

# Bringing Your Portrait to 3D Presence

## Supplementary Material



Figure 1. A conceptual illustration of *Bringing Your Portrait to 3D Presence*. Our pipeline transforms everyday portrait images into fully controllable 3D avatars that can be animated via a tracked proxy mesh. The model is trained entirely on a hybrid synthetic corpus combining rendered and generative sources. Thanks to our dual-UV representation, the system robustly handles inputs of varying completeness—ranging from head-only to half-body or full-body portraits—within a single unified framework.

We provide the model design and training procedure in Sec. A, the details of our dataset curation in Sec. B, the proxy-mesh estimation pipeline in Sec. C, and more experiments in Sec. D. Additional qualitative results are shown in the supplementary videos.

## A. Model Details

### A.1. Model Design

After scattering UV features, we add separate learnable positional embeddings for different UV branches. For the *core-UV* branch, we initialize the positional embedding in UV coordinates. Concretely, we rasterize the vertices of the canonical-space template mesh into the UV plane and obtain a position map shown in Fig. 2, with the same spatial resolution as the UV feature map. It is worth noting that, although GSM [?] also adopts a shell-based design, our Shell-UV is different in design. While GSM adds extra 3D Gaussian layers for geometric expressiveness, our Shell-UV adds no Gaussians and instead uses canonical UV projection to reduce pose- and framing-induced misalignment under partial visibility. We then apply an  $L$ -frequency si-

nusoidal encoding to each UV coordinate, with  $L = 8$ , and pass the encoded features through a linear layer to project them to the Sapiens feature dimension. For the *shell-UV* branch, we initialize the learnable tokens with Gaussian noise.

The UV features with positional embeddings are linearly projected to 1024 and processed by 8 self-attention blocks with 16 attention heads each, to model interactions among tokens. We then perform unpatchify with patch size 8 to form Gaussian attribute maps. For different Gaussian attributes, we use separate decoder heads, each implemented as a two-layer MLP with 256 hidden dimension and SiLU activation. We uniformly sample from the decoded Gaussian attribute maps and slightly retopologize the UV layout so that the utilization of tokens is as high as possible.

During training, we randomly mask from 0% to 50% of the scattered UV features to improve robustness. At test time, we use the input image mask to filter out points that fall outside the masked region due to mesh misalignment. This simple trick effectively prevents artifacts caused by proxy-mesh misalignment.

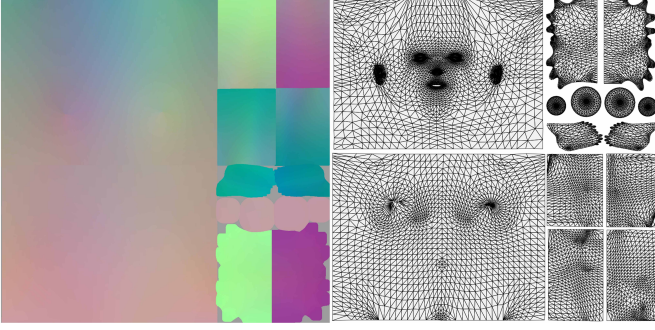


Figure 2. **UV Topology Visualization and Position Map.** We visualize the modified UV topology and the corresponding position map used for sinusoidal encoding.

## A.2. Loss Function

Given a reference–target pair, we predict canonical Gaussian attributes from the reference, rig them to the target mesh, and render both views. The total objective combines reconstruction and regularization terms:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{regu}}. \quad (1)$$

**Reconstruction loss.** For each view  $v \in \{\text{ref}, \text{tgt}\}$ , we supervise image fidelity using pixel and perceptual losses:

$$\mathcal{L}_{\text{rec}}^{(v)} = \lambda_{\text{L1}} \left\| \hat{I}^{(v)} - I^{(v)} \right\|_1 + \lambda_{\text{LPIPS}} \mathcal{L}_{\text{LPIPS}} \left( \hat{I}^{(v)}, I^{(v)} \right), \quad (2)$$

with  $\lambda_{\text{L1}} = 1.0$  and  $\lambda_{\text{LPIPS}} = 0.5$ . The reconstruction loss sums over both views:

$$\mathcal{L}_{\text{rec}} = \mathcal{L}_{\text{rec}}^{\text{ref}} + \mathcal{L}_{\text{rec}}^{\text{tgt}}. \quad (3)$$

**Geometry regularization.** We constrain Gaussian geometry parameters to ensure stability. Offsets are penalized to prevent drift:

$$\mathcal{L}_o = \|\mathbf{d}\|_2, \quad (4)$$

and scales  $\mathbf{s} = [s_x, s_y, s_z]$  are regularized for compactness and isotropy:

$$\mathcal{L}_s = \sum_{i \in \{x, y, z\}} s_i, \quad (5)$$

$$\mathcal{L}_r = \max \left( \frac{\max(\mathbf{s})}{\min(\mathbf{s})} - r, 0 \right), \quad (6)$$

where  $r = 9$ . The geometry regularization term is

$$\mathcal{L}_{\text{geo}} = \lambda_o \mathcal{L}_o + \lambda_s \mathcal{L}_s + \lambda_r \mathcal{L}_r. \quad (7)$$

**Texture regularization.** We further regularize Gaussian appearance to avoid implausible color and opacity distributions. A hand-consistency loss aligns hand and facial color

statistics by matching each hand patch  $\mathbf{P}_i^h$  to its nearest facial patch  $\mathbf{P}_j^f$  (gradient detached):

$$\mathcal{L}_h = \sum_i \min_j \|\mathbf{P}_i^h - \mathbf{P}_j^f\|_2. \quad (8)$$

In addition, we impose a patch-level regularization on the opacity. For each Gaussian opacity map, we divide it into patches and enforce that the average opacity in each patch is close to a target value  $\alpha_{\text{ref}}$ , which we set as 0.8. The opacity loss for each patch is defined as:

$$\mathcal{L}_\alpha = -\frac{1}{N} \sum_{k=1}^N (\alpha_{\text{ref}} \cdot \log(\mu_k) + (1 - \alpha_{\text{ref}}) \cdot \log(1 - \mu_k)), \quad (9)$$

where  $\mu_k$  is the mean opacity of the  $k$ -th patch, and  $N$  is the number of patches.

