Portrait4D: Learning One-Shot 4D Head Avatar Synthesis using Synthetic Data —— Supplementary Material

Yu Deng Duomin Wang Xiaohang Ren Xingyu Chen Baoyuan Wang
Xiaobing.AI

https://yudeng.github.io/Portrait4D/

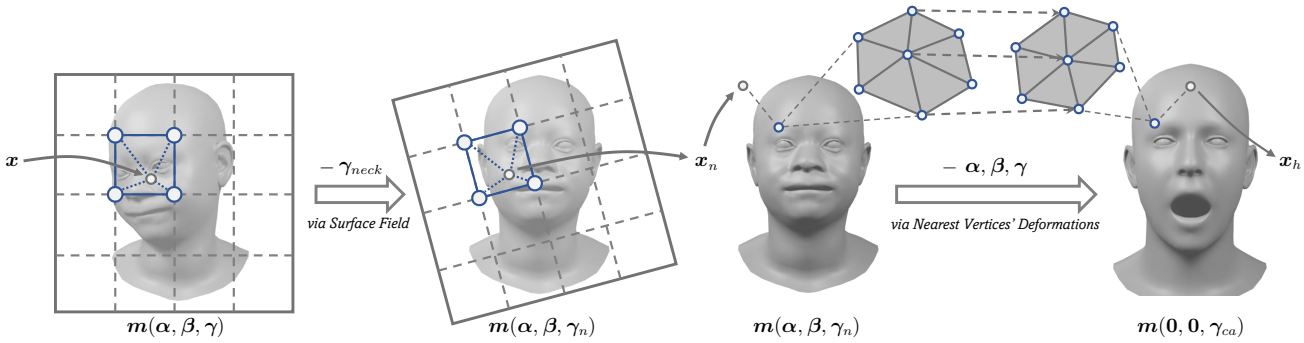

Figure I. Overview of the part-wise 3D deformation field $\mathcal{D}$ in GenHead. We first derive deformation caused by neck pose via the Surface Field approach [2]. The deformation of an arbitrary 3D point is obtained via tri-linear interpolation between those of pre-defined voxel grids. Then, we eliminate the deformation caused by shape and expression variations, via weighted deformations of its nearest vertices on the FLAME mesh. Here, we only show the derivation of the head region deformation for an illustration.

I. Overview

We first present more implementation details in Sec. II, including those of GenHead, 4D data synthesis, and the one-shot 4D head reconstruction pipeline. Then, we provide evaluations of GenHead and additional results of one-shot 4D head synthesis in Sec. III. Finally, we discuss limitations and ethics consideration in Sec. IV.

II. More Implementation Details

II.1. Part-wise Generative Head Model

In this section, we describe more details about the GenHead model, including the shape-aware canonical triplane generator, the part-wise deformation field, the image rendering process, and the learning strategy. We also provide details about data preprocessing and training.

Shape-aware canonical triplane generator. As described in the main paper, our canonical tri-plane generator G_{ca} also takes the shape code $\boldsymbol{\alpha}$ as input for synthesizing shape-related canonical appearance. To achieve this, we simply concatenate $\boldsymbol{\alpha}$ with the random noise $\boldsymbol{z}$, and send them together into the mapping network of G_{ca}'s StyleGAN2 backbone. Considering that the shape and appearance only have weak correlations, we randomly replace $\boldsymbol{\alpha}$ sent into the mapping net with an arbitrary shape code at a possibility of 50% during training to avoid overfitting.

Part-wise 3D deformation field. The part-wise 3D deformation field $\mathcal{D}$ produces observation-to-canonical deformations $[\Delta\boldsymbol{x}_h, \Delta\boldsymbol{x}_p]$ for a 3D point $\boldsymbol{x}$, for modeling shape deformations, as well as animations of neck, face, eyes, and mouth. Illustrations are in Fig. I and II and we describe the details below.

We first calculate the deformation caused by neck joint rotation $\boldsymbol{\gamma}_{neck}$. We leverage Surface Field (SF) proposed by [2], which derives the canonical coordinate $\boldsymbol{x}_n$ via

$$\boldsymbol{x}_n = \boldsymbol{t}^n_{\boldsymbol{x}} \cdot [u, v, w]^\intercal + \langle \boldsymbol{x} - \boldsymbol{t}_{\boldsymbol{x}} \cdot [u, v, w]^\intercal, \boldsymbol{n}_{\boldsymbol{t}_{\boldsymbol{x}}} \rangle \boldsymbol{n}^n_{\boldsymbol{t}_{\boldsymbol{x}}}, \quad \text{(I)}$$

where $\boldsymbol{t}_{\boldsymbol{x}}$ is $\boldsymbol{x}$'s nearest triangle on the mesh $\boldsymbol{m}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$, $[u, v, w]$ is the barycentric coordinates of $\boldsymbol{x}$'s projection onto the triangle, and $\boldsymbol{n}_{\boldsymbol{t}_{\boldsymbol{x}}}$ is the surface normal. $\boldsymbol{t}^n_{\boldsymbol{x}}$ and $\boldsymbol{n}^n_{\boldsymbol{t}_{\boldsymbol{x}}}$ are the corresponding triangle and surface normal on

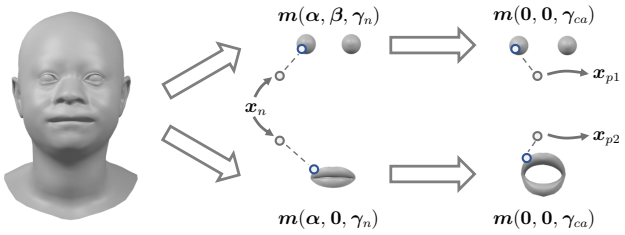

Figure II. We further derive part-region deformations using the cropped-out eye balls and lip meshes, to better deal with motions of eyes and teeth.

a mesh $\boldsymbol{m}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}_n)$ with canonical neck pose, that is, $\boldsymbol{\gamma}_n = [\boldsymbol{0}, \boldsymbol{\gamma}_{jaw}, \boldsymbol{\gamma}_{eye}]$. In practice, we avoid direct SF calculation for every 3D point. Instead, we introduce pre-defined low-resolution voxel grids for SF computation, and approximate the deformation of an arbitrary 3D point with tri-linear interpolation of those of pre-defined voxel grids, as shown in Fig. I. This approximation largely reduces the computational cost produced by nearest triangle search. Moreover, the tri-linear interpolation serves as a low-pass filter which largely alleviates the discontinuity of deformations around hairs that have contact with both the face and the shoulder.

