm), while the protocol remains fair across all 3DGS methods (same extraction for Ours_VAE, Ours_generation, and IDOL). By contrast, the rendering metrics (Table 1) show a clear advantage for our method.

Codec round-trip. Independently of the VAE, we measure the fidelity of the UVRelief codec itself (mesh → UVRelief maps → recovered colored point cloud) over the full X-Humans test set: CD $= 0.4$ cm, confirming that the encoding preserves input geometry with high fidelity.

Table 5: Chamfer Distance (cm) under one shared protocol.

Output	Method	Mean CD ↓	Median CD ↓
3DGS	Ours_VAE	**1.57**	**1.53**
	Ours_generation	1.96	1.92
	IDOL Zhuang et al. (2025)	4.28	3.27
Mesh	Trellis_VAE Xiang et al. (2025)	1.15	1.17
	Trellis_generation Xiang et al. (2025)	3.04	2.91
	SyncHuman Chen et al. (2025)	2.62	2.46
	PSHuman Li et al. (2024)	2.93	2.74
	SiTH Ho et al. (2024)	3.49	3.45

E Method Selection and Evaluation Protocols

In this section, we discuss several related approaches that were not included in our experimental benchmark, along with the rationale for their exclusion:

LCM Li et al. (2026b) utilizes a large transformer architecture to encode human body representations. However, this work is not open-source, and neither the code nor the pretrained models are available for reproducibility or direct comparison. For this reason, LCM could not be included as a baseline in our study.

E3Gen Zhang et al. (2024b) and LayerAvatar Zhang et al. (2025b): Both E3Gen and LayerAvatar employ UV-space Gaussian attribute maps to learn underlying representations and use diffusion-based denoising U-Nets for generative modeling. While these methods are relevant to our research, E3Gen, as described in its original publication, does not support image-conditioned inference. This makes it unsuitable for tasks that require generation based on image inputs. LayerAvatar proposes image conditioning; however, according to its official GitHub repository, it requires users to independently train an image-conditioned U-Net to enable this functionality. Given the absence of ready-to-use, image-conditioned inference pipelines and the significant effort required for additional training, neither E3Gen nor LayerAvatar is appropriate as a primary baseline for our experiments. Consequently, we do not include comparative results for these methods.

SIGMAN Yang et al. (2025): SIGMAN jointly trains a 3D encoder and a 2D decoder using a large dataset of 3DGS human avatars, with the goal of producing high-quality 3DGS models. Although SIGMAN's framework is closely related to our research aims, our evaluation showed that the official implementation and pretrained models fail to provide satisfactory image quality for both VAE reconstruction and generation. As illustrated in Fig. 10, the output renderings from reconstructed models—regardless of whether they were produced using *Render Views (VAE)* or *Render Views (generation)*—exhibited notably poor visual fidelity when generated from the provided input images and SMPL-X models. To address these issues, we contacted the authors to request their evaluation data and details regarding training viewpoints. However, their responses were limited and unclear, and after sharing our own evaluation results generated with the official code, we received no further feedback. Due to these unresolved technical barriers and the insufficient performance of the released models, SIGMAN was excluded from our experimental comparisons.

E.1 Evaluation Protocols: Input Image Preparation

Before evaluation, all input images are processed using a standardized pipeline to ensure consistency across methods, following the approach of DreamGaussian Tang et al. (2023). Each image is loaded at full resolution and undergoes automated background removal using *Rembg* Gatis (2025) with the *bria-rmbg* Zheng et al. (2024) model, resulting in a high-quality RGBA image with a transparent background. After background removal, the foreground mask is used to tightly crop the subject. The cropped area is then recentered and uniformly scaled to fit within a fixed output size of 1024×1024 pixels. Although the pipeline includes an adjustable border parameter to control the margin around the subject, this is set to zero for all experiments. This process ensures that every input is centered and identically scaled, providing a fair basis for avatar synthesis and evaluation. All processed



Figure 10: **SIGMAN Results.** *Render Views (VAE)* shows outputs reconstructed from SIGMAN's VAE using the demo input images and provided SMPL-X model. *Render Views (generation)* shows results from SIGMAN's generated 3DGS based on images from the website folder and the associated SMPL-X model.

images are saved as PNG files with the alpha channel retained, ensuring clean segmentation for subsequent tasks and analyses.

E.2 Evaluation Protocols: Renderer Setting

We use a differentiable Gaussian splatting renderer, *diff-gaussian-rasterization* Kerbl et al. (2023), alongside PyTorch3D Ravi et al. (2020), to render the predicted 3DGS primitives and baseline meshes, respectively. The renderer takes camera parameters in intrinsic-extrinsic format, $\mathbf{K} = \{f_x, f_y, c_x, c_y\}$, along with a scaled camera-to-world matrix $\mathbf{w2c}_{\text{scaled}}$, with the rotation component scaled by a factor of 0.5 following the IDOL convention. For each Gaussian primitive, we calculate the view-space covariance matrix using learnable scale and rotation parameters, then rasterize to produce per-pixel color through α-blending. The renderer can automatically switch between different kernelization backends, supporting both antialiasing and kernel_size/subpixel_offset modes.

