Head Anything: Correspondence-Guided Joint Generation of 3D Heads from Few Images

Umut Kocasarı[1] Bartłomiej Baranowski[1] Richard Shaw[2]
Eduardo Pérez-Pellitero[2] Matthias Nießner[1]
[1]Technical University of Munich, Germany [2]Huawei Noah's Ark Lab, London, UK



Figure 1. **Head Anything.** From a few in-the-wild video frames of a person, our method jointly generates one detailed mesh per frame, shown below it. Each mesh follows its own image while benefiting from the other views and expressions. Website: https://kocasariumut.github.io/HeadAnything/

Abstract

Large image-to-3D generators recover detailed heads from a single photograph but treat every image as a new object. For several images of one person, they hallucinate unseen regions independently without using information from the other images. We present Head Anything, which generates detailed head meshes from a few images with changing expressions and viewpoints. Instead of training a multi-image prior, we turn a pretrained single-image generator with a native 3D latent space into a joint generator that denoises all images together. Since global attention between images is hard to learn from scarce 3D head data, lightweight correspondence-attention adapters let every 3D token attend only to its corresponding points in the other images. We obtain correspondences by fitting a neural parametric head model to the generator's own single-image outputs, requiring no ground truth at test time. Training on both ground-truth and fitted correspondences makes the adapters robust to fitting errors. Each mesh stays pixel-aligned with its own image and borrows the unseen regions from all others. On head scans, Head Anything lowers the head Chamfer distance of the best image-to-3D generator by 32% and its cross-frame inconsistency by 47%, and its meshes improve face tracking. Code and models will be released.

1. Introduction

Telepresence, games, film production and virtual try-on need the 3D geometry of a person's head, including the hair, the ears and the back of the head, at a level of detail that today requires a multi-view capture studio [17, 26]. Most im-

ages of people, however, come from casual videos or photo collections, such as those in Fig. 1, and depict the head from varying viewpoints and with different expressions. Recovering complete, detailed and mutually consistent meshes from such images would make studio-quality head geometry available from everyday data, with every mesh following its own image and benefiting from what the others reveal.

This problem is underconstrained. Each image shows only part of the head, and since expression and pose change, multi-view reconstruction does not apply. A strong learned prior is indispensable, but high-quality head scans exist for only a few hundred people [11, 39], far fewer than the hundreds of thousands of objects used to train general image-to-3D generators [8]. A prior over multiple images is even harder to learn, since the same point on the head appears at a different position in each image as expressions and poses change, and global attention across images cannot reliably learn these correspondences from so few identities.

Existing methods work around this lack of 3D data with different trade-offs. Parametric head models [3, 11, 21] and their trackers [12, 13, 28, 49] naturally combine many frames, but their geometry is coarse and misses hair and fine detail. Feed-forward methods predict per-pixel geometry and correspondences [19], but only for the visible surface. Large image-to-3D generators [20, 22, 36, 41–43] produce complete, detailed meshes, but from one image at a time, and their multi-image variants assume a static object. Video and multi-view diffusion models share information along optical flow [7, 10], epipolar lines [15, 37] or a morphable head model [6, 35], but only between 2D images.

We introduce Head Anything for multi-image 3D head generation by adapting a pretrained single-image generator with a native 3D latent space to jointly generate 3D heads from multiple images of the same person. As Fig. 2 shows, it jointly denoises all 3D latents using lightweight adapters letting each 3D token attend only to corresponding points in the other images. Unlike global attention, correspondence attention knows where to look from the first training step. The correspondences are deformable and bootstrapped from the generator itself by fitting a neural parametric head model [11, 12] to its single-image outputs. This eliminates ground-truth requirements at test time, while training on both ground-truth and fitted correspondences makes the adapters robust to fitting errors. Routing attention through these correspondences within a 3D latent generator distinguishes our method from image-space correspondence attention and from simply combining a generator with a parametric fit. Experiments show our meshes are more accurate than all baselines, far more consistent across frames than image-to-3D generators' meshes, and improve a parametric face tracker's performance. Joint generation yields greater accuracy and consistency than frame-by-frame generation using our backbone, Pixal3D [20], fine-tuned on head scans.

In summary, our contributions are:

- We introduce a joint multi-image head generator that couples the sampling processes of a pretrained single-image 3D generator, so that every mesh follows its own image and benefits from all others.
- We propose a novel correspondence attention mechanism in which every 3D token attends to its corresponding points in the other images, given by a neural parametric head model fitted to the generator's own outputs and made robust by mixed ground-truth and fitted correspondences.
- We demonstrate that joint generation produces more accurate 3D heads than all baselines, far more consistent unseen regions than image-to-3D generators, and meshes that improve parametric face tracking.

2. Related Work

Single-image 3D generation. 3D generators trained on object collections [8] either lift novel-view and multi-view diffusion models to 3D [24, 33] or generate directly in native 3D latent spaces. TRELLIS [42] denoises a sparse voxel structure and then structured latents on the occupied voxels. TRELLIS.2 [43] makes these latents compact at high resolution. TripoSG [22] and Hunyuan3D-2.1 [36] use rectified flows [23, 25] over vector-set latents, and Direct3D-S2 [41] scales sparse attention to high-resolution volumes. Pixal3D [20] back-projects image features into the voxel grid, so that the generated geometry is aligned with the input pixels and preserves fine details. All of these methods generate one object from one image, and their multi-image modes assume one rigid object.

3D head reconstruction. Morphable models [3] such as FLAME [21] form the basis of most monocular face reconstruction and tracking methods [9, 13, 28, 49]. They provide a consistent topology, but their geometry is coarse and covers neither hair nor accessories. NPHM [11] models identity and expression of complete heads with neural fields learned from scans, and MonoNPHM [12] adds appearance and fits it to monocular videos, with detail bounded by its latent space. Feed-forward methods regress per-frame geometry and correspondences from images [19], but only for the visible surface. Capture setups [17, 26] and the avatars built on them [29] need calibrated camera rigs. We use a parametric head model only as a source of dense correspondences, while the geometry comes from the generator.

Correspondence-guided multi-image generation. Video diffusion models attend between frames at the same spatial location [4, 40], while video editing methods follow correspondences. FLATTEN restricts attention to patches on the same optical-flow path [7], and TokenFlow propagates diffusion features through nearest-neighbour correspondences [10]. Multi-view diffusion models attend along epipolar lines [15, 37], and for heads, Morphable Diffu-

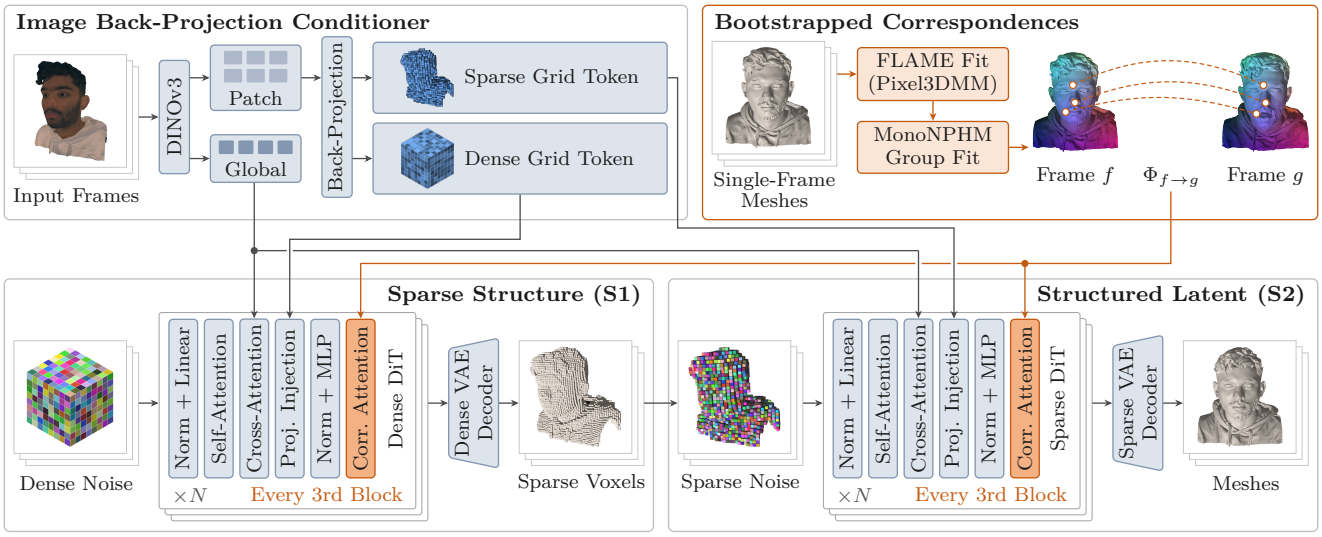

Figure 2. **Architecture overview.** Each frame conditions the fine-tuned generator Pixal3D-FT through DINOv3 features back-projected onto the dense (S1) and sparse (S2) grid tokens. All n frames are denoised jointly, and correspondence-attention adapters couple them at the corresponding points $\Phi_{f\to g}$ of a FLAME and MonoNPHM fit to the single-frame meshes. The details of correspondence attention and how it is applied in each stage are shown in Fig. 3.

sion [6] and CAP4D [35] condition multi-view diffusion on a morphable model and reconstruct an avatar from the generated views. 4D methods reconstruct dynamic objects from video [14, 30, 45] or animate one generated mesh [5], and training-free approaches average several diffusion processes where they overlap [1, 16]. All correspondence-guided methods above exchange information between 2D images, with correspondences from optical flow, camera geometry or a rendered morphable model, and obtain 3D only through a separate reconstruction. We instead exchange information between sparse tokens within a native 3D latent generator. Deformable correspondences bootstrapped from the generator's single-image outputs connect tokens across different head poses and expressions. Each frame retains its own detailed mesh. As in ControlNet [46], the adapters are zero-initialised modules on a pretrained backbone, which keeps its prior intact at the start of training.

