CVTHead: One-shot Controllable Head Avatar with Vertex-feature Transformer

Haoyu Ma, Tong Zhang, Shanlin Sun, Xiangyi Yan, Kun Han, Xiaohui Xie
University of California, Irvine
{haoyum3, tongz27, shanlins, xiangyy4, khan7, xhx}@uci.edu

Abstract

Reconstructing personalized animatable head avatars has significant implications in the fields of AR/VR. Existing methods for achieving explicit face control of 3D Morphable Models (3DMM) typically rely on multi-view images or videos of a single subject, making the reconstruction process complex. Additionally, the traditional rendering pipeline is time-consuming, limiting real-time animation possibilities. In this paper, we introduce CVTHead, a novel approach that generates controllable neural head avatars from a single reference image using point-based neural rendering. CVTHead considers the sparse vertices of mesh as the point set and employs the proposed Vertex-feature Transformer to learn local feature descriptors for each vertex. This enables the modeling of long-range dependencies among all the vertices. Experimental results on the VoxCeleb dataset demonstrate that CVTHead achieves comparable performance to state-of-the-art graphics-based methods. Moreover, it enables efficient rendering of novel human heads with various expressions, head poses, and camera views. These attributes can be explicitly controlled using the coefficients of 3DMMs, facilitating versatile and realistic animation in real-time scenarios. Codes and pre-trained model can be found at https://github.com/HowieMa/CVTHead.

1. Introduction

Personalized head avatars play a crucial role in a wide range of applications, including AR/VR, teleconferencing, and the movie industry. Over the past few decades, there has been an extensive exploration of personalized head avatars in the fields of computer graphics and computer vision. Traditional solution [1] reconstructs a personalized mesh and texture for the source actor explicitly with 3D head scans [61, 64]. To perform full face control, 3D Morphable Models (3DMM) [4, 34] are used as a strong prior of face geometry. 3DMMs is a parametric model and uses PCA-based linear blendshapes to explicitly control face shape, expressions, texture, and head pose independently. However, 3DMM does not model the facial detail and hair region of the human face [21]. Recently, with the development of Neural Radiance Fields (NeRF) [41], reconstructing avatars with implicit models becomes popular as it can reconstruct detailed regions [3, 17, 43]. However, all these methods are subject-specific and they usually require video inputs or multi-view images of the same subject, which limits their usage in practice.

Hence, acquiring human avatars from a single image (i.e., one-shot face reenactment) becomes more and more popular [14, 31, 49, 52, 54, 59, 60, 67, 70, 74]. Given a facial image of an actor, the synthetic images can be driven by videos from other actors. A key step behind these methods is to decouple the facial appearance and motion information from the source and driven images. As a result, mesh-guided face animation has gained significant attention, primarily due to the inherent disentanglement of identity and expression offered by 3DMM. Generally, one-shot mesh-guided face animation can be roughly divided into warp-based and graphics-based. Warp-based methods [14, 65, 67, 72] employ the motion field to transfer the driving pose and expression into the source face. These methods effectively preserve fine facial details and produce high-fidelity results but only work well for a limited range of head poses. Graphic-based methods [16, 31] learn texture maps [5] from single-image and apply computer graphics pipelines to render the animated face image. Thus, it can maintain performance under large head rotations and guarantee the 3D consistency of rendered images. However, the rendering pipeline is usually computationally heavy [29], which makes efficient rendering unachievable.

Alternatively, point-based graphics [22] get rid of the surface mesh and directly use point clouds to model the 3D geometry. Later on, the point-based neural rendering techniques [2, 46, 62] augment each RGB point with a learnable neural descriptor that is interpreted by the neural renderer. The recent SMPLpix [45] further extends these techniques from static scenes to dynamic scenes and enables the efficient rendering of human body avatars under novel subject identities and human poses. Although efficient, these methods still require multi-view images with calibrated cameras

to reconstruct the accurate point cloud first. Consequently, applying these methods to one-shot face reenactment is infeasible.

In this paper, we utilize point-based neural rendering to achieve an efficient and realistic generation of head avatars from a single image. We direct utilize the sparse vertices from the FLAME head model [34] as our point set, instead of reconstructing a dense point cloud of the subject tediously. Specifically, given the vertices from pre-trained 3D face reconstruction networks [16], we learn a local feature descriptor aligned with each vertex. When learning the local descriptor of a 3D point from a 2D reference image, pixel-aligned features [11, 23, 26, 48] are a popular choice. However, these aligned features often become incorrect when the projected 2D location is occluded in the source image. To address this challenge, we propose the *Vertex-feature Transformer*. This approach treats each vertex as a query token [12] and utilizes transformers [56] to directly learn the canonical vertex features from the reference image. By incorporating a global attention mechanism, our model can capture long-range dependencies within the features of all vertices. Thus, the feature descriptor of invisible 3D points can still be reconstructed correctly. Next, we project the feature descriptor and depth of each vertex into image space and employ a UNet-like neural rendering to generate the RGB image. Since the feature descriptor is aligned with the vertices of the FLAME head model, the rendered face can be explicitly controlled by the shape, expression, and head pose coefficients. We name this end-to-end framework as CVTHead, in short for Controllable head avatar with Vertex-feature Transformer.

Our major contributions are summarized as follows:

- We propose CVTHead, a one-shot controllable head avatar framework using point-based neural rendering that can efficiently render novel human heads under novel expressions and camera views. To the best of our knowledge, this is the first work that performs point-based neural rendering from a monocular face image.
- We propose Vertex-feature Transformer to learn the vertex descriptor in canonical space from a single image with transformers, and demonstrate its superiority beyond projection methods.
- By conducting experiments on VoxCeleb1 and VoxCeleb2, we establish that our method achieves performance that is on par with the state-of-the-art approaches, while additionally improving efficiency.

2. Related Work

Mesh-guided Face Reenactment Extensive research has been conducted on employing 3DMM for the explicit animation of human face images [14, 15, 18, 31, 51, 57, 65, 67, 72]. Mesh-guided face reenactment can be divided into warp-based and graphic-based. Warping-based methods [14, 57, 65, 67] warp the source image with explicit motion fields. For example, given both source and driving meshes, Yao et al. [67] extract the motion features with Graph Convolutional Networks. HeadGAN [14] learns the dense flow field with PNCC [77] and SPADE. Face2Face$^{\rho}$ [65] calculates the motion with a set of pre-specified 3D keypoints. However, when faced with significant head rotations, the quality of these approaches drops significantly. Meanwhile, other methods [16, 21, 31] obtain the animated face images from the head mesh with the classic graphics rendering pipeline. In detail, DECA [16] simultaneously learns both the head mesh and the linear albedo subspace of the Basel Face Model [44]. To create realistic face photos, ROME [31] estimates a neural texture and offset for each vertex from the source image and renders the rigged mesh with deferred neural rendering technique [53]. Nevertheless, these methods still require the time-consuming classic differentiable rendering [47].

