GaussianAvatar-Editor: Photorealistic Animatable Gaussian Head Avatar Editor

Xiangyue Liu[1] Kunming Luo[1] Heng Li[1] Qi Zhang[2] Yuan Liu[1] Li Yi[3] Ping Tan[1†]

[1] Hong Kong University of Science and Technology
[2] Tencent AI Lab [3] Tsinghua University

Figure 1. We introduce GaussianAvatar-Editor, a method for text-driven editing of animatable Gaussian head avatars with fully controllable expression, pose, and viewpoint. We show qualitative results of our GaussianAvatar-Editor at the inference time above. Our edited avatars can achieve photorealistic editing results with strong spatial and temporal consistency.

Abstract

We introduce GaussianAvatar-Editor, an innovative framework for text-driven editing of animatable Gaussian head avatars that can be fully controlled in expression, pose, and viewpoint. Unlike static 3D Gaussian editing, editing animatable 4D Gaussian avatars presents challenges related to motion occlusion and spatial-temporal inconsistency. To address these issues, we propose the Weighted Alpha Blending Equation (WABE). This function enhances the blending weight of visible Gaussians while suppressing the influence on non-visible Gaussians, effectively handling motion occlusion during editing. Furthermore, to improve editing quality and ensure 4D consistency, we incorporate conditional adversarial learning into the editing process. This strategy helps to refine the edited results and maintain consistency throughout the animation. By integrating these methods, our GaussianAvatar-Editor achieves photorealistic and consistent results in animatable 4D Gaussian editing. We conduct comprehensive experiments across various subjects to validate the effectiveness of our proposed techniques, which demonstrates the superiority of our approach over existing methods. More results and code are available at: https://xiangyueliu.github.io/GaussianAvatar-Editor/.

1. Introduction

The 3D reconstruction of head avatars using the radiance field-based representation [19] has shown unparalleled photorealistic rendering quality and impressive animatable results. This is critical for visual communications, immersive telepresence, movie production, and augmented or virtual reality. Recently, 3D Gaussian Splatting (3DGS) [14] proposes a GPU-friendly differentiable rasterization pipeline that employs an explicit point-based representation, achiev-

ing superior rendering quality compared to NeRF for novel view synthesis while maintaining real-time performance. 3DGS has been utilized in various downstream applications, particularly head avatar reconstruction [5, 25, 30, 37, 39] with real-time rendering for novel poses and expressions.

Although 3DGS-based avatar reconstruction exhibits remarkable animations, it is essential to incorporate advanced customization options, such as texture editing, shape manipulation, and accessory generation, to accommodate the diverse needs of users. With the rapid advancement of 2D diffusion-like text-to-image (T2I) techniques [27, 28], generative text-driven 3D editing [9, 43, 44] has emerged as a novel approach, complementing previous 3D style transformation and shape manipulation methods [4, 23, 38]. Specifically, Instruct-NeRF2NeRF [9] employs an image-based diffusion model to modify the rendered image by the text prompt, and subsequently updates the 3D radiance field with the edited image. Text-driven 3D editing framework produces promising results on view consistency, enabling more flexible and enhanced editing through text control.

To enable the editing of head avatars, a straightforward solution is to introduce such text-driven editing strategy in Gaussian head avatars. However, challenges remain in editing animatable 3D head avatars using text instructions, particularly regarding anti-occlusion editing in motion-occlusion regions (*e.g.*, teeth occluded by the mouth, eyeballs occluded by eyelids, nosehole occluded by the nose tip) and maintaining spatial-temporal consistency in the editing region throughout the animation process as shown in Fig. 3 and Fig. 10. Specifically, motion occlusions occur when certain parts of the avatar are temporarily obscured by other parts, such as when the lips obscure the teeth as shown in Fig. 10. The occluders can easily affect the Gaussians of the occluded part, leading to artifacts and inconsistencies when animating edited avatars. Meanwhile, the edited images at different timesteps and viewpoints may not be consistent with each other, which also greatly degenerates the generation quality.

To address the above challenges, we introduce our method, *GaussianAvatar-Editor*, to edit animatable head avatars. Specifically, to overcome the incorrect editing caused by occlusions, we propose a novel activation function applied in Gaussian alpha blending for anti-occlusion. To improve the 4D consistency, we apply adversarial learning in the editing framework to reduce the impact of inconsistent supervision signals from diffusion-based editors, greatly improving editing quality. Some results from our GaussianAvatar-Editor in several challenging scenarios are shown in Fig.1. In both qualitative and quantitative comparisons, our method consistently outperforms existing methods in novel views, poses, and expressions.

To summarize, our main contributions are threefold.

- We propose an innovative activation function applied in the Gaussian alpha blending, making our framework robust to multi-layer surfaces.
- We introduce an adversarial learning framework to learn from the 2D diffusion-based editor, which reduces the impact of inconsistent supervision signals and improves the quality of animatable head editing.
- Building on the proposed activation function and adversarial learning, we introduce GaussianAvatar-Editor, which achieves high-quality editing and ensures spatio-temporal consistency in challenging scenarios.

