AvatarPointillist: AutoRegressive 4D Gaussian Avatarization

Hongyu Liu[1,2,*] Xuan Wang[2,§] Zijian Wu[2] Yating Wang[2] Ziyu Wan[3]
Yue Ma[1] Runtao Liu[1] Boyao Zhou[2] Yujun Shen[2] Qifeng Chen[1,§]

[1]HKUST [2]Ant Group [3]City University of Hong Kong

https://kumapowerliu.github.io/AvatarPointillist

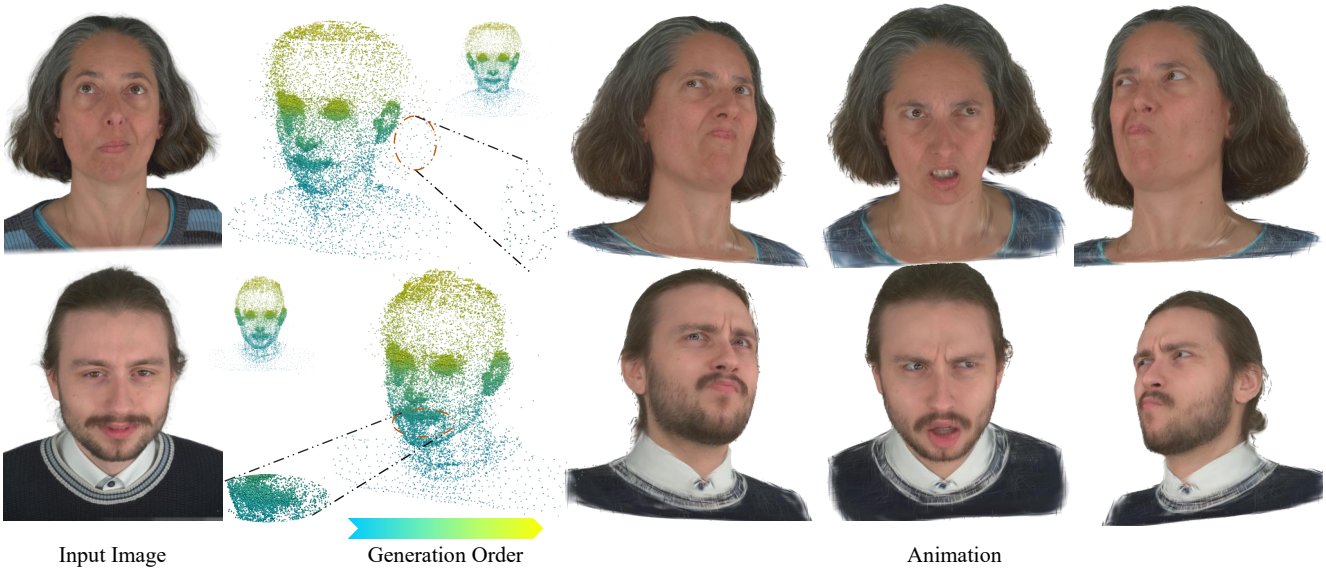

Figure 1. **Gallery of the proposed AvatarPointillist**.The leftmost column shows the input image, the middle column displays the Gaussian point cloud generated by our AR model, and the rightmost column presents the final drivable 4D Gaussian avatar. The generation order proceeds from bottom to top and left to right. It can be seen that our AR model directly models the Gaussian point cloud, allowing it to simulate the adaptive point adjustment capability of Gaussian Splatting to produce precise geometry (e.g., hair and dense beards).

Abstract

We introduce `AvatarPointillist`*, a novel framework for generating dynamic 4D Gaussian avatars from a single portrait image. At the core of our method is a decoder-only Transformer that autoregressively generates a point cloud for 3D Gaussian Splatting. This sequential approach allows for precise, adaptive construction, dynamically adjusting point density and the total number of points based on the subject's complexity. During point generation, the AR model also jointly predicts per-point binding information, enabling realistic animation. After generation, a dedicated Gaussian decoder converts the points into complete, renderable Gaussian attributes. We demonstrate that conditioning the decoder on the latent features from the AR generator enables effective interaction between stages and markedly improves fidelity. Extensive experiments validate that AvatarPointillist produces high-quality, photorealistic, and controllable avatars. We believe this autoregressive formulation represents a new paradigm for avatar generation, and we will release our code inspire future research.*

1. Introduction

The creation of photorealistic and animatable digital humans, often referred to as avatars, is a significant and highly active research area in computer vision and graphics. This

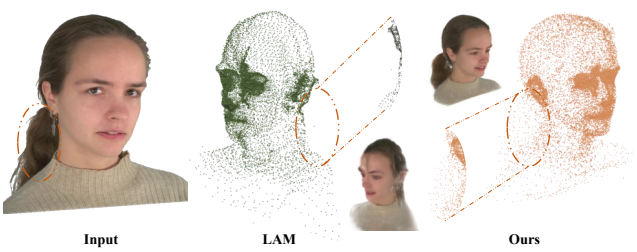

Figure 2. Comparison of different Gaussian point cloud modeling approaches. LAM [22] constructs Gaussian point clouds based on a point cloud template, which fails to reconstruct fine details from the image, such as ponytails. In contrast, our method utilizes an AR model to directly model the Gaussian point cloud. It effectively learns the capability to adaptively adjust point density and count, enabling precise modeling. Moreover, we also include final rendering results for comparison. LAM produces distorted geometry and shows noticeable artifacts.

technology holds the key to transformative applications in virtual reality (VR), telepresence, filmmaking, and immersive gaming. Broadly, existing approaches to avatar animation can be categorized into two main paradigms: 2D-based animation and 3D-aware (or 4D) avatarization. 2D methods typically operate in the image domain to generate expressive talking heads, while 4D methods focus on building full 3D geometric representations that ensure consistency across varying poses and viewpoints.

Early 2D methods in this domain primarily leveraged Generative Adversarial Networks (GANs) [17, 28, 29] and adopted a warping-then-generation scheme [19, 56–58, 81, 82, 84] to synthesize facial expressions and pose changes. With the recent emergence of powerful diffusion models [20, 25, 55], a new wave of 2D methods has demonstrated impressive results in generation quality and generalization capabilities [44, 68, 71, 72]. However, diffusion-based techniques often require substantial computational resources and suffer from long inference times due to the need for multiple sampling steps. More fundamentally, all 2D-based methods lack a sense of 3D structure. This inherent limitation leads to poor handling of extreme pose variations, noticeable geometric distortions, and an inability to render avatars from arbitrary viewpoints.

