Appendix

A. Implementation Details

A.1. Model Structure

Transformer for Gaussian Deformation We use 2-layers of transformer blocks, each with a cross-attention layer and Feed-Forward layer. Unlike the vanilla transformer [13], we use gated MLP [8] for the Feed-Forward layer.

Encoder Encoder is purely convolutional. It accepts both our modified 8-dimensional Gaussian representations and 6-dimensional UV-based texture features. We obtain the projection layer features from the encoder and the initial input to StyleGAN, as seen in Fig. 8

Volumetric Projection Volumetric Projections utilizes only two convolutions to fuse the 3D feature with the StyleGAN intermediate features. Please see Fig. 8 for more information.

Triplane Generator We use a light-weight StyleGAN to generate the Triplane for Gaussian representation. The structure is similar to EG3D [3]. The latent dimension is 64 as the embedding for the frame index. During the self-reenactment or cross-reenactment, we fixed the frame index to be 0 for inference.

A.2. Training details

Training Strategy We applied StyleGAN-ADA's geometric transformation during the training to improve the robustness. Fig. 9 shows the effectiveness of geometric transformation applied to UV maps during training, which allows the model to learn the relative position between the facial and the torso regions based on the UV map. This strategy significantly improves the self/cross-reenactment during extreme poses. For unseen poses, without geometric transformation, the generated portrait always contains a wrong facial shape.

Training/Inference Time To present a fair comparison between our methods and others, we present the training time and inference time in Tab. 4 for volumetric rendering and editing separately.

A.3. Editing details

We applied Instruct-Pixel2Pixel [2] (IP2P) as the guidance tool for editing following Instruct-NeRF2NeRF [6]. We discover that the raw IP2P model does not present consistency for different views. To address this problem, we first take the novel view synthesis based on our model and feed these data with sampled 200 images to IP2P for finetuning. The finetuning process significantly improves the editing quality.

Method	Training Time	Inference Time
Reconstruction		
IM-Avatar [18]	48h	0.1 fps
PointAvatar [18]	4h	15fps
INSTA [1]	2h	20fps
DVP [7]	12h	25fps
StyleAvatar [14]	6h	25fps
FlashAvatar [15]	0.5h	300 fps
SplattingAvatar [11]	0.5h	80fps
Next3D [12]	10h	20fps
StyleHeat [17]	8h	30fps
OTAvatar [9]	8h	20fps
Ours	2.5h	35fps
Editing		
TokenFlow [5]	30 min	0.5 fps
RAV [16]	30 min	0.8 fps
CoDeF [10]	30 min	40fps
IN2N[6] + GaussianStyle	10 min	35 fps

Table 4. Training/Inference Time Comparison for avatar rendering methods and editing methods

During editing, we freeze all other parameters except the projection layers to the StyleGAN module.

B. Analysis of StyleGAN

We evaluate StyleGAN's ability to generate animatable video portraits, which involves capturing varying expressions, continuous facial motions, and cohesive upper body movement during head rotations. Unlike the aligned images in the pre-trained FFHQ dataset, animatable portraits are often unaligned and captured in diverse settings, with a variety of head positions and orientations.

To assess StyleGAN's effectiveness, we applied the GAN inversion method on both aligned and unaligned portraits, comparing the rendering results. This was crucial to determine if StyleGAN could accurately represent a dynamic portrait video. Our evaluation focused on frames showing extreme left and right head poses from videos as inputs for GAN inversion. This approach tested StyleGAN's limits in rendering realistic, continuous motion and its ability to capture the nuanced changes in facial orientation and expression. The insights gained from this assessment were instrumental in shaping the GaussianStyle framework, enhancing our understanding of the capabilities and limitations of StyleGAN in animatable portrait generation.