3. Method

Given n images $\{I_f\}_{f=1}^{n}$ of a person's head, we generate one detailed mesh M_f per image, pixel-aligned with I_f and completed from $I_{g\neq f}$ in regions unseen in I_f. We fine-tune a single-image generator on head scans (Secs. 3.1 and 3.2) and couple n of its sampling processes (Sec. 3.3) through correspondence attention (Sec. 3.4), routed by correspondences bootstrapped from its own outputs (Sec. 3.5).

3.1. Preliminaries

We build on Pixal3D [20], which generates a mesh in two stages. A *sparse-structure* flow (S1) denoises a dense $16^3\times8$ latent, which a dense VAE decoder maps to occupied voxels on a 64^3 grid. A *structured-latent* flow (S2) then denoises latents at these voxels at 512^3 resolution, followed by cascaded refinement to 1024^3 and mesh extraction with a sparse VAE decoder. Both flows use diffusion transformers [27] with N=30 blocks. They receive DINOv3 [34] features through cross-attention and projection injection, which back-projects patch features onto grid tokens to align the output with the input pixels. Each flow is trained with the rectified-flow objective [23, 25]

$$\mathcal{L}(\theta) = \mathbb{E}_{t,\,x_0,\,\epsilon}\left\|v_\theta(x_t, t, c) - (\epsilon - x_0)\right\|^2, \qquad (1)$$

where $x_t = (1-t)\,x_0 + t\,\epsilon$ interpolates a clean latent x_0 and Gaussian noise ϵ at flow time $t \in [0,1]$, c is the image condition and v_θ the predicted velocity.

3.2. A Single-Image Head Prior

Object-level generators miss the specific shape of a face and often fail on hair, so we fine-tune Pixal3D on photorealistic renders of the NPHM head scans [11]. The resulting Pixal3D-FT is a strong single-image head prior, but it generates every frame on its own, so the frames do not benefit from each other. It also hallucinates the back of the head and the hair independently for every frame.

3.3. Joint Generation

Instead of n independent samples, our method draws the latents of all frames as one sample from their joint distribution given all images $\{I_f\}_{f=1}^{n}$. Each frame keeps its own noise, condition c_f, tokens and, in S2, occupied voxels, so that every mesh stays aligned with its own image. The Pixal3D sampler advances all frames in lockstep. The transformer

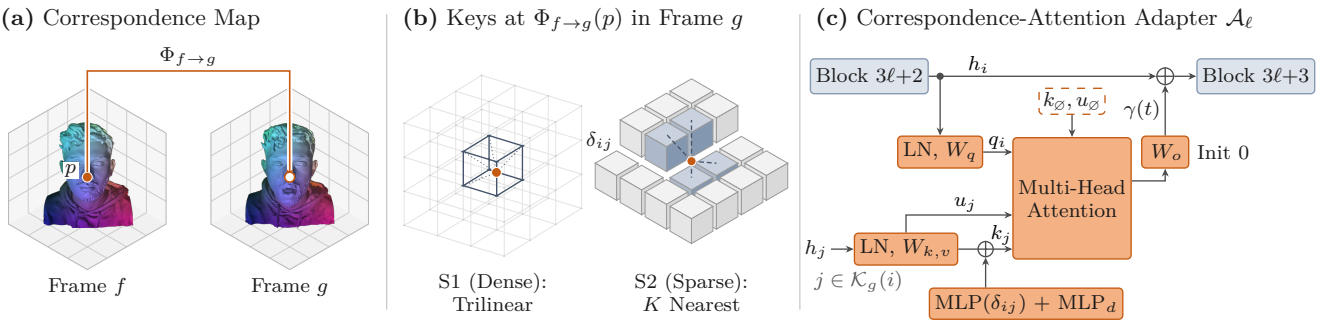

Figure 3. **Correspondence attention.** (a) $\Phi_{f\to g}$ maps a token position p of frame f to its corresponding point in frame g. (b) Keys are sampled trilinearly in S1 and taken from the K nearest occupied tokens in S2. (c) The adapter attends to these keys alongside a learned null key-value pair, then adds its gated output to the residual stream.

blocks process each frame independently. After block 3ℓ+2 in both flows, with $\ell = 0, \dots, 9$, an adapter $\mathcal{A}_\ell$ updates the tokens H_f of each frame f using the tokens H_g of the other frames $g \neq f$,

$$H_f \leftarrow H_f + \gamma(t)\, W_o\, \mathrm{Attn}_\ell\big(H_f, \{H_g\}_{g\neq f}, \Phi\big), \qquad (2)$$

where Attn_ℓ is the correspondence attention of the adapter, Φ the correspondences between the frames, W_o a zero-initialised output projection and $\gamma(t)$ a learned gate on the timestep embedding. At initialisation, the joint generator thus equals the single-image generator. Since both stages are joint, the occupied voxels of S1 already benefit from the other frames. The backbone is fine-tuned together with the adapters at a lower learning rate, so that it can adapt to the incoming information without forgetting its prior.

3.4. Correspondence Attention

Different frames observe different regions of the head, so the tokens of a region that one image does not show can take their shape from the frames that do. Since the head turns and the face deforms between frames, the same voxel holds different points of the head in different frames, and a token can only profit from another frame if it knows where its corresponding point lies. Global attention to all tokens of other frames would have to learn from a few hundred identities where to attend among tens of thousands of tokens, which it cannot do reliably (Sec. 4.4). We thus restrict attention to corresponding positions, as Fig. 3 illustrates, so the adapters only learn what to take from each frame.

Correspondence map. Let A_f map points from the parametric head model's coordinate system into the latent grid of image frame f. Let $\varphi_{f\to g}$ map a head point in frame f to its corresponding point in frame g, with both points expressed in head-model coordinates. A grid point p of frame f is mapped into frame g by

$$\Phi_{f\to g}(p) = A_g\, \varphi_{f\to g}\big(A_f^{-1} p\big). \qquad (3)$$

For training groups with ground-truth correspondences, A_f is known from the scan and $\varphi_{f\to g}$ comes from the scan registrations $r_f \in \mathbb{R}^{V\times 3}$, whose vertices with the same index v are corresponding points,

$$\varphi_{f\to g}(y) = y + \textstyle\sum_{v\in\mathcal{N}(y)} w_v\,(r_{g,v} - r_{f,v}), \qquad (4)$$

where $\mathcal{N}(y)$ are the nearest vertices of y and w_v normalised inverse-distance weights, so points off the surface, such as hair and clothing, follow the nearest surface and the map covers the whole head. At test time, A_f and $\varphi_{f\to g}$ come from a parametric model fitted to the generator's outputs (Sec. 3.5). We use NPHM [11] rather than FLAME [21], since it gives accurate correspondences on the complete head, including the scalp, hair and neck that the other frames have to complete.

Keys at corresponding positions. Token i of frame f at position p_i forms the query $q_i = W_q\,\mathrm{LN}(h_i)$, with layer normalisation LN, and gathers a key set $\mathcal{K}_g(i)$ at $\Phi_{f\to g}(p_i)$ in every other frame g. The S1 latent is a dense 16^3 grid, so its keys and values are sampled trilinearly at $\Phi_{f\to g}(p_i)$, with an embedding of whether the correspondence is valid and the depth features defined below. The S2 tokens are sparse, so we take the K=4 nearest occupied tokens j of frame g at p_j, with values $u_j = W_v\,\mathrm{LN}(h_j)$ and keys that embed the offset $\delta_{ij} = p_j - \Phi_{f\to g}(p_i)$ and two depths,

$$k_j = W_k\,\mathrm{LN}(h_j) + \mathrm{MLP}(\delta_{ij}) + \mathrm{MLP}_d(d_i, d_j), \qquad (5)$$

where d_i and d_j are the coordinates of the query and the key along the viewing axes of their frames' latent grids. These depth features indicate on which side of the head a point lies as seen from each frame, so the adapter can learn which frames to rely on. A learned null key-value pair $(k_\varnothing, u_\varnothing)$ completes the set, allowing tokens to ignore occluded or incorrect correspondences; those outside the grid or without a valid fit are masked. Multi-head attention over these $(n{-}1)K{+}1$ keys gives Attn_ℓ, which connects corresponding points from the first training step. With T tokens per frame and $K \ll T$, a token attends to $(n{-}1)K{+}1$ keys instead of all $(n{-}1)T$ tokens, so the total cost of $\mathcal{O}(n(n{-}1)KT)$ is a factor T/K below global attention.