4D-based methods generate animatable, multi-view-consistent avatars by leveraging 3D geometry. Many approaches use Neural Radiance Fields (NeRF)[49] as the 3D representation[10, 36, 40, 47, 85], achieving good quality but suffering from slow rendering due to NeRF's inefficiency. Recently, 3D Gaussian Splatting (3DGS) [30] has emerged as a faster alternative, enabling real-time performance with photorealistic results. Some methods, such as GAGAvatar [6] and LAM [22], leverage 3DGS for single-image avatar generation, achieving good overall performance but with limited fidelity in fine-grained and identity-specific details. We argue that this issue arises from a fundamental problem in how these methods model the explicit geometry in the 3DGS representation. GAGAvatar [6], for instance, attempts to lift input 2D features directly into 3D, bypassing a complete 3D point cloud for representing the head. This design may limit its ability to handle large-angle views and occluded regions, requiring an auxiliary 2D network for final refinement. LAM[22] addresses this by placing Gaussians directly in a 3D canonical space. However, it relies on a fixed point cloud template (e.g., FLAME vertices[35]) and uses a constant number of Gaussians for all subjects. As shown in Fig. 2, this rigid setup limits the model's ability to adaptively adjust the density or position of Gaussians to capture subject-specific features like beards or unique hairstyles. As a result, it loses one of the core advantages of 3DGS: adaptive control over point distribution based on geometry. This observation leads to our central question: Can we design a generative model that learns the 3DGS point cloud distribution directly, without relying on a fixed template? Such a model would be free to decide where to place points and how many to use, fully capturing the flexible and adaptive nature that gives 3DGS its power.

In this paper, we propose AvatarPointillist, a novel framework that directly tackles this challenge by casting 3DGS avatar generation as an autoregressive (AR) sequential task. Unlike existing methods that rely on fixed templates, our approach learns to generate the 3DGS point cloud distribution from scratch. This point-by-point generation paradigm fully embraces the adaptive and dynamic nature of 3DGS, enabling the model to adjust the spatial distribution of Gaussians on the fly—placing points with higher density and finer precision in geometrically complex regions. To train this, we first employ a fitting procedure [53] to construct Gaussian point data for each identity in a 4D avatar dataset [32], creating dynamically densified 3DGS data with animation binding for each subject. We then quantize this data and train a decoder-only Transformer using a next-token prediction objective. To incorporate identity-specific features, we introduce cross-attention mechanisms that inject identity embeddings into the Transformer. Our model effectively learns to fit this structured data, enabling it to adaptively adjust the spatial distribution and scale of Gaussian points based on the input image, thus supporting high-quality and identity-aware avatar generation.

Once the sequential generation of the point cloud geometry is complete, we utilize a separate Transformer based Gaussian decoder to translate these points into their full Gaussian parameters (e.g., color, opacity, etc.) for

* This work is done partially when Hongyu is an intern at Ant Group.
§ Joint corresponding authors.

rendering. We found that by conditioning this decoder on the latent features from the AR generator, we significantly enhance the final rendering quality. Comprehensive experiments demonstrate that our method significantly outperforms all baselines, both quantitatively and qualitatively. We believe this exploration of autoregressive generation for explicit avatar geometry represents a promising new direction for the community.

2. Related Works

This section briefly reviews related work, including 2D and 3D-aware avatar generation methods, as well as recent approaches on autoregressive geometry generation.

2.1. 2D-Based Animatable Avatar

Image-driven talking head synthesis has seen rapid advancements in recent years, particularly within the 2D generation paradigm [3, 12, 16, 38, 43, 45, 46, 56–58, 66, 73, 78, 81, 82]. Many work leverages Generative Adversarial Networks (GANs) to produce realistic speaking face videos, then applies motion-driven warping, and finally renders the output frames. To guide the warping process, different motion cues such as facial landmarks [57, 81], depth maps [26], and latent representations [3] have been utilized to ensure accurate expression and motion transfer from the driving source. With the emergence of diffusion-based generative models, several recent approaches [44, 68, 71] have incorporated pre-trained diffusion backbones into the one-shot talking face generation pipeline. These methods benefit from strong priors learned on large-scale image datasets. However, due to their inherently two-dimensional modeling assumptions, these approaches often struggle with large pose variations, leading to visible geometric artifacts. Furthermore, they lack explicit 3D awareness, making view control and consistent head movement synthesis particularly challenging.

2.2. 3D-Aware Animatable Avatar

Fitting-Based Methods. Given a monocular video as input, some per-subject optimization method utilizing representations like meshes [18], NeRFs [14, 74, 91], SDFs [88], points [89], and 3D Gaussians [5, 39, 67, 69]. However, the optimization-based nature of these methods often leads to overfitting on the input viewpoint, resulting in poor extrapolation to novel views. Some research [27, 49, 80] leverages large-scale multi-view datasets [2, 32, 52, 75, 76, 87] to learn rich, generalizable priors for geometry and appearance. However, these approaches are fundamentally fitting-based—they are primarily designed to reconstruct or adapt a model to a specific subject, often from scratch. While they can achieve impressive reconstruction quality, their procedures are typically rigid and lack flexibility for broader use cases. More recently, methods such as CAP4D [62] and GAF [61] have introduced diffusion models to synthesize multi-view images from a single input portrait, which are then used to drive the avatar fitting process. Although this strategy improves identity generalization, it still requires considerable time for optimization, limiting its practicality in real-time or one-shot scenarios.

End-to-End Methods. To address the need for generalization, end-to-end methods learn a powerful prior from large-scale monocular [70] or multi-view datasets [32, 48], enabling them to generate an animatable avatar from a single or very few images. A significant milestone in this direction is the advent of Neural Radiance Fields (NeRF) [4, 7, 36, 37, 47, 49, 63, 77, 79, 90], which support high-fidelity 3D reconstruction and fine-grained camera control. Some approaches incorporate 3D supervision from monocular 3D face reconstruction [8, 9, 13] or synthetic multi-view data[10, 11, 40] for better perfomance. NeRF-based pipelines have been widely integrated into one-shot talking head generation frameworks, improving the realism and 3D alignment of the synthesized results. More recently, GAGAvatar [6], LAM [22], and Avat3R [33] demonstrated the effectiveness of 3D Gaussian Splatting (3DGS) [31] in this context, offering faster rendering while preserving high visual quality. However, these methods still suffer from some limitations. GAGAvatar [6], for instance, requires an auxiliary neural network for refinement and its 2D-to-3D lifting strategy struggles to realistically model unseen regions. While LAM [22] addresses these particular issues, it is constrained by a fixed-template point cloud [35]. This static topology inherently limits its fidelity, as it cannot adaptively adjust the Gaussian count to match subject-specific features. Avat3R [33], on the other hand, is not a one-shot method, requiring multiple input images, and its network must be re-executed to generate the Gaussian splatting for each new expression. In contrast, our method, AvatarPointillist, addresses these limitations by formulating the task as an autoregressive (AR) generative process. As a one-shot generative model, it is not constrained by a fixed template or topology. This AR approach allows our model to dynamically and adaptively adjust the Gaussian distribution and total count, enabling the high-fidelity synthesis of complex, subject-specific features.

