8. Appendix

In the Appendix, we show additional extensive experimental results. First, we show details of the quantitative comparison of both datasets: DynamicFace and SplattingAvatar [44]. Second, we depict the qualitative results on various self- and cross-reenactment and novel-view synthesis scenarios. Then, we conduct additional ablation studies on APS, FLAME mouth modification, and isolating the contribution of each component. Followed by the visualization of APS during mouth deformation, we also demonstrate training and inference details and the preprocessing of our model. Then, we elucidate the details of the mesh modification process, dataset, and baselines. Finally, we discuss the broader impacts of GeoAvatar, along with two examples: interactable digital human and virtual presentation.

8.1. Additional Quantitative Comparison

In Table 12 and Table 13, we compared the self-reenactment results of each video in SplattingAvatar [44] and DynamicFace, respectively. For a fair and thorough comparison, we utilized every 10 subjects in each video dataset. We denote the name of each subject with the quantitative results in Table 12 and Table 13. In both datasets, our model shows the best performance in almost every video. Specifically, ours shows the best results in 19 videos out of 20 videos.

Moreover, we demonstrated quantitative comparisons with other models for a more comprehensive evaluation as shown in Table 4. For evaluation metrics, we measured a cosine similarity by employing off-the-shelf models: 1) insightface[*] [10] for "ID preservation" and 2) Facial-Expression-Recognition model[*] trained on FER2013 for "Expression". Still, ours shows the superior results on both ID preservation and expression scores.

	INSTA	3DGS	SplattingAvatar	FlashAvatar	GaussianAvatars	**Ours**
ID preservation ↑	0.850	0.783	0.873	0.889	0.831	**0.906**
Expression ↑	0.712	0.458	0.667	0.717	0.681	**0.750**

Table 4. Quantitative evaluation on cross-reenactment.

8.2. Additional Qualitative Comparison

In Figure 9(a), we compare cross-reenactments by utilizing the EMO-1-shout+laugh sequence of id061 in NeRSemble for challenging expressions. Ours shows stable results, while the baselines suffer from severe artifacts. In Figure 9(b), we show novel-view synthesis under the same identity and expression. Ours successfully generates high-quality images while maintaining consistency in identity.

In Figure 15, we show extensive self-reenactment results of ours, compared to baselines. Indeed, ours shows notably better performance on the mouth region, *e.g.*, handling artifacts on the second row, and resolution of teeth in the fourth and sixth row. Moreover, our model shows robust and high-resolution outputs on other facial regions, *e.g.*, the iris in the first and the fifth rows, and the ear in the third row.

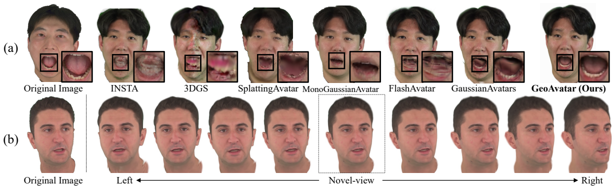


Figure 9. Cross-reenactment and novel-view synthesis.

In Figure 16, we show extensive cross-reenactment results of ours, compared to baselines. To evaluate the robustness of each model more thoroughly, we utilize the source and target actors from different datasets, *e.g.*, the source from SplattingAvatar when the target is from DynamicFace, and vice versa. Even in this harsh scenario, ours shows consistently robust reenactment results, while preserving the identity of the source actor and mimicking the expression of the target actor well. On the other hand, other models suffer from severe artifacts, occurring by notable distribution differences between the source and the target actors.

In Figure 17, we show extensive novel-view synthesis results of ours, compared to baselines. Ours shows consistently robust results on various actors. Especially, we emphasize that in the case when the original image has the extreme facial degree, *e.g.*, the fourth row in the Figure 17, ours can generate the face robustly in more extreme viewpoints. In contrast, other models suffer from artifacts, *e.g.*, INSTA, 3DGS, and GaussianAvatars, spiking artifacts, *e.g.*, FlashAvatar, ghosting effects, *e.g.*, INSTA. Please check our project page for more visualization results.

8.3. Additional Ablation Results

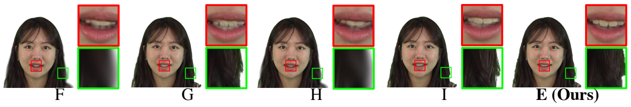


Figure 10. **Additional qualitative ablation results.** We show qualitative results for additional ablation models for thorough evaluations. Models that only utilizes the rigid set,*i.e.*, F and H, show notably blurred results for the region where the FLAME mesh cannot cover the ground truth geometry, *e.g.*, hair. On the other hand, setting every region to flexible set, *i.e.*, G and I, introduces undesirable artifacts for the mouth animation.

In the extension from Table 3, we add more settings for thorough ablations. First, to evaluate the effectiveness of training initialization strategy, we train the baseline model with

[*] https://github.com/deepinsight/insightface

[*] https://github.com/WuJie1010/Facial-Expression-Recognition.Pytorch

Configs	Sets	FLAME mouth	MSE $(10^{-3})\downarrow$	PSNR ↑	SSIM ↑	LPIPS $(10^{-1})\downarrow$
F	Rigid		0.960	30.291	0.923	0.975
G	Flexible		0.991	30.356	0.919	0.653
H	Rigid	✓	0.901	30.471	0.923	0.840
I	Flexible	✓	0.853	31.318	0.930	0.629
E (Ours)	APS	✓	**0.748**	**32.697**	**0.942**	**0.513**

Table 5. **Additional quantitative ablation results.** We show quantitative results for additional ablation models for thorough evaluations. While applying either APS or FLAME mouth slightly improve the performance, they show the synergestic effect when applied together.

	Configuration	MSE $(10^{-3})\downarrow$	PSNR ↑	SSIM ↑	LPIPS $(10^{-1})\downarrow$
A	**Ours** w/o APS	0.853	31.318	0.930	0.629
B	**Ours** w/o FLAME mouth	0.912	30.551	0.929	0.561
C	**Ours** w/o Part-wise deformation	0.815	32.162	0.941	0.540
D	**Ours** w/o $\mathcal{L}_{angle}$	**0.733**	**32.751**	0.941	0.519
E	**Ours**	0.748	32.697	**0.942**	**0.513**

Table 6. Quantitative ablation results by isolating the contribution.

assuming every Gaussian as the rigid set, *i.e.*, F, and the flexible set, *i.e.*, G. Since the original GaussianAvatars [40] utilizes threshold for position loss same with the threshold of the flexible set, A and G is originally same. Then, we apply FLAME mouth modification, *i.e.*, mesh modification and part-wise deformation, to each model, *i.e.*, H and I, respectively. Table 5 shows the average quantitative results of each model, including our final model, *i.e.*, E. We utilize every model in SplattingAvatar and DynamicFace to obtain the average result in Table 5.