Inability for Unaligned portrait generation In Figure 10,

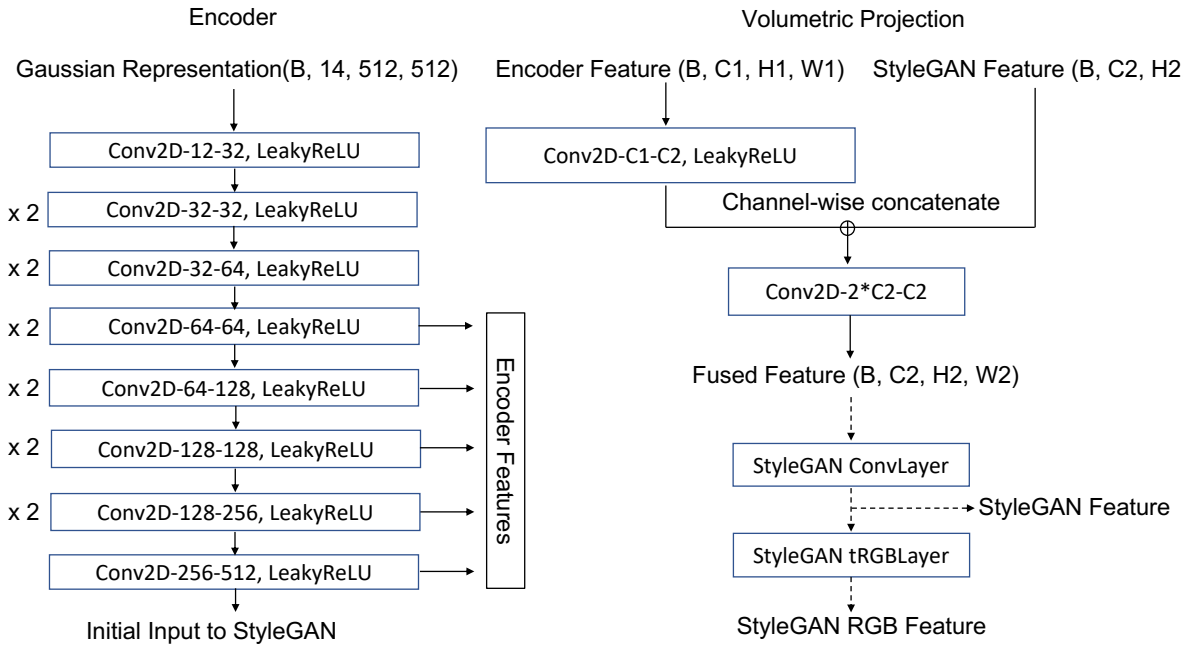

Figure 8. Both encoder and projections are purely convolutional. We obtain intermediate features from the encoder providing StyleGAN with dynamic Gaussian representations

Figure 9. Geometric transformation helps improve the performance of unseen novel views for self/cross-reenactment settings.

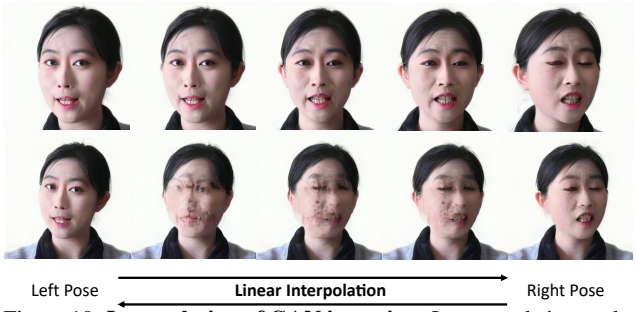

Figure 10. **Interpolation of GAN inversion**: Latent code interpolation between extreme pose parameters along the x-axis for aligned (upper) and unaligned (lower) video portraits.

the linear interpolation of latent codes for extreme poses is presented in two rows: the first for aligned and the second for unaligned inversion. With aligned inversion, interpolating between two style codes yields images that maintain texture quality and exhibit consistent, smooth transitions in facial expressions and poses. This demonstrates StyleGAN's capability in handling aligned facial data. In contrast, the unaligned inversion results reveal StyleGAN's limitations. When processing unaligned faces, particularly in the animatable portrait domain, the model struggles, leading to blurred images. This blurring highlights its difficulty in accurately reconstructing the complex, varied aspects of unaligned faces, including nuanced head movements and expressions. This comparison underlines a key finding: while pre-trained StyleGAN is effective for aligned facial portraits, it falls short in encoding complete portraits with upper body information, unable to capture the full range of portrait dynamics.