2.3. Autoregressive Geometry Generation

Inspired by the profound success of autoregressive (AR) models in natural language processing [1, 65], a significant trend has emerged in applying sequential generation techniques to 3D geometry. This paradigm typically treats a 3D shape (e.g., a mesh or point cloud) as a sequence of discrete tokens. A common strategy involves a two-stage process: first, a Vector Quantized Autoencoder (VQ-VAE) [64] learns a discrete vocabulary of geometric features; second,

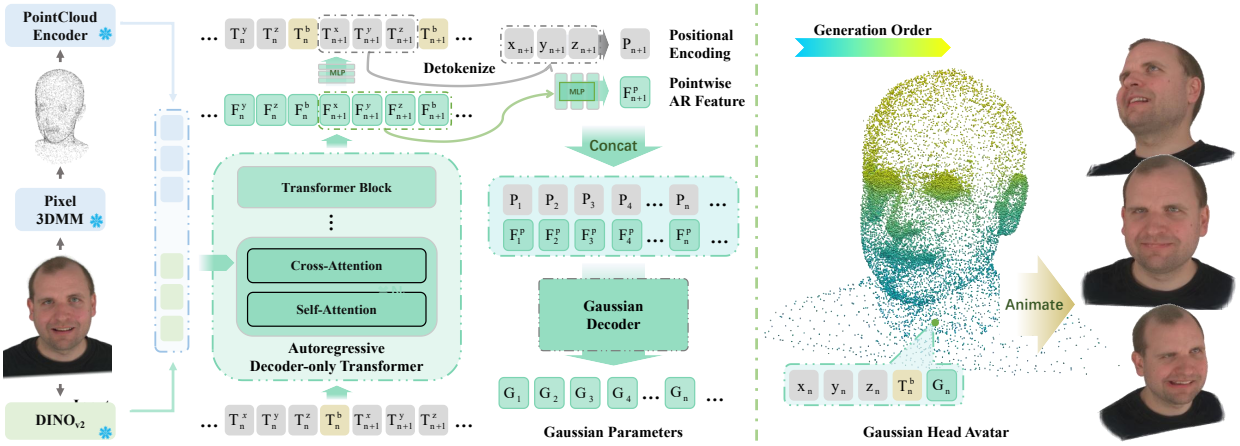

Figure 3. Overview of our framework. It consists of two modules: an autoregressive (AR) model for Gaussian geometry generation and a Gaussian Decoder for predicting rendering attributes. The AR model takes image features from DINOv2 [51] and point cloud features as input. The point cloud feature extract via Pixel3DMM [15] and a point cloud encoder [86]. The AR model is trained to generate a Gaussian point cloud via next-token prediction, where each point is represented by four quantized tokens $(T_n^x, T_n^y, T_n^z, T_n^b)$ corresponding to coordinates and binding information. After generation, the tokens are de-quantized to obtain the actual coordinates. We then combine the positional embeddings P_n with the internal features F_n^p from the AR model as input to the Gaussian Decoder to predict the final accurate Gaussian attributes. Finally, the result is animated using Linear Blend Skinning (LBS) and the binding information.

a Transformer [65] is trained to autoregressively predict the next token in the sequence. This approach is famously demonstrated by MeshGPT [59] for triangle meshes, and similar concepts have been applied to point clouds [60] and implicit representations [50]. To overcome the challenge of modeling long sequences, more recent methods like Meshtron [21] and ARMesh [34] propose hierarchical or coarse-to-fine AR generation, significantly improving the fidelity of the resulting geometry.

3. Preliminary

In this section, we provide a brief overview of some essential prerequisites that are closely related to our Avatar-Pointillist in Section 4. We first introduce our 3DGS data structure and explain how we construct the training data. Then, we present how we quantize the data to make it compatible with AR training.

3.1. Data Construction

We build our training data using the GaussianAvatars [69] method and the Nersemble dataset [32]. Specifically, for each identity in Nersemble, we first fit a complete GaussianAvatars model. This method creates a 3DGS representation where each Gaussian is bound to a specific face of the FLAME mesh [35]. As described in their paper, a 3D Gaussian is static in the local space of its parent triangle but dynamic in the global space as the triangle moves. For each Gaussian, the model defines its location μ, rotation r, and scaling s in this local space. At rendering time, these properties are converted to the global space using the face's transformation (rotation R, translation T, and scale k):

$$r' = Rr, \quad \mu' = kR\mu + T, \quad s' = ks.$$

We refer the reader to the original paper [69] for additional implementation details. To construct our training data, we use the canonical FLAME mesh of each identity to compute the corresponding global canonical Gaussian point cloud. Specifically, let N be the total number of Gaussians in the point cloud, the final point cloud is defined as P as:

$$P = (x_1, y_1, z_1, b_1, x_2, y_2, ...x_N, y_N, z_N, b_N). \quad (1)$$

Here, x_n, y_n, z_n are the global coordinates of the Gaussian in the canonical space for each point, and b_n is the binding index, which indicates the FLAME face to which the point is attached.

3.2. Quantization and Order of Coordinates

Following the approach introduced in [21], we adopt a specific ordering strategy for our Gaussian point cloud. We establish this order by sorting all points in a given cloud based on their coordinate values. The primary sorting key is the vertical y-axis, followed by the z-axis, and finally the x-axis (a yzx sort order). This fixed sorting strategy ensures that identical point clouds will always produce identical input sequences for our model.

In addition to the point cloud coordinates, we structure the sequences using three reserved token types similar

to [21]: Start-of-Sequence (S), End-of-Sequence (E), and Padding (P). For each sequence, we prepend a block of 4 start-of-sequence (S) tokens and append a block of 4 end-of-sequence (E) tokens. This design choice reflects the fact that each point consists of four values: the 3D coordinates (x, y, z) and a binding index. Grouping the special tokens in blocks of four ensures structural consistency within the sequence representation.