First, in F and G, both model shows inferior results in both qualitative and quantitative ways. In specific, as shown in Figure 8.3, F shows severe artifacts in the hair region, which needs high flexibility during training. By applying FLAME modification and deformation to F and G, *i.e.*, H and I, respectively, improves the performance, it still shows artifacts in hair or mouth.

Then, in H and I only applies FLAME mouth modification and part-wise deformation, without applying APS. Though only applying these improves both quantitative and qualitative results notably, still it shows worse result than our final model. We argue that since APS helps model to train each part flexibly, it is helpful to improve the overall quality of every part, either rigid or flexible. Indeed, our final model, *i.e.*, E, shows the best result on both quantitative and qualitative ways.

To better clarify the contribution of each module, we also perform ablations by isolating each module, as shown in Table 6. Especially, in Config C, *i.e.*, excluding part-wise deformation based on MLP, shows a notable degradation compared to the original model, *i.e.*, Config E, which clearly shows the effectiveness of the MLP deformation.

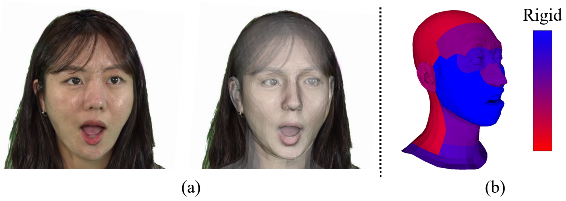


Figure 11. (a) Rendered results with the mesh, (b) Distribution of rigid and flexible sets.

8.4. Visualization of APS and mouth deformation

We visualized the fitted mesh during mouth deformation in Figure 11(a). We also plot the distributions of rigid and flexible sets in Figure 11(b). We indeed check that the distribution follows the human intuition, *e.g.*, rigid for facial regions, flexible for scalp and neck. Several regions, *e.g.*, forehead and ears, show varying results among subjects, *i.e.*, denoted as purple region, which justifies the dynamic allocation of rigid and flexible sets of APS.

8.5. Training and Inference Details

Efficiency	FlashAvatar	GaussianAvatars	MonoGaussianAvatar	GaussianHeadAvatar	**Ours**
Time (hrs) ↓	**1.66**	9.25	7.90	19.92	4.90
Speed (FPS) ↑	**291.20**	19.11	5.91	6.25	71.52

Table 7. Efficiency comparisons on training time (hrs) and inference speed (FPS).

Steps (10^3)	50	100	APS	150	200	250	300
SplattingAvatar	33.95	42.22	-	34.67	43.61	46.12	47.63
DynamicFace	35.17	44.86	-	36.48	47.80	51.03	51.42

Table 8. Gaussian numbers (10^3) on each dataset.

In Table 7, we compare training and inference speeds with a single RTX 3090. Ours shows enough inference efficiency, *i.e.*, > 60 FPS, for real-time scenario. We show numbers of Gaussians at training and inference (300,000 steps) stages in Table 8.

8.6. Preprocessing

We can coarsely divide preprocessing into two steps; masking and FLAME tracking. In the following paragraph, we explain the details.

Masking. Since we target to generate human avatars, we have to distinguish human parts from the others, *e.g.*, clothes and background, as other baselines [44, 51, 65] did. We conjugate two masking logics, Background Matting [30] and BiSeNet [56]. Though Background Matting can distinguish the foreground objects from the background, it still contains non-human parts, *e.g.*, chairs that the subject

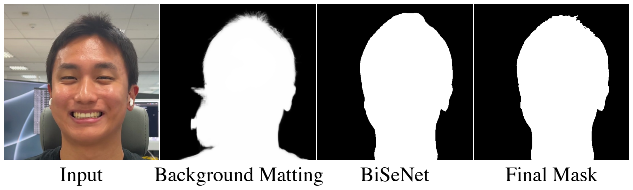


Figure 12. **Comparison of each mask.** Background Matting yields a noisy mask, *i.e.*, containing non-human parts, while BiSeNet yields an over-smoothed mask. We intersect two masks and obtain the final mask for training.

is sitting in. On the other hand, BiSeNet can distinguish human parts and non-human parts accurately, but it returns over-smoothed masks. Consequently, by utilizing the overlapped region of each mask, we can obtain an accurate and sharp mask for the human part. Figure 12 shows the visualization of each mask. We apply the aforementioned mask logic for preprocessing DynamicFace and SplattingAvatar datasets, but not for the NeRSemble dataset since they offer the preprocessed mask together with the images.

FLAME Tracking. We utilize the modification of MICA [64], which utilizes a pre-trained model that returns FLAME shape parameters from a single image, and an additional image-wise FLAME tracking model. We modified the original MICA to work on the 2023 version of FLAME to utilize its revised eye region mesh, which originally worked on the 2020 version. Moreover, we optimize the FLAME neck, global rotation, and translation parameters, which are excluded in the original MICA, along with other FLAME parameters, *e.g.*, expression, jaw, and eye pose, for better tracking results.

Hyperparameters. We set the hyperparameters for the training as in Table 9. Except for the hyperparameters mentioned, we follow the settings of GaussianAvatars [40].

Symbol	Parameter Description	Value
n	number of FLAME parts	10
τ_r	threshold for the rigid set	0.1
τ_f	threshold for the flexible set	2.0
τ_φ	threshold for $\mathcal{L}_{angle}$	0.78
λ	weight for D-SSIM	0.2
N	iterations before APS	100000
-	total iteration	200000

Table 9. **Hyperparameter settings of GeoAvatar.** We utilize the hyperparameters mentioned above for training GeoAvatar.

8.7. Structural Details

Mesh Modification. Though FLAME [28] covers various expressions and joint movements of the face, the absence of mouth interior structures deteriorates the expressiveness of teeth and mouth interior [27, 40, 51]. Consequently, GaussianAvatars [40] adds teeth by duplicating the vertices trajectory of lip rings of FLAME. As shown in Figure 13, the mesh corresponding to frontal teeth can be generated. However, this cannot represent the geometry of molar teeth and mouth interior structure, *e.g.*, palate or tongue.

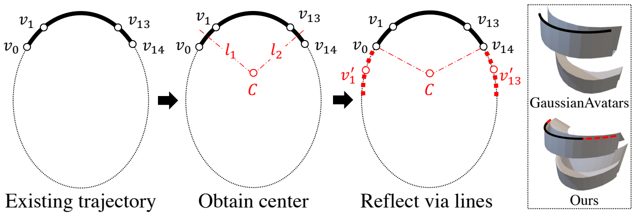


Figure 13. **Teeth trajectory extension.** Assuming that the existing teeth trajectory forms an arc of the circle, we calculate the pseudo-center C of the circle by using the perpendicular bisector line of $\overline{v_0 v_1}$ and $\overline{v_{13} v_{14}}$. Then, we reflect the existing vertices of the teeth by either $\overline{Cv_0}$ or $\overline{Cv_{14}}$, to extend the trajectory smoothly.