2. Related Works

Text-driven Editing. Diffusion models for text-to-image generations [27, 28] have impressive capability in generating diverse, high-quality images from textual prompts. This innovation has led to a variety of applications, such as image-to-image translation [2, 3, 10, 13, 18, 21] and controllable generation [34, 40]. The advancements have also brought significant progress to numerous 3D tasks, such as text-driven editing, benefiting from the abundant prior knowledge of pre-trained text-to-image models. Some works [1, 33, 35, 36] leverage a CLIP model to edit reference images and lift to 3D space through NeRF optimization. Instruct-NeRF2NeRF [9] and Instruct 3D-to-3D [12] distill 3D scenes from a pretrained text-driven image editing model[3]. TextDeformer [8] and Texture [26] achieve geometry and texture modification according to text prompts, respectively. Vox-E [31] and DreamEditor [43] leverage the SDS loss [24] to perform local editing in 3D space. TIP-Editor [44] introduces a novel approach for accurately controlling the appearance of specified 3D regions with both text and image prompts. The editing ability is further enhanced by upscaling to 4D with dynamic scene representations like 4D NeRF. Control4D [32] combines 4D representation with GAN to achieve better spatial-temporal consistency in dynamic scene editing. Our method capitalizes on 3DGS, which achieves real-time renderings with high-quality and text-driven editing ability of InstructPix2Pix [3], and achieves spatial-temporal consistent editing with given textual instructions.

Head Avatar Reconstruction and Editing. A main line of head avatar reconstruction integrates human priors with neural representations. For instance, NerFACE [7] conditions a dynamic NeRF on extra facial expression parameters from the 3DMM model, to reconstruct the 4D facial avatar from monocular video. IMAvatar [42] represents the expression- and pose-related deformations from the canonical space via learned blendshapes and skinning fields, allowing generalization to unseen poses and expressions. INSTA [45] reconstructs a deformable radiance field based on neural graphics primitives and greatly accelerates the training and inference. Recently, many 3DGS-based methods [6, 25, 29, 41] have shown superior performance

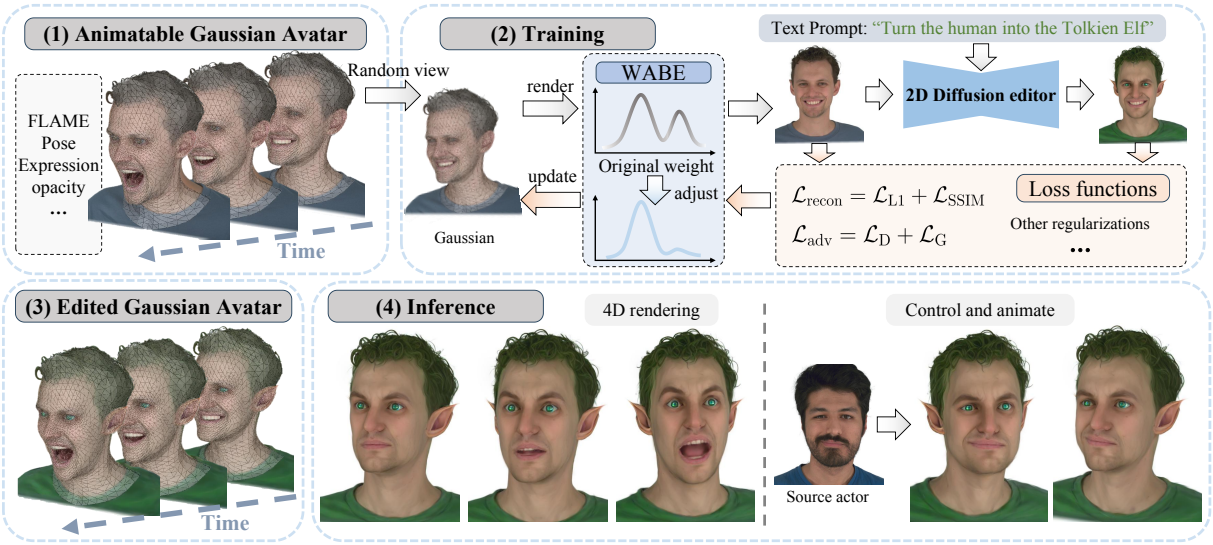

Figure 2. The overview of our method. We follow a render-edit-aggregate optimization pipeline as in Instruct-NeRF2NeRF [9]. We introduce a Weighted Alpha Blending Equation (WABE) to overcome the motion occlusion problem and our novel loss functions to enhance the spatial-temporal consistency. Our edited avatars can generate high-quality and consistent 4D renderings and can be controlled by other actors.

in speed and texture. AvatarStudio [22] reconstructs dynamic digital avatars from multi-view videos and achieves editing by applying a text-driven diffusion model individually on multiple keyframes and optimizing to a unified appearance volume. Thus, its editing cannot be generalized to new expressions and may result in artifacts while handling expressions with significant changes. We address these challenges by considering the differences arising from expressions and poses and achieving high-quality editing that maintains spatial-temporal and spatial-animatable consistency.

