GeneAvatar: Generic Expression-Aware Volumetric Head Avatar Editing from a Single Image
- Supplementary Material -

Chong Bao[1*§] Yinda Zhang[2*] Yuan Li[1*] Xiyu Zhang[1] Bangbang Yang[4]
Hujun Bao[1] Marc Pollefeys[3] Guofeng Zhang[1] Zhaopeng Cui[1†]
[1]State Key Lab of CAD&CG, Zhejiang University [2]Google [3]ETH Zürich [4]ByteDance
https://zju3dv.github.io/geneavatar/

In this supplementary material, we first present an ethics declaration in Section A, followed by detailed implementation aspects in Section B, which covers our model architecture, geometry and texture distillation, and the user study. More experimental results are shown in Section C. Additionally, we include a short video summarizing the method with video results, and an offline webpage for interactive visualization of our editing results.

A. Ethics Declaration

In this paper, we present this ethics declaration to underline our commitment to responsible scientific inquiry within the field of computer vision. Our work uses open-sourced datasets, carefully chosen to ensure that they were collected with the full consent of the participants involved. The privacy and rights of individuals are paramount in our research, and we have taken steps to safeguard these by implementing strict guidelines that govern the use of our research outputs. We acknowledge the importance of diversity and have selected our datasets with the aim of preventing bias, ensuring that our methods are fair and inclusive across various demographics. Our research is purely academic, and any head editing carried out is for the purpose of validating the effectiveness of our methods. We explicitly state that our research does not involve human experimentation and that all human-derived data has been responsibly sourced and vetted for ethical compliance. We affirm that our research is intended solely for scientific advancement and to test the robustness of our methods. There is no intention to vilify or harm any individual or group. Our aim is to contribute to the field of computer vision in a way that is ethically sound, socially responsible, and cognizant of the long-term implications of our work. We embrace open discussions about our ethical approach and are committed to transparency and ethical integrity in all aspects of our research.

*Authors contributed equally.
†Corresponding authors.
§The work was partially done when visiting ETHZ.

B. Implementation Details

B.1. Model Architecture

Our model follows the architecture of 3DMM-based 3DGAN [7] that contains a StyleGAN-based feature generator and a feature decoder. Specifically, the feature generator takes a modification code $\mathbf{z}_{g/n} \in R^{1024}$ as input, and has a mapping network and a feature synthesis network. A mapping network is employed to transform the modification code in $\mathcal{Z}$ space to the code $\mathbf{w} \in R^{14\times512}$ in $\mathcal{W}$ space. The mapping network consists of 3 fully-connected layers with 512 hidden sizes. Then the code w conditions the feature synthesis network following the StyleGAN [4]. The feature synthesis network consists of 7 synthesis convolution blocks, each of which contains 2 convolution layers and a 1×1 convolution layer. The resolutions of 7 synthesis convolution blocks are 4, 8, 16, 32, 64, 128, 512 respectively. The codes in $(2i)$-th and $(2i+1)$-th row of code $\mathbf{w}$ modulate the weights of (i)-th synthesis block. The output of the feature synthesis network is $256 \times 256 \times 32$ neural feature map. We pre-define the UV mapping between the vertices of the 3DMM mesh and neural feature map, and rasterize the neural feature map to the four axis-aligned plane (one parallel to the positive face, two parallel to the side face, one parallel to the top of the head) to generate the tri-plane features. The two side planes are used to collect the features in left-side and right-side faces which will be summed up to generate the final side-plane feature. The modification feature of input query point $\mathbf{x}$ is collected by projecting $\mathbf{x}$ to the tri-plane and summing up the bi-linear interpolated feature from the tri-plane. For geometry editing, the geometry modification decoder takes the modification feature as input and outputs a translation vector to shift the $\mathbf{x}$ to $\mathbf{x}'$. The geometry modification decoder consists of 4 fully connected

Figure A. We show some avatars sampled in the geometry modification learning.