The texture regularization term is

$$\mathcal{L}_{\text{tex}} = \lambda_h \mathcal{L}_h + \lambda_\alpha \mathcal{L}_\alpha. \quad (10)$$

Finally, the total regularization is

$$\mathcal{L}_{\text{regu}} = \mathcal{L}_{\text{geo}} + \mathcal{L}_{\text{tex}}, \quad (11)$$

with  $(\lambda_o, \lambda_s, \lambda_r, \lambda_h, \lambda_\alpha) = (1.0, 0.1, 1.0, 0.1, 0.1)$  in all experiments.

## B. Dataset Details

### B.1. Synthetic Rendering Branch

Our synthetic rendering pipeline begins by sampling shape and pose parameters from the SOMA parametric human model [? ]. Given the sampled body and pose coefficients, we apply texture, hair and clothing assets upon posed mesh. And further sample a HDR image to set up environment lightning. The full scene is rendered using the Cycles rendering engine. For camera configuration, we adaptively determine the look-at point based on the posed subject rather than using a fixed center. Let  $v_{\min}$  and  $v_{\max}$  denote the minimum and maximum vertex positions in world space along the vertical axis, and let  $\mathbf{p}_{\text{pel}}$  and  $\mathbf{p}_{\text{head}}$  be the pelvis and head joint locations, respectively. We define the horizontal look-at target as

$$\mathbf{t}_{xy} = \frac{\mathbf{p}_{\text{pel}} + \mathbf{p}_{\text{head}}}{2}, \quad (12)$$

and the vertical target coordinates as

$$t_z = v_{\min} + \lambda \cdot (v_{\max} - v_{\min}), \quad \lambda = 0.75. \quad (13)$$

Thus, the final camera look-at position is  $\mathbf{t} = (\mathbf{t}_{xy}, t_z)$ , which biases the view toward the upper-body and yields perceptually stable framing across diverse poses and body shapes.

In addition to the rendered images, we export the corresponding SOMA mesh. We then convert it into an SMPL-X using a pre-computed regression matrix, followed by a parameter inversion step in which we optimize the SMPL-X parameters via trust-region Newton conjugate gradient method [?] to best match the converted mesh. We show samples from the rendered dataset in Fig. 6.

## B.2. Generative Branch

### B.2.1. Semantic Factorization

We construct a factorized latent space by decomposing the generative prior into semantically meaningful axes. For each factor, we curate a dedicated vocabulary that captures its underlying variation (e.g., appearance attributes, scene composition, or stylistic cues). These vocabularies collectively define controllable semantic directions along which data diversity can be systematically spanned. These vocabularies are automatically expanded using GPT-5 to ensure broad coverage while preserving the semantic purity of each factor.

- *Actions* have 131 carefully constrained micro-gestures grouped implicitly (speaking, explanatory, facial, grooming, fit-check) and phrased as short dynamic–return patterns to stabilize temporal synthesis while avoiding occlusion or exaggerated motion.
- *Hair* splits into 26 natural color variants and 54 face-visible styles spanning length, texture, braids, locs, curls, and updos.
- *Lighting control* uses 27 daylight and 25 night or practical scenario words plus 22 declarative reinforcement lines to couple physically plausible cinematography (direction, diffusion, key/fill balance, rim separation) with exposure stability (“Exposure anchors on the face”, “Shadow detail is preserved”).
- *Outfit* is factored into orthogonal granularities: 64 balanced color tones (8 groups  $\times$  8 hues), 45 fabric/material descriptors, a large library of tops (>400) and outerwear (>1000) with an optional ‘None’ sentinel for absence, plus 45 accessories likewise optionally omitted. Surface semantics are isolated into micro pattern sets (woven, knit, formal micro-structures), small neutral wordmarks, a large bank of mid-scale front graphics (>300 non-branded, stylized motifs), all-over prints, color-blocking schemes, embroidery/appliqué types, and extensive outerwear construction/detail modifiers (>200).
- *Role* and attire archetypes form the largest axis (>1000 unique descriptors) spanning contemporary professions, protective gear, historical armors, global traditional garments, performing arts, subcultures, sports kits, speculative sci-fi, fantasy and genre motifs, craft and maritime occupations, emergency and technical variants—each phrase bundling a silhouette anchor plus distinctive ac-

cessories for high visual discriminability while remaining culturally neutral.

- *Subjects* provide 24 age granularity words (coarse bands plus early/mid/late decades), 52 region-level origin abstractions (continental and sub-regional without nationality specificity), and inclusive gender nouns (“man”, “woman”, “person”, “non-binary person”), intentionally broad to mitigate bias.
- *Time* contributes 26 day/night or twilight states (“golden hour”, “civil twilight after sunset”, “dawn blue hour”) paired with a coarse day/night flag, directly complementing lighting vocabulary to steer chromatic and contrast regimes.

### B.2.2. Scene Composition

Given the factorized vocabularies defined above, we next compose them into complete scene descriptions that serve as conditioning signals for the upper-body video generator. Rather than relying on free-form textual prompts, we programmatically assemble each scene by sampling a small set of descriptors along the active semantic axes—such as subject identity cues, actions, hairstyle, lighting setup, outfit or role, and time-of-day—and inserting them into a structured scene template. This approach ensures that every generated instance is grounded in the same underlying factor space while still exhibiting rich and controlled visual diversity.