For better representation, we incorporate molar teeth and mouth interiors into the FLAME structure. First, we generate the frontal teeth by utilizing the vertex trajectory of the lip rings [40]. Empirically, we observe that the teeth vertices lie on the xz-plane, *i.e.*, they share the same y-axis value, and their trajectory approximates the shape of an ellipsoid. Given that the curvature of the teeth trajectory is relatively small, we hypothesize that it can be approximated as an arc of a circle. Due to the symmetry of a circle about its center, the arc can be extended smoothly by reflecting it across the center. We apply this approach to the pseudo-arc trajectory of the teeth. To this end, we identify the pseudo-center of the circle and extend the teeth trajectory to generate the molar teeth, as illustrated in Figure 13.

First, we utilize the two leftmost vertices, *i.e.*, $v_0 = (x_0, y_0, z_0)$ and $v_1 = (x_1, y_0, z_1)$, and the two rightmost vertices, *i.e.*, $v_{13} = (x_{13}, y_0, z_{13})$ and $v_{14} = (x_{14}, y_0, z_{14})$, out of 15 vertices that constructs the teeth trajectory ring. First, to obtain the pseudo-center, we obtain the intersection point C between the perpendicular bisector of $\overline{v_0 v_1}$, *i.e.*, l_1, and the perpendicular bisector of $\overline{v_{13} v_{14}}$, *i.e.*, l_2. Each perpendicular bisector can be obtained as follows:

$$l_1 : z - \frac{z_1 + z_2}{2} = -\frac{x_2 - x_1}{z_2 - z_1}\left(x - \frac{x_1 + x_2}{2}\right),$$
$$l_2 : z - \frac{z_{13} + z_{14}}{2} = -\frac{x_{14} - x_{13}}{z_{14} - z_{13}}\left(x - \frac{x_{13} + x_{14}}{2}\right).$$

Then, we reflect the 5 vertices located on the left side, *i.e.*, $v_{i \in \{1, \cdots 5\}}$, with the line $\overline{Cv_0}$. In the same way, we reflect the 5 vertices located on the right side, *i.e.*, $v_{i \in \{9, \cdots 13\}}$, with the line $\overline{Cv_{14}}$. Finally, we can obtain the new teeth trajectory which includes the molar teeth, denoted as the red dot line in Figure 13. After generating the trajectory of the teeth, we shift it backward to generate vertices for the palate and the mouth floor.

8.8. Dataset Details

Dataset	#ID	#Expressions	Resolution	Total Time (min)	Disk Space (GB)
NerFace (CVPR 2021)	3	1	1920×1080	6.44	3.79
IMAvatar (CVPR 2022)	4	11	512×512	7.08	3.39
DynamicFace	**10**	**20**	**3840×2160**	**32.25**	**18.92**

Table 10. Comparison of monocular video-based datasets.

Our proposed dataset, **DynamicFace** is designed to capture a wide range of facial movements, enabling the generation of avatars capable of dynamic motion. DynamicFace consists of 10 videos, each recording a single actor performing various facial expressions provided by the instruction. During recording, actors are instructed to shake their heads slowly to record the various facial degrees with a single camera. Nine subjects are recorded by a single Sony AX700 camcorder with a chromakey background. The remaining subject is recorded by a single iPhone14 with a normal background. In Figure 18, we show the sample frames of DynamicFace. We also compare the details of monocular video-based dataset features in Table 10.

8.9. Baseline Descriptions

INSTA. The instant volumetric head avatars (INSTA) framework embeds a dynamic neural radiance field into a surface-aligned multi-resolution grid around a 3D parametric face model. It employs a deformation field guided by FLAME to map points between deformed and canonical spaces and uses 3DMM-driven geometry regularization for depth alignment. The approach utilizes neural graphics primitives with multi-resolution hash encoding to represent the radiance field, enabling reconstruction and rendering based on monocular RGB videos.

3D Gaussian Splatting. Unlike an existing 3D representation module[35, 37] which implicitly encodes the color and density information of the volume inside MLP, 3D Gaussian Splatting (3DGS) explicitly represents the 3D volume using the mean and 3D variance of Gaussian distribution. Moreover, the efficient tile-based rasterizer of 3DGS enables remarkably faster rendering than the existing module. However, 3DGS requires accurate pre-computed camera poses [13, 46], *e.g*., obtained by COLMAP [43]. Moreover, primitive 3DGS can be applied only to static scenes, which is definitely not appropriate for dynamic avatar generation. To adapt its property in the avatar generation, we utilize FLAME meshes instead of COLMAP to initialize the Gaussian points.

SplattingAvatar. SplattingAvatar proposes a binding strategy between Gaussian and FLAME mesh, which forces Gaussians to move together with FLAME mesh, which is deformed by FLAME coefficients. Since SplattingAvatar does not utilize additional regularization terms to locate Gaussians nearby bonded triangles, it utilizes the walking triangles strategy to adaptively change the bonded triangle of Gaussian.

MonoGaussianAvatar. MonoGaussianAvatar proposes a point-based 3D Gaussian head avatar generation framework. They enhance the point insertion and deletion strategy which prunes away invisible points via thresholding of opacity. They also utilizes the Gaussian deformation field to preserve the accessories.

FlashAvatar. FlashAvatar also utilizes the strategy of binding Gaussians to the FLAME mesh by using its UV map. Then they utilize Gaussian offset models to deform Gaussians by the animation of FLAME meshes. To enhance the mouth interior generation performance, FlashAvatar adds additional faces to fill the mouth cavity, using vertices on the lip. Moreover, they utilize the masked loss which focuses on the mouth region.

GaussianAvatars. GaussianAvatars proposes a multi-view-based Gaussian head avatar creation framework by rigging 3D Gaussian splats to 3DMM faces. This framework employs adaptive density control with binding inheritance, ensuring that newly created or pruned 3D Gaussian splats remain consistently attached to their parent triangles on the FLAME mesh during densification.

In GaussianAvatars, the linear blend skinning weight of upper teeth is rigged to the neck, *i.e*., head movements, while that of lower teeth is rigged to the jaw. Consequently, upper and lower teeth movements are determined by depending on head and jaw movements, respectively. To mitigate this, we propose a deformation network that offers offsets part-wisely, to translate each part independently from the FLAME parameters using the offsets, which is unavailable in GaussianAvatars.

8.10. Additional Comparison Study on a One-shot-based Method

Dataset	Setting	Method	MSE $(10^{-3})\downarrow$	PSNR ↑	SSIM ↑	LPIPS $(10^{-1})\downarrow$	ID preservation ↑
SplattingAvatar	One-shot	GAGAvatar [8]	5.108	23.541	0.875	1.121	0.832
	Video	**Ours**	**0.884**	**32.635**	**0.965**	**0.367**	**0.944**
DynamicFace	One-shot	GAGAvatar [8]	2.421	26.406	0.850	1.485	0.827
	Video	**Ours**	**0.612**	**32.760**	**0.919**	**0.660**	**0.931**

Table 11. Quantitative comparisons on one-shot and video settings.