After eliminating the neck pose, we tackle the deformation produced by $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}_{jaw}$ to deform the 3D point to the head canonical space. Specifically, for the 3D point $\boldsymbol{x}_n$ after neck pose canonicalization, we search for its nearest vertex $\boldsymbol{v}_n$ on the mesh $\boldsymbol{m}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}_n)$ and further obtain the one-ring neighborhood $\mathcal{N}(\boldsymbol{v}_n)$ of the vertex. We then calculate the deformation of $\boldsymbol{x}_n$ via weighted summation of the offsets produced by the neighborhood:

$$\Delta(\boldsymbol{x}_n) = \frac{1}{Z} \sum_{\boldsymbol{v}_i \in \mathcal{N}(\boldsymbol{v}_n)} \omega_i \cdot (\boldsymbol{v}_i^{ca} - \boldsymbol{v}_i), \tag{II}$$

where $\boldsymbol{v}_i^{ca}$ denotes $\boldsymbol{v}_i$'s corresponding vertex on mesh $\boldsymbol{m}(\boldsymbol{0}, \boldsymbol{0}, \boldsymbol{\gamma}_{ca})$, $\omega_i = 1/\|\boldsymbol{x}_n - \boldsymbol{v}_i\|_2$ is the weighting coefficient proportional to the inverse distance between the 3D point and the vertices, and Z is a normalizing scalar. The coordinate $\boldsymbol{x}_h$ in the head canonical space can then be obtained via $\boldsymbol{x}_h = \Delta(\boldsymbol{x}_n) + \boldsymbol{x}_n$. That is, the observation-to-head-canonical deformation $\Delta\boldsymbol{x}_h = \Delta(\boldsymbol{x}_n) + \boldsymbol{x}_n - \boldsymbol{x}$.

Finally, we tackle the eye balls' rotation as well as relative movements between lips and teeth. As shown in Fig. II, we crop out the eye region and lip region from the FLAME mesh. For the eye region, we search for the closest vertices on the eye balls for the 3D point $\boldsymbol{x}_n$, and follow the same procedure as described in Eq. (II) to obtain canonical point $\boldsymbol{x}_{p1}$. For teeth, we notice that their motions are related only to the jaw movements [12]. Therefore, we derive their deformation via an expressionless lip-region mesh $\boldsymbol{m}(\boldsymbol{\alpha}, \boldsymbol{0}, \boldsymbol{\gamma}_n)$. We use the offsets between the vertices on this lip mesh and the corresponding vertices on the canonical mesh $\boldsymbol{m}(\boldsymbol{0}, \boldsymbol{0}, \boldsymbol{\gamma}_{ca})$ to derive the deformation via Eq. (II), and obtain the canonical point $\boldsymbol{x}_{p2}$. Therefore, the observation-to-part-canonical deformation $\Delta\boldsymbol{x}_p = \boldsymbol{x}_{p1} - \boldsymbol{x}$ or $\boldsymbol{x}_{p2} - \boldsymbol{x}$, where we use $\boldsymbol{x}_{p1} - \boldsymbol{x}$ for points inside the bounding boxes of the eye balls, and $\boldsymbol{x}_{p2} - \boldsymbol{x}$ for the remainings.

Image rendering. Given the part-wise deformations $[\Delta\boldsymbol{x}_h, \Delta\boldsymbol{x}_p]$, we can obtain the corresponding features $\boldsymbol{f}_h$ and $\boldsymbol{f}_p$ from the triplanes T_h and T_p, respectively. An MLP then decodes the features to their radiance $(\sigma_h, \boldsymbol{c}_h)$ and $(\sigma_p, \boldsymbol{c}_p)$, where $c. \in \mathbb{R}^{32}$ is a color feature and its first three dimensions correspond to RGB, as in [4]. We perform volume rendering [14, 18] to obtain two feature maps I_h and I_p via the following equation:

$$I(\boldsymbol{r}) = \sum_{i=1}^{N} t_i(1 - \exp(-\sigma_i\delta_i))\boldsymbol{c}_i, t_i = \exp(-\sum_{j=1}^{i-1} \sigma_j\delta_j), \tag{III}$$

where i is the point index along ray $\boldsymbol{r}$ from near to far, and δ denotes adjacent point distance. We blend the two feature maps to a single foreground feature map I_f via the rendered FLAME mask at the same view point:

$$I_f = I_h \odot (1 - M_p) + I_p \odot M_p, \tag{IV}$$

where $\odot$ is element-wise multiplication and M_p is the mask of eyes and inner mouth obtained via rasterization of the FLAME mesh $\boldsymbol{m}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$. Similarly, we can obtain the foreground opacity image I_{opa} by setting $\boldsymbol{c}_i$ of all points to 1 in Eq. (III). Then, we fused the foreground with a 2D background feature map I_{bg} generated by another StyleGAN2:

$$I_{lr} = I_f \odot I_{opa} + I_{bg} \odot (1 - I_{opa}). \tag{V}$$

Finally, the obtained low-resolution feature map I_{lr} is sent into a 2D super-resolution module [4] to synthesize the final image I.

Learning strategy. We adopt the recent 3D-aware GAN training framework [4, 22] to learn GenHead using monocular real images. During training, we randomly sample $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$ and camera pose $\boldsymbol{\theta}$ extracted from the training set, as well as noise code $\boldsymbol{z}$ from normal distribution, and enforce the GenHead model G to generate a corresponding image I. An extra discriminator D then takes the generated image I as well as a real one $\bar{I}$ from the training set to conduct image-level adversarial learning [13, 15]:

$$\begin{aligned}\mathcal{L}_{adv} = &\mathbb{E}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{z},\boldsymbol{\theta}}[f(\mathrm{D}(I_{cat}))] \\ &+ \mathbb{E}_{\bar{I}\sim p_{real}}[f(-\mathrm{D}(\bar{I}_{cat})) + \lambda\|\nabla\mathrm{D}(\bar{I}_{cat})\|^2],\end{aligned} \tag{VI}$$

where $f(u) = \log(1 + \exp(u))$ is the Softplus function and $|\nabla\mathrm{D}(\cdot))\|^2$ denotes the R1 regularization [17]. $I_{cat} =$

$[I, I_{lr}, I_{opa}, U]$ is a concatenation of the synthesized images I, I_{lr}, the opacity image I_{opa}, and a rasterized correspondence map from $\boldsymbol{m}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$ similarly as in [22]. $\bar{I}_{cat}$ is the corresponding concatenation of I_{cat}, where $\bar{I}_{opa}$ is predicted by [6] and $\bar{U}$ is obtained from a reconstructed mesh using [3, 10] and an extra landmark-based optimization step. The additional concatenation of the opacity images and the correspondence maps help with better foreground-background separation and more accurate deformation control.