Neural Head Avatars Recently, several works extend NeRF [2] to model dynamic objects such as virtual avatars with implicit neural representations [3, 17, 25, 28, 43, 78]. For example, subject-dependent methods such as NerRACE [17] and RigNeRF [3] use 3DMM-guided deformation neural fields to enable control over head pose, facial expression, and viewpoints. Typically, these approaches employ an optimization-based head tracker [54] as a preprocessing step to extract accurate 3DMM coefficients from a monocular video of the subject. Meanwhile, subject-agnostic methods such as HeadNeRF [25] and MofaNeRF [78] learn the radiance fields from large-scale multi-view images [64]. When generating an avatar for a novel subject, these methods require time-consuming inverse rendering optimization to obtain the latent codes. Most recently, HiDe-NeRF [35] and OTAvatar [40] employ tri-plane representations [8] to efficiently extract multi-scale features for each query 3D point and also use volume rendering to reconstruct images. Although promising, a limitation of volumetric rendering is the necessity to sample hundreds of 3D points per ray and then feed them through the network to render a single pixel or feature patch. In contrast, our method utilizes a set of points to represent the head avatar, thereby necessitating only a single forward pass for rendering.

Neural Point-based Rendering Neural point-based rendering [2, 30, 33, 46, 62] has gained significant attention in recent years for its ability to generate high-quality images by directly rendering point clouds from static scenes. Aliev et al. [2] introduced Neural Point-Based Graphics (NPBG), which employs learnable neural descriptors to enhance each point for better rendering. Later on, NPBG++ [46] fur-

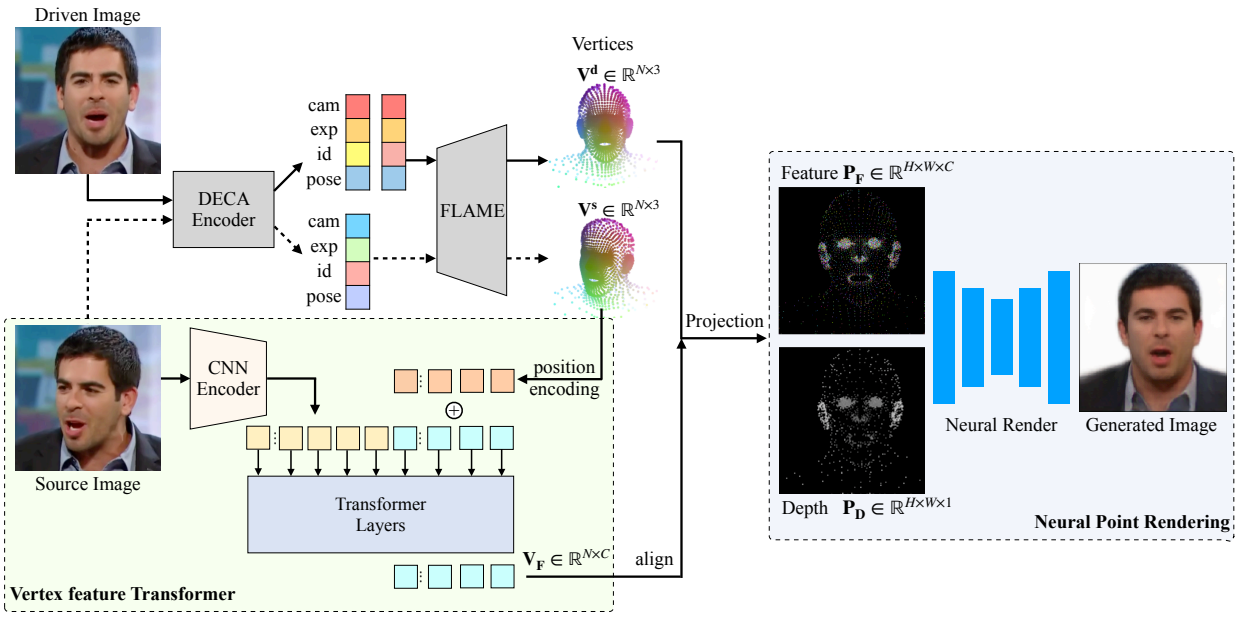

Figure 1. Overview of the CVTHead framework. We employ a pretrained face reconstruction network [16] to obtain the face mesh (Section 3.2) and utilize the proposed verte feature transformers to obtain the feature descriptor of each vertex from the source image (Section 3.3). We then consider the sparse vertex as point set and use point-based neural rendering to synthesize the image (Section 3.4).

ther predicts the descriptors with a single pass to accelerate rendering. Meanwhile, SMPLpix [45] extends point-based neural rendering to generate human avatars under the control of SMPL [38]. Although SMPLpix is a subject-agnostic method, it requires registering the SMPL model to ground truth 3D scans to obtain the RGB color of each vertex. The most recent PointAvatar [76] models the human head as an explicit canonical point cloud and continuous deformation to create realistic and relightable head avatars. However, it is still subject-dependent and requires a video caption of the subject to train the model. Our method also employs neural point-based rendering but simplifies the setting as we only require a single image of the novel subject.

Transformers in Mesh Over the past few years, transformers [56] have made significant progress in many computer vision tasks [7, 12, 39, 55, 63, 66]. There are a few works that also apply transformers to mesh data [9, 13, 36, 37, 68, 75]. In detail, METRO [36] apply transformers to predict the mesh coordinates and 3D joints simultaneously of the human body [38]. Mesh Graphormer [37] further utilizes the topology of the mesh with graph convolution to improve the mesh reconstruction. Both works consider each vertex as a query token and use transformers to learn the non-local relationships among vertices. In our work, we also use transformers to learn the correspondence among vertex features.