Our work focuses on a single video-based setting for preserving identity under expressive variations, *i.e*., high-quality personalization. Unlike one-shot or few-shot-based methods that risk entangling identity, video sequences provide sufficient intra-subject variation and temporal consistency to robustly learn identity-specific structures such as the mouth interior. Furthermore, image-based methods

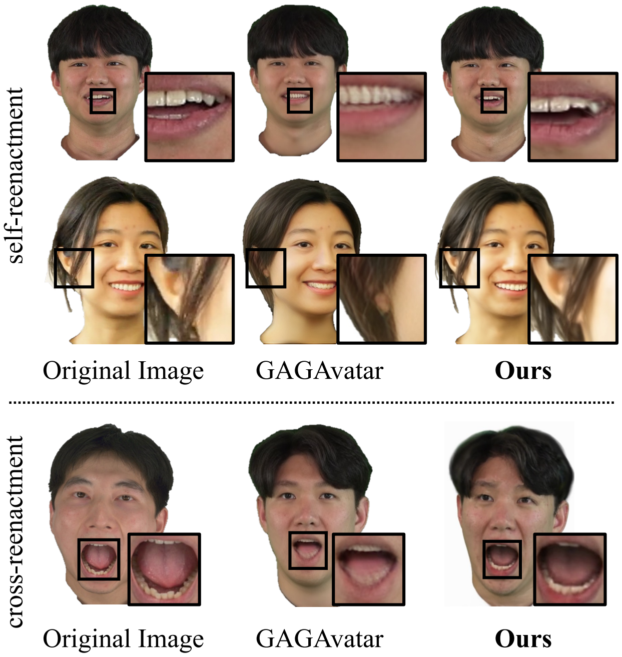


Figure 14. Comparisons on a one-shot-based baseline (GAGAvatar [8]) via self- and cross-reenactment.

often require large-scale pretraining on external datasets, whereas our method achieves high-quality results without any pretraining, making it practical for real-world applications. In Table 11 and Figure 14, we compared GeoAvatar (video) and state-of-the-art image-based avatar generation method, GAGAvatar [8] (one-shot). Ours achieves significantly higher identity preservation, while GAGAvatar struggles to preserve identity.

8.11. Broader Impacts

The proposed GeoAvatar framework demonstrates remarkable versatility, offering a broad spectrum of potential applications across diverse industries. By leveraging its capability to generate realistic, speech-driven 3D avatars, GeoAvatar has the potential to significantly enhance user experiences in domains such as entertainment, education, customer service, and healthcare. Its modular design allows seamless integration with complementary technologies, including large language models (LLMs), text-to-speech (TTS) models, and advanced 3D animation techniques, positioning it as a robust solution for creating engaging, interactive digital human experiences.

The following sections present two examples of industrial-like applications built on GeoAvatar as shown in Figure 19. Notably, these demonstrations were developed using training data captured with a single smartphone, emphasizing the scalability and accessibility of our method.

8.12. Application #1: Interactable Digital Human

To highlight the capabilities of GeoAvatar, we integrated a large language model (LLM), a text-to-speech (TTS) model, and speech-driven 3D facial animation modules from the NVIDIA Audio2Face framework [1]. This configuration enabled the development of a real-time interactable digital human demo capable of engaging in natural, dynamic conversations with users.

In this system, the LLM processes textual user inputs and generates contextually appropriate responses. These responses are then converted into natural-sounding speech by the TTS model. Finally, the Audio2Face module drives a 3D avatar, synchronizing facial expressions, lip movements, and head gestures to match the speech output. This seamless integration results in an immersive, life-like digital human experience, illustrating GeoAvatar's potential for real-time applications in virtual environments, customer engagement, and interactive storytelling. The demo video is in the project page. We utilized CHANGER [26] for seamless head blending into the original scenes.

8.13. Application #2: Virtual Presentation

Expanding on recent advancements in speech-driven animation, we incorporated the state-of-the-art speech-to-facial animation module (S2F) into the GeoAvatar framework. This integration facilitated the generation of highly realistic talking head avatars characterized by natural head movements, expressive facial animations, and precise lip synchronization.

The use of S2F allows GeoAvatar to produce avatars with distinct personalities and speaking styles, making them suitable for a variety of applications, such as virtual presentations, personalized virtual assistants, and digital influencers. The module's ability to capture nuanced head motions and deliver high-quality lip-syncing enhances the realism and engagement of these avatars, particularly in use cases where authentic communication and emotional expressiveness are critical. This example underscores the framework's flexibility and its capacity to deliver diverse, high-fidelity digital human experiences.

Model	INSTA	3DGS	SplattingAvatar	MonoGaussianAvatar	FlashAvatar	GaussianAvatars	Ours
Subject	subject_001						
MSE (10^{-3}) ↓	0.741	0.987	0.666	0.987	1.112	0.776	**0.350**
PSNR ↑	31.402	30.127	31.817	30.154	29.646	31.191	**34.670**
SSIM ↑	0.909	0.896	0.894	0.900	0.890	0.894	**0.929**
LPIPS (10^{-1}) ↓	0.984	1.437	1.270	1.447	0.695	0.728	**0.633**
Subject	subject_002						
MSE (10^{-3}) ↓	1.689	1.857	1.772	2.544	2.160	1.497	**0.787**
PSNR ↑	27.746	27.351	27.531	25.974	26.716	28.284	**31.063**
SSIM ↑	0.820	0.814	0.795	0.821	0.814	0.792	**0.874**
LPIPS (10^{-1}) ↓	1.723	2.020	1.728	1.479	0.881	0.982	**0.775**
Subject	subject_003						
MSE (10^{-3}) ↓	0.553	0.757	1.036	1.217	1.256	0.865	**0.310**
PSNR ↑	32.617	31.294	29.930	29.340	29.088	30.701	**35.155**
SSIM ↑	0.918	0.904	0.884	0.891	0.888	0.905	**0.940**
LPIPS (10^{-1}) ↓	0.905	1.408	1.215	1.286	0.715	0.626	**0.518**
Subject	subject_004						
MSE (10^{-3}) ↓	2.173	1.680	1.328	1.330	2.548	1.080	**0.507**
PSNR ↑	27.039	28.147	28.975	28.866	25.957	29.733	**33.192**
SSIM ↑	0.880	0.865	0.871	0.893	0.869	0.868	**0.914**
LPIPS (10^{-1}) ↓	1.355	1.704	1.160	1.131	0.817	0.627	**0.547**
Subject	subject_005						
MSE (10^{-3}) ↓	1.283	1.128	1.109	1.380	1.219	1.062	**0.440**
PSNR ↑	29.017	29.566	29.650	28.715	29.244	29.826	**33.703**
SSIM ↑	0.882	0.867	0.865	0.874	0.865	0.869	**0.913**
LPIPS (10^{-1}) ↓	1.444	2.029	1.328	1.464	0.679	0.893	**0.643**
Subject	subject_006						
MSE (10^{-3}) ↓	2.064	1.649	2.008	2.390	1.892	1.091	**0.688**
PSNR ↑	27.115	27.995	27.056	26.268	27.350	29.688	**31.673**
SSIM ↑	0.879	0.864	0.855	0.880	0.861	0.872	**0.916**
LPIPS (10^{-1}) ↓	1.618	2.215	1.686	1.527	0.803	0.850	**0.702**
Subject	subject_007						
MSE (10^{-3}) ↓	1.756	0.915	0.994	1.346	1.573	0.840	**0.374**
PSNR ↑	27.608	30.518	30.092	28.735	28.092	30.843	**34.357**
SSIM ↑	0.912	0.905	0.897	0.905	0.895	0.903	**0.939**
LPIPS (10^{-1}) ↓	1.147	1.342	1.251	1.233	0.680	0.536	**0.522**
Subject	subject_008						
MSE (10^{-3}) ↓	2.124	2.995	2.322	2.166	3.127	3.030	**1.564**
PSNR ↑	26.994	25.368	26.435	26.710	25.287	25.288	**28.198**
SSIM ↑	0.863	0.844	0.852	0.871	0.857	0.844	**0.900**
LPIPS (10^{-1}) ↓	1.563	2.003	1.653	1.455	**0.910**	1.269	1.020
Subject	subject_009						
MSE (10^{-3}) ↓	2.119	2.952	1.957	1.324	1.735	2.008	**0.666**
PSNR ↑	27.029	25.444	27.162	28.892	27.965	27.023	**31.835**
SSIM ↑	0.925	0.910	0.901	0.923	0.917	0.908	**0.940**
LPIPS (10^{-1}) ↓	0.944	1.132	1.200	1.005	**0.601**	0.810	0.629
Subject	subject_010						
MSE (10^{-3}) ↓	0.946	1.113	1.065	1.494	1.489	**0.427**	0.435
PSNR ↑	30.310	29.657	29.780	28.302	28.383	**33.840**	33.753
SSIM ↑	0.887	0.877	0.876	0.898	0.880	**0.933**	0.924
LPIPS (10^{-1}) ↓	1.204	1.702	1.287	1.189	0.666	**0.595**	0.614