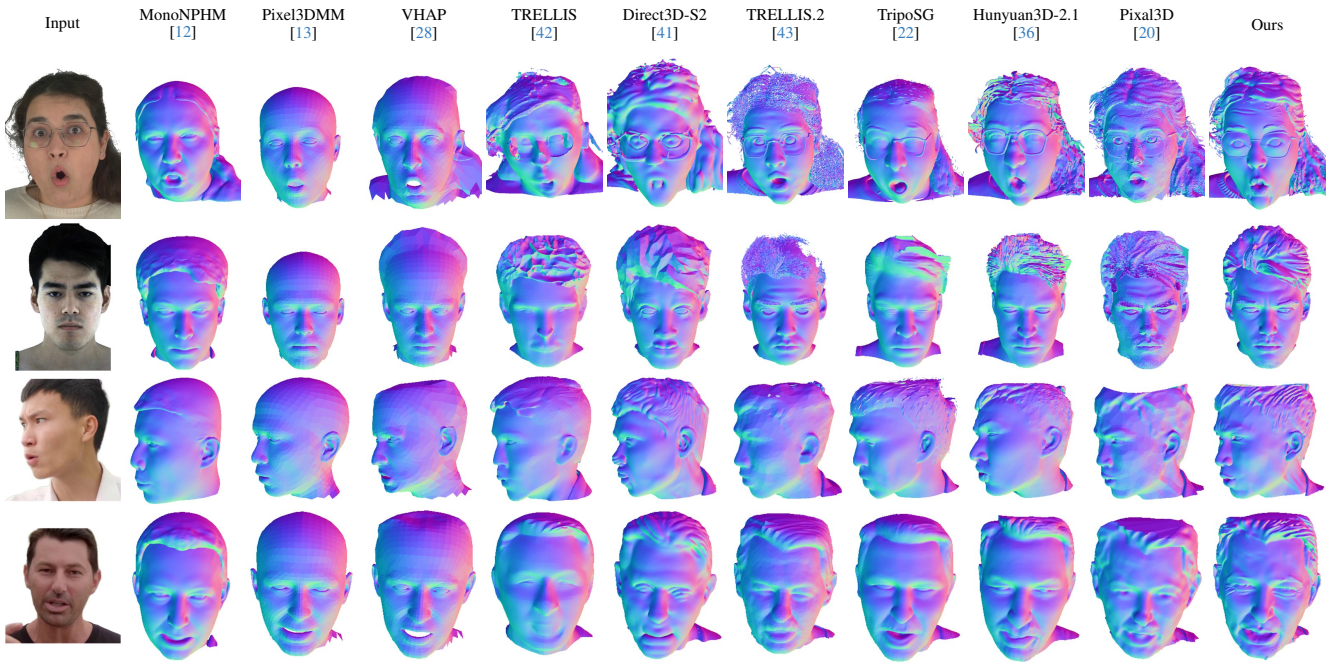

Figure 4. **Mesh generation comparison on NeRSemble, AVA-256, VFHQ and CelebV-HQ.** Our meshes, rendered with surface normals in the input camera, have a better surface quality than those of the baselines.

3.5. Bootstrapped Correspondences

At test time, no scan is available. As shown in Fig. 2 (top right), we bootstrap correspondences by fitting the head model to Pixal3D-FT's per-frame outputs, whose detailed, pixel-aligned meshes provide better targets than pixels.

FLAME fitting. Fitting a parametric model to a generated mesh requires a good initial scale and pose. Since the mesh lacks template topology, we obtain dense FLAME [21] correspondences from the image. Pixel3DMM [13] fits FLAME to each image. We then project the rasterised template coordinates at face pixels onto the single-frame mesh, pairing mesh points with template coordinates. We jointly fit a similarity (s_f, R_f, t_f) and the FLAME shape, expression, and jaw parameters to these pairs using a robust point loss. The similarity places every mesh in a canonical head frame with a consistent scale and serves as A_f. For real images and our FaceVerse renders, A_f combines per-frame head pose and scale, estimated from Face Anything [19], with a single FLAME fit to the first frame. This ensures consistent placement across all frames.

Group fitting. We then fit one MonoNPHM [12] to all meshes of the group, with a shared identity code z^{id}, per-frame expression codes z_f^{ex} and per-frame similarities S_f initialised from the FLAME fit,

$$\min_{z^{\mathrm{id}},\{z_f^{\mathrm{ex}},S_f\}} \sum_f \frac{1}{|P_f|} \sum_{y\in P_f} \left|\mathrm{SDF}\big(S_f(y);\, z^{\mathrm{id}}, z_f^{\mathrm{ex}}\big)\right| + \lambda\,\mathcal{R}\big(z^{\mathrm{id}}, \{z_f^{\mathrm{ex}}\}\big), \tag{6}$$

where SDF is the signed distance of MonoNPHM, P_f are points sampled on the mesh of frame f, and $\mathcal{R}$, weighted by λ, regularises the codes towards the prior. The shared identity lets the fit of every frame use the evidence of all frames. MonoNPHM models each expression as a backward deformation into a shared canonical space, so $\varphi_{f\to g}$ of Eq. (3) maps a point of frame f into the canonical space and from there into frame g. The second step requires inverting frame g's deformation, whose inverse has no closed form. We approximate it with eight fixed-point iterations initialised at the canonical point. The residual exceeds 1 mm for only 0.06% of points; see the appendix for details.

Training with fitted correspondences. A training group consists of scans of one identity with different expressions, each rendered from its own random viewpoint, and Eq. (1) is summed over the group. The two instances of Eq. (3) differ in kind, since registration maps are exact on the scan surface and interpolate elsewhere, whereas fitted maps follow the learned deformation field of MonoNPHM everywhere but inherit the errors of the fit. Adapters trained on registrations alone therefore never see the kind of correspondences that they receive at test time. We run this procedure once to precompute fitted correspondences for a fixed pool of training groups. During training, half the groups come from this pool, while the other half use registration correspondences. Fitted correspondences expose adapters to fitting errors, while registration groups provide accurate supervision. Changing group size from 2 to 8 scans lets one model handle varying numbers of images. At test time,

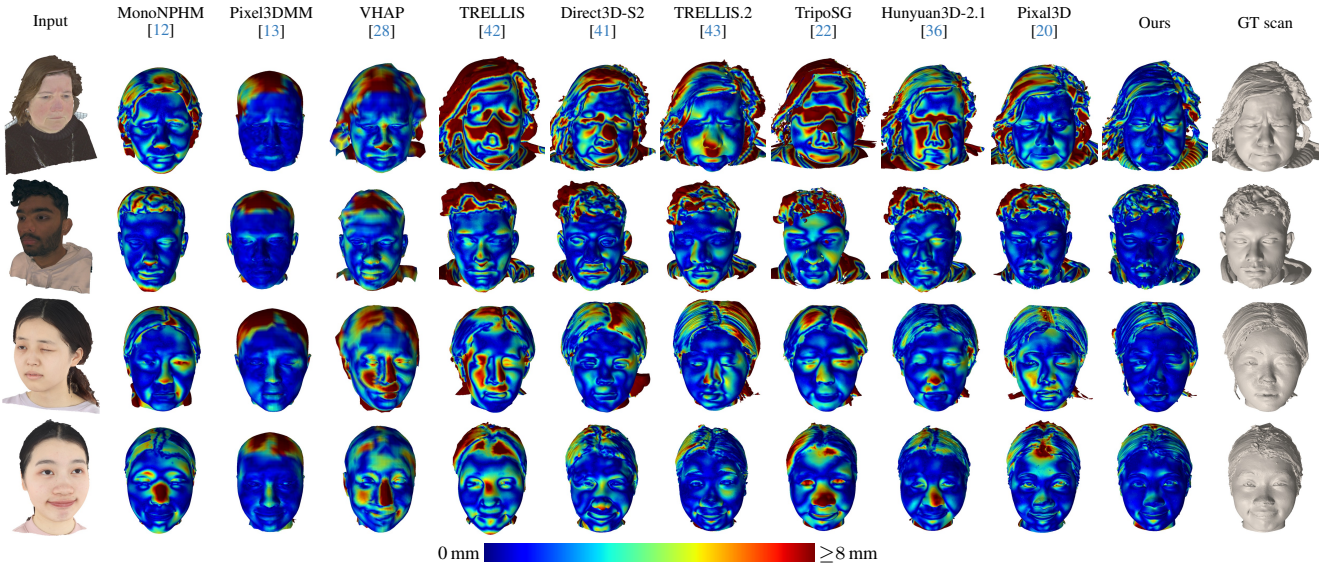

Figure 5. **Mesh generation error comparison on NPHM and FaceVerse.** Colours show the distance to the GT scan.

Table 1. **Mesh generation evaluation on NPHM and FaceVerse.** Head-region metrics and cross-frame consistency (Cons.), best in **bold**, second best underlined. Parametric models share a fixed template and are consistent by construction, so their consistency is shown in gray and not ranked.

	NPHM					FaceVerse				
Method	Chamfer↓	Compl.↓	EMD↓	F-score↑	Cons.↓	Chamfer↓	Compl.↓	EMD↓	F-score↑	Cons.↓
MonoNPHM [12]	8.73	12.57	27.4	33.0	4.91	7.51	9.35	20.9	29.3	5.94
Pixel3DMM [13]	8.41	12.11	29.4	38.8	2.07	9.24	11.81	27.5	28.3	5.66
VHAP [28]	6.32	8.13	24.0	31.8	4.93	7.95	9.92	22.9	25.2	10.36
TRELLIS [42]	6.26	6.10	19.8	31.1	12.64	7.48	7.68	21.9	29.4	12.36
Direct3D-S2 [41]	6.17	6.03	21.0	31.8	11.71	7.65	6.62	24.3	29.4	12.07
TRELLIS.2 [43]	5.99	5.98	19.3	34.1	12.37	5.96	6.57	17.1	35.7	11.31
TripoSG [22]	5.95	5.84	20.6	31.6	11.69	7.42	7.71	22.4	29.5	11.35
Hunyuan3D-2.1 [36]	5.18	5.41	16.7	33.7	9.74	6.72	7.53	18.2	30.8	12.38
Pixal3D [20]	4.83	5.32	17.3	38.0	9.48	6.15	6.71	17.9	32.9	11.09
Pixal3D-FT	3.86	3.77	13.3	45.6	6.73	5.14	5.35	17.6	39.4	9.58
Ours	**3.27**	**3.23**	**11.8**	**50.9**	**5.03**	**4.65**	**5.07**	**15.9**	**43.1**	**7.87**

our method first generates per-frame meshes for fitting, then jointly generates all meshes of a person. A modified version of the standard clean-up avoids introducing holes.

4. Experiments

4.1. Implementation Details

For Pixal3D-FT, we fine-tune Pixal3D on 10,180 scans of the 477 NPHM identities outside our test set, each rendered from two viewpoints. S1, S2 and the cascade are trained for 8k, 8k and 12k steps, respectively, with learning rates of 10^{-5}, 10^{-5} and $5{\cdot}10^{-6}$ and batch sizes of 16, 16 and 4. For joint generation, the adapters follow every third block of both flows, with width 512, 8 heads and $K{=}4$ keys per frame in S2, which amounts to 55 M parameters per stage. We train them with the backbone on groups of 2 to 8 scans from the 459 of these 477 identities with at least four scans. Each stage is trained for 4k steps with two groups per step in S1 and one in S2 and learning rates of 10^{-4} for the adapters and 10^{-5} for the backbone. Both trainings use AdamW, 200 warm-up steps and a condition dropout of 0.1 for classifier-free guidance. On NVIDIA A100 GPUs, the fine-tuning uses two GPUs per stage, and the adapter training takes about 7 h for S1 and 6 h for S2 on one GPU. For eight images, our method takes about 2.3 times as long as generating them one by one with Pixal3D-FT.