3. Methodology

3.1. Overview

Fig 1 illustrates the overall framework of our CVTHead. Given both source image $\mathbf{I^s}$ and driven image $\mathbf{I^d}$, we utilize a pre-trained face reconstruction model [16] to obtain the source and driven vertex coordinates $\mathbf{V^s}$ and $\mathbf{V^d} \in \mathbb{R}^{N\times3}$ of the FLAME model [34] (Sec. 3.2), where N is the number of vertices in FLAME model. Simultaneously, we employ the proposed vertex feature transformer to learn the feature descriptor for all vertices in canonical space $\mathbf{V_F} \in \mathbb{R}^{N\times C}$ from the source image (Sec. 3.3), where C is the number of channels of feature descriptor. Then we project driven vertices and their corresponding feature descriptors onto the vertex feature image $\mathbf{P_F^d} \in \mathbb{R}^{H\times W\times C}$ and the depth image $\mathbf{P_D^d} \in \mathbb{R}^{H\times W\times1}$, where H and W is the height and width of the original image. Next, we conduct neural rendering with a U-Net $\mathcal{G}(\cdot)$ to generate the synthetic image $\hat{\mathbf{I}} = \mathcal{G}(\mathbf{P_F}, \mathbf{P_D}) \in \mathbb{R}^{H\times W\times3}$ (Sec. 3.4). Our framework enables end-to-end training, allowing the entire process to be optimized jointly. During inference, our system enables the rendered image to be animated with novel shapes, expressions, head poses, and viewpoints by manipulating the FLAME parameters. This flexibility allows for the generation of diverse and customizable head avatars.

3.2. Head Mesh Reconstruction

FLAME [34] is a parametric 3D head model with $N = 5023$ vertices. It encompasses a mean template $V_b \in \mathbb{R}^{N\times3}$,

along with shape blendshapes $\mathcal{S} \in \mathbb{R}^{N\times 3\times L}$, and expression blendshapes $\mathcal{E} \in \mathbb{R}^{N\times 3\times K}$. These blendshapes are derived from a vast collection of 4D scans of human heads, allowing FLAME to capture a wide range of facial variations. Given parameters of facial identity $\beta \in \mathbb{R}^L$, expression $\phi \in \mathbb{R}^K$ and pose $\theta \in \mathbb{R}^{3k+3}$ (with $k = 4$ joints for neck, jaw, and eyeballs), FLAME first apply β and ϕ to corresponding blendshapes, resulting in modified vertex positions. Next, the linear blend skinning (LBS) technique $W(\cdot,\cdot)$ is employed to rotate the vertices based on θ. The final reconstruction of FLAME in world coordinates is calculated by:

$$M(\beta,\phi,\theta) = W(V_b + \mathcal{S}\beta + \mathcal{E}\phi, \theta) \in \mathbb{R}^{3n} \quad (1)$$

We employ the pre-trained DECA [16] $f_D(\cdot)$ to obtain β,ϕ,θ and camera parameters c from both source images and driven images with a single forward, i.e, $\beta^s, \phi^s, \theta^s, c^s = f_D(\mathbf{I^s})$ and $\beta^d, \phi^d, \theta^d, c^d = f_D(\mathbf{I^d})$. We also obtain the deformation of hair and shoulder regions from the source image with the pre-trained linear deformation model $f_H(\cdot)$ [31] to refine the vertices locations. Then we obtain the driven vertex coordinates by

$$\mathbf{V^d} = M(\beta^s, \phi^d, \theta^d) + f_H(\mathbf{I^s}) \in \mathbb{R}^{N\times 3} \quad (2)$$

3.3. Vertex-feature Transformer

Motivations In previous approaches that utilize pixel-aligned features [11,48], the feature descriptor of a given 3D point is determined by the feature located at its corresponding 2D projection. In detail, given the 3D point $\mathbf{k}^s \in \mathbf{V^s}$, we project it into the 2D image space by $(u^s, v^s, d^s) = \Pi(\mathbf{k}^s, c_s)$, where $\Pi(\cdot)$ represents the orthographic projection function and c_s is the camera parameters of the reference image obtained from pre-trained DECA. The descriptor of $\mathbf{k}^s$ is defined as $I'[u^s, v^s]$, where I' is the 2D feature map of the source image. However, these methods have several limitations. First, it requires accurate mesh reconstruction to locate the correct 2D pixels. Moreover, when the point is invisible, the feature at the 2D projection cannot represent the real features of that point. For instance, if the ear is occluded by the face, the projection may result in capturing features from the eye or nose instead. As a result, relying solely on the feature at the 2D projection can lead to incomplete or misleading feature descriptors.

Vertex Feature as Tokens To tackle the aforementioned problem, we propose a solution wherein we treat each vertex as an individual query token and leverage the attention mechanism of transformers to acquire its corresponding features from the image feature tokens. This approach avoids the need for a fixed 2D projection and allows for more flexible learning. Specifically, we employ N learnable embedding vectors $\mathbf{X_v} \in \mathbb{R}^{N\times C'}$ to represent the feature descriptors associated with each vertex in canonical space and name it as *Vertex Tokens*, where C' is the number of channels. To further encode the location information of each vertex, we incorporate the sine positional encoding [56] to its corresponding image space coordinates (u^s, v^s) and depth d^s, denoting as $\mathbf{E^s_{uv}}$ and $\mathbf{E^s_{dep}}$, respectively. Finally, the vertex query token is defined as $\tilde{\mathbf{X}}_\mathbf{v} = \mathbf{X_v} + \mathbf{E^s_{uv}} + \mathbf{E^s_{dep}}$. On the other hand, we train a CNN encoder $\mathcal{E}(\cdot)$ to extract feature maps from the source image $\mathbf{I^s}$ and flatten the 2D features into a sequence of tokens $\mathbf{F^s} = \mathcal{E}(\mathbf{I^s}) \in \mathbb{R}^{hw\times C'}$. We also apply the 2D sine positional encodings [12] to encode spatial information, denoted as $\mathbf{E}$. Finally, the image token is defined as $\mathbf{X^s_F} = \mathbf{F^s} + \mathbf{E}$.