For ground-truth geometry, we employ an offscreen rasterizer based on PyTorch3D. Mesh vertices are projected into normalized device coordinates using the same camera intrinsics and extrinsics. Rendering uses perspective-correct projection with z-culling to preserve proper depth relationships. Qualifying vertices are then rasterized and rendered as a point cloud with per-vertex colors. This approach ensures that both the generated 3DGS models and ground-truth visualizations are geometrically consistent when rendered under identical camera settings.

E.3 Evaluation Protocols: Mesh Normalization and Preprocessing

E.3.1 Canonical Alignment

To enable fair geometric comparison between predicted and ground-truth geometries, we normalize all meshes to a canonical frame:

1. **Centering:**

$$\mathbf{v}' = \mathbf{v} - \bar{\mathbf{v}}$$

where $\bar{\mathbf{v}}$ is the centroid.

2. **Scale Normalization:** We compute the radius as $r = \max_i \|\mathbf{v}'_i\|_2$, then scale to target scale s_{target} via:

$$\mathbf{v}'' = \mathbf{v}' \cdot \frac{s_{\text{target}}}{r}$$

3. **Rotation Correction:** We apply a fixed rotation around the Y-axis by angle θ_{rot} (e.g., 180°) to align canonical pose conventions:

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

This normalization ensures that both the predicted and ground-truth geometries are represented within a common coordinate frame prior to ICP alignment.

E.4 Evaluation Protocols: ICP-based Geometric Alignment

Following the evaluation protocol of SiTH Ho et al. (2024), we use ICP to align the predicted meshes with the ground-truth meshes. However, standard point-cloud ICP can easily converge to local minima, especially because human body symmetries often lead to front-back pose ambiguities. To mitigate this, we initialize ICP using multiple yaw rotation hypotheses:

$$\{\theta_0 = 0^\circ, 90^\circ, 180^\circ, 270^\circ\}$$

For each hypothesis, we perform the following steps:

E.4.1 Uniform Scale Estimation

We estimate a global scale factor by comparing point cloud radii:

$$s = \frac{\text{mean radius of GT}}{\text{mean radius of prediction}}$$

E.4.2 Initial Transform Composition

For each yaw candidate θ_i, we compose the following transforms in sequence:

- **Uniform scaling:** $\mathbf{S} = s\mathbf{I}$
- **Rotation around GT centroid:** $\mathbf{R}_i = \mathbf{R}_y(\theta_i)$
- **Translation to GT frame:** $\mathbf{t}_i = \bar{\mathbf{v}}_{\text{gt}} - \mathbf{R}_i(\bar{\mathbf{v}}_{\text{pred}} \cdot s)$

The composed initial transformation is:

$$\mathbf{T}_{\text{init}}^{(i)} = \begin{bmatrix} s\mathbf{R}_y(\theta_i) & \mathbf{t}_i \\ \mathbf{0} & 1 \end{bmatrix}$$

E.4.3 Point Cloud ICP

We apply point-to-point ICP (*trimesh.registration.icp*) to optimize the alignment between meshes, allowing up to 150 iterations per run. To reduce computational cost for large point clouds, we uniformly sample a maximum of $N_{\text{max}} = 60,000$ points from the original set via random selection:

$$\mathcal{P}_{\text{sub}} = \mathcal{P}\left[\text{randperm}(N, N_{\max})\right]$$

The final rigid transformation obtained from ICP is subsequently applied to the entire point set.

E.4.4 BEST HYPOTHESIS SELECTION

To identify the optimal alignment among multiple yaw hypotheses, we compute the mean nearest-neighbor distance (MNN) between the aligned prediction and ground truth:

$$\text{MNN}^{(i)} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \min_{g \in \mathcal{G}} \|p - g\|_2$$

The hypothesis yielding the lowest MNN is selected as the final alignment:

$$i^* = \arg\min_i \text{MNN}^{(i)}$$

E.5 COVARIANCE UPDATE FOR 3DGS

For meshes, the derived rigid transformation from ICP can be directly applied to all vertices. However, for 3DGS models, which involve additional parameters such as per-Gaussian rotation and scale, further adjustment is required to maintain correct spatial relationships after alignment.

After obtaining the optimal rigid transformation $\mathbf{T}_{\text{icp}}$ and scale factor s, we update the Gaussian covariance matrices. For each Gaussian with learnable rotation $\mathbf{q}_i$ and scale $\boldsymbol{\sigma}_i$:

Let $\mathbf{M}_i$ denote the 3×3 rotation matrix derived from quaternion $\mathbf{q}_i$, and $\boldsymbol{\Sigma}_i = (\mathbf{M}_i \text{diag}(\boldsymbol{\sigma}_i))(\mathbf{M}_i \text{diag}(\boldsymbol{\sigma}_i))^T$ be the original covariance matrix.

After alignment with rotation component $\mathbf{R}_{\text{icp}}$ and scale s:

$$\mathbf{M}'_i = \frac{1}{s} \mathbf{R}_{\text{icp}} \mathbf{M}_i$$

$$\boldsymbol{\sigma}'_i = \boldsymbol{\sigma}_i \cdot s$$

$$\boldsymbol{\Sigma}'_i = (\mathbf{M}'_i \text{diag}(\boldsymbol{\sigma}'_i))(\mathbf{M}'_i \text{diag}(\boldsymbol{\sigma}'_i))^T$$

This process ensures that the adjusted 3DGS geometry remains geometrically consistent with the ground truth under the applied transformation, enabling accurate per-pixel comparisons between predicted and reference renderings.