StyleGAN's latent Space In addition, we discover that StyleGAN can obtain a consistent neural representation of the target avatar. From the first line in Figure 10, we observe that even though only two images from extreme poses in the left and right directions are used for GAN inversion, StyleGAN is still capable of rendering relatively good intermediate images when interpolating the latent codes. This suggests that after GAN Inversion, the latent space encoded in StyleGAN remains continuous, motion-aware, and can be effectively sampled. Therefore, we can sample a small number of images from the video to perform GAN inversion,

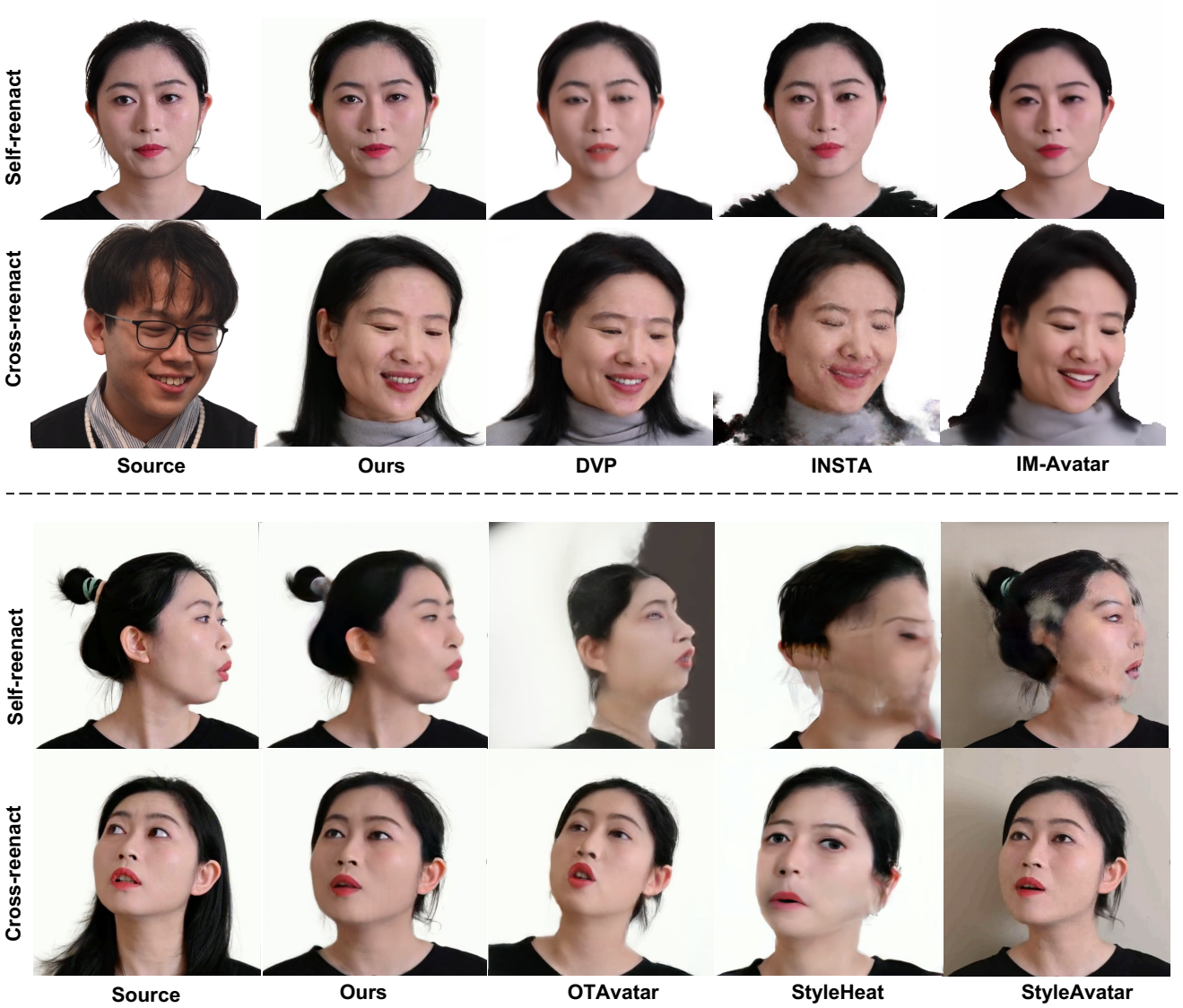

Figure 11. Upper: Comparison with monocular video portrait rendering methods. Lower: Comparison with StyleGAN-based reenactment methods. The comparison suggests that existing methods are unable to deal with unaligned faces and extreme poses.

thereby obtaining the video's neural representation model.

C. Additional Experiments

In this section, we mainly present the comparison with the NeRF-based or 2D or StyleGAN based models for self/cross-reenactment.

C.1. Self/cross-reenactment

We further compared our method with the existing monocular video portrait rendering techniques, including Deep Video Portrait (DVP) [7], INSTA [1], IM-Avatar [18] and StyleGAN-based reenactment models, including StyleHEAT [17], OTAvatar [9] and StyleAvatar [14]. Specifically, OTAvatar and StyleHEAT are designed for aligned one-shot reenactment. To adapt them to unaligned situations and for a fair comparison, we finetuned their models on our video for 10 epochs. It takes about 1 day on a single A6000 to finish fine-tuning.