Our autoregressive model requires discrete tokens as input. We therefore convert our continuous point coordinates into a discrete format using quantization. This is achieved by dividing the coordinate space into a fixed number of bins. The number of bins determines the granularity of the resulting geometry, creating a trade-off between precision and computational load. We found that 1024 quantization levels (similar to strategies in prior work [21]) provide an effective balance between model accuracy and efficiency for representing our Gaussian points. Finally, after quantization, our point cloud P is flattened into a single integer sequence T for the autoregressive model:

$$T = (T_1^x, T_1^y, T_1^z, T_1^b, \ldots, T_N^x, T_N^y, T_N^z, T_N^b). \quad (2)$$

Here, each coordinate T_n^x, T_n^y, T_n^z is a discrete token in the range $[0, 1023]$, corresponding to our 1024 quantization levels. The binding token T_n^b is offset to occupy a distinct part of the vocabulary, defined as $T_n^b = b_n + 1024$. Given that b_n is the original face index (with a maximum of 10144 faces, so $b_n \in [0, 10143]$), the binding tokens T_n^b fall within the range $[1024, 11167]$.

4. Method

We aim to develop a method that generates a 4D Gaussian animatable avatar from a single source image I_s, driven by the motion of a target individual I_t. In Section 4.1, we introduce an autoregressive mechanism for predicting the canonical 3D Gaussian Splatting point cloud. Based on the output of this AR model, a Gaussian decoder is employed to infer the attributes of each point (e.g., position, scale, and rotation). The input to the AR model consists of both the previously generated 3DGS point and the model's learned implicit features, as described in Section 4.2. Furthermore, since our AR model also predicts the binding between each point and the template mesh, the generated canonical 3D Gaussian representation can be animated by deforming it with the mesh motion (see Section 4.3). An overview of the pipeline is illustrated in Figure 3. We now describe each component in detail.

4.1. Autoregressive Model

The core structure of our AR model is a decoder-only Transformer, the architecture is shown in Figure 3. Specifically, our Transformer contains several layers, where each layer contains a cross-attention layer, a self-attention layer, and a feed-forward network.

For injecting the input image information, we use DINOv2 [51] to extract the feature of the input image directly. Meanwhile, since our AR model focuses on point cloud generation, we also use an off-the-shelf 3D face reconstruction model [15] to get the FLAME parameters. We then use these parameters to get the sample vertices from the FLAME mesh and use a point cloud encoder [86] to obtain their features. Finally, the DINOv2 feature and point cloud feature are concatenated and injected into our decoder via the cross-attention layers.

With our Transformer, the output is generated by sequentially predicting each token T_n based on its conditional probability given all previously generated tokens $T_{<n}$: $p(T_n|T_{<n})$. The complete point cloud with binding information is represented as a sequence of $4N$ tokens (i.e., N points with 4 quantized tokens each for x, y, z, and binding). The joint probability of the entire sequence T is modeled as:

$$p(T) = \prod_{n=1}^{4N} p(T_n|T_{<n}). \quad (3)$$

The entire training process uses the standard cross-entropy loss for next-token prediction.

4.2. Gaussian Decoder

Once the AR model generates the complete output sequence, we use a Transformer-based Gaussian decoder to predict the full set of Gaussian parameters (as shown in Fig. 3). First, we detokenize the token sequence to recover the original coordinates (x, y, z) for each point. Similar to LAM [22], these coordinates are passed through a positional encoding [49] and an MLP to produce a per-point geometric feature, P_n. Importantly, we found that the inherent hidden states from the AR Transformer are also crucial for improving generation quality (see Sec. 5.4). We extract the final hidden state sequence F from AR model:

$$F = (F_1^x, F_1^y, F_1^z, F_1^b, \ldots, F_N^x, F_N^y, F_N^z, F_N^b). \quad (4)$$

Since four tokens correspond to a single 3D point, an MLP is used to assemble these four corresponding hidden features $(F_n^x, F_n^y, F_n^z, F_n^b)$ into a single, comprehensive AR feature, F_n^p.

Finally, the two features for each point, the geometric feature P_n and the AR feature F_n^p are concatenated and used as the input to the Gaussian Decoder. This decoder then outputs the final attributes for each Gaussian point k: $c_k \in \mathbb{R}^3$, opacity $o_k \in \mathbb{R}$, scale $s_k \in \mathbb{R}^3$, rotation $R_k \in \mathrm{SO}(3)$, and a positional offset $\Delta p_k \in \mathbb{R}^3$. More specifically, this predicted offset Δp_k is added to the canonical point positions, allowing the model to make fine-grained geometric adjustments to better capture the target person's geometry.

4.3. Expression Animation

Our AR model predicts accurate binding information for each point, enabling us to animate the canonical 3D point cloud directly using vertex-based Linear Blend Skinning (LBS) and corrective blendshapes, in a manner similar to the FLAME model. First, we interpolate vertex-specific attributes from the FLAME mesh to our output points via barycentric interpolation. For each point $\mathbf{p}_i \in \mathbb{R}^3$, we determine its corresponding triangle f_i on the FLAME mesh with the binding information and retrieve the triangle's vertex indices. Using the vertex positions, we compute the barycentric coordinates (b_0, b_1, b_2) of p_i with respect to the triangle. These coordinates are then used to interpolate the FLAME attributes defined at the vertices. The interpolated LBS weights $\hat{\mathbf{w}}_i$ and expression blendshapes $\hat{\mathbf{s}}_i$ for point p_i are computed as:

$$\begin{aligned} \hat{\mathbf{w}}_i &= b_0\mathbf{W}_0 + b_1\mathbf{W}_1 + b_2\mathbf{W}_2 \\ \hat{\mathbf{S}}_i &= b_0\mathbf{S}_0 + b_1\mathbf{S}_1 + b_2\mathbf{S}_2 \end{aligned} \tag{5}$$

where $\mathbf{W}_j$ and $\mathbf{S}_j$ are the LBS weights and expression directions at vertex $j \in \{0, 1, 2\}$ in the corresponding triangle f_i of the FLAME mesh.

Once equipped with these interpolated properties, our Gaussian avatar is now fully rigged and can be driven by the standard FLAME deformation process using pose ($\boldsymbol{\theta}$) and expression (ψ) parameters.

4.4. Loss Functions and Training Strategy

Our model is trained in two stages. We first optimize the AR model for sequential generation. After this stage is complete, we freeze the AR model and separately train the Gaussian Decoder using a combination of rendering losses.

4.4.1. Autoregressive Model Training

The AR Transformer is trained first, akin to a standard language model. The objective is to accurately predict the next token T_n in the quantized sequence. We optimize this stage using a standard Cross-Entropy (CE) loss.