4.2. Evaluation Protocol

Scan benchmarks. We evaluate on 10 held-out NPHM [11] identities, each with eight expressions. Each scan is rendered photorealistically from a random viewpoint. We also evaluate on 10 FaceVerse [39] identities, each with eight scans. These scans differ from our training data. We render them from varied viewpoints and

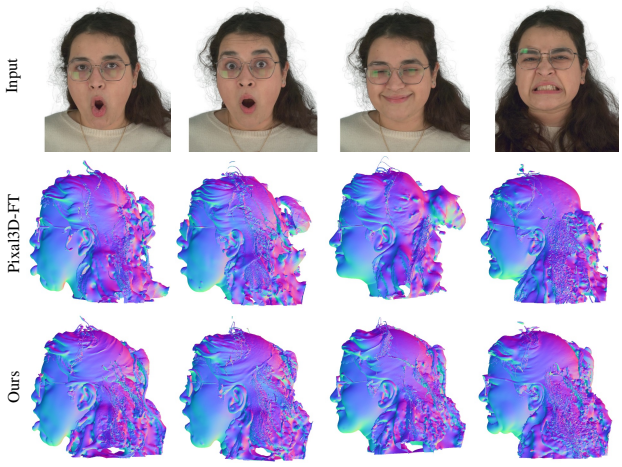

Figure 6. **Back-of-head consistency on NeRSemble.** Side views of the meshes generated from four input frames. Pixal3D-FT hallucinates a different hairstyle for every frame, while ours agree.

process them like real images.

Real images. We evaluate on 10 NeRSemble [17] identities outside the NPHM training and test sets and 10 AVA-256 [26] identities. Each person's eight input images show four expressions captured by two synchronised cameras, one on each side of the face. The remaining cameras provide held-out views and multi-view stereo points from COLMAP [31, 32]. Video frames from VFHQ [44] and CelebV-HQ [48] serve for qualitative results, and all methods receive the same RGBA crops.

Baselines. We compare against the parametric trackers VHAP [28] and Pixel3DMM [13], which fit a shared identity across a person's images. We also evaluate MonoNPHM [12] and six image-to-3D generators [20, 22, 36, 41–43] on each image independently. Finally, Pixal3D-FT uses our fine-tuned backbone to generate a mesh independently for each image. Our method and Pixal3D-FT use only the input camera, as described in Sec. 3.5. Ground-truth geometry and all other cameras are used only for evaluation.

Metrics. Every prediction, ours included, is aligned to its scan with the same similarity search [2, 38] and a final rigid face alignment, so that the metrics compare shapes rather than placements. In the head region above a plane 2 cm below the chin, we report the Chamfer distance, the completeness, the Earth Mover's distance and the F-score at 2 mm [18]. We warp each aligned prediction's non-face surface into every other frame using ground-truth deformation. Cross-frame consistency measures the mean distance to the target predictions. For the captures, we report silhouette IoU in held-out cameras after a similarity fit using the alignment cameras. We also report the Chamfer distance between the camera-visible head and stereo points. We measure consistency by the head Chamfer distance between meshes generated from synchronised images of the same instant. All distances are in mm and F-scores in %.

Table 2. **Mesh generation evaluation on NeRSemble and AVA-256.** Inputs are real images from two synchronised cameras. Held-out IoU, Chamfer (Ch.) and cross-view consistency (Cons.); gray values are not ranked.

	NeRSemble			AVA-256		
Method	IoU↑	Ch.↓	Cons.↓	IoU↑	Ch.↓	Cons.↓
MonoNPHM [12]	0.902	5.07	4.03	0.869	4.74	3.93
Pixel3DMM [13]	0.908	4.50	1.36	0.884	4.80	1.77
VHAP [28]	0.932	4.20	2.70	0.882	4.31	3.22
TRELLIS [42]	0.923	3.93	5.70	0.900	3.92	5.49
Direct3D-S2 [41]	0.931	4.80	7.86	0.911	4.49	7.04
TRELLIS.2 [43]	0.922	4.06	6.09	0.870	4.23	6.76
TripoSG [22]	0.936	3.88	5.31	0.902	3.96	5.30
Hunyuan3D-2.1 [36]	0.923	4.65	8.35	0.899	3.96	5.50
Pixal3D [20]	0.931	3.88	5.36	0.900	3.46	4.80
Pixal3D-FT	0.942	3.29	4.37	0.905	3.36	4.22
Ours	**0.953**	**3.10**	**3.30**	**0.915**	**3.13**	**3.12**

4.3. Comparison with the State of the Art

Qualitative comparison. Fig. 4 compares meshes for frames of NeRSemble, AVA-256, VFHQ and CelebV-HQ. The parametric models recover plausible faces but represent hair as a smooth volume at best, while image-to-3D generators often fail to capture specific facial shape and produce noisy hair geometry. Our meshes have a higher surface quality and more detail, both in facial features such as the eyes, lips and wrinkles and in the structure of the hair.

Scan benchmarks. Tab. 1 evaluates all methods against the ground-truth scans. Our method is the most accurate on both datasets, including FaceVerse, whose scans and inputs differ from our training data, whereas the parametric models produce coarse heads without hair. Their shared template makes them consistent by construction, which is why we do not rank them in consistency. Fine-tuning on head scans already makes Pixal3D-FT the strongest generator, and joint generation improves over it in accuracy and clearly in cross-frame consistency, although both share one backbone. The error maps in Fig. 5 confirm that our meshes are closer to the scans across the whole head.

Real camera images. As Tab. 2 shows, our method outperforms all baselines on the real images of NeRSemble and AVA-256. Our method produces more accurate held-out silhouettes, indicating better reconstruction of unseen regions. Its visible head surfaces are closer to the stereo points, and meshes from synchronised cameras agree more closely than those of every other generator. Joint generation improves all three metrics over Pixal3D-FT, which shares our backbone, on both datasets.

Consistency of unseen regions. Fig. 6 shows the back of the head, which none of the inputs shows. Pixal3D-FT hallucinates a different hairstyle when generating each frame

Table 3. **FLAME tracking accuracy on AVA-256.** Each "+" row adds a 3D term towards a per-frame prediction to the Pixel3DMM fit.

FLAME tracker	Chamfer↓	F-score↑
Pixel3DMM [13]	4.86	46.1
+ Face Anything [19]	4.44	51.9
+ Ours	**4.26**	**52.4**

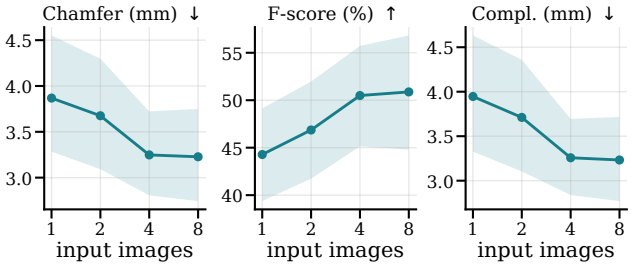

Figure 7. **Number of images.** Head-region metrics of a target frame generated jointly with 1, 2, 4 or 8 images.

independently, whereas our meshes, denoised jointly and coupled at corresponding points, share one hairstyle.

Downstream FLAME tracking. Our meshes can also constrain a tracker. On frontal-camera AVA-256 images, we refine Pixel3DMM's [13] multi-image FLAME fit with a robust point-to-surface term using our meshes or Face Anything's [19] per-pixel geometry. Both use identical placement and weighting, without ground truth. As Tab. 3 shows, both constraints improve the tracker, and our meshes, which cover the whole head, improve it most.

Number of images. Fig. 7 gives the mean and 95% bootstrap interval over 10 NPHM targets. With fixed noise, the target mesh becomes more accurate and complete as images are added, with the largest gains from the first few.

4.4. Ablation Study

Tab. 4 evaluates the ablations on NPHM with the metrics of Tab. 1. Each variant changes one component and uses the same test-time fits. Synchronising Pixal3D-FT without training, by averaging the predicted occupancy of corresponding voxels at every step [1, 16], forces the frames to agree and yields the best consistency. However, consistency measures agreement rather than fidelity. Averaging erodes the heads and yields the largest errors, supporting learned cross-frame information exchange. Global attention over all tokens of other frames is clearly less accurate, as it does not learn where to attend from a few hundred identities. We test the correspondence map against the GT-only variant, trained on ground-truth correspondences only, whose adapters never see the noisy fitted correspondences of test time. Same-voxel attention, as in video models, attends to the same voxel in the other frames and falls clearly behind, since that voxel holds a different point of the head. FLAME correspondences attend to the points that fitted FLAME surfaces match; the more accurate NPHM correspondences of the GT-only variant generate better heads, with lower errors and a clearly higher F-score. Finally, mixing in fitted correspondences, which teaches the adapters to cope with their errors, improves every metric over the GT-only variant.

Table 4. **Ablation study.** Each variant changes one component of the row it is compared with; the full model, which combines all components, is best in every accuracy metric.