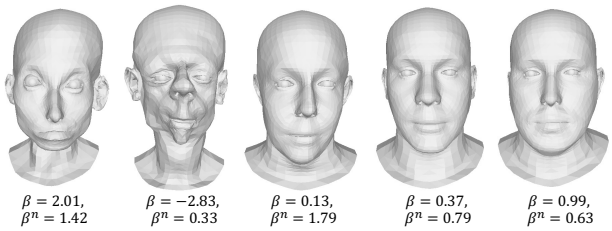

Figure B. We show 3DMM meshes sampled from different shape coefficients β.

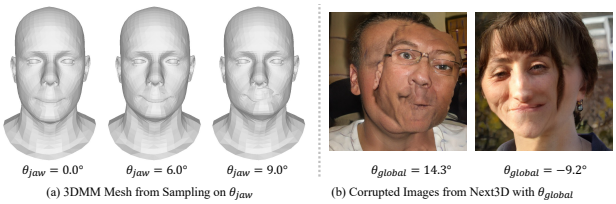

Figure C. We show the 3DMM meshes sampled from different poses of jaw θ_{jaw} and corrupted image generated by Next3D [7] when the 3DMM mesh is sampled with θ_{global}.

layers with 256 hidden sizes and a translation head. For texture editing, the texture modification decoder takes the modification feature as input and outputs a blending weight and modification color value to modify the original color using Eq.(3). The texture modification decoder consists of 4 fully-connected layers with 256 hidden sizes and a blending weight head and a modification color head.

B.2. Geometry Distillation

As illustrated in Fig. B, we observe that the shape of the head is distorted when the mean value $\bar{\beta}$ of 3DMM shape coefficient β is larger than 1.0, e.g., the two heads on the far left deviate from the standard shape definition of the human head. Furthermore, the increasing of the standard deviation β^n of 3DMM shape parameter β will lead to the asymmetry in the shape, e.g., the shape of the first and third head is asymmetrical. Therefore, we sample 3DMM shape parameters β from a normal distribution whose absolute mean and standard deviation are randomly selected within $[0, 1]$, and sample the edit vector β_Δ from the uniform distribution $\mathcal{U}(-3, 3)$ to keep the $\bar{\beta}$ within [-1,1] and $\bar{\beta}$ small as possible.

For 3DMM pose coefficient θ sampling, we only sample different pose coefficients of the jaw θ_{jaw} and keep the others fixed to comply with the 3DMM pose range allowed by Next3D [7], e.g., the generated face is corrupted with θ_{global} since the face is assumed to always locate at the original point without rotation as illustrated in Fig. C(b).

We show some pairs of volumetric avatars that are sampled for geometry modification learning in Fig. A. The proposed geometry distillation scheme can result in a wide range of consistent geometry editing data across expressions and viewpoints, which promotes expression-dependent geometry modification learning. The geometry editing data contains geometry modifications on various facial features across different genders, ages and sex, which promotes the generalization ability of our method.

B.3. Texture Distillation

We show some pairs of volumetric avatars that are sampled for texture modification learning in Fig. D. Our texture distillation scheme enables the generation of a diverse array of texture editing data that is consistent across different expressions and viewpoints. This includes, for instance, partial makeup on the first head, intricate makeup designs on the second head head, and free-style makeup on the third head in Fig. D. Such variety in texture edits greatly enhances the flexibility of our texture modification generator. Furthermore, the texture editing data encompasses modifications on a range of facial features, represented across various genders, ages, and sexes, thereby substantially augmenting the generalizability of our method.

B.4. User Study

Our questionnaire contains 12 editing cases, 6 for geometry editing and 6 for texture editing. These editing cases cover