4.4.2. Gaussian Decoder Training

For each identity in the dataset, we use the trained AR model to generate the latent sequence T and hidden states F_n^p, which are fed into the Gaussian Decoder to predict 3D Gaussian Splatting (3DGS) attributes. The decoder is optimized by comparing the rendered image I_r with the ground-truth view I_{gt} using a combination of photometric and perceptual losses. Specifically, we employ an L_1 loss to ensure pixel-wise color consistency, SSIM to preserve structural similarity, LPIPS to enhance perceptual quality, and a regularization term applied to constrain the predicted offset. The overall objective is defined as:

$$\begin{aligned} \mathcal{L}_{\text{total}} = \lambda_{L1}\mathcal{L}_{L1} + \lambda_{SSIM}\mathcal{L}_{SSIM} \\ + \lambda_{LPIPS}\mathcal{L}_{LPIPS} + \lambda_{Reg}\mathcal{L}_{Reg} \end{aligned} \tag{6}$$

We empirically set the weights to $\lambda_{L1} = 1$, $\lambda_{SSIM} = 0.5$, $\lambda_{LPIPS} = 0.1$ and $\lambda_{Reg} = 0.1$.

5. Experiments

In this section, we first describe our experimental setup, including implementation details, baselines, and evaluation metrics. Then, we present quantitative and qualitative results. Finally, we conduct an ablation study to validate our model and contributions. Additional results are available in the supplementary material.

5.1. Experimental Setup

Implementation Details. We train our model on the NeRSemble dataset [32], which features a total of 419 identities. We randomly select 25 of these identities to form our test set, using the remainder for training. To generate the training data for our AR method, we first fit all identities using the GaussianAvatars [53] method. During the training of the autoregressive model, we utilize the AdamW optimizer [41] with a learning rate of 1e-4. The autoregressive model is trained on 16 NVIDIA H20 GPUs for 50K steps with a batch size of 4. Since the point cloud sequences are very long, we adopt the truncated training strategy from [21]to enhance efficiency. Specifically, the input token sequence is first partitioned into fixed-size context windows, with padding applied to any segments of insufficient length. Then, we utilize a sliding window mechanism to shift the window step-by-step and train each windowed segment accordingly. We set the window size to 12000. For the Gaussian Decoder training, we also use the Adam optimizer and train for 12500 steps on 8 NVIDIA H20 GPUs with a batch size of 4. For more details, please refer to the supplementary material.

Baselines. We compare our method with recent state-of-the-art, single-image-guided 4D avatar reconstruction models, including two NeRF-based methods (AvatarArtist [40] and Portrait4Dv2 [11]) and two Gaussian Splatting-based methods (LAM [22] and GAGAvatar [6]).

Evaluation Metrics. To evaluate perceptual quality, we adopt LPIPS [83] and FID [24]. Expression accuracy is measured using the Average Keypoint Distance (AKD)[42], while pose consistency is assessed by the Average Pose Distance (APD), with pose parameters extracted following[23]. For identity preservation, we employ CLIPScore [54] as our ID metric.

5.2. Qualitative Results

As shown in Figure 4, we provide qualitative comparisons for self-reenactment and cross-reenactment tasks. The first column shows the source image and the target pose (inset). We randomly select diverse viewpoints for a thorough

Figure 4. Qualitative comparison with state-of-the-art methods. The leftmost column shows the input images, with the target image displayed in the bottom-right corner. The first row presents self-reenactment results, while the remaining three rows show cross-reenactment results. Our method demonstrates superior performance in expression and pose consistency, as well as better identity preservation compared to other approaches.

Table 1. Quantitative evaluation of state-of-the-art methods and our approach on the NeRSemble dataset [32]. ↓ indicates lower is better while ↑ indicates higher is better. **Red** highlights the best result, and **Blue** highlights the second-best result.

Method	Self reenactment				Cross reenactment			
	LPIPS ↓	FID ↓	AKD ↓	APD ↓	FID ↓	CLIP ↑	AKD ↓	APD ↓
Portrait4Dv2 [11]	0.20	123.02	5.32	34.53	191.13	0.63	11.94	**142.93**
AvatarArtist [40]	0.21	118.94	6.87	39.58	**175.69**	0.61	9.32	187.31
LAM [22]	0.24	136.01	4.37	61.83	238.54	0.54	**6.68**	210.23
GAGAvatar [6]	**0.18**	**111.76**	**3.93**	**27.94**	181.22	**0.71**	10.01	170.12
Ours	**0.15**	**95.18**	**2.38**	**22.86**	**160.74**	**0.75**	**5.93**	**153.13**

evaluation. The top two rows shows self-reenactment results , and the rest show cross-reenactment . Among baselines, LAM shows clear artifacts, especially in complex facial areas. AvatarArtist works for small pose changes but struggles with larger ones. Portrait4Dv2 and GAGAvatar produce coherent results but often have expression mismatches and over-smoothed hair. In contrast, our method generates more realistic and consistent reenactments, with better alignment in pose and expression. It also preserves fine details like hair texture and facial contours, resulting in sharper and more identity-accurate outputs.

5.3. Quantitative Results

Table 1 summarizes the quantitative results on the test set of the NeRSemble dataset [32] under both self- and cross-reenactment settings. For self-reenactment, the source

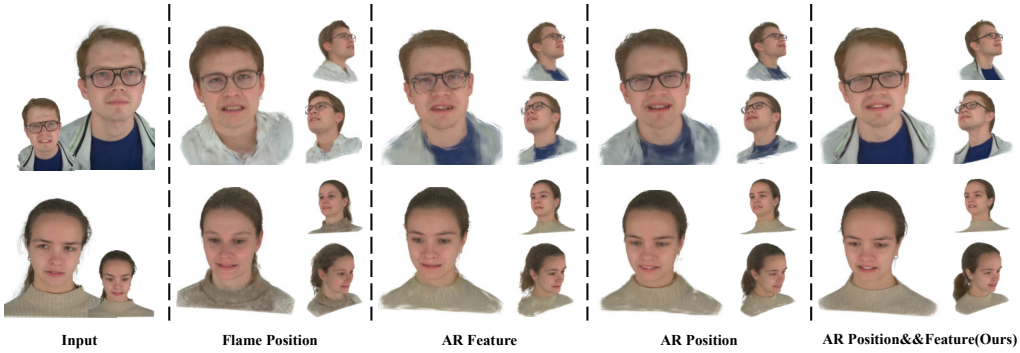

Figure 5. **Visualization of ablation study on input setting of Gaussian decoder.** The leftmost column shows the input. The FLAME Positions baseline, similar to the LAM method, uses the canonical FLAME mesh vertices as a template and only applies decoder-predicted offsets to deform this template into a final Gaussian point cloud. Pointwise AR Feature refers to using only the AR features (F_n^p) without positional information, while Positional Encoding uses only the point embeddings (P_n) without AR features.

Table 2. Ablaiton study on the NeRSemble dataset [32]. ↓ indicates lower is better while ↑ indicates higher is better.

