

Gaussian Head & Shoulders: High Fidelity Neural Upper Body Avatars with Anchor Gaussian Guided Texture Warping

Supplementary Material

In this supplementary material, we provide additional implementation and evaluation details in Sec A, as well as extended results including additional ablation studies, limitations, and a comparison with SMPL-driven body avatar in Sec B. Ethic discussions are in Sec C. We also highly recommend the readers to view our supplementary video.

A IMPLEMENTATION DETAILS

A.1 PREPROCESSING

Our data preprocessing pipeline for extracting FLAME parameters, camera parameters and body landmarks is modified from (Zheng et al., 2022). After obtaining rough FLAME parameters from DECA (Feng et al., 2021), we further optimize the FLAME parameters to minimize the 68 facial landmarks for 3000 iterations. For subject 001, we keep the original training and test split used by PointAvatar (Zheng et al., 2023). For other subjects, we use the last 500 or 1000 frames as test frames, depending on the total frame count in the video. For all subjects, we simply use the first frame as the canonical training frame for initializing anchor Gaussians and updating the anchor correspondences. We use DWpose (Yang et al., 2023) to detach the noise, neck and shoulder landmarks, which are illustrated in Fig 9.

A.2 NETWORK ARCHITECTURE

We have three MLPs in total: MLP_d which predicts the expression blendshapes \mathcal{E} , pose blendshapes \mathcal{P} and LBS weights \mathcal{W} for each regular Gaussian and anchor Gaussian; MLP_f which predicts pose-dependent fine texture; MLP_w which warps view space coordinates to texture space coordinates. All three MLPs have 4 hidden layers and 128 neurons in each hidden layer. The standard Fourier frequency positional encoding (Mildenhall et al., 2020) is applied to the pixel coordinate, FLAME head rotation, camera translation and 2D landmarks before inputting to MLP_f and MLP_w . The pixel coordinate and 2D landmarks are encoded with a frequency of 10, and camera translation and FLAME head rotation are encoded with a frequency of 2. All three MLPs are initialized to predict 0s at the beginning by setting the weights and bias of the output layer to 0. All MLPs use ReLU as the intermediate activations. Tanh is used as the final activation for MLP_f , no final activation is used for MLP_w , and the final activation for MLP_d are the same as (Zheng et al., 2023).

We use a latent dimension $D_t = 32$ for the latent texture \mathbf{T}_f . The coarse texture \mathbf{T}_c is initialized to be the same as the white background, while the fine latent \mathbf{T}_f is initialized and a random and uniform distribution between $[0, 1]$.

A.3 TRAINING DETAILS

For all subjects, we use $\lambda_{head} = 1$, $\lambda_{anchor} = 1$, $\lambda_{warp} = 0.025$, $\lambda_{\hat{\alpha}} = 0.15$. For VGG loss weight λ_{VGG} , we set it to 0 for the first 10K iterations, and then 0.1 for the rest of the training. This is needed as we empirically observe that training the neural texture and warping field with a strong VGG loss from the beginning severely harms their stability. The weights of FLAME regularization are initially set to $\lambda_{\mathcal{E}} = 1000$, $\lambda_{\mathcal{P}} = 1000$, $\lambda_{\mathcal{W}} = 1$ and are reduced by half at 15k, 30k, 45k iteration respectively.

We train our model with Adam optimizer for 70k iterations in total, where the three stages of our training take 4k, 46k and 20k iterations respectively. The learning rate for blendshapes and LBS weight MLP MLP_d , neural texture, anchor Gaussian parameters and neural warping field are set to 10^{-3} , which is halved at 30k-th and 60k-th iterations respectively. The learning rate and density control hyperparameters for regular Gaussians are the same as proposed by the original paper (Kerbl et al., 2023), except that we use a density gradient threshold of 2.5×10^{-4} before we start applying VGG loss, and 8×10^{-3} afterward. For every 10k iterations during the training, we also re-project all