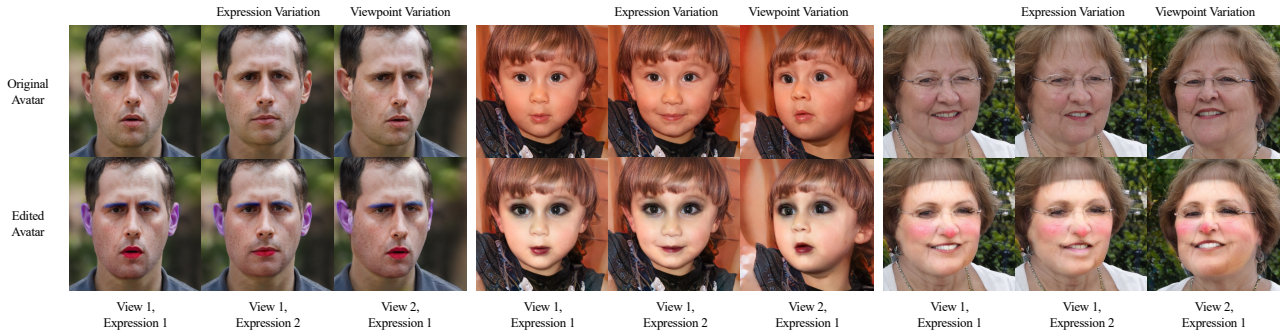

Figure D. We show some avatars sampled in the texture modification learning.

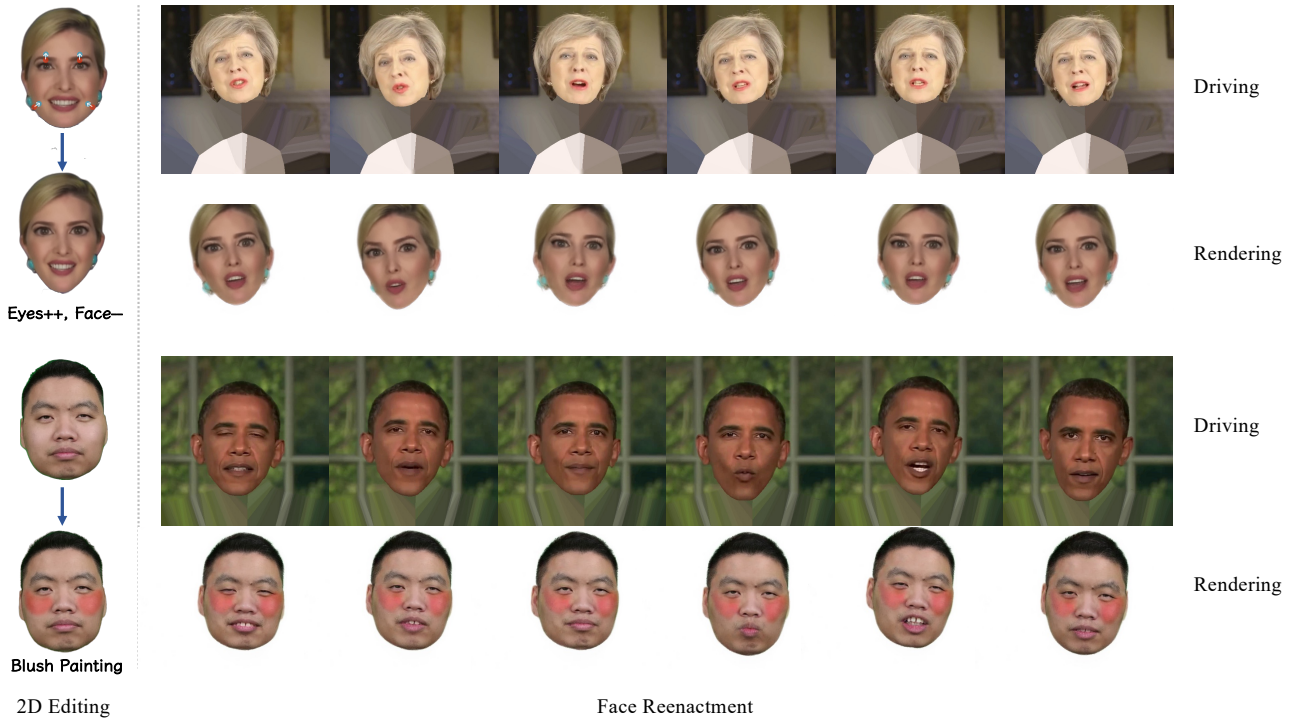

Figure E. We show the face reenactment results on the edited avatars with our modification field.

the editing on 9 heads from the INSTA [8] and NeRFBlendShape [3]. For each editing case, there are 4 questions following the AvatarStudio [6]:

- Which method better follows the given input edited image?
- Which method better retains the identity of the input sequence in the video?
- Which method better maintains temporal consistency in the video?
- Which method is better overall considering the above 3 aspects in the video?