Besides, we introduce a part-region density regularization to encourage the GenHead to leverage T_p instead of T_h for generating eyes and inner mouth:

$$\mathcal{L}_{part} = \sum_{\Pi(\boldsymbol{x}) \in M_p} \sigma_h(\boldsymbol{x}), \tag{VII}$$

where $\Pi(\boldsymbol{x})$ is the 2D projection of point $\boldsymbol{x}$ in the observation space, and σ_h is the corresponding volume density obtained from T_h.

Data preprocessing. We re-align the FFHQ [15] dataset to ensure that the heads have nearly identical scales and are centered in the images. Specifically, we detect facial landmarks of all images using [30]. Then, we re-scale and re-center all heads in the images by performing similarity transforms computed between the detected landmarks and the canonical landmarks of BFM [19]. Finally, we center-crop all images and resize them to a resolution of 512^2. Note that we preserve roll angles of the heads instead of eliminating them as done in [4, 15]. We use Deep3DRecon [10] to extract BFM coefficients from the images and transfer them to FLAME codes via [3]. To improve 3D-to-2D alignment, we conduct extra landmark-based optimization to update the 3D shapes, expressions, eye rotations, and 3D translations. The optimized FLAME codes as well as the camera parameters are used for image synthesis and correspondence map rasterization during training. The camera intrinsics are set identical across all images. In addition, we re-balance the pose distribution of FFHQ based on the estimated head rotations. We duplicate the images by factors of 2, 4, 8, and 16 for those with yaw angles in ranges of $15^\circ \sim 30^\circ$, $30^\circ \sim 45^\circ$, $45^\circ \sim 60^\circ$, and larger than 60°, respectively. We also flip all images and extract the corresponding FLAME parameters. This lead to 210K training images in total compared to the original FFHQ with 70K images.

More training details. We randomly sample $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\theta}$ extracted from a same image, and combine them with a random noise $\boldsymbol{z}$. We perform volume rendering at a resolution of 64^2, and use hirearchical sampling strategy [4, 18] with 48 coarse sampling points and 48 fine points. We train G_{ca} and D via $\mathcal{L}_{adv}$ and $\mathcal{L}_{part}$ to see 25M images in total. The balancing weights for the two losses are set to 1 and 10, respectively. We use Adam optimizer [16] with $(\beta_1, \beta_2) = (0, 0.99)$ and learning rates of 0.0025 and 0.002 for the generator and the discriminator, respectively, and set the batch size to 32. Experiments are conducted on 4 Tesla A100 GPUs with 40GB memory, and the training takes around 2 weeks.

II.2. 4D Data Synthesis

In this section, we describe the data synthesis details for training the 4D head reconstruction pipeline.

Specifically, we follow the data preprocessing procedure described in Sec. II.1 to extract $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\theta})$ from images in FFHQ and VFHQ. For the dynamic data, we sample $\boldsymbol{\alpha}$ extracted from the FFHQ images and $(\boldsymbol{\beta}, \boldsymbol{\gamma})$ from the VFHQ images. For the static data, $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$ are sampled from both the FFHQ and VFHQ images. Note that for γ_{neck}, we sample it from a manually-defined distribution, with pitch in $[-0.2, 0.2]$ rad, yaw in $[-0.5, 0.5]$ rad, and roll in $[-0.1, 0.1]$ rad. For the camera pose $\boldsymbol{\theta}$, we also sample it from a pre-defined uniform distribution that covers most of the camera parameters estimated from FFHQ, with pitch in $[-0.25, 0.65]$ rad, yaw in $[-0.78, 0.78]$ rad, and roll in $[-0.25, 0.25]$ rad. The camera radius are uniformly sampled from $[3.65, 4.45]$, and the camera look-at position from $[-0.01, 0.01] \times [-0.01, 0.01] \times [0.02, 0.04]$. We use fixed camera intrinsics similarly as in GenHead, with a field of view (FoV) equal to 12°.

For a certain identity $(\boldsymbol{\alpha}, \boldsymbol{z})$ with different motions and camera poses, we use the same $\boldsymbol{z}$ to generate the background image. We maintain the intermediate outputs as additional supervisions as described in the main paper. For the sampled triplane features $\bar{T}(\boldsymbol{x})$, we randomly choose 4000 coarse sampling points during the rendering process and record their features $\boldsymbol{f}_h$ from the triplanes T_h. Besides, for all synthesized images $\bar{I}_{re}$, we use an average $\mathcal{W}$ space [1] vector for the modulated convolutional layers in the 2D super-resolution module. Visualizations of the synthetic data can be found in Sec. III.2.

II.3. Animatable Triplane Reconstructor

Canonicalization and reenactment module. The canonicalization and reenactment module Φ consists of a de-expression module Φ_{de} and a reenactment module Φ_{re} sharing the same structure. They each has four transformer blocks with a cross-attention layer, a self-attention layer, and an MLP. An extra MLP is utilized to expand the spatial dimension of the motion feature $\boldsymbol{v}$ for computing the cross-attentions, as shown in Fig. 3 in the main paper. A detailed structure of Φ can be found in Fig. XVI.