Transformers The input to the transformer is the concatenation of both image tokens $\mathbf{X^s_F}$ and vertex tokens $\tilde{\mathbf{X}}_\mathbf{v}$, i.e, $\mathbf{X} = [\tilde{\mathbf{X}}_\mathbf{v}, \mathbf{X^s_F}] \in \mathbb{R}^{(N+hw)\times C'}$. The standard transformer encoder layer [56] consists of alternating layers of the multi-headed self-attention (MHSA) and multi-layer perceptron (MLP). First, three linear projections are applied to transfer $\mathbf{X}$ into three matrices of equal size, namely the query $\mathbf{Q}$, the key $\mathbf{K}$, and the value $\mathbf{V}$. The self-attention is calculated by:

$$\text{SA}(\mathbf{X}) = \text{Softmax}(\frac{\mathbf{QK}^T}{\sqrt{D}})\mathbf{V}, \quad (3)$$

For MHSA, H self-attention modules are applied to $\mathbf{X}$ separately, and each of them produces an output sequence. We utilize the state of the vertex tokens at the output of the transformer encoder and employ a linear transformation to modify its dimensionality, thereby acquiring the vertex descriptor $\mathbf{V_F} \in \mathbb{R}^{N\times C}$.

The vertex-feature transformer has several benefits. Firstly, it eliminates the need for a fixed 2D projection to determine the corresponding feature for each vertex. Instead, it leverages attention mechanisms to identify the relevant feature, introducing a higher degree of flexibility. The transformer incorporates positional encoding to encode location information, further enhancing its adaptability and versatility. Additionally, the global attention mechanism of transformers facilitates long-range correspondence among all vertex features. Even when the projection of a vertex is occluded, the vertex feature can still be obtained from neighboring regions or symmetrical vertices.

3.4. Neural Vertex Rendering

Given the learned vertex feature $\mathbf{V_F}$, we further use neural point-based rendering to generate synthetic images. During the training, we use the driven vertex $\mathbf{V}^d$ to reconstruct the driven image. In detail, we first project the driven vertices $\mathbf{k}^d \in \mathbf{V^d}$ into image space with the driven camera parameter c^d, i.e., $(u^d, v^d, d^d) = \Pi(\mathbf{k}^d, c^d)$. Subsequently, we create the vertex projection features $\mathbf{P^d_F} \in \mathbb{R}^{H\times W\times C}$.

For each vertex $\mathbf{k}^d$, along with its corresponding descriptor $\mathbf{v_F} \in \mathbb{R}^C$, we assign the descriptor to location (u^d, v^d) in the vertex projection features [45]:

$$\mathbf{P_F^d}[\lfloor u^d \rfloor, \lfloor v^d \rfloor] = \mathbf{v_F} \tag{4}$$

We keep the features of the nearest vertex when two vertices are projected into the same pixel on $\mathbf{P_F^d}$. For all pixels without projection (i.e., the background pixel), we assign a constant value. Similarly, we also project the depth d^d value into a depth image P_D which satisfies $\mathbf{P_D^d}[\lfloor u^d \rfloor, \lfloor v^d \rfloor] = d^d$. Finally, we concatenate $\mathbf{P_F^d}$ and $\mathbf{P_D^d}$ and employ a U-Net $\mathcal{G}(\cdot)$ to generate the synthetic image $\hat{\mathbf{I}^d}$ as well as the binary foreground mask $\hat{\mathbf{M}^d}$, i.e.,

$$(\hat{\mathbf{I}^d}, \hat{\mathbf{M}^d}) = \mathcal{G}([\mathbf{P_F^d}, \mathbf{P_D^d}]). \tag{5}$$

3.5. Training

During the training time, we randomly sample $\mathbf{I^s}$ and $\mathbf{I^d}$ from the same video. We fixed the pre-trained DECA and only update the parameters of vertex-feature transformers and neural render. Following [31], we use the L1 loss L_{L1}, VGG perceptual loss L_{vgg} [27], face recognition loss L_{id} [6], and adversarial loss [20, 58] L_a to measure the difference between the reconstructed driven image $\hat{\mathbf{I}^d}$ and the ground truth $\mathbf{I^d}$. We use the Dice loss to match the predicted segmentation masks. The total loss is calculated by:

$$L = \lambda_{L1}L_{L1} + \lambda_{vgg}L_{vgg} + \lambda_{id}L_{id} + \lambda_{seg}L_{seg} + \lambda_a L_a \tag{6}$$

, where λ_{L1}, λ_{vgg}, λ_{id}, λ_{seg} and λ_a is the corresponding weights of each loss term.

4. Experiments

4.1. Experimental Set up

Dataset For a fair comparison with previous works, we conduct experiments on VoxCeleb1 [42] and VoxCeleb2 [10]. VoxCeleb1 contains around 20k video sequences of over 1000 actors and VoxCeleb2 contains around 150k videos of over 6000 actors. Note that, ROME [31] carefully selects a subset of around 15k high-quality video sequences from VoxCeleb2 for training and evaluation, which is not publicly available. We directly use all VoxCeleb2 videos instead. Following [49], each frame is cropped into 256×256 and normalized to $[-1, 1]$. We follow the identity-based split thus all subjects in the validation set are unseen by the model. Besides, we apply an off-the-shelf face parsing network [69] to obtain the foreground mask of each frame, which is considered as the pseudo ground truth.

Implementation Details We use the same CNN encoder $\mathcal{E}(\cdot)$ as in ROME [31], which downsamples $16\times$ of the original image. Naturally, our vertex-feature transformer is able to process arbitrary sizes of mesh. However, due to the quadratic computation complexity w.r.t. the sequence length of the transformer, it's hard to model all $N = 5023$ tokens. Thus, we use the coarse mesh of the FLAME model with $N' = 314$ tokens in our vertex-feature transformer and use the decoder of Spiralnet++ [19] to upsample the vertex features after the transformer, which serializes the neighboring vertices based on triangular meshes. Our vertex-feature transformer has 6 transformer encoder layers and the head of MHSA is set to 4. The feature dimension is set to $C' = 128$ and $C = 32$. Our model is implemented using PyTorch and optimized with the Adam optimizer [32] for a duration of 200 epochs. The learning rate is set to $1e-4$ and the batch size is set to 16. λ_{L1}, λ_{vgg}, and λ_{seg} are set to 1.0, and λ_{id} and λ_a are set to 0.1.