Variant	Chamfer↓	Compl.↓	EMD↓	F-score↑	Cons.↓
Training-free sync	7.61	12.38	35.8	42.1	**4.25**
Global attention	3.99	3.78	13.8	45.1	6.54
Same-voxel attention	3.94	3.80	13.7	46.1	6.31
FLAME corresp.	3.60	3.50	12.4	47.8	5.64
GT-only training	3.45	3.38	12.0	50.5	5.11
Ours	**3.27**	**3.23**	**11.8**	**50.9**	5.03

5. Limitations and Future Work

Despite its strong performance, our method is specialised for human heads because it derives correspondences from a parametric head model. It therefore does not directly generalise to full human bodies or generic objects. Extending correspondence-routed joint generation to these settings, for example through body models or category-level correspondences, is a promising direction. The underlying parametric fit becomes less reliable under heavy occlusion or when only a small portion of the face is visible. Our method also assumes that head geometry remains consistent across images, apart from expression changes. Photo collections taken at different times, with changes in hairstyle or accessories, therefore fall outside its scope. Finally, its reliance on a computationally expensive diffusion-based image-to-mesh generator makes real-time applications infeasible.

6. Conclusion

Head Anything turns a pretrained single-image 3D generator into a consistent joint generator for any number of images of a person. Each 3D token attends to corresponding points in other frames. We obtain these correspondences by fitting a parametric head model to the generator's outputs, requiring no ground truth at test time. Training on both ground-truth and fitted correspondences makes the adapters robust to fitting errors. Each mesh follows its own image, while regions unseen in that image are completed using information from all images. On scan benchmarks and real captures, Head Anything improves accuracy over generator and parametric baselines. It also improves consistency over image-to-3D generators, while its meshes improve parametric face tracking. More broadly, correspondence-routed attention lets a single-image 3D generator exploit several images without training a multi-image prior from scratch.

Acknowledgments

This work was supported by the ERC Consolidator Grant Gen3D (101171131) of Matthias Nießner. We thank Angela Dai for the video voice-over.

LLM Usage Statement

We used Claude (Anthropic), mainly through the Claude Code coding assistant, to support this work. Under the authors' direction, it assisted with implementing and running much of the code, including the training and evaluation pipelines, baseline runs, metric and scoring scripts, and job management on our compute cluster. It also assisted with scripts that generate the tables, plots and figure layouts, including the TikZ sources of Figs. 2 and 3. The authors specified the experiments, inspected the results and checked the reported numbers. It supported discussions of research directions and the analysis of experimental results. The authors chose the research direction, the method and all design decisions. It helped draft and edit the text and verify the citations. All text was revised and approved by the authors, who take full responsibility for the entire content of this paper.

References

[1] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. MultiDiffusion: Fusing diffusion paths for controlled image generation. In *ICML*, pages 1737–1752, 2023. 3, 8, 14

[2] Paul J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. 7, 14

[3] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, pages 187–194, 1999. 2

[4] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, pages 22563–22575, 2023. 2

[5] Jianqi Chen, Biao Zhang, Xiangjun Tang, and Peter Wonka. V2M4: 4D mesh animation reconstruction from a single monocular video. In *ICCV*, pages 11643–11653, 2025. 3

[6] Xiyi Chen, Marko Mihajlovic, Shaofei Wang, Sergey Prokudin, and Siyu Tang. Morphable diffusion: 3D-consistent diffusion for single-image avatar creation. In *CVPR*, pages 10359–10370, 2024. 2, 3

[7] Yuren Cong, Mengmeng Xu, Christian Simon, Shoufa Chen, Jiawei Ren, Yanping Xie, Juan-Manuel Perez-Rua, Bodo Rosenhahn, Tao Xiang, and Sen He. FLATTEN: Optical FLow-guided ATTENtion for consistent text-to-video editing. In *ICLR*, 2024. 2

[8] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3D objects. In *CVPR*, pages 13142–13153, 2023. 2

[9] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3D face model from in-the-wild images. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 40(4):88:1–88:13, 2021. 2

[10] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. TokenFlow: Consistent diffusion features for consistent video editing. In *ICLR*, 2024. 2

[11] Simon Giebenhain, Tobias Kirschstein, Markos Georgopoulos, Martin Rünz, Lourdes Agapito, and Matthias Nießner. Learning neural parametric head models. In *CVPR*, pages 21003–21012, 2023. 2, 3, 4, 6, 13

[12] Simon Giebenhain, Tobias Kirschstein, Markos Georgopoulos, Martin Rünz, Lourdes Agapito, and Matthias Nießner. MonoNPHM: Dynamic head reconstruction from monocular videos. In *CVPR*, pages 10747–10758, 2024. 2, 5, 6, 7, 12

[13] Simon Giebenhain, Tobias Kirschstein, Martin Rünz, Lourdes Agapito, and Matthias Nießner. Pixel3DMM: Versatile screen-space priors for single-image 3D face reconstruction. In *ICLR*, 2026. 2, 5, 6, 7, 8, 12

[14] Yanqin Jiang, Li Zhang, Jin Gao, Weiming Hu, and Yao Yao. Consistent4D: Consistent 360° dynamic object generation from monocular video. In *ICLR*, 2024. 3

[15] Yash Kant, Aliaksandr Siarohin, Ziyi Wu, Michael Vasilkovsky, Guocheng Qian, Jian Ren, Riza Alp Guler, Bernard Ghanem, Sergey Tulyakov, and Igor Gilitschenski. SPAD: Spatially aware multi-view diffusers. In *CVPR*, pages 10026–10038, 2024. 2

[16] Jaihoon Kim, Juil Koo, Kyeongmin Yeo, and Minhyuk Sung. SyncTweedies: A general generative framework based on synchronized diffusions. In *NeurIPS*, pages 95198–95237, 2024. 3, 8

[17] Tobias Kirschstein, Shenhan Qian, Simon Giebenhain, Tim Walter, and Matthias Nießner. NeRSemble: Multi-view radiance field reconstruction of human heads. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 42(4):161:1–161:14, 2023. 1, 2, 7, 13, 15

[18] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36 (4):1–13, 2017. 7

[19] Umut Kocasarı, Simon Giebenhain, Richard Shaw, and Matthias Nießner. Face anything: 4D face reconstruction from any image sequence. In *ECCV*, pages 269–287, 2026. 2, 5, 8, 13

[20] Dong-Yang Li, Wang Zhao, Yuxin Chen, Wenbo Hu, Meng-Hao Guo, Fang-Lue Zhang, Ying Shan, and Shi-Min Hu. Pixal3D: Pixel-aligned 3D generation from images. In *SIGGRAPH Conference Papers*, pages 1–12, 2026. 2, 3, 5, 6, 7, 12

[21] Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017. 2, 4, 5, 12

[22] Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding

Liang, Wanli Ouyang, and Yan-Pei Cao. TripoSG: High-fidelity 3D shape synthesis using large-scale rectified flow models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. Early access. 2, 5, 6, 7

[23] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023. 2, 3

[24] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3D object. In *ICCV*, pages 9298–9309, 2023. 2

[25] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023. 2, 3

[26] Julieta Martinez, Emily Kim, Javier Romero, Timur Bagautdinov, Shunsuke Saito, Shoou-I Yu, Stuart Anderson, Michael Zollhöfer, Te-Li Wang, Shaojie Bai, Chenghui Li, Shih-En Wei, Rohan Joshi, Wyatt Borsos, Tomas Simon, Jason Saragih, Paul Theodosis, Alexander Greene, Anjani Josyula, Silvio Mano Maeta, Andrew I. Jewett, Simon Venshtain, Christopher Heilman, Yueh-Tung Chen, Sidi Fu, Mohamed Ezzeldin A. Elshaer, Tingfang Du, Longhua Wu, Shen-Chi Chen, Kai Kang, Michael Wu, Youssef Emad, Steven Longay, Ashley Brewer, Hitesh Shah, James Booth, Taylor Koska, Kayla Haidle, Matt Andromalos, Joanna Hsu, Thomas Dauer, Peter Selednik, Tim Godisart, Scott Ardisson, Matthew Cipperly, Ben Humberston, Lon Farr, Bob Hansen, Peihong Guo, Dave Braun, Steven Krenn, He Wen, Lucas Evans, Natalia Fadeeva, Matthew Stewart, Gabriel Schwartz, Divam Gupta, Gyeongsik Moon, Kaiwen Guo, Yuan Dong, Yichen Xu, Takaaki Shiratori, Fabian Prada, Bernardo R. Pires, Bo Peng, Julia Buffalini, Autumn Trimble, Kevyn McPhail, Melissa Schoeller, and Yaser Sheikh. Codec avatar studio: Paired human captures for complete, driveable, and generalizable avatars. In *NeurIPS (Datasets and Benchmarks Track)*, pages 83008–83023, 2024. 1, 2, 7, 13, 15

[27] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, 2023. 3

[28] Shenhan Qian. VHAP: Versatile head alignment with adaptive appearance priors. https://github.com/ShenhanQian/VHAP, 2024. 2, 5, 6, 7

[29] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. GaussianAvatars: Photorealistic head avatars with rigged 3D gaussians. In *CVPR*, pages 20299–20309, 2024. 2

[30] Jiawei Ren, Kevin Xie, Ashkan Mirzaei, Hanxue Liang, Xiaohui Zeng, Karsten Kreis, Ziwei Liu, Antonio Torralba, Sanja Fidler, Seung Wook Kim, and Huan Ling. L4GM: Large 4D gaussian reconstruction model. In *NeurIPS*, pages 56828–56858, 2024. 3

[31] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016. 7, 14

[32] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, pages 501–518, 2016. 7, 14

[33] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. MVDream: Multi-view diffusion for 3D generation. In *ICLR*, 2024. 2

[34] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research*, 2026. 3, 12

[35] Felix Taubner, Ruihang Zhang, Mathieu Tuli, and David B. Lindell. CAP4D: Creating animatable 4D portrait avatars with morphable multi-view diffusion models. In *CVPR*, pages 5318–5330, 2025. 2, 3

[36] Tencent Hunyuan3D Team. Hunyuan3D 2.1: From images to high-fidelity 3D assets with production-ready PBR material. *arXiv preprint arXiv:2506.15442*, 2025. 2, 5, 6, 7