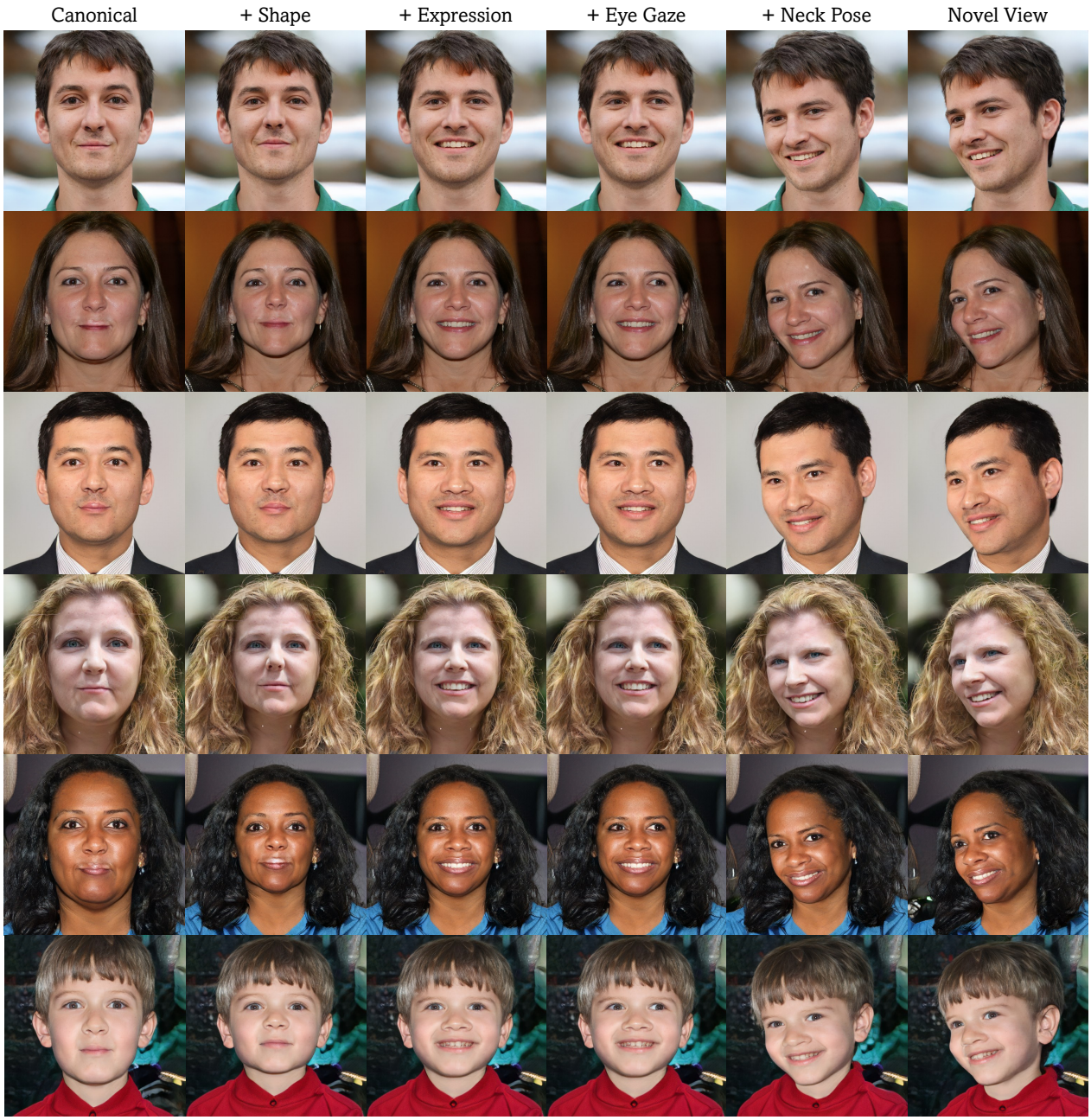

Figure III. Head images synthesized by GenHead. We can generate diverse virtual identities and support individual control over head shapes, expressions, eye gazes, neck poses, and camera poses. **Best viewed with zoom-in.**

Motion feature. We utilize the motion features from [25] as input to Φ. More specifically, we use a concatenation of the features from [25]'s expression encoder E_{exp}, lip encoder E_{lip}, and eye encoder E_{eye}. This leads to a motion vector $\boldsymbol{v}$ of dimension $30 + 512 + 6 = 548$.

Background prediction. We leverage a U-Net to predict the background feature map I_{bg} from the input image. The structure of it is illustrated in Fig. XVII.

Image rendering. We follow a similar procedure as in Sec. II.1 for image synthesis. Specifically, a foreground feature map I_f is rendered from the reconstructed triplane T via Eq. (III), and fused with the predicted background I_{bg} from the U-Net via Eq. (V). Note that for the 4D head synthesis pipeline, we do not use part-wise triplanes as in GenHead but a single triplane T to represent the whole head region and render the corresponding images.

More training details. We train the animatable triplane reconstructor Ψ using the synthetic data described in Sec. II.2 and Sec. 3.2 in the main paper, as well as the training objective in Sec. 3.4 in the main paper. We initialize the projection weights of all cross-attention layers to zeros, and use the pre-trained weights from GenHead for the radiance decoding MLP and 2D super-resolution module in the renderer $\mathcal{R}$, as well as the discriminator D. The balancing weights for each loss term in Eq. (4) in the main paper are set to 1, 1, 0.1, 1, 0.3, 1, and 0.01 for $\mathcal{L}_{re}$, $\mathcal{L}_{f}$, $\mathcal{L}_{tri}$, $\mathcal{L}_{depth}$, $\mathcal{L}_{opa}$, $\mathcal{L}_{id}$, and $\mathcal{L}_{adv}$, respectively. During the first 1000K images, we do not use $\mathcal{L}_{adv}$ and fix the network parameters inside the renderer $\mathcal{R}$. After seeing 1000K images, we activate $\mathcal{L}_{adv}$ and unfrozen the trainable parameters in $\mathcal{R}$. We discard $\mathcal{L}_{f}$, $\mathcal{L}_{tri}$, and $\mathcal{L}_{depth}$ at this stage. Following GenHead, we perform volume rendering with 48 coarse sampling points and 48 fine points per ray. We use a volume rendering resolution of 64^2 at the first 1000K images, and gradually increase the resolution to 128^2 at the next 1000K images. During the entire training process, we use a fixed average $\mathcal{W}$ space vector from GenHead for the 2D super-resolution module as in Sec. II.2.

We train Ψ and $\mathcal{R}$ to see 12M images in total. We use Adam optimizer with $(\beta_1, \beta_2) = (0.9, 0.999)$ and a learning rate of $1e-4$ for all the networks. The batch size is set to 32, half of which are dynamic data and half of which are static data. The model is trained with 8 Tesla A100 GPUs with 80GB memory for 10 days.

III. More Results

III.1. Evaluation of GenHead

Controllable head image generation. Figure III shows the controllable head image generation results of GenHead. We start from canonical appearances synthesized by random noise $\boldsymbol{z}$ with an average shape and neutral expression (*i.e.*, $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma} = \mathbf{0}$). Then, we introduce shape variations as well as expression and pose control to different canonical heads. As shown, GenHead supports individual control over head shape, expression, eye gaze, neck pose, and camera pose. The synthesized images are of high photorealism and can be directly used as training data to facilitate our one-shot 4D head reconstruction pipeline.