Metrics Following previous works [31], we evaluate our CVTHead on both self-reenactment and cross-identity reenactment. In self-reenactment, the source and driving image come from the same video. In this scenario, the driving image can be viewed as the ground truth. We use the following metrics to measure the reconstruction quality between the driving image and the synthesized results: (1)L1 loss on the masked region; (2) peak signal-to-noise ratio (PSNR); (3) learned perceptual image patch similarity (LPIPS) with pre-trained AlexNet [73], and (4) multi-scale structured similarity (MS-SSIM). In the cross-identity reenactment, the source and driven image come from different subjects. Given the source image of one subject, We random sample a different subject in the validation set as the driving image. This evaluation requires the model to fully disentangle the identity and expression information. Since ground truth is unavailable, this task can only be evaluated by some proxy metrics. In detail, we use (1) FID [24] to evaluate the image realism; (2) CSIM [71], which measures the cosine similarity of the identity embeddings from a pre-trained model between the source image and the synthesized image; and (3) image quality assessment (IQA) [50]

4.2. Results of talking-face synthesis

We first evaluate the performance of our method on talking-face synthesis. To the best of our knowledge, ROME [31] is the only method that share the same setting with our method, i.e., one-shot mesh-based face reenactment based on graphics without warping field. Thus, we mainly compare our method with ROME [31]. Besides, we also compare with warping-based methods including First-Order Motion Model (FOMM) [49] and the Bi-Layer [70].

Self-reenactment The quantitative comparison results are summarized in Table 1. It is noteworthy that our CVTHead achieves comparable performance with previous

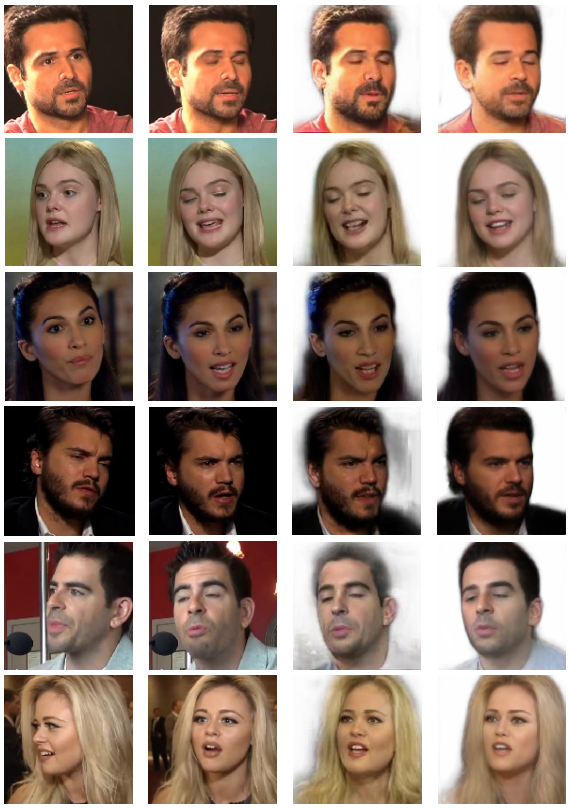

Figure 2. Qualitative comparisons of self-reenactment on VoxCeleb1. The 1st column is the source image. The 2nd column is the driving image, which can be considered as the ground truth. The 3rd column is the results from ROME, and the 4th column is the result from our CVTHead.

Dataset	VoxCeleb1			
Method	L1 ↓	PSNR ↑	LPIPS ↓	MS-SSIM ↑
FOMM [49]	0.048	22.43	0.139	0.836
Bi-Layer [70]	0.050	21.48	0.108	0.839
ROME [31]	0.048	21.13	0.116	0.838
Ours	0.041	22.09	0.111	0.840
Dataset	VoxCeleb2			
Method	L1 ↓	PSNR ↑	LPIPS ↓	MS-SSIM ↑
FOMM [49]	0.059	20.93	0.165	0.793
ROME [31]	0.050	20.75	0.117	0.834
Ours	0.042	21.37	0.119	0.841

Table 1. Results of self-reenactment on the VoxCeleb1 and VoxCeleb2 (↑ means larger is better, ↓ means smaller is better.)

methods over all metrics. Figure 2 illustrates the qualitative comparisons. We also add on the predicted soft mask as in [31] to compare its quality. The first three rows showcase scenarios with minimal head rotations and predominantly frontal source images. In such cases, both ROME and CVTHead exhibit similar performance. However, when the source images depict side views while the driving images present frontal views, ROME tends to generate images with blurry foreground masks in the occluded regions of the source image. Furthermore, ROME often renders these concealed areas in darker colors. These observations indicate that ROME struggles to effectively learn the features of occluded regions and fails to capture the correspondence between mesh vertices. Conversely, our CVTHead addresses these limitations by leveraging transformers to capture long-range dependencies among vertices. These observations suggest that ROME does not effectively learn the features of occluded regions and does not capture the correspondence between mesh vertices. In contrast, our CVTHead addresses these two issues by leveraging transformers to capture long-range dependencies among vertices.

Dataset	VoxCeleb1			
Method	FID ↓	CSIM ↑	IQA ↑	FPS ↑
FOMM [49]	39.69	0.592	37.00	64.3
Bi-Layer [70]	43.8	0.697	41.4	20.1
ROME [31]	29.23	0.717	39.11	12.9
Ours	25.78	0.675	42.26	24.3
Dataset	VoxCeleb2			
Method	FID ↓	CSIM ↑	IQA ↑	FPS ↑
FOMM [49]	61.28	0.624	36.20	64.3
ROME [31]	53.52	0.729	37.34	12.9
Ours	48.48	0.712	40.27	24.3

Table 2. Results of cross-identity reenactment.

Cross-identity Reenactment We proceed to evaluate our method in comparison to other methods for cross-identity reenactment. The quantitative comparison results are presented in Table 2. Strikingly, we achieve similar performance on the assessed metrics as ROME, indicating the effectiveness of our method in cross-identity reenactment tasks. Furthermore, we provide qualitative results in Figure 3, showcasing the ability of our method to generate images with desired expressions, head poses, and other attributes. Notably, warping-based methods usually cannot maintain the identity information such as face shape from the source image. For mesh-guided methods, ROME tends to generate lower-quality images when local regions are occluded in the source image. In contrast, our method demonstrates the capability to maintain the quality of all local regions even in such challenging scenarios.

Inference time comparison We also evaluate the inference time of each model, considering the complete duration

Figure 3. Qualitative comparisons of cross-identity reenactment on VoxCeleb1.

of 3D mesh reconstruction, the vertex deformation model, and the rendering process. To provide a comprehensive analysis, we report the average FPS (Frames per Second) based on 1000 runs performed on a single RTX 3090Ti. The results are presented in the last column of Table 2. Notably, warping-based method is more efficient as they don't need the tedious rendering and mesh reconstruction. ROME achieves a modest 12.9 FPS, while our CVTHead model achieves a significantly higher rate of 24.3 FPS. This outcome highlights the superior efficiency of the point-based neural rendering approach compared to traditional graphic-based rendering methods.

