Supplementary Material for *OFERA: Blendshape-driven 3D Gaussian Control for Occluded Facial Expression to Realistic Avatars in VR*

Seokhwan Yang, Boram Yoon, Seoyoung Kang, Hail Song, Woontack Woo

A Implementation Details

Environment. The training of the EPM model and the MiA model, which is based on the FATE [6] backbone, was conducted on an NVIDIA A40 GPU server. The EPM model was trained with the Adam optimizer using a fixed learning rate of 1×10^{-3}, for 50 epochs with a batch size of 512. The loss function was L1 loss. For MiA training, we directly adopted the training settings of the original FATE model, while replacing the FLAME [3] parameters with those obtained from Mediapipe [4] and EPM. Specifically, for each frame, the FLAME parameters estimated by Mediapipe and EPM were applied as offsets to the preprocessed parameters generated by the INSTA [8] pipeline, such that the FATE Gaussian avatar model could be trained over the EPM output distribution.

The BDA fitting and system inference were performed on a desktop equipped with an Intel Core i9-14900K CPU, 64 GB RAM, and an NVIDIA GeForce RTX 4090 GPU. The VR rendering system was implemented with a local Python server and Unity client communicating via a local WebSocket connection. We used Unity 2021.3.45f1 together with Meta Movement SDK[1] version 71.0.1.

VR Rendering Implementation. Our Unity-based VR renderer builds upon an open-source 3D Gaussian Splatting [1] viewer[2] for VR environments. The original implementation supports stereo rendering of static 3D Gaussian scenes. We extended this renderer to support real-time streaming and dynamic updates of Gaussian attributes, enabling per-frame modification of position, rotation, and scale driven by headset-derived expression signals. These modifications allow the renderer to be integrated into an end-to-end, real-time avatar control pipeline rather than a static scene viewer.

B Dataset Details

EPM Training Dataset. Since the EPM model maps expression blendshapes to mesh-based expression parameters, it must be trained to generalize across diverse identities as well as a wide range of facial expressions. To this end, we constructed a composite dataset by combining three sources: INSTA [8], NeRSemble [2], and Ava-256 [5]. From each dataset, 10 subjects were randomly selected, resulting in a total of 30 subjects used for training and evaluation. The train/validation/test split was performed with an 8:1:1 ratio using a fixed random seed. The details of the selected subjects and splits are summarized in Tab. 1.

For parameter extraction, we employed the MICA [7] model's metrical tracker to obtain FLAME [3] parameters and used Mediapipe [4] to extract ARKit[3]-compatible blendshape coefficients. All parameters were normalized prior to training. To avoid bias caused by varying frame counts across subjects, we balanced the training set by downsampling to match the number of frames of the subject with the fewest samples.

BDA Pseudo Paired Dataset. The pseudo paired dataset used for fitting the BDA module was constructed from four subjects. For each subject, approximately 5,000 frames were captured while the participant performed a wide range of facial expressions wearing a VR headset. For each frame, VR headset–driven blendshape coefficients and the corresponding Mediapipe-style ARKit blendshapes extracted from avatar-rendered frontal images were paired offline. The resulting dataset consists of approximately 20,000 paired samples in total.

C User Study Example

To provide an impression of the user study setup, we present sample stimuli and corresponding avatar renderings. Participants were instructed to follow a set of facial expression prompts (e.g., angry, surprised, smiling) while wearing the VR headset. The captured expressions were then transferred to their personalized Gaussian avatars using our proposed OFERA system. Fig. 1 shows example frames from two participants, illustrating the alignment between reference expressions and the rendered avatars.

References

[1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1

[2] T. Kirschstein, S. Qian, S. Giebenhain, T. Walter, and M. Nießner. Nersemble: Multi-view radiance field reconstruction of human heads. *ACM Transactions on Graphics (TOG)*, 42(4):1–14, 2023. 1, 2

[3] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6):194–1, 2017. 1

[4] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, et al. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*, 2019. 1

[5] J. Martinez, E. Kim, J. Romero, T. Bagautdinov, S. Saito, S.-I. Yu, S. Anderson, M. Zollhöfer, T.-L. Wang, S. Bai, et al. Codec avatar studio: Paired human captures for complete, driveable, and generalizable avatars. *Advances in Neural Information Processing Systems*, 37:83008–83023, 2024. 1, 2

[6] J. Zhang, Z. Wu, Z. Liang, Y. Gong, D. Hu, Y. Yao, X. Cao, and H. Zhu. Fate: Full-head gaussian avatar with textural editing from monocular video. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5535–5545, 2025. 1

[7] W. Zielonka, T. Bolkart, and J. Thies. Towards metrical reconstruction of human faces. In *European conference on computer vision*, pp. 250–269. Springer, 2022. 1

[8] W. Zielonka, T. Bolkart, and J. Thies. Instant volumetric head avatars. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4574–4584, 2023. 1, 2

- *Seokhwan Yang, Seoyoung Kang and Hail Song are with KAIST UVR Lab. E-mail: {ysshwan147 | sy1009kang | hail96}@kaist.ac.kr.*
- *Boram Yoon is with KAIST KI-ITC ARRC. E-mail: boram.yoon1206@kaist.ac.kr.*
- *Woontack Woo is with KAIST UVR Lab and KAIST KI-ITC ARRC. Corresponding Author. E-mail: wwoo@kaist.ac.kr.*

[1] Meta Horizon, "Movement SDK for Unity - Overview", https://developer.oculus.com/documentation/unity/move-overview

[2] "GaussianSplattingVRViewerUnity", https://github.com/clarte53/GaussianSplattingVRViewerUnity

[3] Apple Developer, "ARFaceAnchor.BlendShapeLocation", https://developer.apple.com/documentation/arkit/arfaceanchor/blendshapelocation

Table 1: Subjects used for EPM training from each dataset and their split into train/validation/test sets.