	001			002			003			004		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
INSTA	18.58	0.751	0.269	22.90	0.880	0.177	22.24	0.809	0.175	19.45	0.784	0.310
SplattingAvatar	18.49	0.737	0.307	25.34	0.876	0.171	21.34	0.790	0.220	19.83	0.765	0.351
PointAvatar	22.83	0.822	0.100	30.61	0.924	0.062	28.12	0.874	0.077	23.99	0.837	0.133
FlashAvatar	19.87	0.782	0.133	25.44	0.894	0.082	24.79	0.869	0.063	20.42	0.795	0.216
GaussianAvatars	19.96	0.774	0.184	24.52	0.895	0.094	23.15	0.828	0.107	19.37	0.813	0.308
GS*	23.26	0.814	0.082	32.99	0.937	0.046	29.85	0.888	0.054	24.18	0.836	0.139
Ours	25.95	0.856	0.064	31.98	0.949	0.042	31.26	0.917	0.042	24.68	0.839	0.120
Ours No MLP	24.48	0.840	0.070	31.44	0.942	0.042	28.85	0.892	0.051	24.61	0.837	0.120

	005			006			007			008		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
INSTA	19.47	0.757	0.251	23.44	0.861	0.165	18.68	0.733	0.291	19.97	0.675	0.246
SplattingAvatar	20.06	0.763	0.250	22.78	0.838	0.201	20.15	0.754	0.257	19.97	0.665	0.432
PointAvatar	22.82	0.847	0.142	29.42	0.929	0.043	22.30	0.826	0.088	21.61	0.748	0.174
FlashAvatar	19.65	0.789	0.152	24.25	0.871	0.060	20.02	0.770	0.116	20.56	0.691	0.197
GaussianAvatars	17.72	0.792	0.200	24.48	0.877	0.137	19.72	0.773	0.186	18.86	0.654	0.328
GS*	22.80	0.847	0.129	29.56	0.924	0.039	22.31	0.820	0.099	22.60	0.762	0.173
Ours	24.48	0.895	0.074	30.97	0.943	0.033	23.26	0.856	0.074	21.47	0.726	0.111
Ours No MLP	22.19	0.860	0.078	28.71	0.912	0.037	21.49	0.827	0.081	22.02	0.765	0.116

Table 3: **Quantitative evaluation of full self-reenactment task** We report PSNR \uparrow , SSIM \uparrow , and LPIPS \downarrow , and color the **best** and **second-best** methods for each subject respectively.

anchor Gaussians to the image plane of the canonical image plane, and remove the anchor Gaussians that are out of the view frustum. This is to prevent unconstrained anchor Gaussians from applying noisy regularization on the texture warping field.

Following (Zheng et al., 2023) and (Zheng et al., 2022), we also add a static bone, which does not take any transformation with the FLAME expression and poses.

As our preprocessing pipeline does not track eye movement, for subjects with significant eye movements in the training frames, i.e., subjects 002 and 005, we do not update the opacity and SH of regular Gaussians in the third stage to prevent undesirable view-dependent artifacts. For subjects where the semantic mask fails, i.e., subject 003, the No MLP texture may contain significant noise in the head region. We hence manually define a rough bounding box for this subject to clean the No MLP texture for self-reenactment and cross-reenactment tasks.

The training takes around 2 hours for each subject on an RTX4080 Ti.

A.4 EVALUATION DETAILS

Following (Zheng et al., 2023) and (Grassal et al., 2021), we also fine-tune the pre-tracked FLAME expression, pose parameters, camera translation and body landmarks during the training to account for inaccuracies in the preprocessing pipeline. We use Adam optimizer with a learning rate of 10^{-4} and optimize them from the 30k-th iteration. For test-time tracking optimization, we only use L2 RGB loss. Since we do not have a direct gradient flowing back from the body texture to the FLAME parameters, we also optimize a translation and rotation offset for the body texture mapping.

B ADDITIONAL RESULTS

B.1 VIDEOS

We strongly encourage the readers to watch the videos containing self-reenactment and cross-reenactment results in the supplementary.