4.3. Results of 3DMM-based Face Animation

After obtaining the vertex descriptors using the vertex feature transformer, the resulting face can be further manipulated by adjusting the coefficients of the FLAME model [34], which control expression ϕ, pose θ, face shape β, and camera views c. The ability to explicitly control these coefficients enables us to generate faces of the same subject with different expressions, face shapes, and camera views, as illustrated in Figure 4. This result demonstrates that the learned feature descriptors exhibit a strong alignment with the vertices in the canonical space. Consequently, neural point-based rendering can serve as a viable alternative to traditional graphic-based rendering methods. Moreover, we intentionally select two distinct source images of the same subject. Interestingly, the generated images, utilizing vertex features from these distinct sources, exhibit a striking resemblance. This intriguing observation further underscores the effectiveness and robustness of our method.

4.4. Ablation Studies

Vertex deformation We utilize the linear deformation model $f_H(\cdot)$ from ROME [31] to deform the vertices of the hair and shoulder region. In this study, we conduct an ablation experiment where we train CVTHead without this vertex deformation module, instead employing the default FLAME mesh with a bald head. The results presented in Table 3 demonstrate that the removal of the vertex deformation ("D." in short) has only a minor impact on the performance. Interestingly, Figure 5 reveals that the synthesized images from CVTHead, both with and without vertex deformation, appear nearly identical. Furthermore, even in cases where the subject has fluffy or long hair that extends beyond the head area, the absence of vertex deformation in CVTHead does not hinder its ability to generate the correct hairstyle. These results indicate that the local vertex descriptor can effectively capture the necessary features.

Method	L1 ↓	PSNR ↑	LPIPS ↓	MS-SSIM ↑
CVTHead (w/o D.)	0.041	22.47	0.121	0.842
CVTHead	0.041	22.09	0.111	0.840

Table 3. Ablation study on the vertex deformation module. We evaluate the performance of self-reenactment on the VoxCeleb1.

Method	L1 ↓	PSNR ↑	LPIPS ↓	MS-SSIM ↑
Pixel-aligned features	0.045	21.81	0.107	0.841
CVTHead	0.041	22.09	0.111	0.840

Table 4. Ablation study on the pixel-aligned features

Pixel-aligned features In our work, we design the vertex feature transformers to learn the vertex feature. In this study, we consider the pixel-aligned features as the baseline, which project the 3D vertex into 2D and choose the corresponding pixel from the image. We follow the architecture design in S3F [11] and use a UNet-like feature extractor and sample features of each vertex with its corresponding 2D projection. Table 4 indicates that this approach yields a marginally lower PSNR, but a slightly improved LPIPS score. As shown in Figure 6, this design can maintain more detailed local features such as hair due to the high-resolution features. Thus, a slightly better LPIPS is achieved. However, when the point is occluded in the source image, the synthesized image tends to generate blur and shadow in these areas if they are visible in the driving

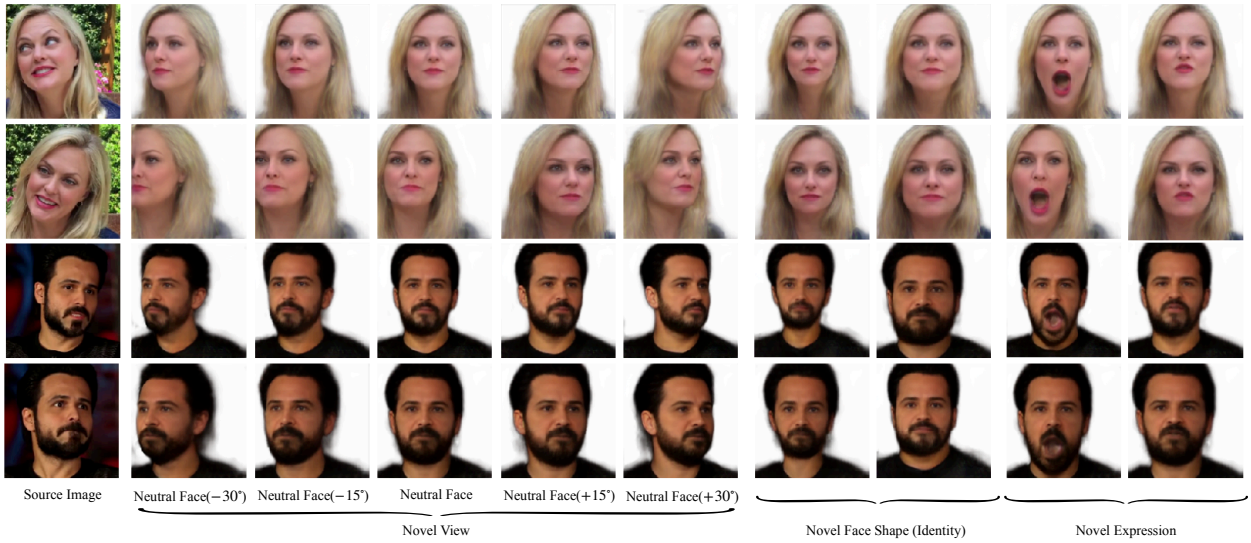

Figure 4. Qualitative results of face animation with novel views, novel face shapes (identity), and novel expressions.

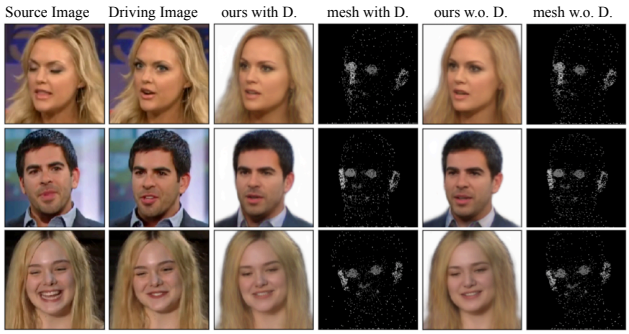

Figure 5. Ablation study of CVTHead with and without the vertex deformation model for hair and shoulder region (D. in short)

pose, which is the reason of the worse PSNR. These results suggest that pixel-aligned methods cannot capture the correct features due to the ambiguity of depth. In this case, when a large head rotation happens, this method encounters the same issue as warp-based methods.