Table 12. **Additional quantitative results.** Detailed quantitative comparison results of each subject from DynamicFace. For a fair comparison, we utilized all 10 subjects without omission. **Bold** indicates the best and underline indicates the second.

Model	INSTA	3DGS	SplattingAvatar	MonoGaussianAvatar	FlashAvatar	GaussianAvatars	Ours
Subject	bala						
MSE (10^{-3}) ↓	1.396	1.2691	4.428	0.968	0.576	0.661	**0.485**
PSNR ↑	28.609	29.010	23.550	30.426	32.510	31.822	**33.193**
SSIM ↑	0.936	0.914	0.922	0.928	0.938	0.957	**0.968**
LPIPS (10^{-1}) ↓	0.816	1.424	0.715	0.819	0.346	0.350	**0.320**
Subject	biden						
MSE (10^{-3}) ↓	0.661	0.804	1.898	0.749	0.713	0.783	**0.291**
PSNR ↑	31.994	31.205	27.308	31.356	31.531	31.303	**35.586**
SSIM ↑	0.969	0.958	0.955	0.959	0.969	0.956	**0.982**
LPIPS (10^{-1}) ↓	0.359	0.452	0.412	0.433	0.239	0.239	**0.169**
Subject	malte_1						
MSE (10^{-3}) ↓	1.273	1.644	2.779	0.926	1.033	0.559	**0.387**
PSNR ↑	29.186	28.076	25.631	30.669	30.418	32.665	**34.224**
SSIM ↑	0.946	0.933	0.937	0.944	0.949	0.970	**0.976**
LPIPS (10^{-1}) ↓	0.579	0.807	0.509	0.561	0.343	0.305	**0.273**
Subject	marcel						
MSE (10^{-3}) ↓	1.180	2.794	2.228	1.089	1.850	2.902	**1.149**
PSNR ↑	29.635	25.807	26.497	29.791	27.404	24.457	**29.639**
SSIM ↑	0.947	0.929	0.940	0.948	0.931	0.913	**0.960**
LPIPS (10^{-1}) ↓	0.556	1.075	**0.530**	0.591	0.634	0.706	0.588
Subject	nf_01						
MSE (10^{-3}) ↓	1.705	1.904	3.643	1.505	1.644	1.961	**1.054**
PSNR ↑	27.891	27.470	24.458	28.484	28.114	27.399	**30.027**
SSIM ↑	0.945	0.933	0.935	0.939	0.951	0.931	**0.966**
LPIPS (10^{-1}) ↓	0.677	1.030	0.631	0.694	0.473	0.722	**0.431**
Subject	nf_03						
MSE (10^{-3}) ↓	1.171	1.962	2.001	1.576	1.383	0.636	**0.492**
PSNR ↑	29.637	27.249	27.063	28.226	28.857	30.403	**33.286**
SSIM ↑	0.941	0.923	0.934	0.935	0.941	0.943	**0.968**
LPIPS (10^{-1}) ↓	0.565	0.940	0.465	0.596	0.392	0.360	**0.353**
Subject	obama						
MSE (10^{-3}) ↓	3.264	1.546	3.245	1.045	1.371	0.854	**0.234**
PSNR ↑	27.231	29.317	25.073	30.212	30.387	29.243	**36.801**
SSIM ↑	0.951	0.937	0.945	0.958	0.963	0.935	**0.981**
LPIPS (10^{-1}) ↓	0.487	0.565	0.558	0.427	0.335	0.267	**0.157**
Subject	person_0004						
MSE (10^{-3}) ↓	6.881	6.726	3.478	6.078	9.676	**2.579**	2.308
PSNR ↑	24.045	24.536	24.648	22.345	24.503	26.377	**30.349**
SSIM ↑	0.909	0.904	0.937	0.913	0.928	0.937	**0.947**
LPIPS (10^{-1}) ↓	1.102	1.365	0.697	1.803	0.704	0.632	**0.431**
Subject	wojtek_1						
MSE (10^{-3}) ↓	1.203	0.738	3.422	0.645	0.423	0.364	**0.282**
PSNR ↑	29.269	31.534	24.683	32.119	33.867	34.454	**35.535**
SSIM ↑	0.957	0.949	0.938	0.956	0.964	0.979	**0.981**
LPIPS (10^{-1}) ↓	0.563	0.630	0.541	0.521	0.234	0.220	**0.217**
Subject	yufeng						
MSE (10^{-3}) ↓	5.559	8.312	4.145	4.373	3.727	5.497	**2.061**
PSNR ↑	23.331	21.227	24.412	24.501	25.463	23.122	**27.708**
SSIM ↑	0.878	0.849	0.888	0.893	0.900	0.860	**0.924**
LPIPS (10^{-1}) ↓	1.084	1.896	0.823	0.885	0.741	1.334	**0.729**

Table 13. **Additional quantitative results.** Detailed quantitative comparison results of each subject provided by SplattingAvatar [44]. For a fair comparison, we utilized all 10 subjects without omission. **Bold** indicates the best and underline indicates the second.