Shape-aware canonical appearance. Figure IV shows the synthesized canonical appearances given shape codes $\boldsymbol{\alpha}$ extracted from different source images as condition. In each row, we fix the shape code and vary the random noise $\boldsymbol{z}$, and compute an image-space average appearance as well. As shown, the distribution of the canonical appearance is influenced by the given shape code. In the ablation study, we show that this strategy largely improves the image generation quality in terms of FID without sacrificing the controllability. Note that at inference time, we can use different shape codes for the appearance and the deformation field for more diverse virtual head image generation.

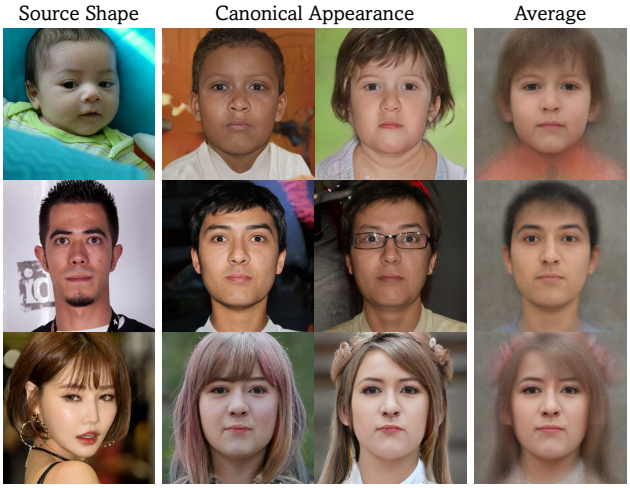

Figure IV. Generated canonical appearances with shape codes from different source images.

Comparisons with previous head GANs. We compare GenHead with existing 3D head GANs: DiscoFaceGAN [11], AniFaceGAN [27], 3DFaceShop [23], GNARF [2], OmniAvatar [29], Next3D [22], and EG3D [4]. Since GNARF and OmniAvatar do not release their codes and models for head generation, we only compare with their reported FID. For a fair comparison, we re-train GenHead using FFHQ images aligned by [4] instead of our new alignment described in Sec. II.1. We train the model at a resolution of 256^2 for efficiency. The following ablation study also adopts the same configuration.

Table I shows the quantitative results. For image synthesis quality, we measure the FID score between 50K generated images and all available real images in the training set. For control accuracy, we measure APD, AED, Landmark Distance (LMD), Average Shape Distance (ASD), and Average Shape Variance (ASV). For APD, we compute the distance between input camera angles and those reconstructed by [10] using 1000 generated images. For AED, we manually extract expressions from 30 reference images with typical and distinct expressions, and combine them with 50 generated appearances for image synthesis. We leverage EMOCA [9] to compute AED between the reference images and the synthesized ones. For ASD and LMD, we measure the vertex and landmark distances between input shapes and those reconstructed from synthesized images using EMOCA, respectively. We randomly sample 500 shape codes from the training set, and generate 10 different appearances for each shape. Since DiscoFaceGAN, 3DFaceShop, and AniFaceGAN require BFM

Table I. Comparisons between head GANs on controllable items, generation quality&diversity (Q. & Div.), and control accuracy.

Method	Independent Control Item					Q. & Div.	Control Accuracy				
	Exp.	Neck	Gaze	Teeth	BG	FID ↓	APD ↓	AED ↓	LMD ↓	ASD ↓	ASV ↓
DiscoFaceGAN [11]	✓					12.9	0.031	0.829	-	-	66.6
3DFaceShop [23]	✓				✓	21.7	0.022	0.865	-	-	7.3
AniFaceGAN [27]	✓					20.1	0.039	0.687	-	-	19.9
GNARF [2]	✓					6.6	-	-	-	-	-
OmniAvatar [29]	✓	✓			✓	5.8	-	-	-	-	-
Next3D [22]	✓		✓			3.9	0.029	0.868	23.1	20.8	18.6
A Baseline EG3D [4]						4.8	N/A	N/A	N/A	N/A	N/A
B + Deformation & BG	✓	✓			✓	6.5	0.031	0.783	9.6	9.1	9.3
C + Correspondence Map	✓	✓			✓	7.8	0.030	0.757	10.5	9.9	5.5
D + Opacity Image	✓	✓			✓	9.5	0.030	0.722	9.1	8.7	8.1
E + Shape Condition	✓	✓			✓	4.7	0.032	0.700	9.5	9.0	6.4
F + Part Model (Ours)	✓	✓	✓	✓	✓	4.6	0.028	0.699	9.1	8.5	6.1

Table II. Comparison on 3D consistency of different head GANs using the evaluation metrics of [28].

Method	PSNR ↑	SSIM ↑
EG3D [4]	34.0	0.928
Next3D [22]	34.5	0.941
Ours	34.2	0.940

shapes as condition which differ from our FLAME topology, we only compare with Next3D for these two metrics. For ASV, we calculate the vertex variance between reconstructed shapes of different images synthesized with the same shape code. Similarly, we sample 500 shape codes and 10 different appearances for each shape.

As shown, GenHead achieves the best overall control accuracy, with competitive image generation quality. What's more, our method supports full control over expression, neck pose, eye gaze, relative motions between lips and teeth, and background separation, which cannot be achieved by previous methods. Figure V shows a comparison between GenHead and Next3D for separate teeth control. Ideally, lip motions should not influence the position of the upper teeth. However, the upper teeth synthesized by Next3D move with the lip variations. By contrast, our method maintains the position of the upper teeth during expression changes, which is more consistent with reality.

In Tab. II, we further compare the 3D consistency of our method with EG3D and Next3D. We use the evaluation metric from GRAM-HD [28] which measures the reconstruction fidelity of a 3D reconstruction method NeuS [26] on multi-view images generated by different generators. As shown, our method yields comparable results with the two baselines. The high 3D consistency of GenHead guarantees reasonable synthetic 4D data for learning the subsequent one-shot 4D head synthesizer. For further improvement of the 3D consistency, a possible way is to use the

Figure V. GenHead better captures the relative movements between lips and teeth, where the upper teeth should stay steady under expression changes.

3D-to-2D imitative strategy proposed by Mimic3D [5] to generate high-resolution tri-planes for direct volume rendering. This way, the synthetic 4D data of GenHead will have even higher 3D consistency which can further facilitate the learning of the 4D head synthesizer.

Ablation study. We conduct ablation studies to validate different components in GenHead. ***A*** is a baseline identical to EG3D. ***B*** adds the deformation field (non-part-wise) and the background network on top of *A*. ***C*** introduces the correspondence map condition to the dual discriminator and ***D*** further introduces the opacity image condition. ***E*** leverages the shape condition for synthesizing canonical appearance. ***F*** utilizes the part-wise deformation field and canonical tri-planes which is our final configuration.