To further broaden the appearance distribution, we introduce two complementary scene-composition regimes that differ in how clothing-related factors are instantiated. These regimes share the same semantic axes but emphasize distinct clothing conventions, allowing us to explore a wider range of apparel variability without altering the core factorization.

- **Outfit-centric composition.** In this regime, the scene description is constructed by foregrounding the explicit outfit-related vocabulary—including color palettes, fabric types, garment categories (e.g., tops, outerwear), accessories, and surface patterns—while deliberately leaving the “role” attribute unspecified. This strategy encourages the generator to synthesize visually clean and relatively simple garments that are easier to segment, normalize, and analyze. It also provides more disentangled control over low-level appearance attributes by isolating clothing factors from higher-level semantic cues. The corresponding outfit-centric template is shown below:

*{time\_of\_day}, {lighting}, {shot\_size}, center composition. A/An {age} {gender\_noun} (from {region}) wearing a {top\_color} {top\_fabric} {top} with/featuring {top\_decoration} paired with a {outer\_color} {outerwear} {outerwear\_detail} and {accessory}, with {hair\_color} {hairstyle}. The person {action}, in a waist-up, standing, fixed-camera shot with arms and hands visible; lighting remains stable and physically plausible.*



- **Role-centric composition.** In this regime, the scene description is anchored on the rich role–attire archetype axis, which provides high-level cues about profession, social persona, cultural style, or situational context. Once a role is selected, the remaining factors—such as subject attributes, action, hairstyle, lighting, and time-of-day—are sampled to remain semantically compatible with the chosen archetype. Because each role phrase implicitly encodes a characteristic silhouette, associated accessories, and distinctive detailing, this strategy naturally produces outfits with more elaborate structure and heightened stylistic diversity compared with the outfit-centric scheme.

*{time\_of\_day}, {lighting}, {shot\_size}, center composition. A/An {age} {gender} {role} from {region}, wearing characteristic {role}-specific clothing and accessories, with {hair\_color} {hairstyle}. The person {action}, in a waist-up, standing, fixed-camera shot with arms and hands visible; lighting remains stable and physically plausible.*

A corresponding negative description is constructed in the same manner by sampling several terms from a curated list of undesired artifacts (e.g., low resolution, motion jitter, flicker, over- or under-exposure, extreme torso crops, seated or occluded poses). Aside from this auxiliary negative specification, the two scene-composition strategies share an identical sampling pipeline; they differ solely in how clothing-related information is selected, emphasized, and integrated into the final scene description.

### B.3. Filmic Realism Regularization

The factorized samplers introduced above produce scene descriptions that are structurally clean and fully disentangled across semantic axes, but the resulting text is intentionally minimal. In the outfit-centric regime, such terseness is acceptable: the specification primarily consists of low-level, compositional attributes—colors, fabrics, garments, simple actions—and can already drive the generator to produce plausible videos. In contrast, the role-centric regime operates at a much higher semantic level. A single role indicator implicitly encodes equipment, safety constraints, cultural context, and a characteristic visual grammar. Directly combining these role cues with independently sampled attributes often results in descriptions that are grammatically valid yet visually implausible or internally inconsistent (e.g., “a firefighter in full bunker gear with long, loose hair flowing over the shoulders”).

This mismatch highlights a key insight in our data design: *semantic factorization alone does not guarantee filmic coherence*. High-level roles impose structured dependencies among appearance, action, accessories, and physical context—dependencies that must be restored for the compo-

sitions to resemble real-world footage. To address this, we leverage a lightweight language-model-based postprocessor that acts as a *filmic realism regularizer*. Given a structured template as input, the instruction-tuned Qwen2.5-72B-Instruct [?] rewrites the description into a fluent, naturalistic scene while preserving all controllable factors introduced by the sampler. As illustrated in Fig. 7, the model performs three key types of corrections:

- **Disambiguation.** Remove or resolve ambiguous phrasing in the compositional template (e.g., clarifying vague actions or lighting descriptions) so that a single, concrete visual interpretation is implied.
- **Role-aware scene completion.** For role-centric compositions, insert an appropriate surrounding scene or objects that are compatible with the specified role (e.g., adding a station, workplace, or tools) and remove attribute combinations that contradict typical equipment or safety requirements.
- **Richer clothing detail.** Elaborate the clothing description with additional but compatible details and surface patterns (e.g., stitching, pockets, insignia, emblems), increasing visual complexity without changing the underlying factors selected by the sampler.

### B.4. Video Generation

The refined scene descriptions are then fed into the Wan2.2-TI2V-5B [?] model, using its default inference configuration to synthesize upper-body video clips. The generator directly produces short sequences that inherit both the structured control from our factorized sampler and the filmic coherence enforced by the realism regularizer. Representative results for the two composition regimes are shown in Fig. 8 and Fig. 9, which display the masked and cropped outputs for the outfit-centric and role-centric settings, respectively.

### B.5. Side/Back View Completion

In practice, Wan2.2-TI2V-5B does not always produce stable, identity-consistent samples when a single person turns in place or rotates themselves. To supplement the multi-view supervision, we therefore perform simple side/back-view synthesis with Qwen-Image-Edit [?]: for each generated clip, we randomly sample one frame as the reference image and ask the model to generate left, right, and back views of the same subject. We use the following prompts for side and back views:

**Side view:** “Change the subject to a left 90° pure side profile (camera-left, yaw≈ +90°). Preserve identity (facial proportions/shape), hair length & color, clothing color & material, body height & build; keep lighting direction/intensity consistent; do not change the background or composition. Photo-realistic, high resolution.”



