AvatarPointillist: AutoRegressive 4D Gaussian Avatarization

Supplementary Material

In the supplementary materials, we first discuss the limitations of our proposed method (Sec.A). We then offer a perspective on the future directions of our approach (Sec.B). Following that,We conduct additional ablation studies to explore various model architectures for our AR model(Sec.C). Additional implementation details are provided, including the training data, specific procedures for each model, and other relevant training configurations (Sec.D). We also include further comparisons and visual results to showcase the effectiveness of our method (Sec. E). Finally, we present more qualitative results in the supplementary video.

A. Limitations

The limitation of our method is the generation time for the canonical 3DGS point cloud. Due to the sequential nature of our auto-regressive model, this process takes approximately 30 minutes. It is important to note, however, that this is a one-time, upfront cost. Once the canonical point cloud is generated, it is fixed, and this initial cost does not impact the subsequent animation and driving speeds.

B. Future Work

We believe the most significant potential of our auto-regressive framework lies in its alignment with scaling laws [4], akin to Large Language Models (LLMs). Currently, our generalization to in-the-wild data is constrained by the limited scale of the available training set. In the future, we aim to overcome this by leveraging massive-scale synthetic data and expanding to larger, diverse 4D datasets. This data scaling will enable us to move beyond simple supervised learning toward a complete "Pre-training to Post-training" paradigm [8], incorporating Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL) [1, 5] to align the model with human aesthetic preferences. We hope this roadmap—treating avatar generation as a scalable, generalist foundation model problem—serves as a key inspiration for the community.

C. More Ablation Studies

Impact of AR architecture. We further investigated the influence of the auto-regressive architecture by replacing our standard decoder-only Transformer with a hierarchical Hourglass Transformer, inspired by Meshtron [2]. As shown in Figure S1, the Hourglass architecture yields inferior performance compared to the conventional Transformer decoder. We attribute this performance gap to the fundamental difference between data modalities. The Hourglass architecture relies on a hierarchical downsampling mechanism, where the input sequence length for each Transformer layer is compressed by a fixed factor relative to the previous layer to accelerate processing. While this design is highly effective for meshes with explicit topology and hierarchical surface structures, our Gaussian point clouds lack such inherent connectivity and order. Consequently, the aggressive downsampling in the Hourglass architecture leads to a loss of critical geometric information rather than an efficiency gain in our context.

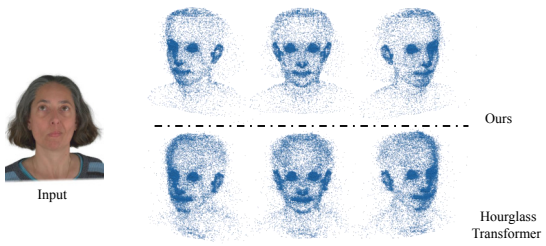

Figure S1. Influence of the auto-regressive architecture. We use Hourglass Transformer to generate the Gaussian point for comparison.

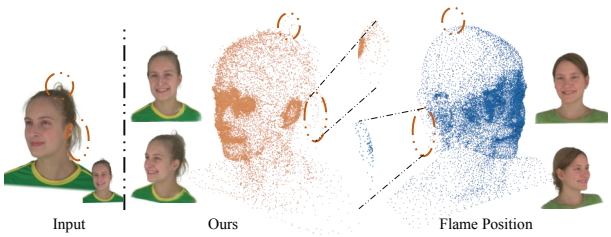

Figure S2. The impact of our AR model. The Flame Positions method skips our AR model and directly uses 3D vertices from the canonical FLAME mesh as input to the Gaussian decoder, which then predicts offsets to refine these fixed points.

Impact of whether to use an AR model. We further provide a visual comparison in Figure S2 to demonstrate the impact of using our AR model for Gaussian point cloud generation. The baseline, denoted as FLAME Position (consistent with the main paper), mimics the strategy of LAM [3]: it utilizes a point cloud with a fixed topology and relies solely on the Gaussian Decoder to predict offsets for refining these static points. As observed, the FLAME Position method fails to generate plausible geometric details. Furthermore, due to the lack of informative point-wise features (which are inherently provided by our AR

Figure S3. The impact of different fitting processes during data generation. The Original is the GaussianAvatars [6]. Ours means limiting the number of Gaussians per face to a range of $[4, 15]$

model) to guide the decoding process, this baseline exhibits a noticeable identity shift and inferior texture quality.