Participants are shown an original image, an edited image, and four videos rendered from four methods side by side, and asked to select one of four methods to answer each question.

B.5. Comparison to 3DMM-based Geometry Editing

Optimization-based 3DMM fitting typically requires dense landmarks (better in 3D) and/or multi-view images to achieve reconstruction quality. However, our goal is to achieve single view-based volumetric avatar editing, where

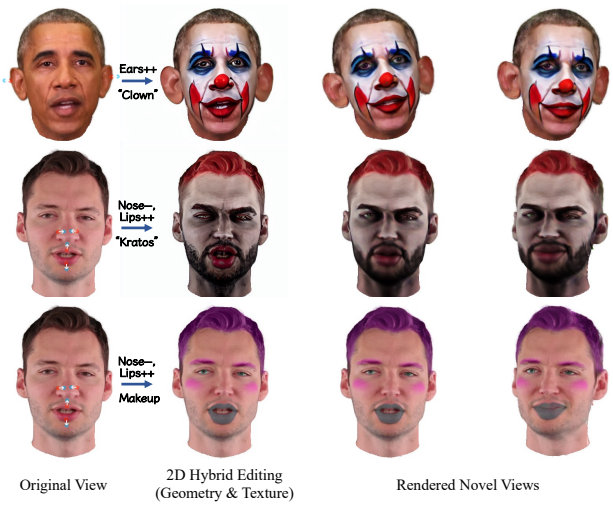

Figure F. We show hybrid editing results with geometry and texture editing.

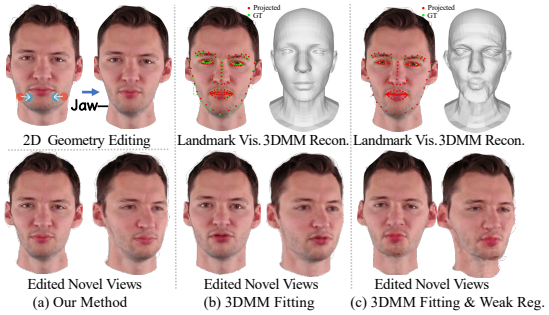

Figure G. We show a more thorough comparison with the 3DMM-based geometry editing. The parametric regularization in 3DMM fitting is tuned to enhance landmark alignment, albeit at the expense of introducing distortions to the resulting 3D geometry.

we only have access to one perspective view. The fitting is error-prone, especially for out-of-domain cases in this setting. As illustrated in Fig. G, 3DMM fitting with 2D landmarks from a single image cannot well constrain the 3D shape no matter with (b) regular regularization or (c) weak regularization (for better landmark fitting). In contrast, (a) our 3D editing uses the learned prior to faithfully guide the editing from limited constraints.

B.6. Editing Efficiency and Model Complexity

Our method takes 75 seconds for geometry editing and 164 seconds for texture editing over a Next3D-based avatar on an RTX 4090 GPU. The editing speed is largely determined by the backbone architecture. Designing an efficient backbone for real-time editing is out of the scope of this paper but an interesting future direction. Our model size is 234 MB. For avatar editing, it requires 9.1 GB GPU memory to perform auto-decoding optimization.

Image identity similarity	Geometry				Texture			
	Roop	PVP	Next3D	Ours	Roop	PVP	Next3D	Ours
Mean ↑	0.8373	0.8704	0.8547	**0.8845**	0.7320	0.8476	0.8500	**0.9147**
Median ↑	0.8447	0.8836	0.8680	**0.8854**	0.7828	0.8608	0.8674	**0.9181**
SD ↓	**0.0264**	0.0400	0.0449	0.0448	0.1173	0.0407	0.0340	**0.0310**
RSD (%) ↓	**3.16**	4.60	5.25	5.06	16.03	4.80	4.00	**3.39**

Table A. We show the mean, median, standard deviation(SD), standard deviation (SD), and relative standard deviation (RSD) of the quantitative comparisons with the PVP [5], Roop [1], Next3D [7] on image identity similarity [2].