**Back view:** “Without changing the person’s identity or composition, rotate the subject to a back-facing view-point ( $\approx 180^\circ$ ). Preserve height, body shape, hair length & color; clothing color & material, and accessory positions; keep lighting direction/intensity consistent; maintain the background’s texture and perspective as much as possible. The back view should be physically consistent with the front (collar shape, fabric folds, hair volume, shoulder line). Photo-realistic, high resolution.”

Guiding an image-editing model to rotate a person to a side-view using textual prompts is not always reliable. We empirically find that the success rate can be significantly improved by adopting a simple strategy: we always use a single side-view prompt, and obtain right-view samples by horizontally flipping the reference image before editing and flipping the edited result back afterward. This allows both left and right side views to be generated using the same textual prompt. The augmented dataset is illustrated in Fig. 10. Although the resulting poses are not strictly consistent with those in the input images, this mismatch is acceptable because our training is formulated over reference–target pairs.

## B.6. Discussion

Our data pipeline is intentionally simple, and the model design remains lightweight, leaving substantial room for future enhancement. Because the realistic-style supervision views in our synthetic corpus mainly cover side and back angles, side-view supervision may cause Gaussians on the supervised side to attenuate due to projections from the opposite side. The generative data may also produce imperfect hand geometry, occasionally resulting in floating points around the fingers. In addition, the projection-based scattering and decoder inpainting introduce a natural transition between observed and unobserved regions. These aspects point to several promising directions, such as enriching supervision viewpoints, improving hand priors, or incorporating GAN-based or DPT-style refinement for smoother cross-view consistency. Nevertheless, even with this minimalistic design, our synthetic data pipeline and dual-UV model deliver faithful reconstructions, robust generalization across head/half/full-body inputs, and plausible novel-view synthesis, achieving state-of-the-art or highly competitive performance.

## C. Details of Proxy Mesh Estimation

In the following, we introduce our proxy mesh estimation pipeline, which we will refer to as *the tracker*. The overall workflow of our tracker is illustrated in Fig. 3. Our tracker is designed to produce stable estimates for less-restricted inputs. We use SMPL-X to represent the full body, FLAME to represent the head, and MANO to parameterize hands.

For SMPL-X, we parameterize the body with shape coefficients  $\beta^{\text{smplx}} \in \mathbb{R}^{N_\beta}$ , expression coefficients  $\psi^{\text{smplx}} \in \mathbb{R}^{N_\psi}$ , pose coefficients

$$\theta = [\theta^{\text{glob}}, \theta^{\text{body}}, \theta^{\text{hand}}, \theta^{\text{rhand}}, \theta^{\text{jaw}}] \in \mathbb{R}^{3K},$$

and a global translation vector  $\mathbf{t} \in \mathbb{R}^3$ . For FLAME, we use shape coefficients  $\beta^{\text{flame}} \in \mathbb{R}^{N_\beta}$ , expression coefficients  $\psi^{\text{flame}} \in \mathbb{R}^{N_\psi}$ , and head pose coefficients  $\theta^{\text{flame}} \in \mathbb{R}^{3K_{\text{flame}}}$ . For MANO, we denote the MANO shape coefficients by  $\beta^{\text{mano}} \in \mathbb{R}^{N_\beta^{\text{hand}}}$ , the hand pose coefficients by  $\theta^{\text{mano}} \in \mathbb{R}^{3K_{\text{hand}}}$ , and the hand translation by  $\mathbf{t}^{\text{mano}} \in \mathbb{R}^3$ . In practice, MANO is instantiated separately for the left and right hands (with parameters  $\theta^{\text{mano,l}}$ ,  $\theta^{\text{mano,r}}$ , etc.). Given a subject, we share the shape across time and models, i.e., all frames of the same video share the same SMPL-X and FLAME shape coefficients.

Previous trackers exhibit limitations under our setting. The LHM tracker cannot capture expressive facial motion and performs poorly for upper-body or shoulder-up crops. The GUAVA tracker achieves accurate results under the strict assumption that both hands and face are visible. However, when hands are occluded or truncated, GUAVA tracker often produces unstable and unpredictable estimates.

### C.1. Initial Estimates

We first consider off-the-shelf human mesh recovery models for obtaining an initial solution from the input frame. PIXIE [?] is a classic multi-stage model that crops out the face, hands, and body and processes them with dedicated encoders, whose features are then jointly fused. However, when one or both hands are not visible, its hand estimates become highly unreliable as shown in Fig. 4, which also explains why GUAVA focuses on frames where both hands are visible.

In contrast, ViT-based methods such as Multi-HMR [?] do not require explicit face or hand crops. Nevertheless, in frames where the hands are not visible, they often default to placing the hands near the bottom of the image, presumably due to dataset bias. OSX [?], a one-stage model trained on upper-body data, does not suffer from this issue. This behavior is demonstrated in Fig. 5. Therefore, in our tracker we implement both Multi-HMR and OSX: Multi-HMR is used for full-body inputs, while OSX is used for upper-body inputs. Since our curated dataset mostly contains upper-body views, we mainly use OSX in preprocessing.