Method	LPIPS ↓	FID ↓	AKD ↓	APD ↓
Flame Position	0.23	120.34	4.82	41.22
AR Feature	0.22	110.93	5.89	32.96
AR Position	0.19	103.80	5.81	41.49
Ours	**0.15**	**95.18**	**2.38**	**22.86**

image is randomly chosen from intermediate frames with minimal occlusion. For cross-reenactment, a fixed motion sequence is used as the target across all methods. All baselines use the same source image per identity, and input camera views are aligned across methods. We also align the input camera views as closely as possible across methods and use each method's own tracking pipeline to obtain ground-truth poses, further ensuring fairness in evaluation. As shown in Table 1, our method consistently outperforms all baselines across metrics in both tasks, showing superior identity preservation and motion transfer accuracy.

5.4. Ablation Study

Effectiveness of autoregressive Model. We compare our full method with a baseline called FLAME position, which adopts a static-topology approach similar to LAM [22]. This baseline skips our AR model and directly uses 3D vertices from the canonical FLAME mesh as input to the Gaussian decoder, which then predicts offsets to refine these fixed points. As shown in Figure 5, this static point cloud fails to capture subject-specific geometry and shows limited resemblance to the input image. It struggles with complex regions like hair and cannot adaptively allocate points to important areas. Additionally, since it relies only on point embeddings without rich per-point features, the rendered results often lack identity consistency. In contrast, our AR model generates the 3D point cloud directly, allowing more accurate geometry reconstruction. The AR features also help the decoder produce higher-quality Gaussian attributes.

Effectiveness of Input Setting of Gaussian Decoder. We further analyze the impact of different inputs to the Gaussian Decoder, as shown in Figure 5. Using only the final AR hidden state F_n^p (Pointwise AR Feature) yields suboptimal results due to the lack of spatial information. Using only de-quantized 3DGS coordinates P_n (Positional Encoding) performs better by providing spatial context, but misses the semantic richness of the AR features. Our full method combines both P_n and F_n^p, allowing the decoder to leverage spatial guidance and deep semantic cues, resulting in more accurate attribute prediction and the best overall quality.

6. Conclusion

We propose AvatarPointillist, a novel framework for one-shot 4D Gaussian avatar generation. At its core is an autoregressive model that learns to generate Gaussian point clouds point by point, removing fixed topology constraints. This enables dynamic control over the number and placement of Gaussians, focusing more on complex, identity-specific areas and fully exploiting the adaptive nature of 3D Gaussian Splatting (3DGS). Our two-stage architecture feeds the AR model's output and hidden features into a Gaussian decoder to predict high-quality rendering attributes. Experiments show that AvatarPointillist outperforms prior methods in both quantitative metrics and visual quality. We believe this autoregressive approach offers a promising direction for explicit 3D avatar generation.

7. Acknowledgment

The work was supported by HKUST under grant number WEB25EG01.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in neural information processing systems*, pages 1877–1901, 2020. 3

[2] Marcel C Buehler, Gengyan Li, Erroll Wood, Leonhard Helminger, Xu Chen, Tanmay Shah, Daoye Wang, Stephan Garbin, Sergio Orts-Escolano, Otmar Hilliges, et al. Cafca: High-quality novel view synthesis of expressive faces from casual few-shot captures. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–12, 2024. 3

[3] Egor Burkov, Igor Pasechnik, Artur Grigorev, and Victor Lempitsky. Neural head reenactment with latent pose descriptors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13786–13795, 2020. 3

[4] 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, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16123–16133, 2022. 3

[5] Yufan Chen, Lizhen Wang, Qijing Li, Hongjiang Xiao, Shengping Zhang, Hongxun Yao, and Yebin Liu. Monogaussianavatar: Monocular gaussian point-based head avatar. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–9, 2024. 3

[6] Xuangeng Chu and Tatsuya Harada. Generalizable and animatable gaussian head avatar. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2, 3, 6, 7

[7] Xuangeng Chu, Yu Li, Ailing Zeng, Tianyu Yang, Lijian Lin, Yunfei Liu, and Tatsuya Harada. Gpavatar: Generalizable and precise head avatar from image (s). *arXiv preprint arXiv:2401.10215*, 2024. 3

[8] 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. 3

[9] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 3

[10] Yu Deng, Duomin Wang, Xiaohang Ren, Xingyu Chen, and Baoyuan Wang. Portrait4d: Learning one-shot 4d head avatar synthesis using synthetic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7119–7130, 2024. 2, 3

[11] Yu Deng, Duomin Wang, and Baoyuan Wang. Portrait4d-v2: Pseudo multi-view data creates better 4d head synthesizer. *arXiv preprint arXiv:2403.13570*, 2024. 3, 6, 7

[12] Nikita Drobyshev, Jenya Chelishev, Taras Khakhulin, Aleksei Ivakhnenko, Victor Lempitsky, and Egor Zakharov. Megaportraits: One-shot megapixel neural head avatars. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2663–2671, 2022. 3

[13] 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)*, 40(4): 1–13, 2021. 3

[14] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8649–8658, 2021. 3

[15] Simon Giebenhain, Tobias Kirschstein, Martin Rünz, Lourdes Agapito, and Matthias Nießner. Pixel3dmm: Versatile screen-space priors for single-image 3d face reconstruction, 2025. 4, 5

[16] Yuan Gong, Yong Zhang, Xiaodong Cun, Fei Yin, Yanbo Fan, Xuan Wang, Baoyuan Wu, and Yujiu Yang. Toontalker: Cross-domain face reenactment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7690–7700, 2023. 3

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[18] Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. Neural head avatars from monocular rgb videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18653–18664, 2022. 3

[19] Jianzhu Guo, Dingyun Zhang, Xiaoqiang Liu, Zhizhou Zhong, Yuan Zhang, Pengfei Wan, and Di Zhang. Liveportrait: Efficient portrait animation with stitching and retargeting control. *arXiv preprint arXiv:2407.03168*, 2024. 2

[20] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023. 2

[21] Zekun Hao, David W Romero, Tsung-Yi Lin, and Ming-Yu Liu. Meshtron: High-fidelity, artist-like 3d mesh generation at scale. *arXiv preprint arXiv:2412.09548*, 2024. 4, 5, 6

[22] Yisheng He, Xiaodong Gu, Xiaodan Ye, Chao Xu, Zhengyi Zhao, Yuan Dong, Weihao Yuan, Zilong Dong, and Liefeng Bo. Lam: Large avatar model for one-shot animatable gaussian head. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–13, 2025. 2, 3, 5, 6, 7, 8

[23] Thorsten Hempel, Ahmed A. Abdelrahman, and Ayoub Al-Hamadi. Toward robust and unconstrained full range of rotation head pose estimation. *IEEE Transactions on Image Processing*, 33:2377–2387, 2024. 6