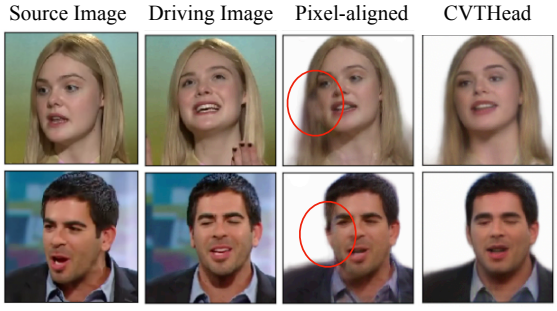

Figure 6. Ablation study of vertex features.

5. Limitations

While our method demonstrates effective face animation capabilities from a single image, one potential limitation is that the performance of our approach heavily relies on the accuracy of the 3D mesh reconstruction, specifically utilizing DECA [16] in our setup. In certain challenging scenarios, DECA may struggle to fully disentangle the shape and expression factors from the driving images. Consequently, CVTHead may generate images that differ in expressions or head poses from the intended outcome. This highlights the need for further advancements in the accuracy and robustness of 3D mesh reconstruction techniques to address such limitations.

6. Conclusion

In this paper, we propose a novel approach for generating explicitly controllable head avatars from a single reference image, utilizing point-based neural rendering. We treat the sparse vertices of the head mesh as a point set and leverage the vertex-feature transformer to learn the local feature descriptor for each vertex. Through our research, we demonstrate that point-based rendering can effectively replace traditional graphic-based rendering methods, offering enhanced efficiency. Moreover, we envision that our method can be seamlessly integrated with various generative tools, such as diffusions, to further enhance the quality of generated images and we consider this as future work.

References

[1] Oleg Alexander, Mike Rogers, William Lambeth, Jen-Yuan Chiang, Wan-Chun Ma, Chuan-Chang Wang, and Paul Debevec. The digital emily project: Achieving a photorealistic

digital actor. *IEEE Computer Graphics and Applications*, pages 20–31, 2010. 1

[2] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In *ECCV*, pages 696–712, 2020. 1, 2

[3] ShahRukh Athar, Zexiang Xu, Kalyan Sunkavalli, Eli Shechtman, and Zhixin Shu. Rignerf: Fully controllable neural 3d portraits. In *CVPR*, pages 20364–20373, 2022. 1, 2

[4] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999. 1

[5] James F Blinn and Martin E Newell. Texture and reflection in computer generated images. *Communications of the ACM*, pages 542–547, 1976. 1

[6] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 67–74. IEEE, 2018. 5

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 3

[8] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, pages 16123–16133, 2022. 2

[9] Junhyeong Cho, Kim Youwang, and Tae-Hyun Oh. Cross-attention of disentangled modalities for 3d human mesh recovery with transformers. In *ECCV*, 2022. 3

[10] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622*, 2018. 5

[11] Enric Corona, Mihai Zanfir, Thiemo Alldieck, Eduard Gabriel Bazavan, Andrei Zanfir, and Cristian Sminchisescu. Structured 3d features for reconstructing controllable avatars. In *CVPR*, pages 16954–16964, 2023. 2, 4, 7

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2, 3, 4

[13] Zhiyang Dou, Qingxuan Wu, Cheng Lin, Zeyu Cao, Qiangqiang Wu, Weilin Wan, Taku Komura, and Wenping Wang. Tore: Token reduction for efficient human mesh recovery with transformer. In *ICCV*, pages 15143–15155, 2023. 3

[14] Michail Christos Doukas, Stefanos Zafeiriou, and Viktoriia Sharmanska. Headgan: One-shot neural head synthesis and editing. In *ICCV*, pages 14398–14407, 2021. 1, 2

[15] Nikita Drobyshev, Jenya Chelishev, Taras Khakhulin, Aleksei Ivakhnenko, Victor Lempitsky, and Egor Zakharov. Megaportraits: One-shot megapixel neural head avatars. In *MM*, 2022. 2

[16] 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 (ToG)*, pages 1–13, 2021. 1, 2, 3, 4, 8

[17] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *CVPR*, pages 8649–8658, 2021. 1, 2

[18] Partha Ghosh, Pravir Singh Gupta, Roy Uziel, Anurag Ranjan, Michael J Black, and Timo Bolkart. Gif: Generative interpretable faces. In *3DV*, pages 868–878, 2020. 2

[19] Shunwang Gong, Lei Chen, Michael Bronstein, and Stefanos Zafeiriou. Spiralnet++: A fast and highly efficient mesh convolution operator. In *ICCVW*, 2019. 5

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 5

[21] Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. Neural head avatars from monocular rgb videos. In *CVPR*, pages 18653–18664, 2022. 1, 2

[22] Markus Gross and Hanspeter Pfister. *Point-based graphics*. Elsevier, 2011. 1

[23] Tong He, Yuanlu Xu, Shunsuke Saito, Stefano Soatto, and Tony Tung. Arch++: Animation-ready clothed human reconstruction revisited. In *ICCV*, pages 11046–11056, 2021. 2

[24] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 30, 2017. 5

[25] Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. Headnerf: A real-time nerf-based parametric head model. In *CVPR*, pages 20374–20384, 2022. 2

[26] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. Arch: Animatable reconstruction of clothed humans. In *CVPR*, pages 3093–3102, 2020. 2

[27] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711, 2016. 5

[28] Kacper Kania, Kwang Moo Yi, Marek Kowalski, Tomasz Trzciński, and Andrea Tagliasacchi. Conerf: Controllable neural radiance fields. In *CVPR*, pages 18623–18632, 2022. 2

[29] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey. *arXiv preprint arXiv:2006.12057*, 2020. 1

[30] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 2023. 2

[31] Taras Khakhulin, Vanessa Sklyarova, Victor Lempitsky, and Egor Zakharov. Realistic one-shot mesh-based head avatars. In *ECCV*, pages 345–362, 2022. 1, 2, 4, 5, 6, 7

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[33] Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. In *Computer Graphics Forum*, pages 29–43, 2021. 2

[34] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, pages 194–1, 2017. 1, 2, 3, 7

[35] Weichuang Li, Longhao Zhang, Dong Wang, Bin Zhao, Zhigang Wang, Mulin Chen, Bang Zhang, Zhongjian Wang, Liefeng Bo, and Xuelong Li. One-shot high-fidelity talking-head synthesis with deformable neural radiance field. In *CVPR*, 2023. 2