As shown in the videos, existing methods either fail to model the body properly (INSTA (Zielonka et al., 2022), SplattingAvatar (Shao et al., 2024)), or fail to learn the details on head and body

	001			002			003			004		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
INSTA	26.62	0.898	0.082	34.89	0.963	0.032	29.80	0.922	0.071	28.35	0.941	0.069
SplattingAvatar	24.29	0.876	0.109	32.66	0.958	0.034	25.06	0.881	0.098	27.13	0.932	0.073
PointAvatar	26.17	0.904	0.079	34.93	0.968	0.021	30.90	0.923	0.053	29.65	0.948	0.045
FlashAvatar	27.44	0.911	0.069	35.61	0.973	0.021	30.30	0.939	0.037	28.09	0.942	0.046
GaussianAvatars	25.52	0.896	0.078	33.11	0.960	0.038	27.52	0.886	0.058	27.75	0.945	0.064
GS*	27.10	0.906	0.062	37.61	0.975	0.015	32.26	0.928	0.038	30.55	0.950	0.042
Ours	29.31	0.926	0.047	36.91	0.981	0.013	33.36	0.943	0.030	31.58	0.957	0.039
Ours No MLP	29.16	0.924	0.048	36.89	0.981	0.013	32.06	0.939	0.034	31.45	0.956	0.041

	005			006			007			008		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
INSTA	29.15	0.940	0.054	33.43	0.977	0.022	22.98	0.871	0.119	33.18	0.975	0.022
SplattingAvatar	28.75	0.938	0.061	31.93	0.967	0.030	23.56	0.873	0.121	32.92	0.976	0.024
PointAvatar	31.39	0.952	0.036	34.94	0.981	0.016	24.85	0.893	0.062	32.32	0.977	0.025
FlashAvatar	31.03	0.957	0.030	34.00	0.982	0.017	23.14	0.881	0.073	33.03	0.980	0.018
GaussianAvatars	29.71	0.956	0.039	33.47	0.978	0.020	26.32	0.937	0.050	30.88	0.971	0.023
GS*	32.36	0.959	0.030	35.62	0.983	0.014	25.00	0.892	0.064	33.99	0.980	0.020
Ours	33.90	0.967	0.027	36.90	0.987	0.012	26.35	0.921	0.045	36.14	0.988	0.013
Ours No MLP	33.74	0.967	0.026	36.77	0.987	0.012	25.00	0.909	0.048	35.27	0.988	0.012

Table 4: **Quatitative evaluation of head-only self-reenactment task.** We report the metrics with the body region masked out. Note that the body region is still used during the training.

	002			005			007		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
No Anchor Loss	24.96	.910	.088	22.91	.854	.117	19.30	.773	.134
No Warp Loss	32.86	.949	.041	24.19	.891	.081	22.74	.848	.076
Ours	31.98	.949	.042	24.48	.895	.074	23.26	.856	.074

Table 5: **Quatitative ablation.** We show the anchor constraint is necessary for learning sharp and correct body texture. While the warp loss might not necessarily improve the performance for the self-reenactment task, it is needed for cross-reenactment with out-of-distribution poses.

(PointAvatar (Zheng et al., 2023)). While the pure Gaussian Splatting baseline (GS*) could learn the face and body with much better details, it still learns blurry textures and presents severe artifacts when the subject is moving in extreme head rotation. It is most obvious for the self-reenactment and cross-reenactment videos of subject 005 – many Gaussians modeling the cloth texture are not well-aligned with each other, as a result, they cannot move naturally with the head motion. In comparison, our method can learn extremely sharp textures with robust performance under novel poses and motions.

B.2 ABLATION

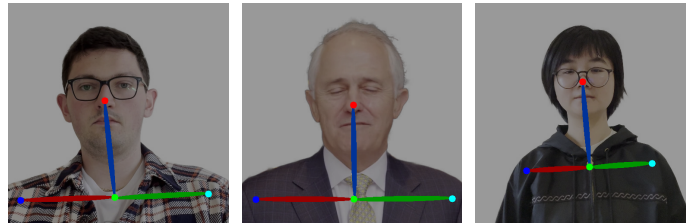
Additional ablation results are presented in Table 5 and Figure 10, demonstrating the critical role of the anchor loss in achieving sharp and precise textures. Although the warp loss \mathcal{L}_{warp} does not necessarily improve the numerical metrics for the self-reenactment task, Fig ?? illustrates its importance in preventing the significant failure when combining neural warping with additional Euclidean transformation.