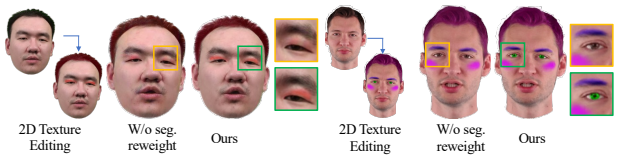

Figure H. We inspect the efficacy of the segmentation-based loss reweighting strategy.

B.7. Statistical analysis of quantitative comparisons

As shown in Tab. A, We show the mean, median, standard deviation (SD), and relative standard deviation (RSD) of image identity similarity below. Our method surpasses other methods in mean and median but also has a small deviation in SD and RSD.

C. More Experiments

C.1. Hybrid Editing

We show the hybrid editing results in Fig. F. We can edit the geometry of the avatar while changing the texture with a text prompt or makeup image. The rendered novel views are consistent across multiple viewpoints and expressions and present vivid appearances, e.g., clown makeup and enlarged eyes on the first head, and "Kratos" makeup and enlarged lips and reduced nose give a fierce appearance on the second head in Fig. F.

C.2. Face Reenactment

We show the results of face reenactment in Fig. E. Our geometry and texture modification seamlessly follow the expressions from the driving video, and present consistent results across various viewpoints and expressions. This provides great potential for the VR/AR and live broadcasts of digital avatars.

C.3. Geometry Visualization on Geometry Editing

We visualize the normal of meshes extracted from the volumetric avatar under various expressions in Fig. K. Given a single edited image, our method faithfully modifies the geometry of the avatars with multi-view consistency, e.g., the enlarged ears with consistent geometry across multiple viewpoints in the last row of Fig. K. Furthermore, our expression-dependent geometry modification

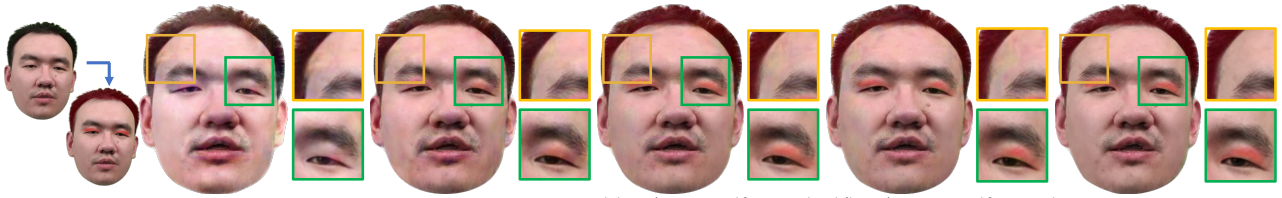

Figure I. We inspect the efficacy of different loss terms in Eq. (5) when performing avatar editing.

Figure J. We inspect the efficacy of the implicit latent space guidance.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
# mod. code = 32+512	25.47	0.8508	0.0966
# mod. code = 128+512	27.69	0.8674	0.0803
# mod. code = 512+512 (ours)	**27.75**	**0.8685**	**0.0798**

Table B. We quantitatively inspect the efficacy of dimensions of the modification latent code on avatar editing.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
W/o Code Guidance	21.30	0.7543	0.6066
Ours	**35.42**	**0.9398**	**0.0308**

Table C. We quantitatively inspect the efficacy of the implicit latent space guidance on the novel view synthesis of edited avatars in training.

seamlessly adapts to different expressions, e.g., the enlarged nose and lips present consistent results across multiple expressions in the second row of Fig. K.

C.4. Ablations

Segmentation-based Loss Reweighting Strategy We inspect the efficacy of the segmentation-based loss reweighting strategy by replacing this strategy with averaging the L2 loss of the whole image during auto-decoding optimization. As depicted in Fig. H, the absence of the reweighting strategy results in an inability to reconstruct fine-grained makeup since these makeups occupy small regions that have a negligible impact on the loss, e.g., the missing red eye shadow on the left head and untouched color of eyes on the right head in Fig. H. In contrast, our method can accurately reconstruct the makeup from a single edited image and present consistent results across multiple expressions.