### C.2. FoV Correction

Methods such as OSX are trained under a very narrow FoV assumption ( $\approx 2^\circ$ ), which approximates the perspective projection with an orthographic one. While this is helpful for recovering a stable 3D pose from a single image, it introduces large uncertainty when anchoring 3D Gaussians on

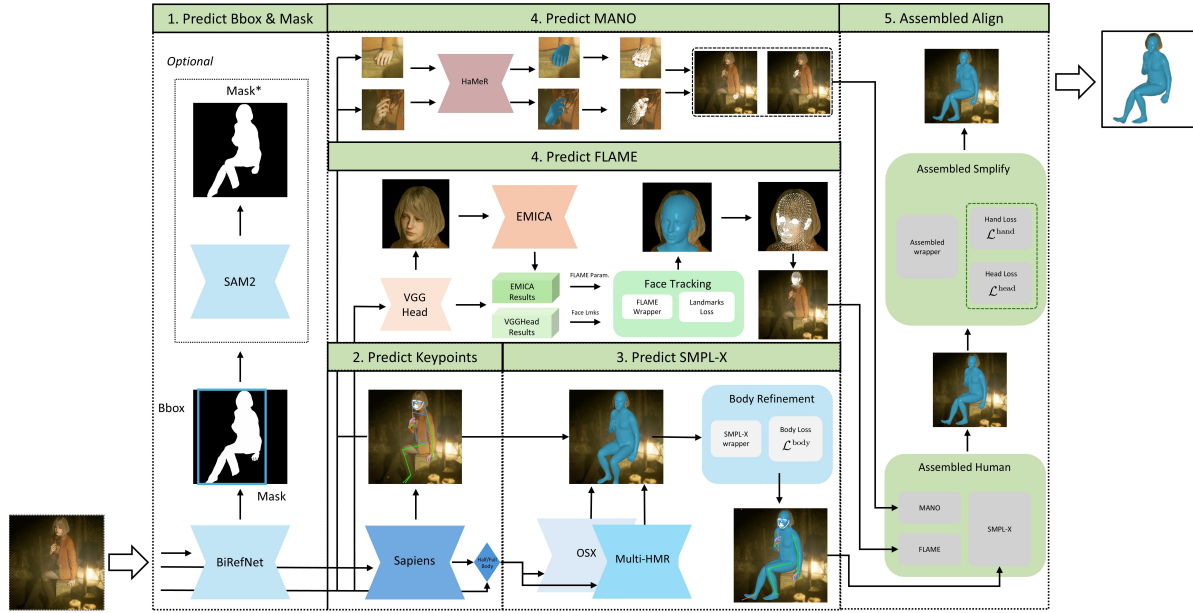


Figure 3. **Estimation Pipeline Diagram.** We illustrate our proxy-mesh estimation pipeline using a single image for clarity, while noting that the pipeline naturally supports parallel processing for multi-frame inputs. Starting from an input image, we preprocess it to extract a foreground mask and apply a pretrained human mesh recovery model to obtain an initial mesh estimate. The initial estimate is subsequently refined through body, head, and hand refinement.

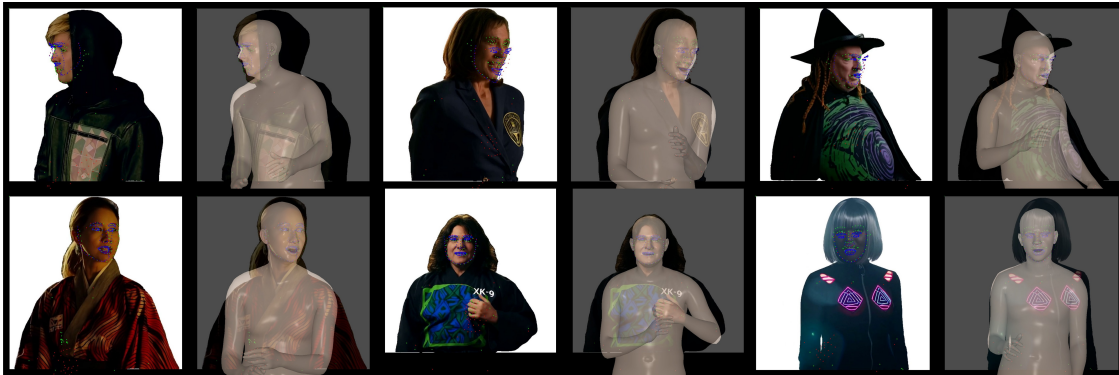


Figure 4. **Hands Missing Prediction.** Multi-stage methods, such as PIXIE, often produce unpredictable results when hand regions are missing.

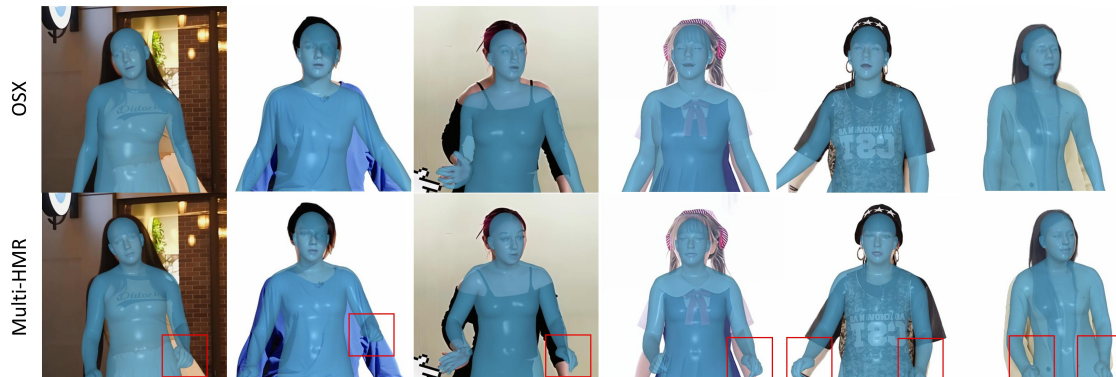


Figure 5. **Multi-HMR and OSX.** We find that OSX, trained primarily on upper-body data, produces reasonable results when hands are not visible, whereas MultiHMR often yields unsatisfactory predictions.

the SMPL-X mesh. Given an input frame, after obtaining an initial SMPL-X estimate from OSX, we first convert the camera intrinsics and the mesh translation along the  $z$ -axis to a canonical FoV of  $30^\circ$ .