3D Gaussian Head Avatar. Various methods [5, 25, 30, 37, 39] attempt to bring Gaussian Splatting to dynamic 3D human head avatar reconstruction. GaussianAvatars [25] proposes binding 3D Gaussian to the FLAME [16] model mesh. Specifically, GaussianAvatars [25] initializes a 3D Gaussian at the center of each FLAME [16] model triangle and uses a binding strategy to support Gaussian splats that densify and prune while maintaining the binding relations. Then, it optimizes the 3D Gaussian and the FLAME [16] model in an end-to-end fashion. In this work, we pioneer the adaptation of 3D Gaussian splatting to Animatable Head Avatar Editing tasks, aiming to achieve photorealistic editing and reenactment to different actors, executing the advantages of Gaussian Splitting representation for the first time in this context. Considering gradients from visible pixels (non-occluded regions) may erroneously propagate to non-visible Gaussians (occluded parts), we specifically designed an activation function for Gaussian alpha blending to handle the motion-occluded regions.

3. Preliminary

3.1. 3D Gaussian Splatting

Gaussian Splatting [14] represents 3D scenes using Gaussian spheres $\{G_k \mid k = 1, \ldots, K\}$, where each Gaussian G_k is defined by the point center $\mu_\mathbf{k}$, and a covariance matrix $\mathbf{\Sigma_k}$ as,

$$G_k(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\mu_\mathbf{k})^\top \mathbf{\Sigma_k}^{-1}(\mathbf{x}-\mu_\mathbf{k})}. \tag{1}$$

The covariance matrix $\mathbf{\Sigma_k}$ is parameterized by a rotation matrix $\mathbf{R_k}$ and a scaling matrix $\mathbf{S_k}$ as $\mathbf{\Sigma_k} = \mathbf{R_k}\mathbf{S_k}\mathbf{S_k}^\top\mathbf{R_k}^\top$.

During rendering, 3DGS [14] employs spherical harmonics $\mathbf{c_k}$ to model view-dependent color and applies α-blending of different Gaussians according to the depth order $1, ..., K$ as,

$$\mathbf{C}(\mathbf{x}) = \sum_{k=1}^{K} \mathbf{c_k}\alpha_k \prod_{j=1}^{k-1}(1-\alpha_j). \tag{2}$$

3.2. Gaussian Avatar

The work of GaussianAvatars [25] binds 3D Gaussians $\mathcal{G}$ to the underlying animatable FLAME [16] model to represent a head avatar M by

$$M(\beta, \theta, \psi, \mathcal{G}) = W(T(\beta, \theta, \psi), J(\beta), \theta, \mathcal{G}), \tag{3}$$

where β is the shape, θ is the pose, ψ is the expression, W means deformation with predefined skinning weights, T is the pose-dependent shap, and J is the joint points. Gaussians $\mathcal{G}$ are defined on the FLAME model and will be transformed along with the motion of the head avatar. It achieves

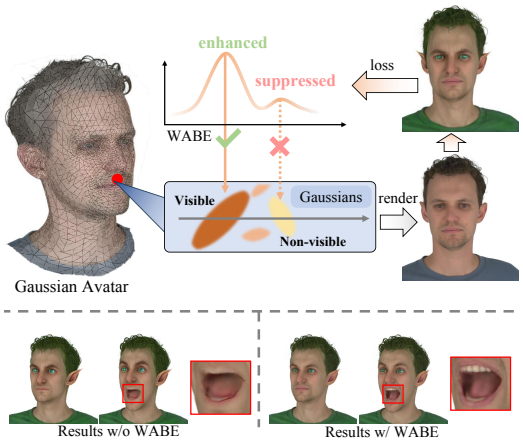

Figure 3. Illustration of the Weighted alpha blending equation (WABE), which is adjusted to suppress non-visible parts while enhancing visible parts. Lower left: results when WABE is disabled. Lower right: when WABE is enabled, motion-occluded regions like teeth and tongue can be successfully optimized.

photorealistic rendering and controllable animation at the same time. In this paper, we design a text-driven method to edit Gaussian avatars.

4. Method

The overview of our proposed method is illustrated in Fig. 2. Given an animatable Gaussian avatar built according to the method in GaussianAvatars [25], we follow the render-edit-aggregate method similar to Instruct-NeRF2NeRF [9] to update the avatar gradually. Specifically, we first randomly sample a training view and render an image using the Gaussian avatar. We then edit this rendered image with 2D diffusion-based editors [3] according to the text prompt provided by users. Finally, we compute loss functions between the rendered and edited images and back-propagate the gradients to refine the Gaussian avatar.

4.1. Challenges in Gaussian Avatar Editing

However, unlike reconstructing Gaussian avatars from multi-view videos, text-driven editing of these Gaussian avatars presents significant challenges.

Motion occlusion. A key challenge is brought by the occlusions in the motion sequence, which complicates the convergence of the optimization process. Specifically, when optimizing 3D Gaussians using gradients derived from supervision images, the α-blending rendering technique in 3DGS [14] updates all 3D Gaussians indiscriminately, despite whether these Gaussians are visible from the current viewpoint. In our scenario, gradients from visible pixels (e.g., pixels on occluders) may erroneously be propagated to invisible Gaussians (e.g., pixels on occluded parts). For example, as shown in Fig. 10, the lip might occlude the teeth, the eyelid might occlude the eyeball, the nose might occlude the nostril, etc. When occlusion happens, the gradient should be stopped at occluders without affecting the occluded parts.