[37] Hung-Yu Tseng, Qinbo Li, Changil Kim, Suhib Alsisan, Jia-Bin Huang, and Johannes Kopf. Consistent view synthesis with pose-guided diffusion models. In *CVPR*, pages 16773–16783, 2023. 2

[38] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991. 7, 14

[39] Lizhen Wang, Zhiyuan Chen, Tao Yu, Chenguang Ma, Liang Li, and Yebin Liu. FaceVerse: A fine-grained and detail-controllable 3D face morphable model from a hybrid dataset. In *CVPR*, pages 20333–20342, 2022. 2, 6, 13

[40] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *ICCV*, pages 7623–7633, 2023. 2

[41] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Yajie Bao, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, and Yao Yao. Direct3D-S2: Gigascale 3D generation made easy with spatial sparse attention. In *NeurIPS*, pages 170778–170804, 2025. 2, 5, 6, 7

[42] 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 *CVPR*, pages 21469–21480, 2025. 2, 5, 6, 7, 14

[43] Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, and Jiaolong Yang. Native and compact structured latents for 3D generation. In *CVPR*, pages 14419–14429, 2026. 2, 5, 6, 7, 12

[44] Liangbin Xie, Xintao Wang, Honglun Zhang, Chao Dong, and Ying Shan. VFHQ: A high-quality dataset and benchmark for video face super-resolution. In *CVPR Workshops*, pages 657–666, 2022. 7, 15

[45] Yiming Xie, Chun-Han Yao, Vikram Voleti, Huaizu Jiang, and Varun Jampani. SV4D: Dynamic 3D content generation

with multi-frame and multi-view consistency. In *ICLR*, 2025. 3

[46] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, pages 3836–3847, 2023. 3

[47] 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*, 3:9150038, 2024. 14

[48] Hao Zhu, Wayne Wu, Wentao Zhu, Liming Jiang, Siwei Tang, Li Zhang, Ziwei Liu, and Chen Change Loy. CelebV-HQ: A large-scale video facial attributes dataset. In *ECCV*, pages 650–667, 2022. 7, 15

[49] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Towards metrical reconstruction of human faces. In *ECCV*, pages 250–269, 2022. 2

Appendix

A. Overview

This appendix complements the main paper. Appendix B describes our supplementary video, Appendix C gives additional implementation details, Appendix D describes the benchmarks and the evaluation protocol, and Appendix E reports additional comparisons. Appendix F adds ablations, Appendix G reports the runtime and memory of every step, and Appendix H shows further results of our method.

B. Supplementary Video

We highly recommend watching our supplementary video, a narrated presentation of our method that contains additional qualitative results. The video shows renderings of our generated head meshes, in which an orbiting camera turns around each mesh and reveals all of its details, including the hair, the ears and the back of the head. It further compares our meshes with those of the baselines, showing the reconstruction quality and fidelity of each method as well as the consistency of the meshes generated from different images of the same person.

C. Additional Implementation Details

C.1. Base Model and Sampling

We build on the public code and the released weights of Pixal3D [20], which follows the two-stage design of TRELLIS.2 [43] with DINOv3 [34] conditioning, and sample all stages with its released sampler settings. The noise of every stage is seeded per frame, so a frame receives the same noise in the single-image pass and in every joint group. The cascade from resolution 512^3 to 1024^3 and the decoding run per frame on the jointly generated S2 latent.

C.2. Single-Image Fine-Tuning

Pixal3D-FT is Pixal3D fine-tuned in all three geometry stages with Eq. (1) and no architectural change. It is fine-tuned on renders of the scans, and our adapters are trained on the same renders. We rasterise each scan with its captured vertex colours from random viewpoints with varying fields of view. We match foreground colour statistics to real video frames and apply slight Gaussian blur, sensor noise and a JPEG round trip. Finally, we crop around the head as in the real-image pipeline. Among several variants of this renderer, this one was closest to real photographs in DINOv3 feature statistics.

C.3. Adapters and Joint Training

Architecture. The adapters follow every third block of both flows, starting with the third block. Each adapter uses LayerNorm, width-512 query, key and value projections with 8 heads, the key embeddings of Eq. (5), and a learned null key-value pair. A zero-initialised projection maps back to the backbone width, with its output scaled by a gate on the timestep embedding. The key embeddings combine a depth MLP, used in both stages, with a validity embedding in S1 and an offset MLP in S2, and every stage has 55 M adapter parameters. The depth features d_i and d_j are the coordinates of the query and the key token along the viewing axis of the view-aligned latent grid of their own frame.

Training. Registration groups and fitted groups are drawn with equal probability. All frames of a group share the flow time, which S1 draws from a logit-normal distribution. The backbone is fine-tuned together with the adapters, with a learning rate ten times lower than that of the adapters.

C.4. Correspondences

Registration correspondences. For registration groups, $\varphi_{f\to g}$ in Eq. (4) uses the 8 nearest registration vertices with inverse-distance weights. A_f is the known similarity transform from the scan to the render's view-aligned latent grid. In S1, a correspondence is valid if $\Phi_{f\to g}(p)$ lies inside the latent grid; in S2, a key token is valid if it lies within two tokens of $\Phi_{f\to g}(p)$. Invalid keys are masked, and the null key is always available.

FLAME fitting. The test-time FLAME fit minimises

$$\min_{s_f,R_f,t_f,\beta_f} \sum_{(y,\hat{c})\in\mathcal{C}_f} \rho\big(s_f R_f\, F_{\beta_f}(\hat{c}) + t_f - y\big) + \lambda_\beta \|\beta_f\|^2, \tag{S1}$$

over the pairs $\mathcal{C}_f$ of a mesh point y and a FLAME template coordinate $\hat{c}$, where $F_{\beta_f}(\hat{c})$ is the posed FLAME surface point at $\hat{c}$, ρ a Huber penalty and λ_β a prior weight on the shape, expression and jaw parameters β_f of FLAME 2023 [21]. The similarity is initialised by a Procrustes alignment. The pairs come from face, eye, nose, lips and forehead pixels of the Pixel3DMM [13] fit whose rays intersect the single-frame mesh in the generator's camera. We do not use the fit's depth or scale.

MonoNPHM group fit. We optimise Eq. (6) with Adam. The identity code starts at the mean of the identity codes of MonoNPHM [12] and the expression codes at zero, and $\mathcal{R}$ penalises both codes after normalising them by the standard deviations of the code banks of MonoNPHM. Every iteration uses random surface points of the single-frame meshes above the neck. The fitted pool of the training groups was computed once with the same procedure, from single-frame meshes of the backbone after its first fine-tuning stage.

Inverse deformation. $\varphi_{f\to g}$ maps y to $\bar{y} = S_f(y) + \mathcal{D}(S_f(y); z_f^{\mathrm{ex}})$ using MonoNPHM's backward deformation

Table S1. **Holes on NPHM.** Mean number of open boundaries per frame in the head region and the head metrics of Tab. 1 for the meshes of Head Anything.

Clean-up	Holes↓	Chamfer↓	F-score↑
Standard clean-up	63.2	3.26	50.9
Topology-preserving + filling	4.2	3.27	50.9

$\mathcal{D}$, then returns $S_g^{-1}(\tilde{y})$. We solve $\tilde{y} + \mathcal{D}(\tilde{y}; z_g^{\text{ex}}) = \bar{y}$ approximately with eight iterations of $\tilde{y} \leftarrow \bar{y} - \mathcal{D}(\tilde{y}; z_g^{\text{ex}})$, starting from $\tilde{y} = \bar{y}$. Deformation blends to zero on the rigid torso. Correspondences outside the grid or without nearby tokens are masked; the null key lets tokens ignore incorrect matches. On the test-time fits of the 10 NPHM groups, the final residual has a median below 0.01 mm and exceeds 1 mm for 0.06% of the points, and 14.8% of the S1 correspondences leave the latent grid.

Relation of the two kinds of correspondences. Both instances of Eq. (3) map a point of the head in one frame to the same point in another frame, but they are built differently. Registration maps interpolate the displacements of corresponding vertices, so they are exact on the scan surface and extrapolate off the surface by inverse-distance weighting. Fitted maps evaluate MonoNPHM's learned deformation field, which is smooth and defined everywhere, but inherit errors from the FLAME and MonoNPHM fits. Their placement A_f comes from the FLAME fit rather than the scan. Training on registrations alone would leave a gap between training and test-time correspondences, which the fitted pool closes. The GT-only variant in Tab. 4 and the fitted-only variant in Tab. S4 evaluate the two extremes.

C.5. Clean-Up

The pipeline's standard mesh clean-up closes small holes, cuts the decoded surface along non-manifold edges and removes connected components below an area threshold. Removing pieces disconnected from the head by these cuts opens holes, most visibly around the mouth. Our clean-up applies the same repairs but removes floating components only after re-welding the cut, then fills small loops. Above the head region's lower boundary, it closes each remaining boundary loop of perimeter 10–250 mm with a flat ring patch over the hole border. For real images, the minimum perimeter is 3 mm. This removes most holes and leaves the geometry metrics essentially unchanged (Tab. S1). We apply this clean-up to both Head Anything and Pixal3D-FT meshes. It is not a contribution of this work but makes the meshes usable as hole-free head surfaces.

C.6. Real-Image Pipeline

For a sequence of real images, as for the FaceVerse renders, all frames are cropped with one square crop around the matte of the first frame, enlarged by 10%. This crop defines one virtual camera in which every frame is generated. Its intrinsics come from capture calibration, the FaceVerse render camera or, for in-the-wild images, the focal length predicted by Face Anything [19]. Face Anything predicts metric depth and canonical coordinates for each frame, from which we obtain its rigid head motion relative to this camera. Scaled ray-ICP against the predicted depth then estimates the metric scale of each single-frame mesh. On NPHM, every frame is generated in its own render camera, and A_f comes from per-frame FLAME fits.