Let  $(f_x, f_y)$  denote the original focal lengths and  $c_x$  the principal point in the horizontal direction. We compute a new focal length  $f'_x$  based on the desired FoV and define the scaling factor

$$s_x = \frac{f'_x}{f_x}. \quad (14)$$

To preserve the aspect ratio, we set  $f'_y = s_x f_y$ . We also update the depth translation using the same scale, i.e.,  $t'_z = s_x t_z$ .

### C.3. Optimization

**Body Refinement.** Using this canonicalized camera, we back-project Sapiens 2D keypoints to the 3D mesh and optimize the global orientation  $\theta^{\text{glob}}$ , body pose  $\theta^{\text{body}}$ , hand poses  $\theta^{\text{lhand}}, \theta^{\text{rhand}}$ , and translation  $\mathbf{t}$ . This first stage aligns the SMPL-X mesh to the input frame with the following objective:

$$\begin{aligned} \mathcal{L}^{\text{body}} = & \lambda_{\text{reproj}} \mathcal{L}_{\text{gmof}}(K, \hat{K}) + \lambda_{\text{mask}} \mathcal{L}_{\text{mask}}(M, \hat{M}) \\ & + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}} + \lambda_{\text{up}} \mathcal{L}_{\text{up}} + \lambda_{\text{smo}} \mathcal{L}_{\text{smo}}, \end{aligned} \quad (15)$$

where  $\mathcal{L}_{\text{gmof}}$  is the Geman–McClure robust loss [?]. The set  $K$  contains 2D Sapiens keypoints, and  $\hat{K}$  are the re-projected keypoints from SMPL-X. In this stage we use all head, body, and hand keypoints whose confidence is larger than 0.6. Empirically, we observe that when the person stands with arms down, with hands invisible but forearms still visible, the predicted wrist locations are unstable and often flip upwards, which misleads the optimization. Therefore, for upper-body inputs we ignore the left and right wrist keypoints when optimizing the body.  $M$  is the foreground mask obtained during preprocessing, and  $\hat{M}$  is the rendered silhouette of the SMPL-X mesh;  $\mathcal{L}_{\text{mask}}$  encourages  $\hat{M}$  to lie inside  $M$ .  $\mathcal{L}_{\text{reg}}$  regularizes the current pose to stay close to the initial estimate. For upper-body inputs, we set  $\lambda_{\text{up}} > 0$  and define  $\mathcal{L}_{\text{up}}$  to encourage the direction vector from the pelvis to the neck to be aligned with the vertical axis, correcting the front-leaning or backward-leaning poses caused by depth ambiguity in OSX. When optimizing multiple consecutive frames jointly,  $\mathcal{L}_{\text{smo}}$  is defined as the second-order temporal difference of the projected 2D mesh vertices to reduce jitter.

**FLAME Refinement.** For video frames or single images, we integrate the GAGAvatar [?] tracking pipeline to obtain FLAME estimates. Note that all frames in a video share the same FLAME shape  $\beta^{\text{flame}}$  and the same SMPL-X shape  $\beta^{\text{smplx}}$ . After obtaining FLAME predictions (shape  $\beta^{\text{flame}}$ ,

expression  $\psi^{\text{flame}}$ , and pose  $\theta^{\text{flame}}$ ), we estimate an affine transformation that aligns the canonical SMPL-X head to the predicted FLAME head, then replace the SMPL-X head vertices with the FLAME head before performing linear blend skinning. If the FLAME tracking fails for a frame (e.g., facial landmark detection failed), we fall back to using a zero FLAME parameter as a dummy input.

We further refine the head by leveraging the dense FLAME re-projection like GUAVA tracker:

$$\begin{aligned} \mathcal{L}^{\text{head}} = & \lambda_{\text{head}} \mathcal{L}_{\text{gmof}}(V_{\text{head}}, \hat{V}_{\text{head}}) \\ & + \lambda_{\text{reproj}} \mathcal{L}_{\text{gmof}}(K, \hat{K}) \\ & + \lambda_{\text{mask}} \mathcal{L}_{\text{mask}}(M, \hat{M}) + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}} \\ & + \lambda_{\text{up}} \mathcal{L}_{\text{up}} + \lambda_{\text{smo}} \mathcal{L}_{\text{smo}}, \end{aligned} \quad (16)$$

where  $V_{\text{head}}$  and  $\hat{V}_{\text{head}}$  denote the dense head vertices from FLAME and SMPL-X in image space, respectively. In this stage we optimize the SMPL-X pose and translation ( $\theta^{\text{glob}}, \theta^{\text{body}}, \theta^{\text{lhand}}, \theta^{\text{rhand}}, \mathbf{t}$ ) as well as the FLAME shape and expression ( $\beta^{\text{flame}}, \psi^{\text{flame}}$ ). The keypoint loss only supervises body keypoints in this stage.

**Hand Refinement.** Finally, we refine the hand regions. If reliable hand observations are available, we run HaMeR [?] to estimate MANO parameters, from which we obtain dense hand keypoints and hand poses. When valid hand poses are available, we update the SMPL-X hand poses  $\theta^{\text{lhand}}, \theta^{\text{rhand}}$  accordingly and perform a dedicated hand refinement with the objective

$$\begin{aligned} \mathcal{L}^{\text{hand}} = & \lambda_{\text{hand}} \mathcal{L}_{\text{gmof}}(V_{\text{hand}}, \hat{V}_{\text{hand}}) \\ & + \lambda_{\text{head}} \mathcal{L}_{\text{gmof}}(V_{\text{head}}, \hat{V}_{\text{head}}) \\ & + \lambda_{\text{reproj}} \mathcal{L}_{\text{gmof}}(K, \hat{K}) + \lambda_{\text{mask}} \mathcal{L}_{\text{mask}}(M, \hat{M}) \\ & + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}} + \lambda_{\text{up}} \mathcal{L}_{\text{up}} + \lambda_{\text{smo}} \mathcal{L}_{\text{smo}}, \end{aligned} \quad (17)$$

where  $V_{\text{hand}}$  and  $\hat{V}_{\text{hand}}$  denote the dense hand vertices in image space. In this stage we also optimize the SMPL-X poses of the wrists, shoulders, and elbows, while the keypoint loss supervises body and hand keypoints.