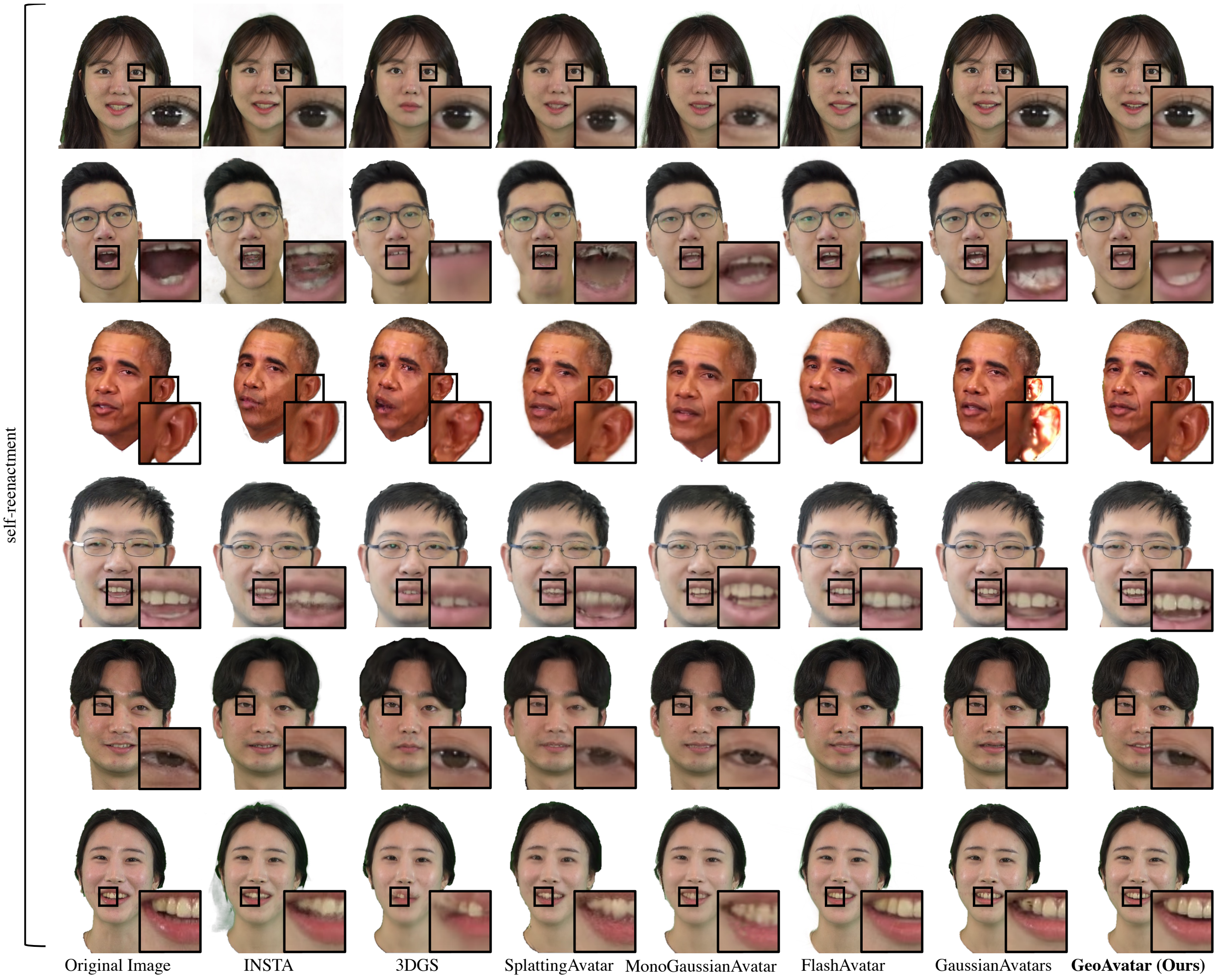


Figure 15. **Additional self-reenactment synthesis results.** We show additional self-reenactment results of ours, compared to baselines, on SplattingAvatar and DynamicFace datasets. Our shows high-resolution results not only on regions where FLAME geometry is accurate, *e.g.*, eyes, but also on the regions where FLAME geometry is erroneous or absent, *e.g.*, ears and mouth interior. However, baselines struggle to generate high-resolution results, *e.g.*, the first and fifth rows, or to prevent artifacts, *e.g.*, the second, fourth, and sixth rows.