D. More Implementation Details

Data Construction Details. To construct our training dataset, we first employ VHAP tracking [6] to obtain the FLAME coefficients and camera parameters for each identity in the dataset. Subsequently, we perform per-subject fitting using the GaussianAvatars framework [6].

During this fitting process, we observed that the original GaussianAvatars strategy tends to aggressively split or clone Gaussians on specific mesh faces. This results in a highly skewed point distributio, where a few faces might contain an excessive number of points (e.g., up to 8,000) while the majority contain only a few. Such severe data imbalance poses significant convergence challenges for our auto-regressive network training. To address this, we introduced a density constraint during the fitting stage, strictly limiting the number of Gaussians per face to a range of $[4, 15]$. Empirically, we observed that this regularization stabilizes the distribution. Consequently, the total point count per subject was optimized from approximately 80,000 to a more manageable range of 22,000 to 30,000, which significantly reduces the learning complexity for the AR model. Crucially, we verified that this reduction did not degrade the fitting performance as shown in Figure S3.

Network Architecture. We implement our auto-regressive Transformer using the `x-transformers` library. The model is configured with an embedding dimension of 1024, a depth of 24 layers, and 16 attention heads. To enhance performance and stability, we incorporate Rotary Positional Embeddings (RoPE) [7]and Root Mean Square Normalization (RMSNorm) [9]. For the Gaussian decoder, we adopt a framework similar to LAM [3]. This architecture consists of $L = 12$ Multi-Modal Transformer (MM-Transformer) blocks, where each block is equipped with 16 attention heads and operates with a hidden feature dimensionality of 1024.

E. Additional Comparisons and Visual Results

E.1. Visual Comparisons on VFHQ and in the wild images

As illustrated in Fig S4 and Table S1, our proposed method demonstrates superior robustness and generalization across both the VFHQ dataset and various in-the-wild images, with quantitative results in Table S1 showing that our approach achieves the best performance in key metrics such as $LPIPS$ (0.22), AKD (3.21), and APD (4.24) for self-reenactment, while also leading in cross-reenactment FID (65.84) and APD (5.26), which collectively underscore our model's ability to maintain high visual fidelity and precise motion control compared to existing baselines like Portrait4Dv2, GAGAvatar, and AvatarArtist; qualitatively, the results in the first four rows of Fig. S4 showcase our method's capacity to synthesize consistent facial structures and sharp textures even under extreme poses and varied lighting conditions, outperforming other methods that often suffer from blurring or geometric distortions, yet as highlighted in the fifth row, our model occasionally encounters challenges when faced with extremely atypical hair geometries, leading to a subtle identity shift during the reenactment process—a limitation we attribute to the limited coverage of such complex geometries in our current training distribution, which we aim to address in future work by incorporating even larger-scale datasets like VFHQ and other diverse high-quality video repositories to further enhance the model's geometric modeling capabilities and overall robustness.

E.2. Visual Comparisons with Large-scale 3D Models

In addition to the primary baselines, we provide extensive visual comparisons with recent large-scale 3D generative models, specifically Hunyuan3D 2.0 and Seed3D 1.0, as illustrated in Fig. S5; while these general-purpose 3D generators are capable of producing impressive static geometries, our results demonstrate that AvatarPointillist excels in capturing the fine-grained details and structural consistency essential for animatable 4D avatars. The fundamental motivation for adopting an Autoregressive (AR) Transformer architecture stems from its unique ability to handle variable sequence lengths, which is a critical prerequisite for implementing adaptive sampling in the context of Gaussian avatars.