Table I and Fig. VI show the comparisons between different configurations. Naively introducing the 3D deformation (*B*) cannot achieve satisfactory motion control accuracy as indicated by the relatively higher AED, LMD, ASD and ASV, as well as the blurry image-space average of the canonical appearance in Fig. VI. Adding the correspondence map condition (*C*) improves the alignment of

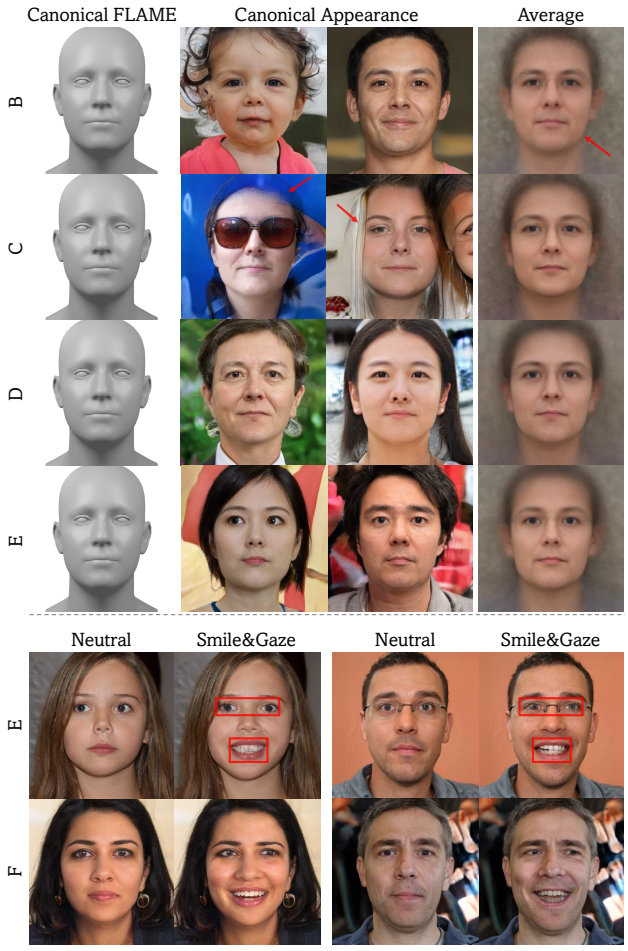

Figure VI. Ablation study. **Top:** synthesized canonical appearances and their image-space averages. **Bottom:** synthesized images under expression and gaze variations.

the canonical appearances and leads to better controllability in terms of AED and ASV, but sacrifices image quality and leads to transparent foregrounds. These artifacts also lead to inaccurate shape reconstruction results as indicated by the higher LMD and ASD. Further introducing the opacity image (*D*) resolves the transparency issue and improves the overall controllability, but leads to a further quality drop in terms of FID. Conditioning the canonical appearance on shape code (*E*) largely improves the image quality and maintains competitive control accuracy. Finally, adopting the part-wise deformation and canonical tri-planes (*F*) improves the controllability of eye gaze and the quality of inner mouth, and yields the best overall control accuracy.

III.2. Synthetic 4D Data

We showcase our synthetic data for training the one-shot 4D head synthesis pipeline in Fig. X and XI. As shown, the dynamic data contain virtual identities each with different motions and camera poses, while the static data contain pose variations only. The static data have a wider range of identity distribution to enhance the model's generalizability. Backgrounds are fixed for each identity to facilitate learning the foreground-background separation.

Figure VII. Reconstruction results by switching off all cross-attention layers in Φ_{de} and Φ_{re}.

III.3. One-Shot 4D Head Synthesis

We provide additional one-shot 4D head synthesis results in Fig. XII and XIII. Our method can faithfully reconstruct head avatars from the given portraits and control their expressions and poses for photorealisitic image synthesis.

III.4. Comparisons with the Prior Art

Figure XIV and XV shows more visual comparisons on one-shot head reenactment with previous methods. Our method yields the best visual quality, and can well preserve the identities and geometries of the source images under large pose variations compared to the alternatives.

III.5. Φ_{de} in Different Configurations

Figure VII shows the reconstruction results of an input image by switching off all cross-attention layers in Φ_{de} and Φ_{re} in different configurations (corresponding to the ablations in the main paper). Without cross-attentions, config. ***A*** and ***C*** still canonicalize expression of the input, which indicates that the self-attention and feed-forward layers in Φ_{de} take the responsibility of expression neutralization. By contrast, ours handles expression neutralization only through the cross-attentions thus the reconstructed expression is unchanged under this circumstance.

III.6. Out-of-Distribution Results

We show reenactment results on out-of-distribution subjects in Fig. VIII, where the sources are generated from Stable Diffusion [20] and the drivings from Unity Engine. Our method produces reasonable results on these cases.

IV. Discussions

IV.1. Limitations and Future Works

While our method can synthesize high-fidelity 4D head avatar at a single shot, it still has some limitations.

Our method cannot well handle complex accessories and makeups as shown in Fig. IX. It also struggles to reconstruct high-frequency details in the background (*e.g.*, last row in

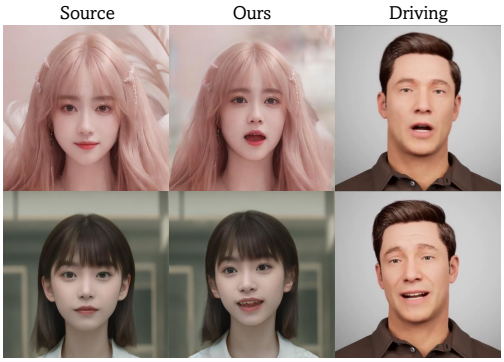

Figure VIII. Reenactment results on out-of-distribution subjects.

Figure IX. Limitations of our method. It can produce inferior results for input with heavy makeups and large viewing angles.

Fig. XII). We believe this problem can be mitigated by increasing the volume rendering resolution to allow for more intricate information flow. This would also help with the texture flickering issue brought by the 2D super-resolution module. Learning with synthetic data of more diverse appearance can also be helpful.

When the input image is nearly profile with large yaw angles, our method can produce inferior results due to out-of-distribution issue (see Fig. IX). We are also aware of artifacts under certain expressions such as eye blink, as the GenHead model for data synthesis is learned on FFHQ dataset with relatively less images of closed eyes. It is possible to leverage data with more diverse expressions and poses to improve the model's generalizability.