[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. *Advances in neural information processing systems*, 30, 2017. 6

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[26] Fa-Ting Hong, Longhao Zhang, Li Shen, and Dan Xu. Depth-aware generative adversarial network for talking head video generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3397–3406, 2022. 3

[27] Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. Headnerf: A real-time nerf-based parametric head model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20374–20384, 2022. 3

[28] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2

[29] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021. 2

[30] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2

[31] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 3

[32] Tobias Kirschstein, Shenhan Qian, Simon Giebenhain, Tim Walter, and Matthias Nießner. Nersemble: Multi-view radiance field reconstruction of human heads. *ACM Trans. Graph.*, 42(4), 2023. 2, 3, 4, 6, 7, 8

[33] Tobias Kirschstein, Javier Romero, Artem Sevastopolsky, Matthias Nießner, and Shunsuke Saito. Avat3r: Large animatable gaussian reconstruction model for high-fidelity 3d head avatars. *arXiv preprint arXiv:2502.20220*, 2025. 3

[34] Jiabao Lei, Kewei Shi, Zhihao Liang, and Kui Jia. ARMesh: Autoregressive mesh generation via next-level-of-detail prediction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 4

[35] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017. 2, 3, 4

[36] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17969–17978, 2023. 2, 3

[37] Xueting Li, Shalini De Mello, Sifei Liu, Koki Nagano, Umar Iqbal, and Jan Kautz. Generalizable one-shot 3d neural head avatar. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[38] Hongyu Liu, Xintong Han, Chengbin Jin, Lihui Qian, Huawei Wei, Zhe Lin, Faqiang Wang, Haoye Dong, Yibing Song, Jia Xu, et al. Human motionformer: Transferring human motions with vision transformers. *arXiv preprint arXiv:2302.11306*, 2023. 3

[39] Hongyu Liu, Xuan Wang, Ziyu Wan, Yujun Shen, Yibing Song, Jing Liao, and Qifeng Chen. Headartist: Text-conditioned 3d head generation with self score distillation. In *ACM SIGGRAPH 2024 Conference Papers*, New York, NY, USA, 2024. Association for Computing Machinery. 3

[40] Hongyu Liu, Xuan Wang, Ziyu Wan, Yue Ma, Jingye Chen, Yanbo Fan, Yujun Shen, Yibing Song, and Qifeng Chen. Avatarartist: Open-domain 4d avatarization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10758–10769, 2025. 2, 3, 6, 7

[41] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2019. 6

[42] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*, 2019. 6

[43] Yue Ma, Yingqing He, Xiaodong Cun, Xintao Wang, Siran Chen, Xiu Li, and Qifeng Chen. Follow your pose: Pose-guided text-to-video generation using pose-free videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4117–4125, 2024. 3

[44] Yue Ma, Hongyu Liu, Hongfa Wang, Heng Pan, Yingqing He, Junkun Yuan, Ailing Zeng, Chengfei Cai, Heung-Yeung Shum, Wei Liu, et al. Follow-your-emoji: Fine-controllable and expressive freestyle portrait animation. *arXiv preprint arXiv:2406.01900*, 2024. 2, 3

[45] Yue Ma, Kunyu Feng, Xinhua Zhang, Hongyu Liu, David Junhao Zhang, Jinbo Xing, Yinhan Zhang, Ayden Yang, Zeyu Wang, and Qifeng Chen. Follow-your-creation: Empowering 4d creation through video inpainting. *arXiv preprint arXiv:2506.04590*, 2025. 3

[46] Yue Ma, Zexuan Yan, Hongyu Liu, Hongfa Wang, Heng Pan, Yingqing He, Junkun Yuan, Ailing Zeng, Chengfei Cai, Heung-Yeung Shum, et al. Follow-your-emoji-faster: Towards efficient, fine-controllable, and expressive freestyle portrait animation. *arXiv preprint arXiv:2509.16630*, 2025. 3

[47] Zhiyuan Ma, Xiangyu Zhu, Guo-Jun Qi, Zhen Lei, and Lei Zhang. Otavatar: One-shot talking face avatar with controllable tri-plane rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16901–16910, 2023. 2, 3

[48] Julieta Martinez, Emily Kim, Javier Romero, Timur Bagautdinov, Shunsuke Saito, Shoou-I Yu, Stuart Anderson, Michael Zollhöfer, Te-Li Wang, Shaojie Bai, Chenghui Li, Shih-En Wei, Rohan Joshi, Wyatt Borsos, Tomas Simon, Jason Saragih, Paul Theodosis, Alexander Greene, Anjani Josyula, Silvio Mano Maeta, Andrew I. Jewett, Simon Venshtain, Christopher Heilman, Yueh-Tung Chen, Sidi Fu, Mohamed Ezzeldin A. Elshaer, Tingfang Du, Longhua Wu, Shen-Chi Chen, Kai Kang, Michael Wu, Youssef Emad,

Steven Longay, Ashley Brewer, Hitesh Shah, James Booth, Taylor Koska, Kayla Haidle, Matt Andromalos, Joanna Hsu, Thomas Dauer, Peter Selednik, Tim Godisart, Scott Ardisson, Matthew Cipperly, Ben Humberston, Lon Farr, Bob Hansen, Peihong Guo, Dave Braun, Steven Krenn, He Wen, Lucas Evans, Natalia Fadeeva, Matthew Stewart, Gabriel Schwartz, Divam Gupta, Gyeongsik Moon, Kaiwen Guo, Yuan Dong, Yichen Xu, Takaaki Shiratori, Fabian Prada, Bernardo R. Pires, Bo Peng, Julia Buffalini, Autumn Trimble, Kevyn McPhail, Melissa Schoeller, and Yaser Sheikh. Codec Avatar Studio: Paired Human Captures for Complete, Driveable, and Generalizable Avatars. *NeurIPS Track on Datasets and Benchmarks*, 2024. 3

[49] 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. *Communications of the ACM*, 65(1):99–106, 2021. 2, 3, 5

[50] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. AutoSDF: Shape priors for 3d completion, reconstruction and generation. In *CVPR*, 2022. 4

[51] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 4, 5

[52] Dongwei Pan, Long Zhuo, Jingtan Piao, Huiwen Luo, Wei Cheng, Yuxin Wang, Siming Fan, Shengqi Liu, Lei Yang, Bo Dai, et al. Renderme-360: a large digital asset library and benchmarks towards high-fidelity head avatars. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[53] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20299–20309, 2024. 2, 6

[54] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 6

[55] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2

[56] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2377–2386, 2019. 2, 3

[57] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *Advances in neural information processing systems*, 32, 2019. 3