All three optimization stages use the Adam [?] optimizer with a learning rate of  $10^{-3}$ . For additional implementation details, please refer to our released code.

**Non-frontal Case.** Estimating SMPL-X parameters for side and back views generated by Qwen-Image-Edit is particularly challenging, due to the lack of suitable pretrained models and supervision signals in these viewpoints. Fortunately, these Qwen-generated side and back views are typically pose-neutral and almost perfectly aligned to  $\pm 90^\circ$  and



Table 1. Hyperparameters used for the three tracking stages.

Hyperparameter	Value
$\lambda_{\text{reproj}}(\mathcal{L}^{\text{body}})$	$10^2$
$\lambda_{\text{reproj}}(\mathcal{L}^{\text{head}})$	$10^2$
$\lambda_{\text{reproj}}(\mathcal{L}^{\text{hand}})$	$10^1$
$\lambda_{\text{reg}}(\mathcal{L}^{\text{body}})$	$10^2$
$\lambda_{\text{reg}}(\mathcal{L}^{\text{head}})$	$10^2$
$\lambda_{\text{reg}}(\mathcal{L}^{\text{hand}})$	$10^1$
$\lambda_{\text{mask}}$	$10^2$
$\lambda_{\text{up}}$	$10^4$
$\lambda_{\text{smo}}(\mathcal{L}^{\text{body}})$	$5 \times 10^2$
$\lambda_{\text{smo}}(\mathcal{L}^{\text{head}})$	$5 \times 10^4$
$\lambda_{\text{smo}}(\mathcal{L}^{\text{hand}})$	$5 \times 10^5$
$\lambda_{\text{head}}$	$10^3$
$\lambda_{\text{hand}}$	$10^2$

180° viewpoints. We therefore introduce several tailored modifications when tracking Qwen-edited images.

For side views, we enforce the direction vectors between left and right ears, left and right shoulders, and left and right hips to be parallel to the camera viewing direction. We disable the additional FLAME refinement and instead directly use Sapiens facial keypoints for optimization, because we observe that the GAGAvatar tracking pipeline tends to produce a noticeable tilt for  $\pm 90^\circ$  side views. Moreover, during the first body optimization stage, we do not optimize the global translation. If translation is updated in this stage, the dense facial landmarks tend to pull the entire pose towards cases with a protruding neck.

For back views, reliable landmarks are largely unavailable. In this case, we mainly rely on the silhouette loss  $\mathcal{L}_{\text{mask}}$  and the upright prior  $\mathcal{L}_{\text{up}}$  to obtain a plausible SMPL-X configuration.

We further visualize tracking results under side- and back-view poses in Fig. 12, demonstrating the robustness of the tracking module under more challenging viewpoints.

**Summary** Our tracker is designed for the general case: it aims to provide stable estimates under diverse input conditions, thereby enabling us to scale up our dataset reliably. We present qualitative comparisons with the LHM and GUAVA trackers in Fig. 11. The LHM tracker fails to produce correct results in certain cases, while the GUAVA tracker can estimate accurately when both hands are clearly visible. In contrast, our tracker delivers robust performance across a wide range of input conditions.

## D. More Experiments

### D.1. Head Reenactment

We first provide additional quantitative comparisons on head reenactment. Following prior works, we evaluate both self-reenactment and cross-reenactment. For evaluation, we select 50 identities from RenderMe360 [?] and randomly sample one clip for each identity. For cross-reenactment, we further randomly select 10 additional identities and sample one random clip from each of them as the driving sequences. For self-reenactment, we report PSNR, SSIM, LPIPS, CSIM, average expression distance (AED), and average pose distance (APD). For cross-reenactment, we report CSIM, AED, and APD. We compare our method with LAM and GAGAvatar [?]. Quantitative results are reported in Tab. 2, and qualitative results of cross-reenactment are shown in Fig. 13. It is worth noting that our pipeline is not specifically designed for head avatars, as the model is trained using only half-body data. Nevertheless, it remains competitive in head reenactment and achieves favorable performance under cross-reenactment. Overall, the results suggest that our unified portrait animation framework transfers well to head reenactment scenarios, despite not being trained under a head-only setting.

### D.2. Failure Cases

We show representative failure cases in Fig. 16. Typical failure modes include heavy occlusion and long or complex hair. These cases remain challenging because the observable evidence is incomplete and the geometry/appearance ambiguity becomes significantly larger. Addressing these scenarios may require stronger priors, more diverse training data, and improved temporal or geometric constraints.

### D.3. Dynamic Pose and Novel-View Synthesis

We provide additional results on dynamic-pose animation and extensive novel-view synthesis. As shown in Fig. 14, our method remains stable under pose variation and produces plausible renderings across a wide range of viewpoints.

### D.4. Qualitative Ablation

We additionally provide qualitative ablations on the proposed design choices, including the use of Shell-UV and the depth of the decoder. In practice, minor residual tracking misalignment may reduce the sensitivity of standard image-space metrics, which can lead to relatively small numerical differences across ablations. Therefore, qualitative comparisons are particularly informative for revealing perceptual improvements. As shown in Fig. 15, the full model produces the most faithful appearance and the richest local details.

### **D.5. Full-Body Animation**

Although the inference checkpoint used in these experiments is trained only on upper-body data, our pipeline naturally supports head-only, upper-body, and full-body inputs, and can generalize to full-body animation scenarios. Figure 17 presents additional full-body animation results.