Implicit Latent Space Guidance We ablate the implicit latent space guidance by fully sampling a modification code of 1024 dimensions from a standard normal distribution instead of the concatenation of a teacher code and a reduced modification code of 512 dimensions during training. We take the training of the geometry modification generator as an example. As shown in Tab. C, we quantitatively evaluate the quality of novel view synthesis on the training data. Specifically, we render images of the edited avatar under novel viewpoints as ground truth, and apply the modification fields from two methods to the original avatar, and quantitatively compare the rendered modified images from two methods with the ground truth. Our methods surpass the method without the implicit latent space guidance in all metrics. The implicit latent space guidance improves the convergences on training data. Then, we evaluate two methods in a novel geometry editing case where auto-decoding optimization is performed to infer the modification field from a single edited image. As illustrated in the Fig. J, the method without the implicit latent space guidance fails to generalize on the novel editing case and results in a blurred and corrupted image. In contrast, our method can faithfully render the image of the edited avatar under novel viewpoint and expression.

Hyper-parameters. As shown in the Tab. B, we hereby provide ablation over dimensions of modification latent space. As illustrated in the Fig. I, we also show the impact of loss weights of Eq. (5). **(a-b)**: The fine-grained makeup cannot be faithfully reconstructed without $\mathcal{L}_2$ or with a weak $\mathcal{L}_2$. **(c-d)**: Some color distortion occurs without regularization $\mathcal{L}_{reg}$ or global appearance constraint $\mathcal{L}_{lpips}$.

C.5. Limitations

As illustrated in Fig. L, We show hard cases by (a-b) adding additional objects (e.g., add hat) and (c-d) changing hairstyle (e.g. add fringe) in the following figure. As shown, our method reconstructs rough but incomplete shapes. The texture also looks blurry due to the missing of proper prior.