[58] Aliaksandr Siarohin, Oliver J Woodford, Jian Ren, Menglei Chai, and Sergey Tulyakov. Motion representations for articulated animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13653–13662, 2021. 2, 3

[59] Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. Meshgpt: Generating triangle meshes with decoder-only transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 4

[60] Yongbin Sun, Yue Wang, Ziwei Liu, Joshua Siegel, and Sanjay Sarma. Pointgrow: Autoregressively learned point cloud generation with self-attention. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 61–70, 2020. 4

[61] Jiapeng Tang, Davide Davoli, Tobias Kirschstein, Liam Schoneveld, and Matthias Niessner. Gaf: Gaussian avatar reconstruction from monocular videos via multi-view diffusion. *arXiv preprint arXiv:2412.10209*, 2024. 3

[62] Felix Taubner, Ruihang Zhang, Mathieu Tuli, and David B Lindell. Cap4d: Creating animatable 4d portrait avatars with morphable multi-view diffusion models. *arXiv preprint arXiv:2412.12093*, 2024. 3

[63] Alex Trevithick, Matthew Chan, Michael Stengel, Eric Chan, Chao Liu, Zhiding Yu, Sameh Khamis, Ravi Ramamoorthi, and Koki Nagano. Real-time radiance fields for single-image portrait view synthesis. 2023. 3

[64] Aaron Van Den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in neural information processing systems*, 2017. 3

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3, 4

[66] 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. 3

[67] Yating Wang, Xuan Wang, Ran Yi, Yanbo Fan, Jichen Hu, Jingcheng Zhu, and Lizhuang Ma. 3d gaussian head avatars with expressive dynamic appearances by compact tensorial representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21117–21126, 2025. 3

[68] Huawei Wei, Zejun Yang, and Zhisheng Wang. Aniportrait: Audio-driven synthesis of photorealistic portrait animations. *arXiv:2403.17694*, 2024. 2, 3

[69] Jun Xiang, Xuan Gao, Yudong Guo, and Juyong Zhang. Flashavatar: High-fidelity head avatar with efficient gaussian embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1802–1812, 2024. 3, 4

[70] Liangbin Xie, Xintao Wang, Honglun Zhang, Chao Dong, and Ying Shan. Vfhq: A high-quality dataset and benchmark

for video face super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022. 3

[71] You Xie, Hongyi Xu, Guoxian Song, Chao Wang, Yichun Shi, and Linjie Luo. X-portrait: Expressive portrait animation with hierarchical motion attention. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2, 3

[72] Mingwang Xu, Hui Li, Qingkun Su, Hanlin Shang, Liwei Zhang, Ce Liu, Jingdong Wang, Yao Yao, and Siyu Zhu. Hallo: Hierarchical audio-driven visual synthesis for portrait image animation. *arXiv preprint arXiv:2406.08801*, 2024. 2

[73] Sicheng Xu, Guojun Chen, Yu-Xiao Guo, Jiaolong Yang, Chong Li, Zhenyu Zang, Yizhong Zhang, Xin Tong, and Baining Guo. VASA-1: Lifelike audio-driven talking faces generated in real time. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 3

[74] Yuelang Xu, Lizhen Wang, Xiaochen Zhao, Hongwen Zhang, and Yebin Liu. Avatarmav: Fast 3d head avatar reconstruction using motion-aware neural voxels. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–10, 2023. 3

[75] Yuelang Xu, Lizhen Wang, Zerong Zheng, Zhaoqi Su, and Yebin Liu. 3d gaussian parametric head model. In *European Conference on Computer Vision*, pages 129–147. Springer, 2025. 3

[76] 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 *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 601–610, 2020. 3

[77] Zhenhui Ye, Tianyun Zhong, Yi Ren, Jiaqi Yang, Weichuang Li, Jiawei Huang, Ziyue Jiang, Jinzheng He, Rongjie Huang, Jinglin Liu, et al. Real3d-portrait: One-shot realistic 3d talking portrait synthesis. *arXiv preprint arXiv:2401.08503*, 2024. 3

[78] Fei Yin, Yong Zhang, Xiaodong Cun, Mingdeng Cao, Yanbo Fan, Xuan Wang, Qingyan Bai, Baoyuan Wu, Jue Wang, and Yujiu Yang. Styleheat: One-shot high-resolution editable talking face generation via pre-trained stylegan. In *European conference on computer vision*, pages 85–101. Springer, 2022. 3

[79] Wangbo Yu, Yanbo Fan, Yong Zhang, Xuan Wang, Fei Yin, Yunpeng Bai, Yan-Pei Cao, Ying Shan, Yang Wu, Zhongqian Sun, et al. Nofa: Nerf-based one-shot facial avatar reconstruction. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023. 3

[80] Zhixuan Yu, Ziqian Bai, Abhimitra Meka, Feitong Tan, Qiangeng Xu, Rohit Pandey, Sean Fanello, Hyun Soo Park, and Yinda Zhang. One2avatar: Generative implicit head avatar for few-shot user adaptation. *arXiv preprint arXiv:2402.11909*, 2024. 3

[81] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9459–9468, 2019. 2, 3

[82] Bowen Zhang, Chenyang Qi, Pan Zhang, Bo Zhang, Hsiang-Tao Wu, Dong Chen, Qifeng Chen, Yong Wang, and Fang Wen. Metaportrait: Identity-preserving talking head generation with fast personalized adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22096–22105, 2023. 2, 3

[83] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[84] Jian Zhao and Hui Zhang. Thin-plate spline motion model for image animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3657–3666, 2022. 2

[85] Xiaochen Zhao, Jingxiang Sun, Lizhen Wang, Jinli Suo, and Yebin Liu. Invertavatar: Incremental gan inversion for generalized head avatars. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–10, 2024. 2

[86] Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, BIN FU, Tao Chen, Gang YU, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 4, 5

[87] Xiaozheng Zheng, Chao Wen, Zhaohu Li, Weiyi Zhang, Zhuo Su, Xu Chang, Yang Zhao, Zheng Lv, Xiaoyuan Zhang, Yongjie Zhang, et al. Headgap: Few-shot 3d head avatar via generalizable gaussian priors. *arXiv preprint arXiv:2408.06019*, 2024. 3

[88] Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C Bühler, Xu Chen, Michael J Black, and Otmar Hilliges. Im avatar: Implicit morphable head avatars from videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13545–13555, 2022. 3

[89] 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

[90] Yiyu Zhuang, Hao Zhu, Xusen Sun, and Xun Cao. Mofanerf: Morphable facial neural radiance field. In *European conference on computer vision*, pages 268–285. Springer, 2022. 3

[91] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Instant volumetric head avatars. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4574–4584, 2023. 3