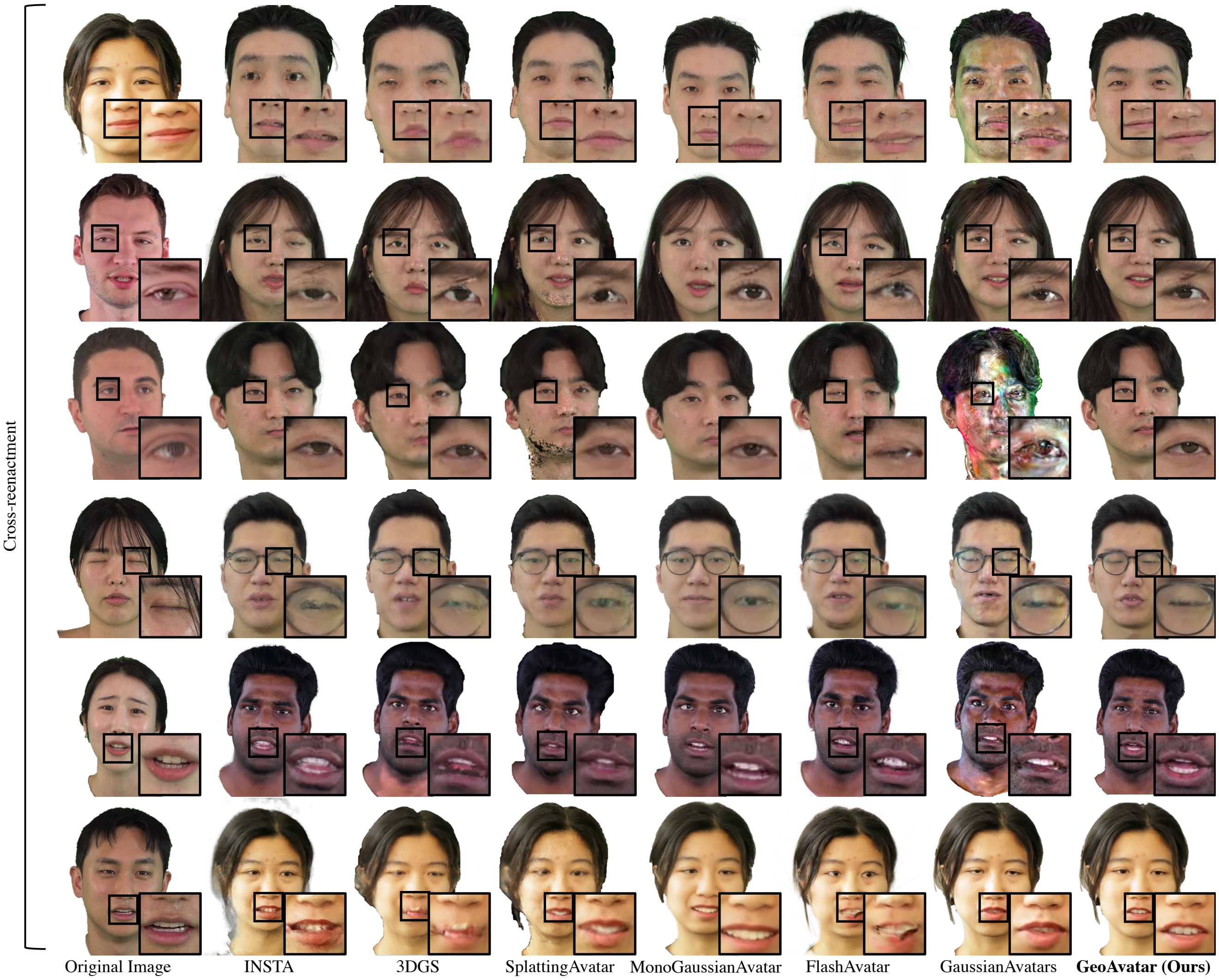


Figure 16. **Additional cross-reenactment synthesis results.** We show additional cross-reenactment results of ours, compared to baselines, on SplattingAvatar and DynamicFace datasets. To evaluate models thoroughly, we utilize the source and target actors from different datasets, *e.g.*, SplattinAvatar and DynamicFace. Ours shows robust generation results without artifacts, while baselines suffer from severe artifacts. In specific, ours shows outstanding results for rendering eye regions, *e.g.*, the first, second, and third rows, including accessories, *e.g.*, eyeglasses in the third row. Best viewed zoom-in.

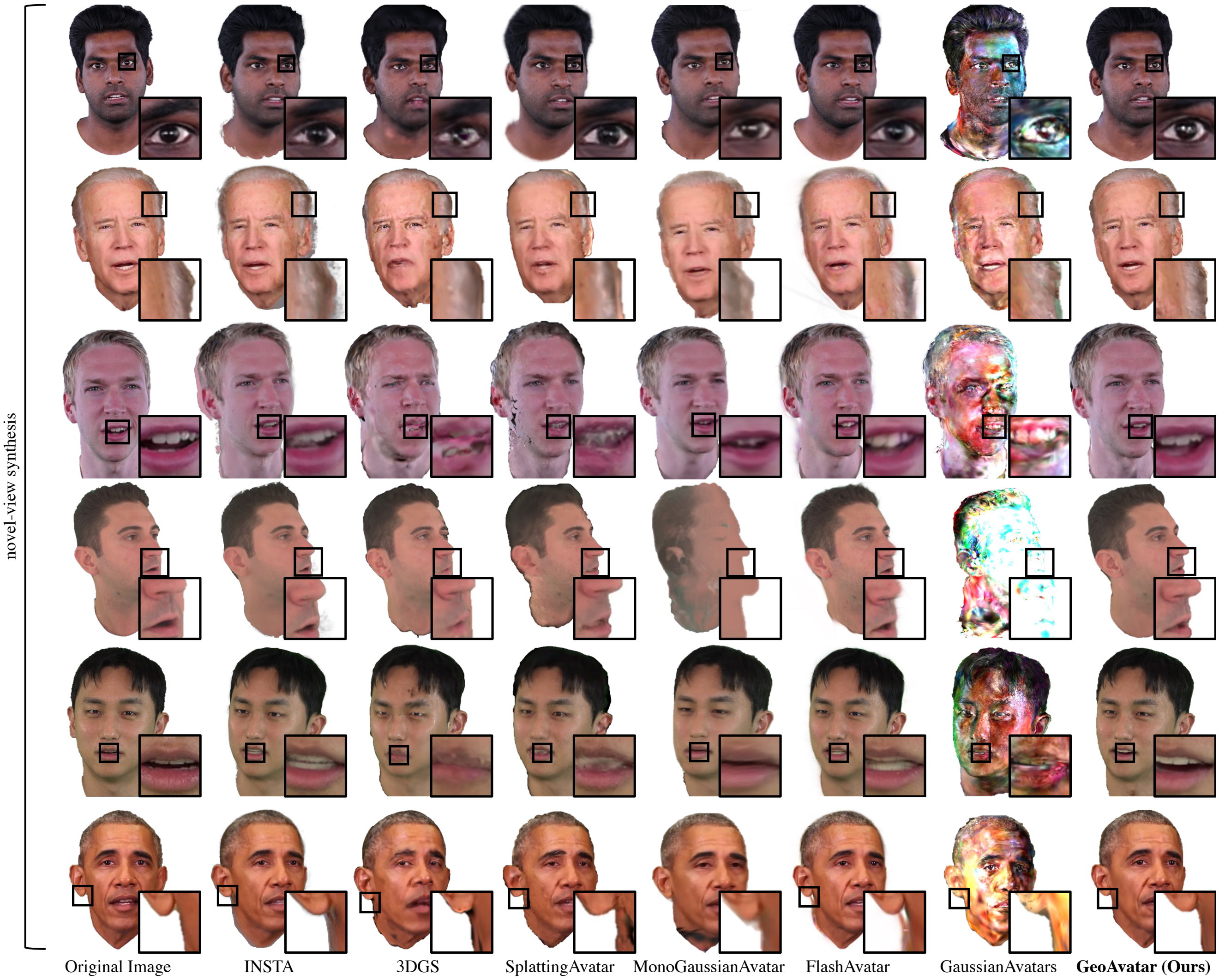


Figure 17. **Additional novel-view synthesis results.** We show additional novel-view synthesis results of ours, compared to baselines, on SplattingAvatar and DynamicFace datasets. Ours shows clean and robust results not only on the facial region, *e.g.*, the first, third, and fifth row, but also on the boundaries, *e.g.*, the second, fourth, and sixth rows. However, baselines suffer from either artifacts and low-resolution results in the facial region, or noisy boundaries in the boundary. Best viewed zoom-in.