B.3 TEXTURE CLEANING

When distilling the pose-dependent fine texture into the coarse texture for our no MLP version, we utilized DeepLabV3 (Chen et al., 2017) to obtain a coarse mask of the background and set the values of those pixels to 1. This is needed because the body texture contains a padding region to account for the body part that is moving in and out during the video. A majority section of the



Figure 7: Qualitative evaluation of self-identity reenactment.

Figure 8: **Qualitative evaluation of cross-identity reenactment.**Figure 9: **Landmarks.** We use DWPose (Yang et al., 2023) to detect nose, neck and shoulder landmarks to use as input to MLP_f and MLP_w .

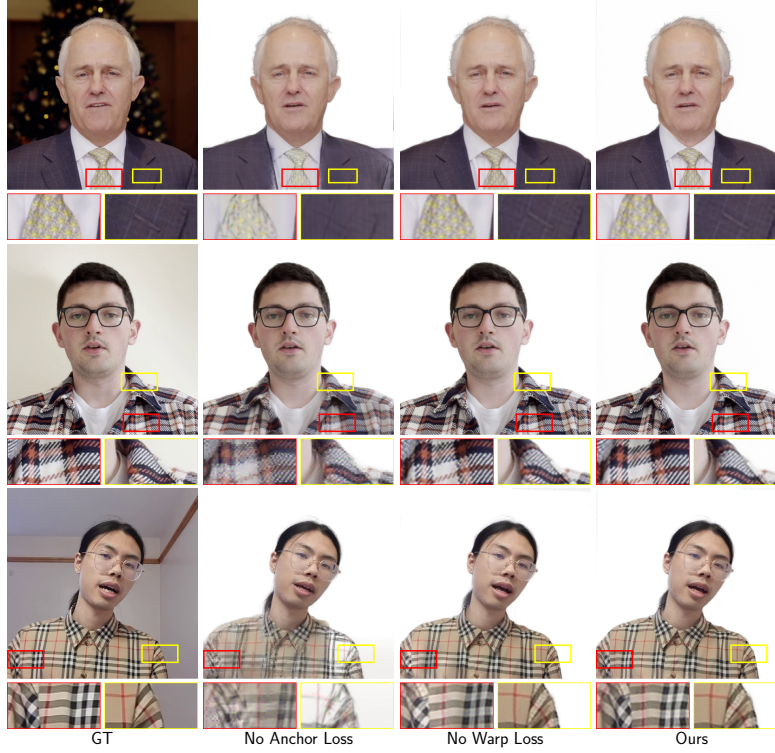
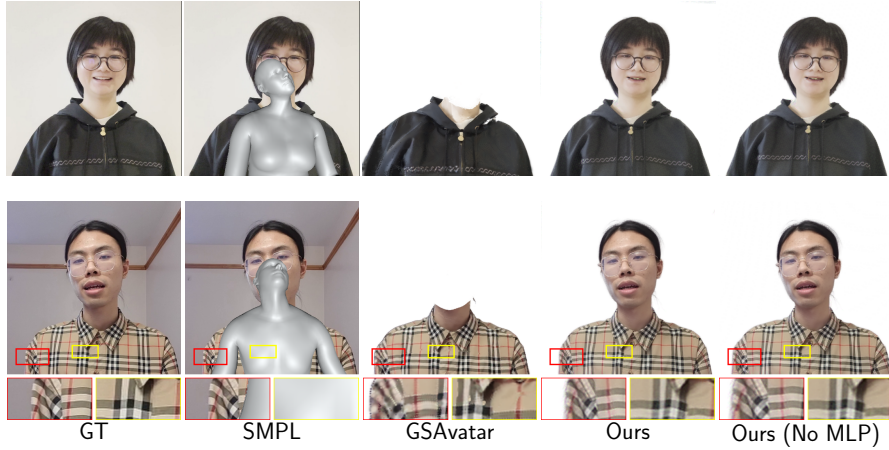
Figure 10: **Qualitative Ablation.**

Figure 11: **Qualitative comparison with full body avatar methods.** Due to the limited landmarks available on the shoulders and chest, existing SMPL tracking methods fail to obtain correct SMPL parameters. Fully body neural avatars that rely on SMPL hence fail to learn accurate and robust body. While our method does not include SMPL 3DMM, the use of static virtual bone and neural texture warping allow us to learn the body texture accurately.