Fig. 11 shows the comparison between our methods with the existing NeRF-based and StyleGAN-based reenactment methods. INSTA has bad predictions for the non-facial areas. IM-Avatar presents over-smoothing results. DVP utilizs PNCC to Image translation, but struggles with the fine-grained details. StyleHeat cannot deal with unaligned faces and thus generates explicit artifacts during both self/cross-reenactment. OTAvatar utilized a Triplane [3] for geometry-

Figure 12. Next3D, after fine-tuning on target person video, is deficient in domain transfer, as visualized by the artifact for mouth regions.

aware 3D modeling of the target portrait. It cannot disentangle the movement of heads from the torso area. StyleAvatar stands out in cross-reenactment, while not as robust as our methods in dealing with extreme poses.

C.2. Editing

For editing comparison, we further include Next3D [12]. Since it cannot deal with unaligned data, we crop the images from videos. We fine-tuned Next3D on the cropped aligned videos for a fair comparison.

In Fig.12, we apply StyleGAN-NADA[4] to Next3D following fine-tuning on the aligned target portrait videos. Unlike Diffusion, the use of CLIP in Next3D does not ensure consistent intensity for editing. Furthermore, in contrast to our approach, which preserves StyleGAN's domain generalization capability by training only the projection layers while keeping StyleGAN frozen, our fine-tuning on Next3D diminishes its ability to render normal mouth areas, as evidenced by explicit artifacts in these regions.

C.3. Novel View Synthesis

We present the novel view results for 3D geometry evaluations. In case our method is trained on a short monocular portrait video without multi-view inputs, we range the reconstructed results under the viewpoints ranging from -30° to +30°, as shown in Fig. 13, the novel views maintain good visual quality within the range.