References

[1] deepfakes. roop. SomdevSangwan, 2023. Accessed: 2023-10-10. 4

[2] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos

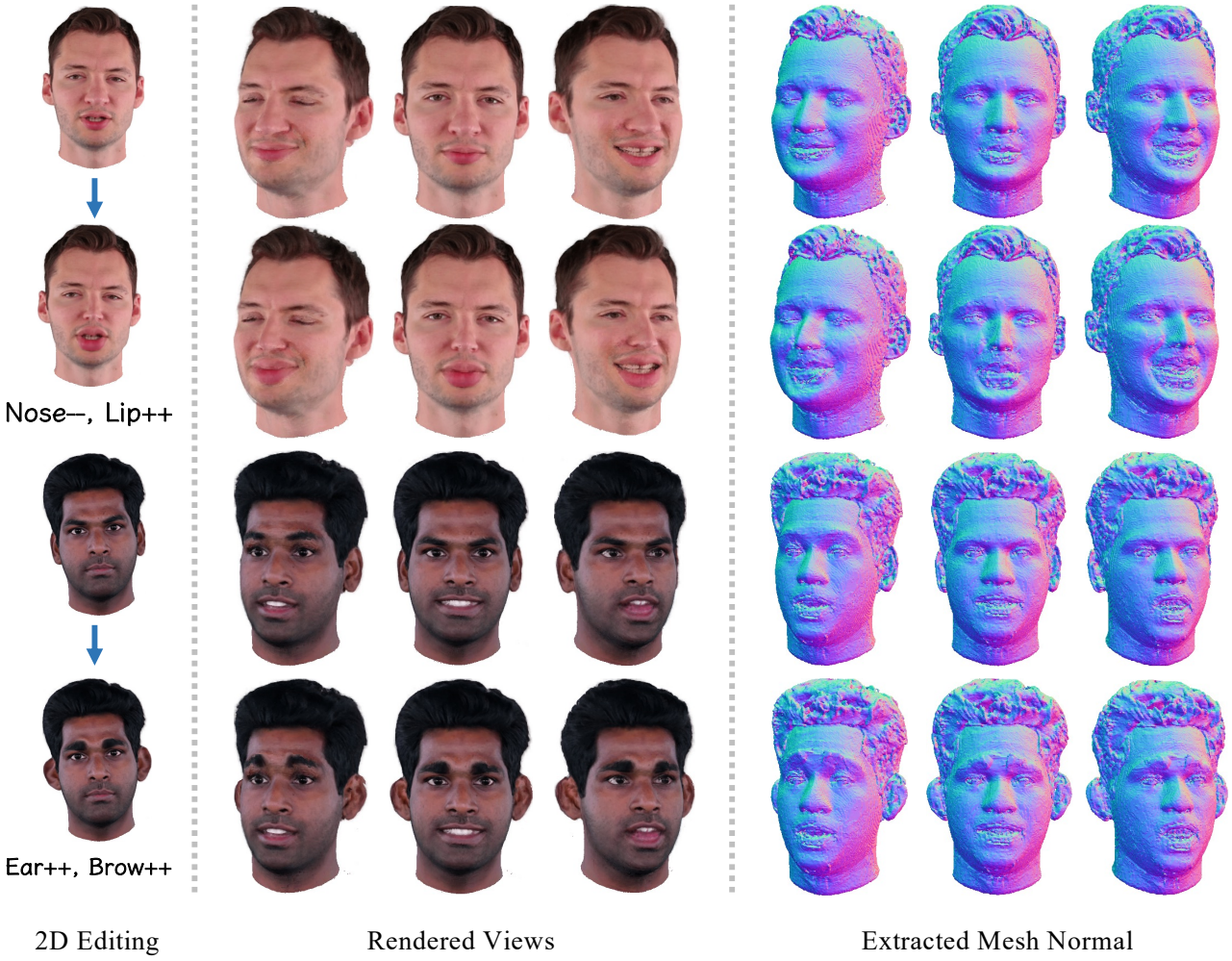

Figure K. We show the mesh normal of original avatars and edited avatars in geometry editing.

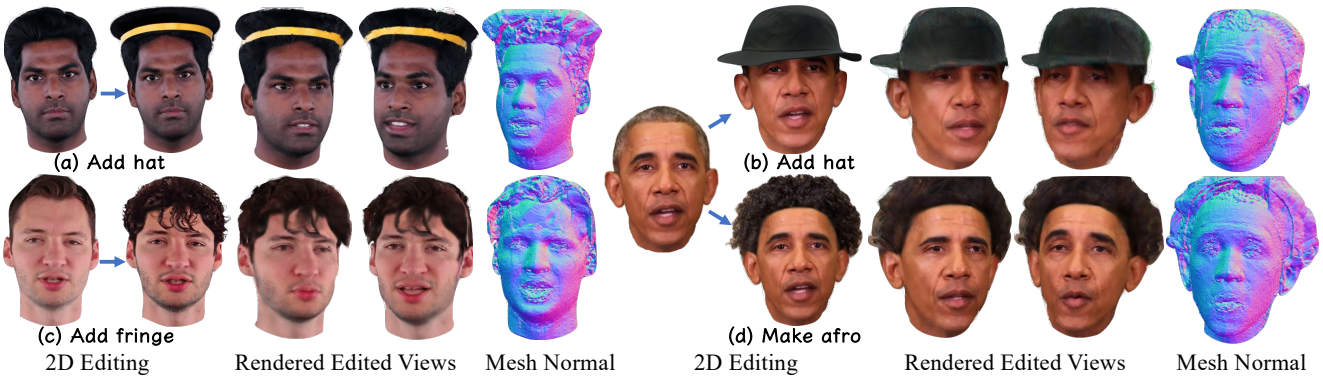

Figure L. We show some failure cases in our method where we add the additional object (a-b) and change the hairstyle (c-d).

Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 4

[3] Xuan Gao, Chenglai Zhong, Jun Xiang, Yang Hong, Yudong Guo, and Juyong Zhang. Reconstructing personalized semantic facial nerf models from monocular video. *ACM Transactions on Graphics (TOG)*, 41(6):1–12, 2022. 3

[4] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 1

[5] K-E Lin, Alex Trevithick, Keli Cheng, Michel Sarkis, Mohsen Ghafoorian, Ning Bi, Gerhard Reitmayr, and Ravi Ramamoorthi. Pvp: Personalized video prior for editable dynamic portraits using stylegan. In *Computer Graphics Forum*, page e14890. Wiley Online Library, 2023. 4

[6] 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

[7] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20991–21002, 2023. 1, 2, 4

[8] 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