4D consistency. Another key challenge is maintaining 4D spatial and temporal consistency after editing, *e.g.*, the same facial point should be the same over time and across views after editing. While some recent works, such as Instruct-NeRF2NeRF [9], introduce a render-edit-aggregate method to mitigate multi-view inconsistency in static 3D scene editing, ensuring 4D consistency is significantly more challenging.

4.2. Occlusion-aware Rendering and Editing

As discussed earlier, the α-blending in 3DGS [14] updates all 3D Gaussians along the ray during the training process, leading to poor editing results in regions with severe motion occlusions, as shown in Fig. 3 and Fig. 10. Ideally, the correct approach would be to update only the visible 3D Gaussians during the editing process while preserving the 3D Gaussians in the invisible regions. Motivated by this, we propose a modified rendering equation, referred to as the weighted alpha blending equation (WABE), specifically tailored for Gaussian avatar editing.

Weighted alpha blending equation (WABE). Ideally, we only want to update the visible Gaussian during editing while keeping the invisible Gaussian unchanged. This inspires us to seek a blending function that can make the editing process aware of the visible and invisible parts. We replace the original α-blending function, i.e. Eq. 2, of the Gaussian splatting as follows,

$$\mathbf{C}(\mathbf{x}) = \sum_{k=1}^{K} w_k \mathbf{c}_k \alpha_k \prod_{j=1}^{k-1} (1 - \alpha_j), \tag{4}$$

where we add an additional term w_k here to model the visibility of the current Gaussian. Invisible Gaussians will have zero weights. To achieve this goal, as illustrated in Fig. 3, we design the weighted function w_k as follows,

$$w_k = e^{-\beta(1-\prod_{j=1}^{k-1}(1-\alpha_j))}, \tag{5}$$

where β controls the distribution of weights between layers. $\prod_{j=1}^{k-1}(1-\alpha_j)$ is the probability of not being occluded by Gaussians in front of the current one, i.e. the visibility of the current Gaussian. According to our definition, w_k will decrease to 0 with the reduction of visibility, and w_k equals 1 if the Gaussian is fully visible. A larger value of β makes the weights w_k change more fast, leading to more pronounced transitions between layers. This produces a stronger blending effect with sharper changes in transparency. We set β to 6 in all of our experiments. As shown in Fig. 3, when our WABE is enabled, the editing 'Turn him into the Tolkien Elf' only affects the skin and does not change the teeth.

Figure 4. Our results on novel view synthesis. We show our edited results using the text prompt *"Turn her into the Tolkien Elf"*.

Figure 5. Comparison on novel view synthesis. Our method produces more high-quality and multi-view consistent results than baselines.

4.3. 4D Consistent Editing

Editing reconstruction loss. We apply the Instruct-Pix2Pix [3] model to generate edited images $\mathbf{E}_i^t$. Then, to edit the Gaussian avatar, we use the reconstruction loss as the L1 norm and SSIM loss [11] between the rendered image $\mathbf{C}_i^t$ and the edited image $\mathbf{E}_i^t$ as follows,

$$\begin{aligned}\mathcal{L}_{\text{recon}} &= \mathcal{L}_{\text{L1}} + \mathcal{L}_{\text{SSIM}} \\ &= \left\|\mathbf{C}_i^t - \mathbf{E}_i^t\right\|_1 + SSIM(\mathbf{C}_i^t - \mathbf{E}_i^t),\end{aligned} \tag{6}$$

where i is the viewpoint index, and t is the time index.

Due to the lack of temporal and spatial consistency in the images edited by instructions, the supervision in Gaussian splatting optimization might lead to conflicts. Inspired by Instruct-NeRF2NeRF [9], we extend its render-edit-aggregate pipeline to the 4D space to gradually optimize the origin avatar towards the final convergent result. Specifically, we first randomly sample a training view i and a time t and render an image $\mathbf{C}_i^t$ using the Gaussian avatar. We then edit this rendered image with 2D diffusion-based editors [3] according to the text prompt provided by users. Finally, we compute loss functions between the rendered image $\mathbf{C}_i^t$ and the edited image $\mathbf{E}_i^t$ and back-propagate the gradients to refine the Gaussian avatar.

Temporal adversarial learning. Since the temporal consistency of the instruction-edited images is not ensured, relying solely on reconstruction loss like Instruct-NeRF2NeRF [9] often leads to blurry or distorted artifacts in results, especially in animations. Thus, we introduce a temporal adversarial learning scheme to improve consistency in different time steps.