D. Baseline Details

D.1. Self/cross-reenactment

To demonstrate the fairness of our comparison with the baselines, we provide specific details on the various baselines and indicate how they differ from the original reports. Since part of the methods do not release source code, we reproduce them by ourselves with fairness.

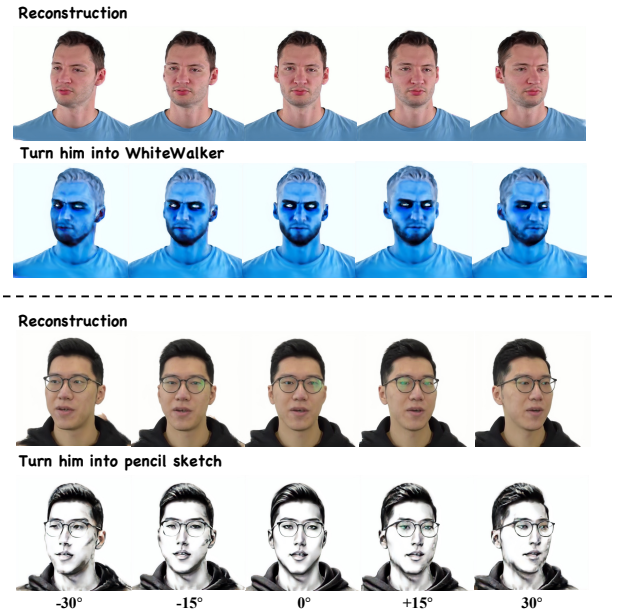

Figure 13. Our reconstruction and editing is consistent for novel views under various conditions.

FlashAvatar We adopt tracking parameters given by the authors and implement the training following the official GitHub repo.

PointAvatar and IM-Avatar The Point-Avatar [19] and IM-Avatar [18] shares the same data preprocessing. We follow the official report to perform the reconstruction.

SplattingAvatar This work adopts the same data preprocessing as in the previous two, we follow the official GitHub repo to reimplement the code.

INSTA We follow the provided official pipeline in the report.

StyleAvatar We reprocess our data via the FaceVerse in StyleAvatar and retrain it from the code in the official repo.

D.2. Portrait Editing

We compared our method with both guidance-based and video-based editing methods. Given the limitations of CoDEF and TokenFlow in handling long video sequences and the increasing GPU memory requirements with video length, respectively, we standardized our evaluation on 3-second video segments, roughly comprising 75 frames for a balanced comparison.

TokenFlow It first did inversion and then editing. We followed the official code provided by TokenFlow [5] for data preprocessing and editing.

Rerender-A-Video We apply the same prompt as that used in TokenFlow for video-based editing following the officially released code by RAV [16]

CoDeF CoDeF's editing process involves modifying a canonical image via Instruct-Pix2Pix and generating the final edited video according to the deformation field. For the

other procedures, we follow the officially released code by CoDeF [10] for data processing, training, and editing.

Insturct-NeRF2NeRF Compared with the original setting in IN2N [6], instead of training the model from scratch and iteratively updating the dataset. We selected a subset containing 200 images with our novel view synthesis as psueduo ground-truth for the model to finetune the model. It takes about 10 epochs to converge.

E. Metrics Detail

Peak Signal-to-Noise Ratio (PSNR). The PSNR is used to eval the generated image quality with ground truth. It is widely used in the field of evaluation image generation

Learned Perceptual Image Patch Similarity (LPIPS). The LPIPS is to apply the perceptual function at the patch level to calculate the feature distance between the generated image and ground truth.

Structural Similarity Index (SSIM). SSIM evaluates the visual impact of three key components: luminance, contrast, and structure.

Blind Image Spatial Quality Evaluator (BIQ). It is a metric to evaluate the generated images without ground truth.

F. Limitations

Although GaussianStyle is able to synthesize photo-realistic and fully animatable head avatars with editing capabilities, there are still areas for improvement:

(1) GaussianStyle relies on video tracking parameters. Inaccurate tracking of landmarks and expressions might introduce potential errors into our model, leading to artifacts and degraded facial details. Our method could benefit from a more accurate video tracking estimation method or corrective operations.