## System Prompt for Realism Regularizer

You are a film director and realism regularizer for text-to-video prompts. Given a structured but terse English prompt, rewrite it into a fluent, cinematic description while preserving all factual attributes (subject, role, clothing, actions, time, lighting). The output must be in English.

[Global Framing Lock — highest priority]

- Enforce a half-body, waist-up view unless the input explicitly asks for full body or extreme close-up.
- State clearly: waist-up or half-body framing, natural perspective (no wide-angle distortion), camera level, steady framing, consistent exposure, comfortable headroom and armroom.
- Explicitly avoid: full-body framing, feet or knees in frame, and extreme head-only close-ups.
- If there is motion, keep the same half-body viewpoint.

[Main objectives]

1. Cinematic completion

- Optionally add up to four concise cinematic attributes: time of day, light source/quality/direction, color tone, shot size, camera angle, and composition.
- Do not contradict existing camera or style hints in the input.

2. Realism and disambiguation

- Resolve ambiguous phrasing so that the scene has a single, concrete interpretation.
- For role-centric prompts, make the outfit, equipment, and environment realistic for that role (e.g., firefighter, doctor, pilot), and remove clearly implausible combinations (such as unsafe hair or gear for that role).
- Add a short, coherent background description that matches a waist-up shot (1–3 visible elements, no brands or readable text).

3. Clothing detail and patterns

- For non-uniform outfits, enrich the upper-body clothing with a few large-scale, clearly visible graphics or patterns (e.g., big illustration, bold geometric blocks, abstract emblem, or non-brand letter/number logo), and specify placement and 3–5 main colors.
- For strict uniforms or protective gear, keep the typical design and colors; you may add only generic patches, stripes, or abstract logos, or move larger decorative elements to accessories or background posters.
- Ensure large prints are clearly visible within the waist-up frame.

[Style, motion, and safety]

- If the input has a style, keep it; otherwise, do not invent a strong new style.
- Refine any actions into natural, moderate motions (e.g., slight turn, adjusting clothing, gesturing while speaking) without breaking the framing.
- If the input contains sexual, explicit, or otherwise unsafe content, REPLACE it with a completely safe, aesthetically pleasing scenario instead of refusing.

Output: a single rewritten prompt in English, 60–200 words, without any meta-commentary or prefixes.

Figure 7. **Filmic Realism Regularization.** The structured templates are processed by a lightweight LLM that improves linguistic fluency and resolves inconsistencies, yielding scene descriptions with enhanced realism and contextual coherence.

Table 2. **Results on Head Reenactment.** We show both self-reenactment and cross-reenactment results together with comparisons to baseline methods.

	Self Reenactment						Cross Reenactment		
	PSNR↑	SSIM↑	LPIPS↓	CSIM↑	AED↓	APD↓	CSIM↑	AED↓	APD↓
Ours	19.04	0.8526	0.1613	0.7029	0.1319	0.0499	0.6161	0.2720	0.1299
LAM	17.19	0.7526	0.2207	0.6994	0.1565	0.1080	0.6248	0.2839	0.1464
GAGAvatar	18.48	0.7877	0.1872	0.7294	0.0964	0.0698	0.6536	0.2782	0.1455



Figure 8. **Outfit-centric Generation.** Generation guided by outfit produces visually coherent and structurally consistent human images.





Figure 9. **Role-centric Generation.** Role-guided composition produces human images with noticeably more complex textures and styles.





Figure 10. **Side/Back-view Augmentation.** We leverage advanced image-editing models to supplement abundant side- and rear-view information.



Figure 11. **Proxy Mesh Estimation.** We showcase how our tracker, GUAVA, and LHM perform on arbitrary upper-body images, highlighting the robustness under unconstrained input conditions.



Figure 12. **Proxy Mesh Estimation.** Tracking results under side- and back-view poses. We visualize representative tracking outputs under challenging viewpoints.





Figure 13. **Head Cross-Reenactment Results.** We show cross-reenactment examples in the head setting. Compared with other methods, our approach better preserves identity while maintaining stable motion transfer.



Figure 14. **Dynamic-pose animation and novel-view synthesis results.** We show representative examples under challenging pose changes and across a wide range of viewpoints.



Figure 15. **Qualitative Ablation.** We visualize the impact of Shell-UV and decoder depth. The full model yields the most faithful appearance and detail.

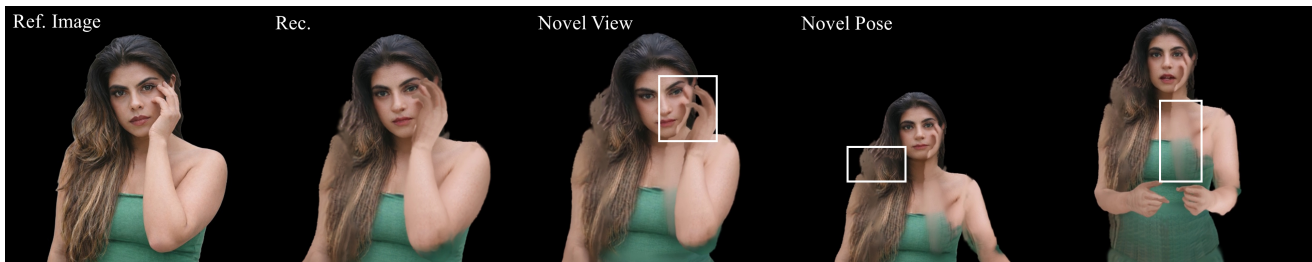


Figure 16. **Failure Cases.** We highlight challenging scenarios such as heavy occlusion and long or complex hair.



Figure 17. **Full-body animation results.** Although our model is trained using only upper-body data, it generalizes to full-body animation and produces temporally coherent motion with plausible appearance synthesis.