Previous work [17] has demonstrated the effectiveness of conditional adversarial training in preventing blurry rendered images by training a discriminator to determine true or fake images of different viewpoints. This inspires us to extend this conditional adversarial loss to enforce temporal consistency, which alleviates blurry artifacts in rendered images. More specifically, we train a discriminator $\mathcal{D}$ to distinguish real and fake image pairs. The real image pairs P_{real} consists of $\mathbf{E}_i^t$ and $\mathbf{E}_i^t - \mathbf{E}_i^k$ where $\mathbf{E}_i$ is the edited image from the 2D image editor InstructPix2Pix [3], and t, k means adjacent timestep. Similarly, a fake pair P_{fake} consists of the rendered images $\mathbf{C}_i^t$ and $\mathbf{C}_i^t - \mathbf{E}_i^k$. The pairs are concatenated in RGB channels and fed into the discriminator $\mathcal{D}$. We optimize the discriminator $\mathcal{D}$ and the edited Gaussian avatar with the following objective functions,

$$\begin{aligned}\mathcal{L}_{\text{D}} &= \mathbb{E}_{\mathbf{R}}[-log(\mathcal{D}(P_{real}))] + \mathbb{E}_{\mathbf{F}}[-log(1-\mathcal{D}(P_{fake}))], \\ \mathcal{L}_{\text{G}} &= \mathbb{E}_{\mathbf{F}}[-log(\mathcal{D}(P_{fake}))].\end{aligned} \tag{7}$$

In this adversarial loss, we compare not only the edited images with the rendered images but also the differences on different timesteps, which forces the model to learn the temporal consistency for better editing quality.

Figure 6. Our results on self-reenactment. Self-reenactment renders held-out unseen head pose and expressions from 16 training camera viewpoints. The bottom part shows the text prompts.

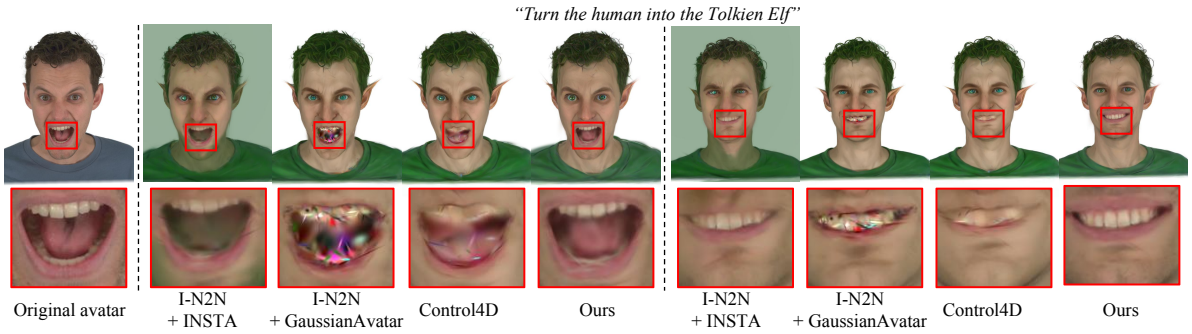

Figure 7. Comparison of self-reenactment. Our edited avatar can correctly produce detailed facial features under unseen expressions and head poses from the same subject.

4.4. Optimization and Regularization

During the training process, we jointly optimize the loss functions mentioned above: $\mathcal{L}_{\text{recon}}$ and $\mathcal{L}_{\text{G}}$ for the edited Gaussian splattings, and $\mathcal{L}_{\text{D}}$ for the discriminator. Inspired by GaussianAvatars [25], we also regularize the Gaussian's position and scales to make Guassians close to the underlying FLAME [16] model by $\mathcal{L}_{\text{const}}$. The total loss formula is expressed as follows:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{recon}} + \lambda_2 \mathcal{L}_{\text{D}} + \lambda_3 \mathcal{L}_{\text{G}} + \lambda_4 \mathcal{L}_{\text{const}}, \qquad (8)$$

where λ means the weights for the loss and we set $\lambda_1 = 10$, $\lambda_2 = 0.01$, $\lambda_3 = 0.01$, and $\lambda_4 = 10$ in all of our experiments. The weights can be adjusted to prioritize different aspects of the training objective, such as reconstruction accuracy, adversarial training, and the perceptual quality.

4.5. Inference

After the optimization of our method, the edited Gaussian head avatar can render the target novel views conditioned on the given expression and pose parameters.

5. Experiments

5.1. Setup

Implementation details. In our pipeline, we first use GaussianAvatars [25] to reconstruct the original animatable Gaussian head avatar from input videos. Then, we edit the original avatar using the input text prompt. Our model is trained using the Adam optimizer with a learning rate of $1e-2$, running for $10,00$ iterations per editing. The total training phase takes about 15 minutes on one 42GB NVIDIA A100 GPU. Given a set of novel expressions, poses, and viewpoints during the inference phase, we can directly drive the edited avatar to the new pose render images of the edited Gaussian avatar.

Dataset. We conducted experiments on the NeRSemble dataset [15], which consists of multi-view videos capturing the front and side views of 8 individuals from 16 camera viewpoints. There are 11 video sequences for each subject, and each video sequence contains approximately 150 frames of different expressions and movements. The first 10 sequences include instructed facial expressions and emotions, while the last sequence records free expressions. During our experiments, all video images are downscaled to a resolution of 802×550. For quantitative evaluation, we use 9 out of 10 video sequences and 15 out of 16 camera views for training and use the last video sequence (free performance) to evaluate the ability of visually cross-identity reenactment.