Dataset	Train IDs	Val IDs	Test IDs
INSTA [8]	{bala, biden, justin, malte, marcel, nf_01, obama, person_0004}	{nf_03}	{wojtek_1}
NeRSemble [2]	{057, 074, 100, 145, 165, 178, 210, 251}	{036}	{037}
Ava-256 [5]	{GTA798, LVD531, CPP930, NTA876, UIQ957, JEN167, RHU956, PKH444}	{QLL122}	{INQ807}

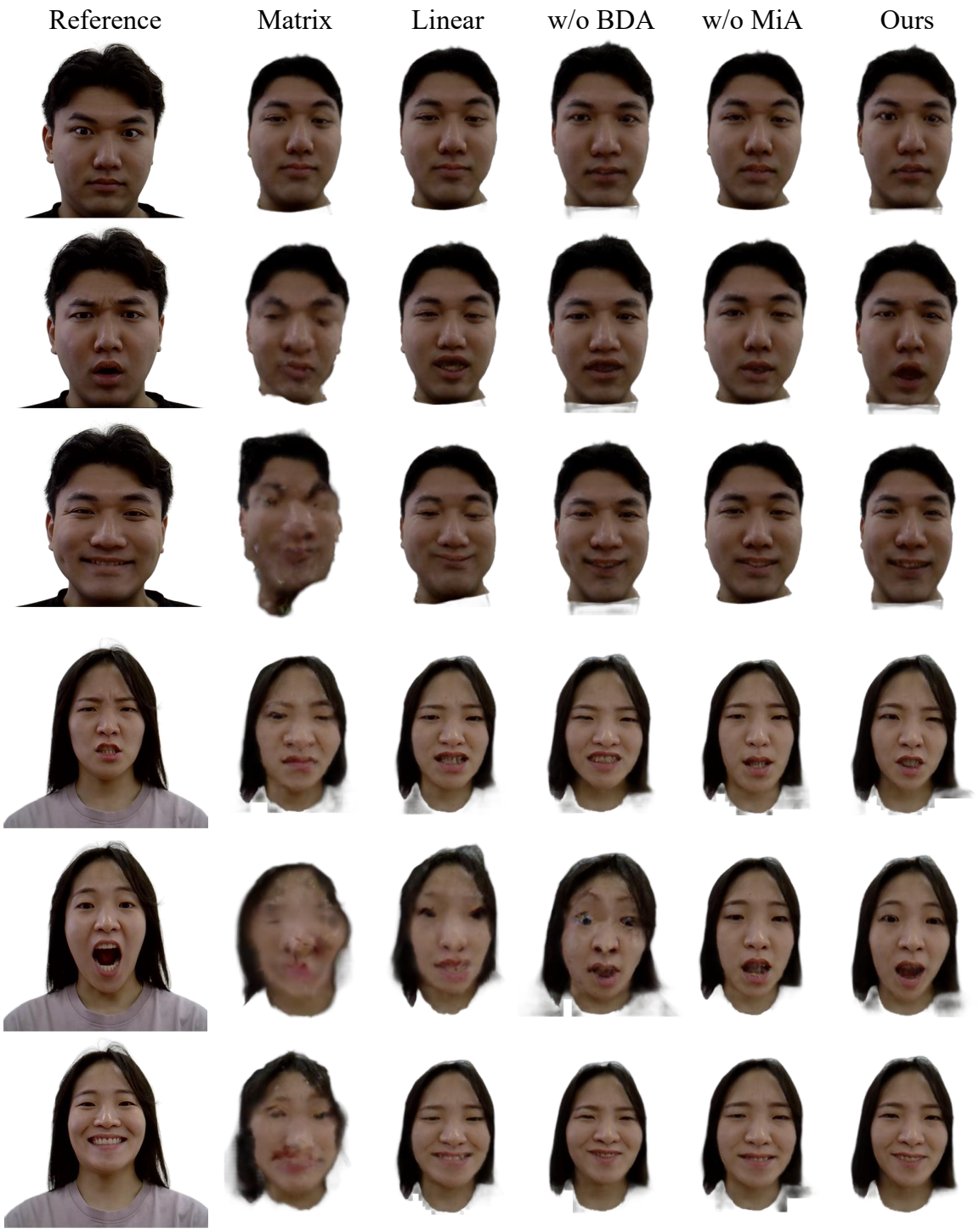

Fig. 1: Examples of user study stimuli and corresponding avatar renderings. Two participants are shown performing various expressions, with the left columns displaying the captured reference images and the right columns showing the rendered avatar results generated by our OFERA system.