D. Benchmarks and Evaluation Protocol

D.1. Datasets

NPHM. We use the 10 held-out NPHM [11] identities 088, 127, 195, 243, 291, 339, 387, 436, 482 and 535, which are not used in training, each with eight expressions. They are rendered with the photorealistic renderer of Appendix C.2 from random viewpoints with a yaw within $\pm 30°$, a pitch between $-8°$ and $20°$ and a field of view between $10°$ and $70°$. Each group of eight frames is generated jointly, and all 80 frames are scored.

FaceVerse. We randomly select 10 FaceVerse [39] identities with artifact-free scans, each with eight expressions. Every scan is rendered from a random viewpoint with a yaw within $\pm 30°$ and a pitch between $-8°$ and $20°$, with one camera distance per identity and a field of view at which the head fits into the image in all eight frames. Our pipeline generates all eight images in one virtual camera (Appendix C.6), so the change of viewpoint appears as head motion. Each ground-truth scan is cut 20 mm below the chin. Since the scans are not metric, a single dataset-wide scale factor, shared by all methods, sets the median outer eye-corner width to 90 mm.

NeRSemble and AVA-256. We use the 10 NeRSemble [17] identities 139, 170, 171, 175, 176, 177, 178, 192, 497 and 510, none of which is an NPHM training or test identity, and the 10 AVA-256 [26] identities AAN112, ADL311, AEY864, AJR151, ANX726, CDR970, DOT682, FXN596, KDA058 and NRE683. For each identity, two synchronised cameras about $20°$ to $33°$ to the left and right of the face observe four timestamps: a near-neutral face, a smile, an open mouth and one further expression, where a cheek expression replaces the smile on AVA-256, which has no smile sequence. The two cameras are chosen by a fixed rule among those that see the whole head at all four timestamps, and NeRSemble timestamps in which the head leaves one of the two views are not used. Since the methods assume one intrinsic matrix per sequence, the AVA-256 images of the two cameras are resampled into one virtual pinhole camera per identity with the real camera rotations and centres. For evaluation, we hold out all other cameras of an identity, seven on NeRSemble and seven or eight on AVA-256, and reconstruct the multi-view stereo points of

each timestamp with COLMAP [31, 32] from the calibrated cameras.

Inputs. All methods receive the same RGBA crops, with BiRefNet [47] mattes in the wild and the dataset mattes on the captures. We binarise mattes for every method because soft edges cause some generators to produce flat cards. Parametric trackers receive the same crops composited on white, as their pipelines require. The image-to-3D generators are run with their official code and default settings on every image.

D.2. Alignment and Metrics

Scan benchmarks. Every prediction is aligned to its scan using the same similarity search [2, 38]. We run scaled trimmed ICP from 72 starting orientations, using a two-sided residual to prevent scale collapse. We then run face ICP from each resulting pose, followed by 12 perturbed restarts from the best poses. We keep the pose with the lowest face residual and finally align the face rigidly, so that the metrics compare shapes rather than placements. The face and lips regions are defined by a FLAME fit of each scan, and the head region by a plane 2 cm below the chin of that fit. To measure cross-frame consistency, we move each aligned prediction's non-face points into every other frame of the same person using the inverse-distance-weighted displacement of the 8 nearest correspondence vertices. Correspondences come from NPHM registrations or FLAME fits to FaceVerse scans. We measure the mean distance to each target frame's aligned non-face surface and average over all target frames.

Captures. For silhouette IoU, we fit a per-frame similarity transform to masks from the alignment cameras, including the input cameras, by maximising a differentiable soft-silhouette IoU from 144 starts. We then measure IoU in held-out cameras within a head region shared by all methods. For the Chamfer distance, every mesh is aligned to the stereo points of its timestamp with the similarity search above, and only the prediction vertices within 10 mm of a camera-visible stereo point are used. The cross-view consistency is the head Chamfer distance between the two meshes generated from the two synchronised images of one timestamp, each placed by its own alignment to the stereo points.

E. Additional Quantitative Results

TRELLIS with all images. TRELLIS [42] can also be conditioned on all eight images of a person at once, in its multi-image mode with multidiffusion [1], which outputs one mesh per person. Tab. S2 compares this mode with TRELLIS applied to every image on its own and with our method on NPHM, with the metrics of Tab. 1. The additional images make TRELLIS more accurate, but our meshes are far more accurate in every metric. The multi-image mode shares one mesh per person, making it consistent by construction, like the parametric models in Tab. 1. However, it cannot follow per-frame expressions, so its consistency is shown in gray and not ranked.

Table S2. **TRELLIS with all images of a person on NPHM.** The multi-image mode receives all eight images and gives one mesh per person, which is scored on every frame with the protocol of Tab. 1. Distances in mm and F-score in %; the consistency of the shared multi-image mesh is shown in gray and not ranked.

Method	Chamfer↓	Compl.↓	EMD↓	F-score↑	Cons.↓
TRELLIS [42]	6.26	6.10	19.8	31.1	12.64
TRELLIS (multi-image)	5.58	5.47	19.1	34.4	4.88
Ours	**3.27**	**3.23**	**11.8**	**50.9**	**5.03**

Table S3. **Number of images.** Head metrics of the target frames of Fig. 7, with distances in mm and F-score in %.

Images	Chamfer↓	Compl.↓	EMD↓	F-score↑
1	3.87	3.95	13.3	44.3
2	3.68	3.71	12.5	46.9
4	3.25	3.26	12.0	50.5
8	3.23	3.23	12.1	50.9

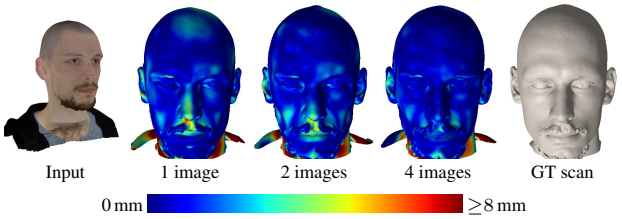

Figure S1. **More images.** Head error of one NPHM frame, generated from its input image alone and together with one and three other images of the same person.

Number of images. Tab. S3 lists the head metrics of Fig. 7, and Fig. S1 shows the head error of one target frame generated from one, two and four images. Every target is generated with the first $n-1$ frames of a fixed random order of the other frames of its identity, and the fitted correspondences are recomputed for every n. With more images, the head becomes more accurate.

F. Ablations

All ablations are evaluated on the 80 NPHM frames with the protocol of Tab. 1, the same test-time fits and the same seeds as Head Anything. Every variant changes one setting of its reference, and both stages are retrained with otherwise identical settings unless the variant needs no training.

Design choices. Tab. S4 varies the joint stages, the keys, the training correspondences and the backbone training. Generating both stages jointly performs best. Most of the gain comes from joint generation in the structure stage, since S1 determines the occupied voxels and thus the hair

Table S4. **Design choices on NPHM.** Distances in mm; groups of rows are separated by rules.

Variant	Chamfer↓	Compl.↓	EMD↓	Face↓	Lips↓
Joint structure stage only	3.31	3.25	12.2	1.430	1.603
Joint shape stage only	3.82	3.76	13.1	1.465	1.644
No depth features	**3.17**	3.21	**11.7**	**1.386**	1.735
No null key	3.33	3.23	**11.7**	1.418	1.756
Fitted-only training	3.32	3.27	**11.7**	1.440	1.698
Frozen backbone	3.30	**3.20**	12.0	1.427	1.645
Ours	3.27	3.23	11.8	1.393	**1.565**

Table S5. **Matched training compute on NPHM.** The metrics of Tab. 1, with distances in mm and F-score in %.

Variant	Chamfer↓	Compl.↓	EMD↓	F-score↑	Cons.↓
Pixal3D-FT (one-by-one)	3.86	3.77	13.3	45.6	6.73
Pixal3D-FT + matched fine-tuning	4.04	3.91	13.3	44.8	6.67
Ours	**3.27**	**3.23**	**11.8**	**50.9**	**5.03**

volume and back of the head. Joint generation in the shape stage further improves the result. Without the depth features or the null key, the overall accuracy is similar, but the lips, the region whose shape changes most between expressions, become less accurate. Training on the fitted pool alone performs slightly worse than mixed training. Freezing the backbone also captures slightly less facial detail than fine-tuning it jointly with the adapters.

Training compute. Our adapters are trained jointly with the backbone for additional steps. For comparison, we fine-tune Pixal3D-FT without adapters using the same number of steps, trainer, data, group sizes and images per step, then generate each frame independently (Tab. S5). The additional fine-tuning does not improve Pixal3D-FT and even slightly reduces its accuracy. This indicates that our method's gains come from joint generation rather than additional training.

G. Runtime and Memory

We measure the time of every step for the eight images of each of 10 NPHM identities on one NVIDIA A100, running Pixal3D-FT and our method one after the other in the same job. Tab. S6 gives the mean time per image of every step and the peak GPU memory. Our method takes 260 s per image on average and Pixal3D-FT 114 s, about 2.3 times as long.

Table S6. **Runtime and memory for eight images.** Mean time per image in seconds over 10 NPHM identities, and peak GPU memory.

Step	Pixal3D-FT	Ours
Single-image generation	27	30
Joint generation	–	31
Pixel3DMM	–	21
FLAME fit	–	15
MonoNPHM group fit	–	62
Mesh decimation	–	14
Clean-up	87	81
Other	–	6
Total time per image (s)	114	260
Peak GPU memory (GB)	20	46

H. Additional Qualitative Results

Fig. S2, Fig. S3, Fig. S4 and Fig. S5 show further results of our method on NeRSemble [17], AVA-256 [26], VFHQ [44] and CelebV-HQ [48], two identities with four images each. Our method produces high-quality, detailed head geometry from studio captures and in-the-wild video frames alike. Every mesh is shown in the head frame from the front, from both sides and from the back, in surface-normal colours. Fig. S6 shows two failure cases with a hand-held microphone in front of the mouth or at the chin. Our method merges the microphone and head into one surface rather than separating them. The microphone also overlaps with the mouth and chin. The rest of the head, including the back, stays plausible and consistent with the input.