Evaluation Settings. We evaluate the quality of the edited head avatar from three aspects: (1) **Novel-view rendering** that uses the edited avatars to render images with training head pose and expressions from held-out camera viewpoints; (2) **Self-reenactment** that renders held-out unseen head pose and expressions from 16 training camera

Figure 8. Our results on cross-identity reenactment. Cross-identity reenactment animates the avatar to render images with unseen head poses and expressions from sequences of a different actor. The bottom part shows the text prompts.

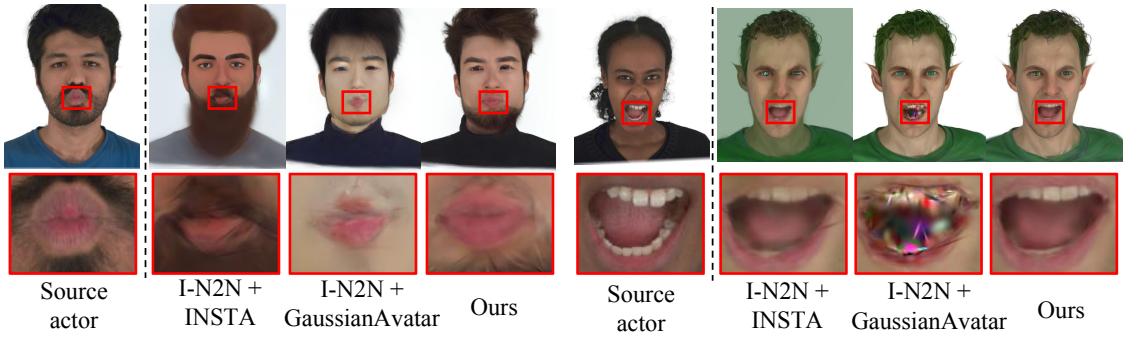

Figure 9. Comparison of Cross-identity reenactment. Different edited avatars are controlled by the same source actor. Our method can render high-quality results with novel expressions, while baseline methods suffer from artifacts.

viewpoints; (3) **Cross-identity reenactment** that uses the avatar to render images with head poses and expressions from sequences of a different subject.

Metrics. To quantitatively evaluate the performance, we employed CLIP Text-Image Direction Similarity (CLIP-S) [9], CLIP Direction Consistency (CLIP-C) [9] to evaluate the edited results, which measure the consistency between renderings of edited avatars and input text prompts.

Baselines. Since no existing method is available to achieve animatable Gaussian avatar editing, we compare our method to the most relevant approaches. (1) **I-N2N+GaussianAvatar**. One important baseline method is to directly apply a static 3D editing approach to 4D Gaussian avatars. Specifically, we apply Instruct-NeRF2NeRF [9] to perform text-drive editing on the reconstructed animatable 4D Gaussians of GaussianAvatars [25]. Though animatable Gaussian avatar editing can be directly achieved in this way, the edited results are far from acceptable, largely due to the motion-occlusion problem presented in Sec. 4.2. (2) **I-N2N+INSTA**. To compare with animatable 4D Gaussian editing based on NeRF [20], we apply Instruct-NeRF2NeRF [9] to perform text-drive editing on the reconstructed animatable 4D NeRF of INSTA [45]. (3) **Control4D**. Another baseline is Control4D [32], a 4D Gaussian editing method designed for GaussianPlanes (spatial triplanes and 3D Gaussian flow). Note that Control4D represents head avatars as implicit parameters, which means Control4D's result cannot be re-animated. Since Control4D's code for dynamic editing has not been released, we reimplement it based on its static Gaussian editing version.

5.2. Head Avatar Editing and Animation

Quantitative results are summarized in Table 1. We refer readers to the video in the supplementary for more qualitative results.

Novel view rendering. As shown in Fig. 4, given a Gaussian avatar and an editing prompt *"Turn the human into the Tolkien Elf"*, our method can produce multi-view consistent and high-quality results. We compare our edited avatars with existing methods for novel view synthesis. Qualitative comparison results in Fig. 5, with two novel view renderings for each method. Both our method and baseline methods can produce multi-view consistent rendering results. However, the results from baseline methods are poorer, especially visible in the added beard, which is also blended on the teeth. In contrast, by addressing the motion occlusion problem during editing, our result can render clear and detailed teeth. Quantitative results are presented in Table 1, which also shows our editing results achieve better consistency with the input text than baselines.

Self-reenactment. Our qualitative results are shown in Fig. 6, and qualitative comparison with baselines are shown in Fig. 7. As we can see, directly applying the method Instruct-NeRF2NeRF [9] to INSTA [45] or Gaus-