[36] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. *CVPR*, 2021. 3

[37] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *ICCV*, pages 12939–12948, 2021. 3

[38] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 3

[39] Haoyu Ma, Zhe Wang, Yifei Chen, Deying Kong, Liangjian Chen, Xingwei Liu, Xiangyi Yan, Hao Tang, and Xiaohui Xie. Ppt: token-pruned pose transformer for monocular and multi-view human pose estimation. In *ECCV*, 2022. 3

[40] Zhiyuan Ma, Xiangyu Zhu, Guo-Jun Qi, Zhen Lei, and Lei Zhang. Otavatar: One-shot talking face avatar with controllable tri-plane rendering. In *CVPR*, pages 16901–16910, 2023. 2

[41] 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 *ECCV*, 2020. 1

[42] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*, 2017. 5

[43] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, pages 5865–5874, 2021. 1, 2

[44] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *2009 sixth IEEE international conference on advanced video and signal based surveillance*, pages 296–301, 2009. 2

[45] Sergey Prokudin, Michael J Black, and Javier Romero. Smplpix: Neural avatars from 3d human models. In *WACV*, pages 1810–1819, 2021. 1, 3, 5

[46] Ruslan Rakhimov, Andrei-Timotei Ardelean, Victor Lempitsky, and Evgeny Burnaev. Npbg++: Accelerating neural point-based graphics. In *CVPR*, pages 15969–15979, 2022. 1, 2

[47] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020. 2

[48] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *ICCV*, pages 2304–2314, 2019. 2, 4

[49] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *NeurIPS*, 32, 2019. 1, 5, 6

[50] Shaolin Su, Qingsen Yan, Yu Zhu, Cheng Zhang, Xin Ge, Jinqiu Sun, and Yanning Zhang. Blindly assess image quality in the wild guided by a self-adaptive hyper network. In *CVPR*, pages 3667–3676, 2020. 5

[51] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *CVPR*, pages 6142–6151, 2020. 2

[52] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. Neural voice puppetry: Audio-driven facial reenactment. In *ECCV*, pages 716–731, 2020. 1

[53] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *Acm Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 2

[54] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *CVPR*, pages 2387–2395, 2016. 1, 2

[55] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *ICML*, 2021. 3

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 3, 4

[57] Qiulin Wang, Lu Zhang, and Bo Li. Safa: Structure aware face animation. In *3DV*, pages 679–688, 2021. 2

[58] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, pages 8798–8807, 2018. 5

[59] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *CVPR*, pages 10039–10049, 2021. 1

[60] Olivia Wiles, A Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *ECCV*, pages 670–686, 2018. 1

[61] Cheng-hsin Wuu, Ningyuan Zheng, Scott Ardisson, Rohan Bali, Danielle Belko, Eric Brockmeyer, Lucas Evans, Timothy Godisart, Hyowon Ha, Alexander Hypes, Taylor Koska, Steven Krenn, Stephen Lombardi, Xiaomin Luo, Kevyn McPhail, Laura Millerschoen, Michal Perdoch, Mark Pitts,

Alexander Richard, Jason Saragih, Junko Saragih, Takaaki Shiratori, Tomas Simon, Matt Stewart, Autumn Trimble, Xinshuo Weng, David Whitewolf, Chenglei Wu, Shoou-I Yu, and Yaser Sheikh. Multiface: A dataset for neural face rendering. In *arXiv*, 2022. 1

[62] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *CVPR*, pages 5438–5448, 2022. 1, 2

[63] Xiangyi Yan, Hao Tang, Shanlin Sun, Haoyu Ma, Deying Kong, and Xiaohui Xie. After-unet: Axial fusion transformer unet for medical image segmentation. In *WACV*, 2022. 3

[64] Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. Facescape: a large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In *CVPR*, pages 601–610, 2020. 1, 2

[65] Kewei Yang, Kang Chen, Daoliang Guo, Song-Hai Zhang, Yuan-Chen Guo, and Weidong Zhang. Face2face ρ: Real-time high-resolution one-shot face reenactment. In *ECCVC*, pages 55–71, 2022. 1, 2

[66] Sen Yang, Wen Heng, Gang Liu, GUOZHONG LUO, Wankou Yang, and Gang YU. Capturing the motion of every joint: 3d human pose and shape estimation with independent tokens. In *ICLR*, 2023. 3

[67] Guangming Yao, Yi Yuan, Tianjia Shao, and Kun Zhou. Mesh guided one-shot face reenactment using graph convolutional networks. In *Proceedings of the 28th ACM international conference on multimedia*, pages 1773–1781, 2020. 1, 2

[68] Yusuke Yoshiyasu. Deformable mesh transformer for 3d human mesh recovery. In *CVPR*, pages 17006–17015, 2023. 3

[69] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV*, pages 325–341, 2018. 5

[70] Egor Zakharov, Aleksei Ivakhnenko, Aliaksandra Shysheya, and Victor Lempitsky. Fast bi-layer neural synthesis of one-shot realistic head avatars. In *ECCV*, pages 524–540, 2020. 1, 5, 6

[71] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *ICCV*, pages 9459–9468, 2019. 5

[72] Bohan Zeng, Boyu Liu, Hong Li, Xuhui Liu, Jianzhuang Liu, Dapeng Chen, Wei Peng, and Baochang Zhang. FNeVR: Neural volume rendering for face animation. In *NeurIPS*, 2022. 1, 2

[73] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 5

[74] Wenxuan Zhang, Xiaodong Cun, Xuan Wang, Yong Zhang, Xi Shen, Yu Guo, Ying Shan, and Fei Wang. Sadtalker: Learning realistic 3d motion coefficients for stylized audio-driven single image talking face animation. In *CVPR*, pages 8652–8661, 2023. 1

[75] Ce Zheng, Xianpeng Liu, Guo-Jun Qi, and Chen Chen. Potter: Pooling attention transformer for efficient human mesh recovery. In *CVPR*, pages 1611–1620, 2023. 3

[76] Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J Black, and Otmar Hilliges. Pointavatar: Deformable point-based head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21057–21067, 2023. 3

[77] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z Li. Face alignment across large poses: A 3d solution. In *CVPR*, pages 146–155, 2016. 2

[78] Yiyu Zhuang, Hao Zhu, Xusen Sun, and Xun Cao. Mofanerf: Morphable facial neural radiance field. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, pages 268–285. Springer, 2022. 2