Figure S2. **More results on NeRSemble.** Two identities with four input images each.

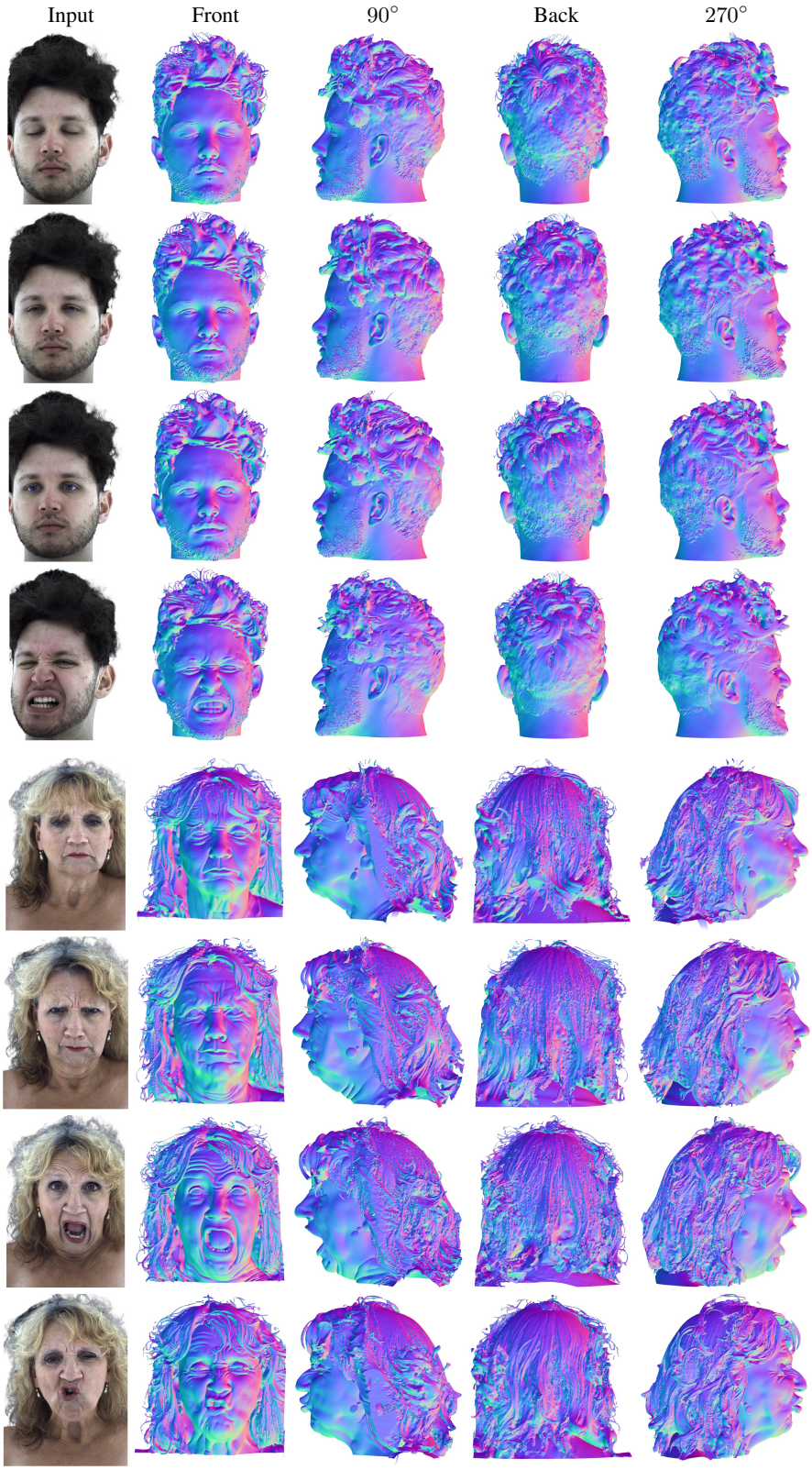

Figure S3. **More results on AVA-256.** Two identities with four input images each.

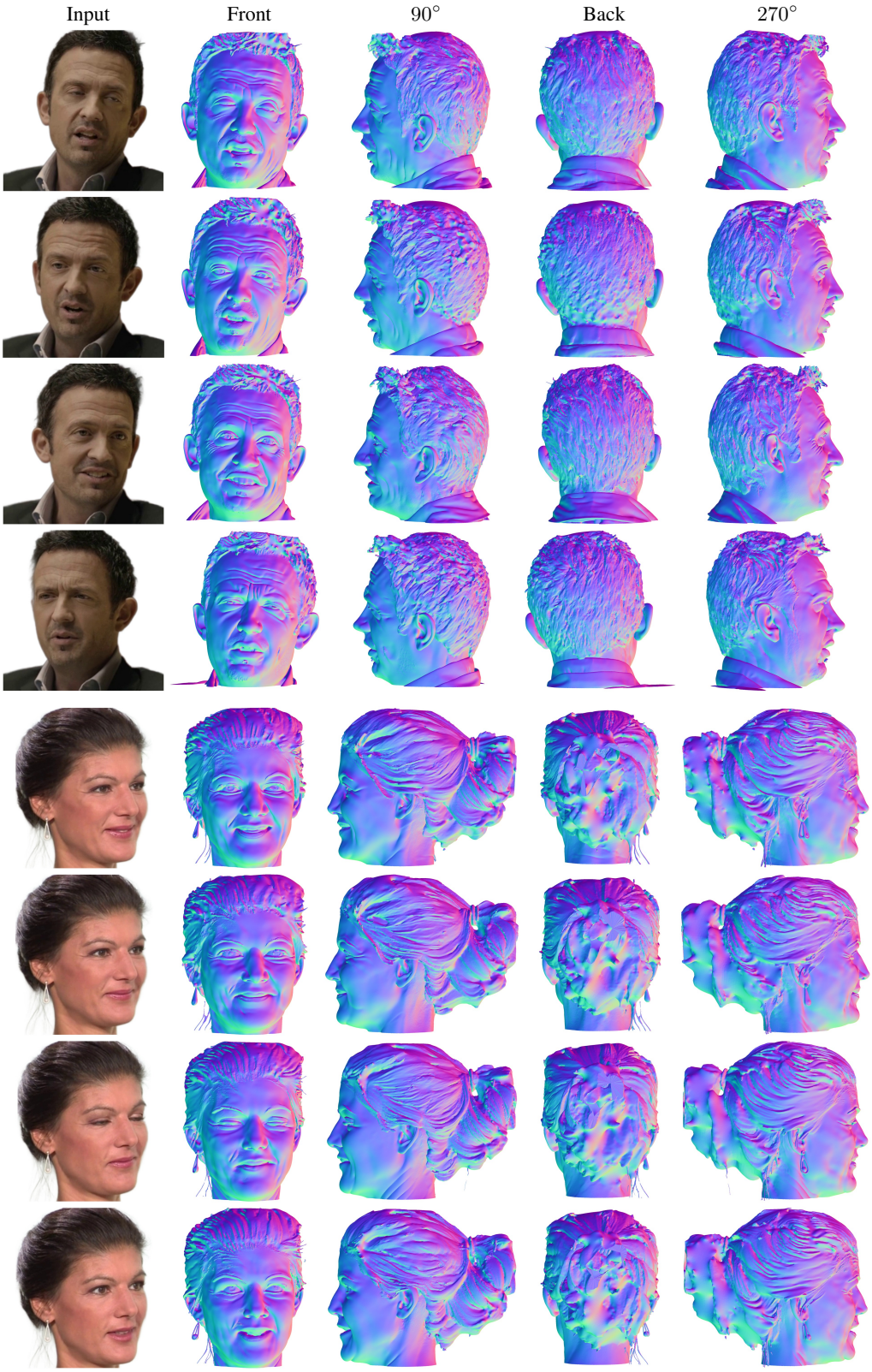

Figure S4. **More results on VFHQ.** Two identities with four frames each from one video.



Figure S5. **More results on CelebV-HQ.** Two identities with four frames each from one video.

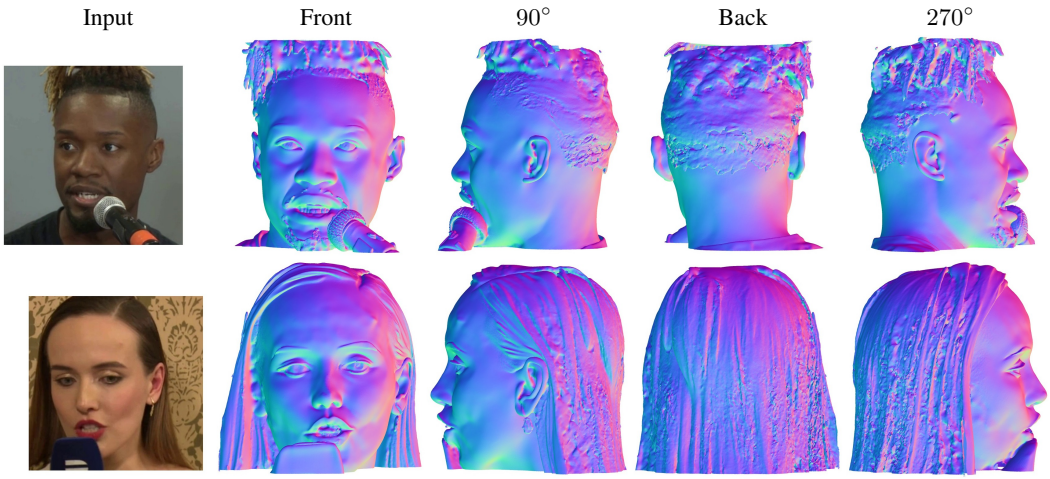

Figure S6. **Failure cases.** A microphone held in front of the mouth or at the chin is joined with the head mesh, and the microphone and the face overlap in one surface.