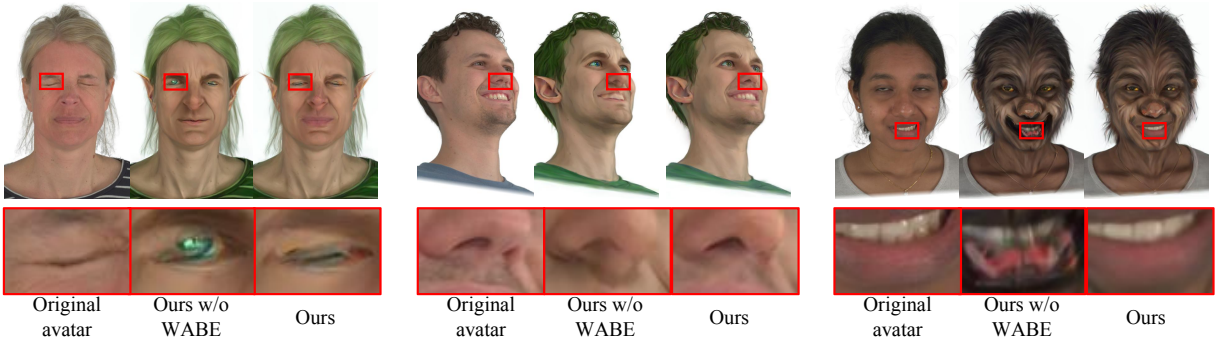

Figure 10. Ablation study of WABE.

	Novel view rendering		Self-reenactment		Cross-identity reenactment	
	CLIP-S ↑	CLIP-C ↑	CLIP-S ↑	CLIP-C ↑	CLIP-S ↑	CLIP-C ↑
INSTA+I-N2N	0.181	0.955	0.042	0.923	0.043	0.936
GA+I-N2N	0.236	0.968	0.044	0.938	0.069	0.941
Control4D	0.222	0.980	0.058	0.938	/	/
Ours w/o WABE	0.236	0.968	0.061	0.948	0.077	0.950
Ours w/o adv	0.266	0.976	0.077	0.950	0.070	0.946
Ours	**0.275**	**0.978**	**0.081**	**0.951**	**0.081**	**0.951**

Table 1. Quantitative comparisons and ablation studies with CLIP-S and CLIP-C. We compare our method with existing methods for novel view rendering, self-reenactment, and cross-identity reenactment. Our method obtains superior results than other methods.

sianAvatars [25] results in serious artifacts at largely different head poses or facial expressions, since it is designed for editing static scenes. Control4D [32] produces better results, but its rendered images are blurry at unseen expressions. Unlike those approaches, our method obtains detailed and realistic rendering results with clear facial features even animated by unseen expressions. Quantity comparisons with baselines in Table 1 also demonstrate the effectiveness of our method.

Cross-identity reenactment. To evaluate the generalization ability of our method, we further drive those edited avatars by expressions and head poses from other actors. As shown in Table 1, our method achieves superior CLIP-S scores and comparable CLIP-C scores as baseline methods. We also show qualitative comparison results in Fig. 8 and Fig. 9. As can be seen, our edited avatars can render better results than baseline methods.

5.3. Ablation Study

WABE. To validate the effectiveness of the proposed WABE for handling motion occlusion, we perform ablation experiments by disabling the WABE in our pipeline. The rendering results without WABE are also shown in Fig. 3 and Fig. 10. The results demonstrate that without WABE, the occluded regions like teeth when open mouth, eyeballs when closed eyes, the lips, the nosehole, etc., produce worse editing results, which leads to worse quantitative results in Table 1. This demonstrates the importance of WABE in handling the occlusion problem.

Adversarial learning mechanism. We also validate the proposed adversarial learning for spatial and temporal consistency. As shown in Table 1, disabling the adversarial learning loss in our pipeline decreases the test scores, especially the CLIP-C score, which demonstrates the importance of our adversarial learning mechanism.

6. Conclusion

In this paper, we have presented GaussianAvatar-Editor, a text-driven framework for realistic animatable Gaussian avatar editing. For the motion occlusion problem where editing gradients would be back-propagated from non-occlusion parts to erroneously update the occlusion parts, we proposed a Weighted alpha blending equation (WABE) to replace the original Gaussian rendering function so as to suppress those erroneous updates. Moreover, to enhance the 4D supervision consistency of the editing supervision, we proposed an adversarial learning framework. By incorporating all these designs together, our method can produce high-quality, realistic editing results for 4D animatable Gaussian avatars. We have conducted comprehensive experiments on various subjects to validate the proposed methods. Both qualitative and quantitative results demonstrated that our method is superior to existing methods.

Limitations. GaussianAvatar-Editor utilizes FLAME model to do animation, which could not animate unmodeled parts like the tongue. We leave this for future exploration.

References

[1] Chong Bao, Yinda Zhang, Bangbang Yang, Tianxing Fan, Zesong Yang, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Sine: Semantic-driven image-based nerf editing with prior-guided editing field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20919–20929, 2023. 2

[2] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *European conference on computer vision*, pages 707–723. Springer, 2022. 2

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 2, 4, 5

[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, et al. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, pages 16123–16133, 2022. 2

[5] Yufan Chen, Lizhen Wang, Qijing Li, Hongjiang Xiao, Shengping Zhang, Hongxun Yao, and Yebin Liu. Monogaussianavatar: Monocular gaussian point-based head avatar. *arXiv*, 2023. 2, 3

[6] Helisa Dhamo, Yinyu Nie, Arthur Moreau, Jifei Song, Richard Shaw, Yiren Zhou, and Eduardo Pérez-Pellitero. Headgas: Real-time animatable head avatars via 3d gaussian splatting. *arXiv preprint arXiv:2312.02902*, 2023. 2