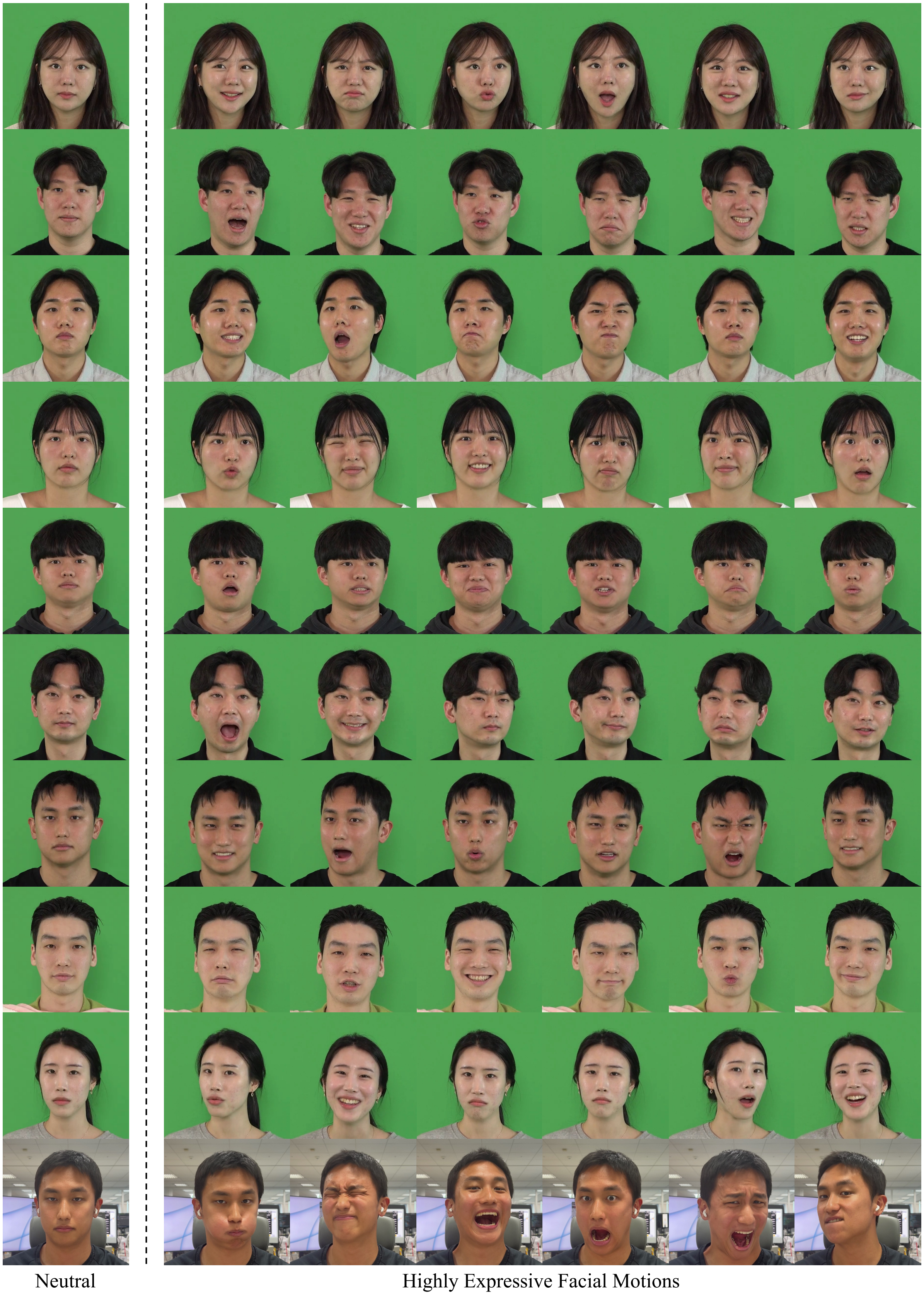
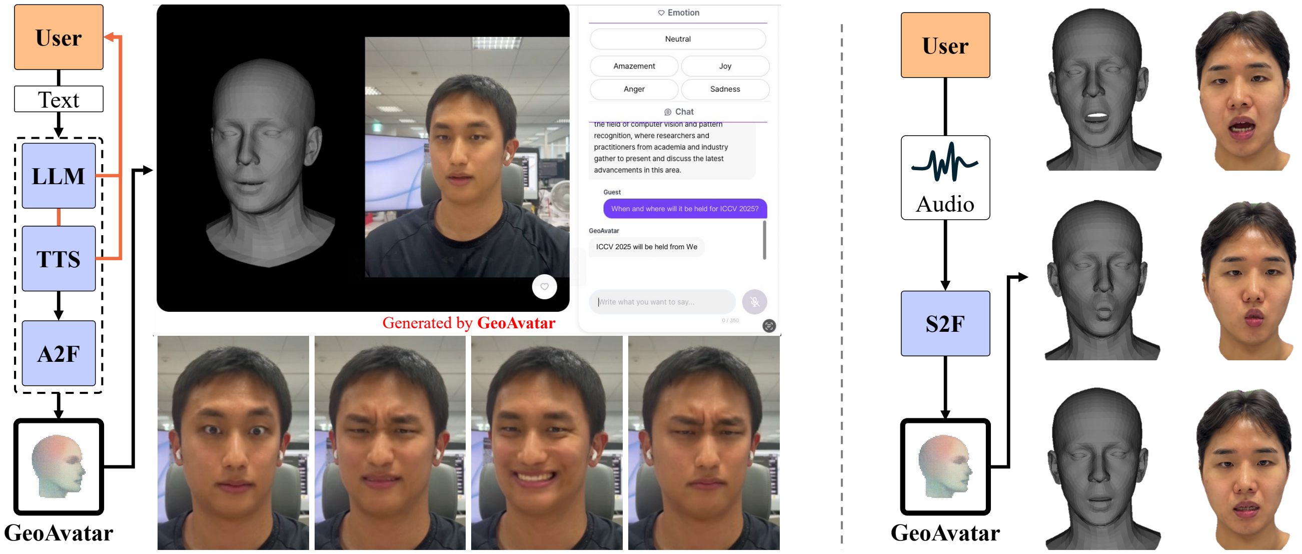


Figure 18. **Examples of DynamicFace sequences.** We show our DynamicFace example sequences for all subjects. Our DynamicFace has diverse highly expressive facial motions.



the field of computer vision and pattern recognition, where researchers and practitioners from academia and industry gather to present and discuss the latest advancements in this area.

Guest: When and where will it be held for ICCV 2025?

GeoAvatar: ICCV 2025 will be held from We

(a) Application #1: Interactable Digital Human

(b) Application #2: Virtual Presentation

Figure 19. **Application examples.** Our proposed GeoAvatar framework shows its versatility across multiple applications. (a) Real-time Interactable Digital Human Demo: By integrating a large language model (LLM), text-to-speech (TTS), and NVIDIA Audio2Face (A2F) module, GeoAvatar enables real-time, interactive conversations with a fully animated digital human. Additional post-processing for backgrounds is used to enhance visual outputs. (b) Virtual Presentation: Given an input audio, GeoAvatar utilizes a speech-driven 3D facial animation module (S2F) to generate a high-quality digital human capable of delivering presentations with natural facial expressions and lip synchronization. Both demo videos are visualized in our submitted project page HTML file.