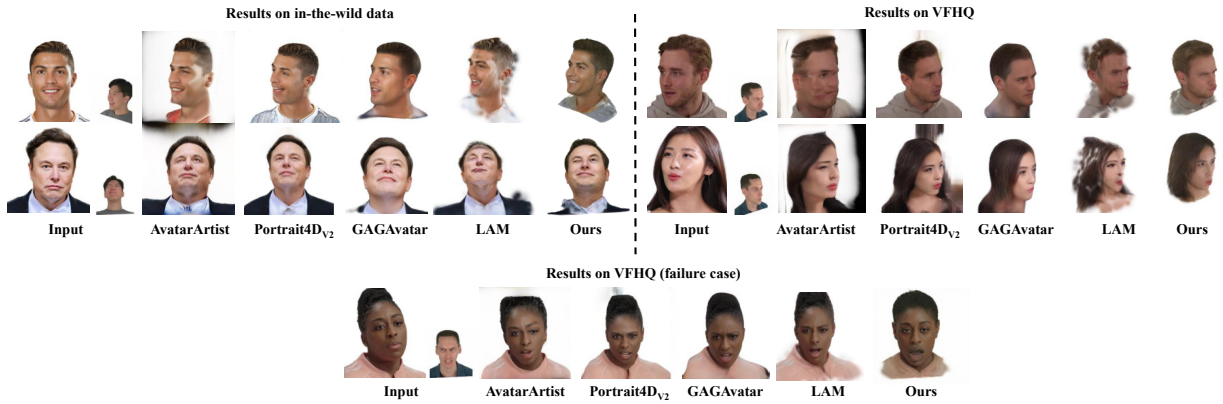

Figure S4. Qualitative comparison on in-the-wild and VFHQ datasets. (Top) Rows 1-2 show results on in-the-wild images with diverse poses and expressions. Rows 3-4 show results on the VFHQ dataset. Our method exhibits superior visual fidelity and identity consistency compared to baseline methods like AvatarArtist, Portrait4Dv2, GAGAvatar, and LAM. (Bottom) Row 5 illustrates a failure case where extreme hair geometry leads to subtle identity distortion, pointing towards future improvements in handling out-of-distribution geometries.

Table S1. Quantitative evaluation on the VFHQ dataset. We report metrics for both self-reenactment and cross-reenactment tasks. ↓ denotes lower is better, while ↑ denotes higher is better. The best and ¡font color="blue"¿second-best¡/font¿ results are highlighted in red and blue, respectively. Our method achieves competitive performance across most metrics, particularly in texture quality (FID/LPIPS) and structural accuracy (AKD/APD).

Method	Self reenactment				Cross reenactment			
	LPIPS ↓	FID ↓	AKD ↓	APD ↓	FID ↓	CLIP ↑	AKD ↓	APD ↓
Portrait4Dv2	0.29	66.60	4.74	**5.08**	78.92	**0.67**	10.84	**6.13**
AvatarArtist	0.26	52.26	4.28	11.72	71.83	0.43	**9.69**	8.93
LAM	0.34	89.32	9.67	20.45	100.54	0.21	18.28	13.10
GAGAvatar	**0.24**	**47.93**	**3.98**	9.64	**70.29**	0.52	10.26	7.89
Ours	**0.22**	**50.24**	**3.21**	**4.24**	**65.84**	**0.57**	**9.38**	**5.26**

Figure S5. Visual comparison between our method, Hunyuan3D 2.0, SeeD3D 1.0.

E.3. Visual Comparisons

In Figure S6, we present additional visual comparisons, demonstrating that our method achieves superior performance. For more visual results, please refer to our video results.

References

[1] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 1

[2] 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. 1

[3] 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. 1, 2

[4] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 1

[5] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 27730–27744, 2022. 1

[6] 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*

Figure S6. Qualitative comparison with state-of-the-art methods. The leftmost column of the figure presents the input images, with the bottom-right corner indicating the target image. The first row illustrates the results of self-reenactment, while the subsequent rows showcase the results of cross-reenactment. Our method demonstrates superior performance in terms of expression and pose consistency, as well as identity preservation. For more visual results, please refer to our video results.

Vision and Pattern Recognition, pages 20299–20309, 2024. 2

[7] Jianlin Su, Yu Lu, Shengfeng Pan, Murtadha Ahmed, Jian Liu, and Wenhan Lv. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 2

[8] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1

[9] Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2