(2) GaussianStyle utilizes tracking parameters for Gaussian Point Deformation, which could introduce errors due to a lack of explicit regularization for landmark matching. In addition, the tracking always present the average expression but cannot capture the extreme expressions. Exploring more robust and accurate techniques could open new directions for future work.

(3) GaussianStyle is still sensitive to extreme views and poses. For out-of-domain camera views and head poses, our methods show degradation in rendering, as illustrated in Fig. 13.

G. Ethical Consideration

Our research primarily focuses on simulating high-fidelity facial avatars. However, due to its photo-realistic facial rendering capabilities, there exists a potential for misuse. For example, creating speech videos of public figures portraying events or statements that never occurred. The risk of such abuses is a longstanding concern in the field of AI-synthesized photo-realistic humans, evident in phenomena like deepfake swapping and talking head generation.

While it is challenging to completely prevent the misuse of this technology, our paper contributes by providing a technical analysis of facial synthesis. This insight allows users to better understand the field and recognize the limitations of AI synthesis to a certain extent, including aspects like tooth detail and temporal consistency.

Furthermore, we advocate for responsible usage practices. These include measures like embedding watermarks in generated videos and employing synthetic face detection technologies for photo-realistic portraits. Such steps are crucial in mitigating the risks associated with this technology.

References

[1] *Instant Volumetric Head Avatars*, 2023. 1, 3

[2] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022. 1

[3] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. 1, 3

[4] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022. 4

[5] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arxiv:2307.10373*, 2023. 1, 4

[6] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. *arXiv preprint arXiv:2303.12789*, 2023. 1, 5

[7] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollöfer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37 (4):163, 2018. 1, 3

[8] Hanxiao Liu, Zihang Dai, David R. So, and Quoc V. Le. Pay attention to mlps, 2021. 1

[9] Zhiyuan Ma, Xiangyu Zhu, Guojun Qi, Zhen Lei, and Lei Zhang. Otavatar: One-shot talking face avatar with controllable tri-plane rendering. *arXiv preprint arXiv:2303.14662*, 2023. 1, 3

[10] Hao Ouyang, Qiuyu Wang, Yuxi Xiao, Qingyan Bai, Juntao Zhang, Kecheng Zheng, Xiaowei Zhou, Qifeng Chen, and Yujun Shen. Codef: Content deformation fields for temporally consistent video processing. *arXiv preprint arXiv:2308.07926*, 2023. 1, 5

[11] Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. SplattingAvatar: Realistic Real-Time Human Avatars with Mesh-Embedded Gaussian Splatting. In *Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

[12] Jingxiang Sun, Xuan Wang, Lizhen Wang, Xiaoyu Li, Yong Zhang, Hongwen Zhang, and Yebin Liu. Next3d: Generative neural texture rasterization for 3d-aware head avatars. In *CVPR*, 2023. 1, 4

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 1

[14] Lizhen Wang, Xiaochen Zhao, Jingxiang Sun, Yuxiang Zhang, Hongwen Zhang, Tao Yu, and Yebin Liu. Styleavatar: Real-time photo-realistic portrait avatar from a single video. *arXiv preprint arXiv:2305.00942*, 2023. 1, 3

[15] Jun Xiang, Xuan Gao, Yudong Guo, and Juyong Zhang. Flashavatar: High-fidelity head avatar with efficient gaussian embedding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

[16] Shuai Yang, Yifan Zhou, Ziwei Liu, , and Chen Change Loy. Rerender a video: Zero-shot text-guided video-to-video translation. In *ACM SIGGRAPH Asia Conference Proceedings*, 2023. 1, 4

[17] 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. 1, 3

[18] 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. 1, 3, 4

[19] 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 (CVPR)*, 2023. 4