Currently, the synthetic data from GenHead relies on 3DMM for expression control which can be less vivid compared to that of real data, and thus restricts the motion control ability of our method. Besides, the data synthesis process requires training an animatable 3D-aware GAN in advance which is also challenging and can suffer from loss of modality issue of GAN. Learning on 4D synthetic data also encounters more severe overfitting issue compared to learning on static 3D data as in [24]. Therefore, synthesizing 4D data of better quality and diversity to facilitate the one-shot reconstruction pipeline is a key problem to be solved. Alternatively, it is worth exploring an effective way to incorporate real data and 3D priors into an end-to-end training framework. Apart from that, extending the current pipeline to support few-shot cases is also an important direction.

IV.2. Ethics Consideration

The goal of this paper is to create animatable head avatars for virtual communications. However, without a proper supervision, it can be misused for creating deceptive contents to people. We do not condone any such harmful behavior. Incorporation of advanced deepfake detectors [7, 8, 21] is a possible way to prevent the potential misuse.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019. 3

[2] Alexander Bergman, Petr Kellnhofer, Wang Yifan, Eric Chan, David Lindell, and Gordon Wetzstein. Generative neural articulated radiance fields. *Advances in Neural Information Processing Systems*, 35:19900–19916, 2022. 1, 5, 6

[3] Timo Bolkart. Bfm to flame. https://github.com/TimoBolkart/BFM_to_FLAME, 2020. 3

[4] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2, 3, 5, 6

[5] Xingyu Chen, Yu Deng, and Baoyuan Wang. Mimic3d: Thriving 3d-aware gans via 3d-to-2d imitation. *arXiv preprint arXiv:2303.09036*, 2023. 6

[6] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021. 3

[7] Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. On the detection of synthetic images generated by diffusion models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 8

[8] Davide Cozzolino, Andreas Rössler, Justus Thies, Matthias Nießner, and Luisa Verdoliva. Id-reveal: Identity-aware deepfake video detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15108–15117, 2021. 8

[9] Radek Daněček, Michael J Black, and Timo Bolkart. Emoca: Emotion driven monocular face capture and animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20311–20322, 2022. 5

[10] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 3, 5

[11] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3D imitative-contrastive learning. In *IEEE/CVF Con-*

ference on Computer Vision and Pattern Recognition, pages 5154–5163, 2020. 5, 6

[12] Stephan J Garbin, Marek Kowalski, Virginia Estellers, Stanislaw Szymanowicz, Shideh Rezaeifar, Jingjing Shen, Matthew Johnson, and Julien Valentin. Voltemorph: Real-time, controllable and generalisable animation of volumetric representations. *arXiv preprint arXiv:2208.00949*, 2022. 2

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014. 2

[14] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH*, 18(3):165–174, 1984. 2

[15] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 2, 3

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 3

[17] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International Conference on Machine Learning*, pages 3481–3490, 2018. 2

[18] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020. 2, 3

[19] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3D face model for pose and illumination invariant face recognition. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 296–301, 2009. 3

[20] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 7

[21] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *ICCV*, page 5, 2019. 8

[22] Jingxiang Sun, Xuan Wang, Lizhen Wang, Xiaoyu Li, Yong Zhang, Hongwen Zhang, and Yebin Liu. Next3d: Generative neural texture rasterization for 3d-aware head avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20991–21002, 2023. 2, 3, 5, 6

[23] Junshu Tang, Bo Zhang, Binxin Yang, Ting Zhang, Dong Chen, Lizhuang Ma, and Fang Wen. Explicitly controllable 3d-aware portrait generation. *arXiv preprint arXiv:2209.05434*, 2022. 5, 6

[24] Alex Trevithick, Matthew Chan, Michael Stengel, Eric Chan, Chao Liu, Zhiding Yu, Sameh Khamis, Manmohan Chandraker, Ravi Ramamoorthi, and Koki Nagano. Real-time radiance fields for single-image portrait view synthesis. *ACM Transactions on Graphics (TOG)*, 42(4):1–15, 2023. 8

[25] Duomin Wang, Yu Deng, Zixin Yin, Heung-Yeung Shum, and Baoyuan Wang. Progressive disentangled representation learning for fine-grained controllable talking head synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17979–17989, 2023. 4

[26] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 6

[27] Yue Wu, Yu Deng, Jiaolong Yang, Fangyun Wei, Qifeng Chen, and Xin Tong. Anifacegan: Animatable 3d-aware face image generation for video avatars. *arXiv preprint arXiv:2210.06465*, 2022. 5, 6

[28] Jianfeng Xiang, Jiaolong Yang, Yu Deng, and Xin Tong. Gram-hd: 3d-consistent image generation at high resolution with generative radiance manifolds. *arXiv preprint arXiv:2206.07255*, 2022. 6

[29] Hongyi Xu, Guoxian Song, Zihang Jiang, Jianfeng Zhang, Yichun Shi, Jing Liu, Wanchun Ma, Jiashi Feng, and Linjie Luo. Omniavatar: Geometry-guided controllable 3d head synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12814–12824, 2023. 5, 6

[30] Baosheng Yu and Dacheng Tao. Heatmap regression via randomized rounding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8276–8289, 2021. 3

Figure X. Synthesized dynamic data from GenHead for learning one-shot 4D head synthesis.

Figure XI. Synthesized static data from GenHead for learning one-shot 4D head synthesis.