[7] 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. 2

[8] William Gao, Noam Aigerman, Thibault Groueix, Vova Kim, and Rana Hanocka. Textdeformer: Geometry manipulation using text guidance. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 2

[9] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19740–19750, 2023. 2, 3, 4, 5, 7

[10] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 2

[11] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016. 5

[12] Hiromichi Kamata, Yuiko Sakuma, Akio Hayakawa, Masato Ishii, and Takuya Narihira. Instruct 3d-to-3d: Text instruction guided 3d-to-3d conversion. *arXiv preprint arXiv:2303.15780*, 2023. 2

[13] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023. 2

[14] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023. 1, 3, 4

[15] Tobias Kirschstein, Shenhan Qian, Simon Giebenhain, Tim Walter, and Matthias Nießner. Nersemble: Multi-view radiance field reconstruction of human heads. *ACM Transactions on Graphics (TOG)*, 42(4):1–14, 2023. 6

[16] 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.*, 36(6):194–1, 2017. 3, 6

[17] Xiangyue Liu, Han Xue, Kunming Luo, Ping Tan, and Li Yi. Genn2n: Generative nerf2nerf translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5105–5114, 2024. 5

[18] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 2

[19] B Mildenhall, PP Srinivasan, M Tancik, JT Barron, R Ramamoorthi, and R Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, 2020. 1

[20] 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. 7

[21] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023. 2

[22] Mohit Mendiratta Pan, Mohamed Elgharib, Kartik Teotia, Ayush Tewari, Vladislav Golyanik, Adam Kortylewski, Christian Theobalt, et al. Avatarstudio: Text-driven editing of 3d dynamic human head avatars. *arXiv preprint arXiv:2306.00547*, 2023. 3

[23] Juan C Pérez, Thu Nguyen-Phuoc, Chen Cao, Artsiom Sanakoyeu, Tomas Simon, Pablo Arbeláez, Bernard Ghanem, Ali Thabet, and Albert Pumarola. Styleavatar: Stylizing animatable head avatars. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 8678–8687, 2024. 2

[24] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2

[25] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. *arXiv preprint arXiv:2312.02069*, 2023. 2, 3, 4, 6, 7, 8

[26] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. *arXiv preprint arXiv:2302.01721*, 2023. 2

[27] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[28] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 2

[29] Shunsuke Saito, Gabriel Schwartz, Tomas Simon, Junxuan Li, and Giljoo Nam. Relightable gaussian codec avatars. *arXiv preprint arXiv:2312.03704*, 2023. 2

[30] Shunsuke Saito, Gabriel Schwartz, Tomas Simon, Junxuan Li, and Giljoo Nam. Relightable gaussian codec avatars. In *CVPR*, 2024. 2, 3

[31] Etai Sella, Gal Fiebelman, Peter Hedman, and Hadar Averbuch-Elor. Vox-e: Text-guided voxel editing of 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 430–440, 2023. 2

[32] Ruizhi Shao, Jingxiang Sun, Cheng Peng, Zerong Zheng, Boyao Zhou, Hongwen Zhang, and Yebin Liu. Control4d: Dynamic portrait editing by learning 4d gan from 2d diffusion-based editor. *arXiv preprint arXiv:2305.20082*, 2023. 2, 7, 8

[33] Hyeonseop Song, Seokhun Choi, Hoseok Do, Chul Lee, and Taehyeong Kim. Blending-nerf: Text-driven localized editing in neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14383–14393, 2023. 2

[34] Andrey Voynov, Kfir Aberman, and Daniel Cohen-Or. Sketch-guided text-to-image diffusion models. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 2

[35] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3835–3844, 2022. 2

[36] Can Wang, Ruixiang Jiang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Nerf-art: Text-driven neural radiance fields stylization. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 2

[37] Jie Wang, Jiu-Cheng Xie, Xianyan Li, Feng Xu, Chi-Man Pun, and Hao Gao. Gaussianhead: High-fidelity head avatars with learnable gaussian derivation, 2024. 2, 3

[38] 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. 2

[39] Yuelang Xu, Benwang Chen, Zhe Li, Hongwen Zhang, Lizhen Wang, Zerong Zheng, and Yebin Liu. Gaussian head avatar: Ultra high-fidelity head avatar via dynamic gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3

[40] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 2

[41] Zhongyuan Zhao, Zhenyu Bao, Qing Li, Guoping Qiu, and Kanglin Liu. Psavatar: A point-based morphable shape model for real-time head avatar creation with 3d gaussian splatting. *arXiv preprint arXiv:2401.12900*, 2024. 2

[42] 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. 2

[43] Jingyu Zhuang, Chen Wang, Liang Lin, Lingjie Liu, and Guanbin Li. Dreameditor: Text-driven 3d scene editing with neural fields. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–10, 2023. 2

[44] Jingyu Zhuang, Di Kang, Yan-Pei Cao, Guanbin Li, Liang Lin, and Ying Shan. Tip-editor: An accurate 3d editor following both text-prompts and image-prompts. *arXiv preprint arXiv:2401.14828*, 2024. 2

[45] 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. 2, 7