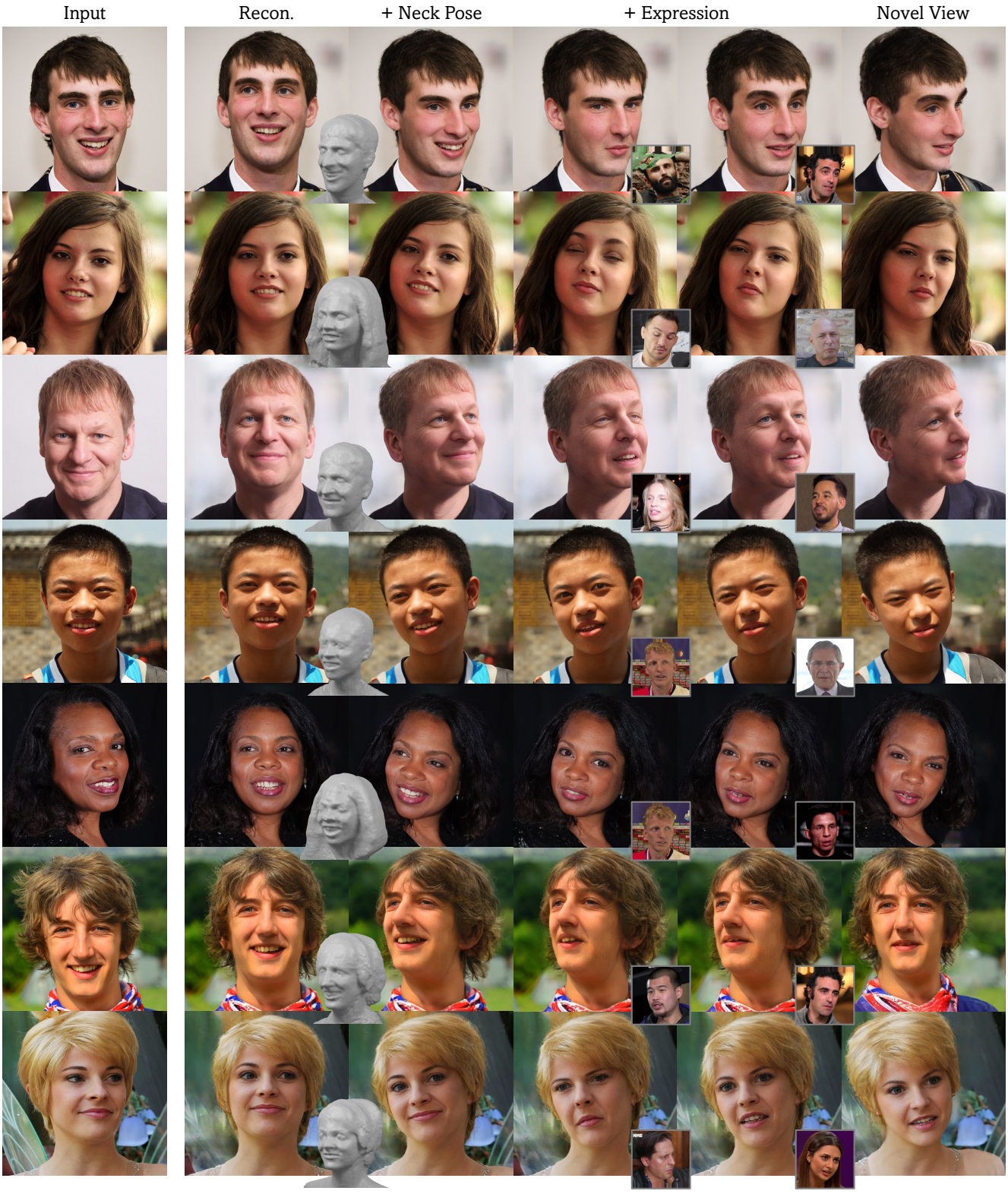

Figure XII. One-shot 4D head synthesis results by our method.

Figure XIII. One-shot 4D head synthesis results by our method.

Figure XIV. Comparison on one-shot head reenactment with previous methods. **Best viewed with zoom-in.**

Figure XV. Comparison on one-shot head reenactment with previous methods. **Best viewed with zoom-in.**

Figure XVI. Network structure of the canonicalization and reenactment module Φ.

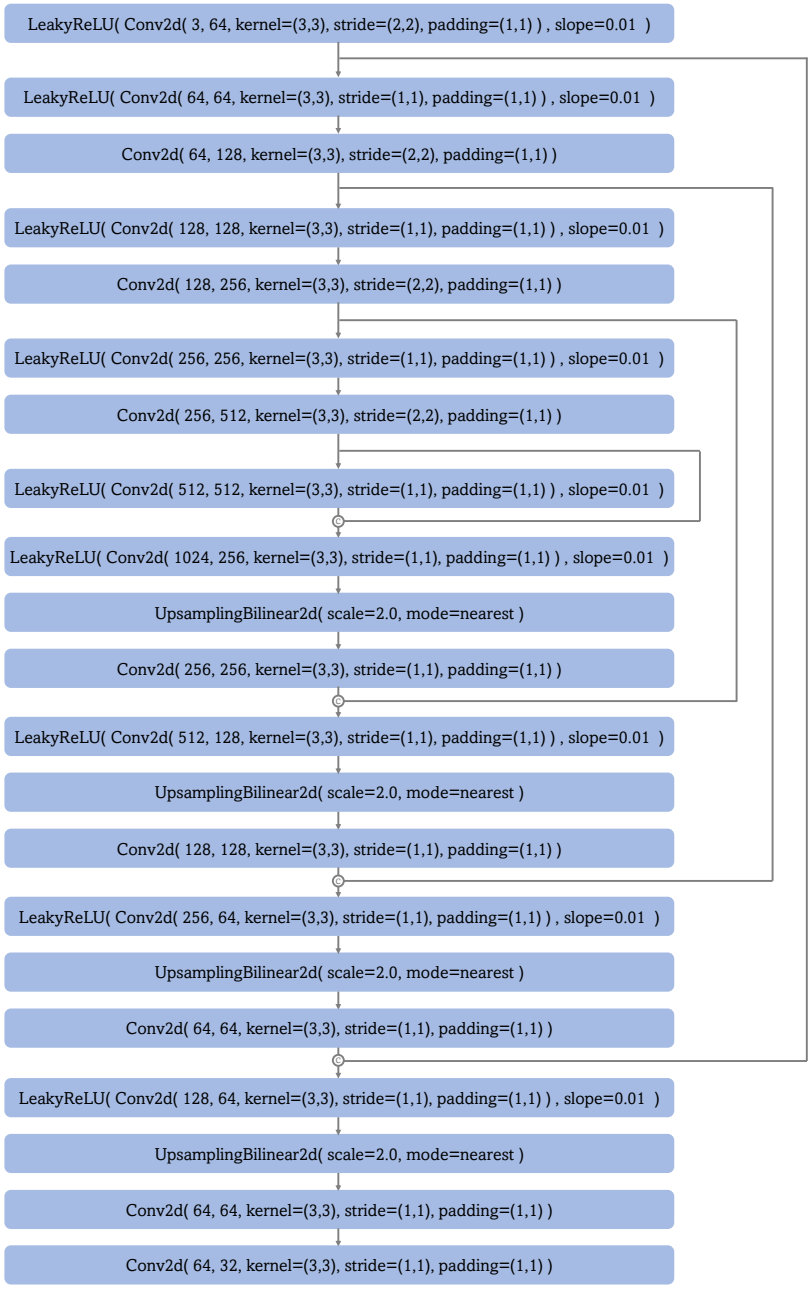

Figure XVII. Structure of the background U-Net.