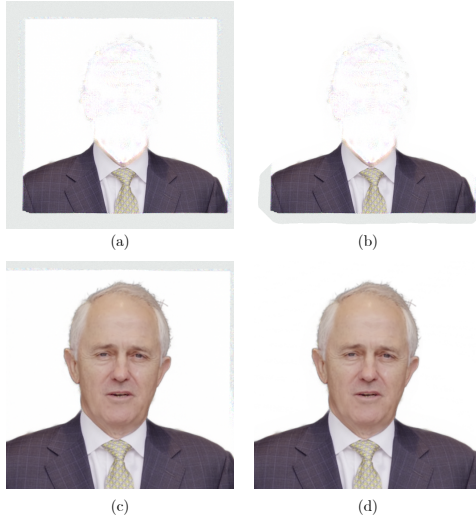


Figure 12: **Texture cleaning.** We show the body texture without masking (a) and with cleaning (b), as well as the rendering without texture cleaning (c) and with texture cleaning (d).

	004			007		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
GSAvatar	17.08	.811	.178	16.64	.744	.143
Ours	26.82	.887	.094	23.87	.885	.052
Ours No MLP	26.70	.885	.094	22.43	.861	.056

Table 6: **Body Only Quantitative Comparison with Full Body Avatars.** We show that existing full body neural avatar methods that rely on SMPL deformation perform significantly worse than our methods. Metrics are computed after masking out the background and head regions.

padding, especially the padding region on the top the left and right sides, are rarely used and trained during optimization. As a result, the fine texture colors obtained in those regions can produce noisy artifacts; see Fig 12.

B.4 COMPARISON WITH FULL BODY AVATARS

To verify our choice of driving anchor Gaussians only with head 3DMM (FLAME), we select two subjects that show a larger portion of the upper body and compare our method with GSAvatar, a Gaussian Splatting based full body neural avatar methods that deform the representation based on SMPL (Hu et al., 2024b). As the code release of GSAvatar only supports SMPL instead of SMPLX, we simply use semantic masks to remove the head region during the training and compare only the reconstruction quality of the body part. As shown in Tab 6 and Fig 11, since the existing SMPL tracking methods for monocular videos are developed only for views that include the whole body, the fitted SMPL is significantly misaligned with the GT (Sun et al., 2021), even after fine-tuning during Gaussian optimization. As a result, the clothed body reconstructed by GSAvatar presents several artifacts under novel poses and are significantly misaligned the GT. Our method is able to reconstruct the chest and shoulders with much better quality and accuracy. We would also like to note that, although we do not include body 3DMM in our method, due to the usage of virtual static bone, technically speaking, the effect is exactly the same as have a SMPLX 3DMM where the body and hand parts (SMPLX and MANO) are kept static during the whole sequences.

B.5 NOVEL VIEW SYNTHESIS

We show novel view synthesis results of our method in Fig. 14. Typically, because our method modeled the body as 2D texture, it would be difficult to render it from novel views, just as StyleA-



Figure 13: **SMPLX Estimation via OSX Lin et al. (2023)**. Some latest SMPLX prediction methods such as OSX Lin et al. (2023) are capable of predicting more accurate body 3DMM annotation than landmark optimization pipeline used in Hu et al. (2024b). However, as they are still mainly trained and optimized on frames with full-body or upper-body portraits with arms visible, their performance can be degraded with our tight framing setting: they tend to struggle with shoulders and can fail to detect any body with extreme poses such as the one shown in last column. Regardless, please note that we do not incorporate SMPLX not only because the annotation accuracy is not guaranteed, but also to keep a fair comparison with our baselines, where only FLAME 3DMM is used for LBS.



Figure 14: **Novel View Synthesis Results**. Since our method is trained only with monocular video where only limited view angles are included for the body, we can only render novel views with small displacement to the training views, similar to all other monocular neural avatar methods.

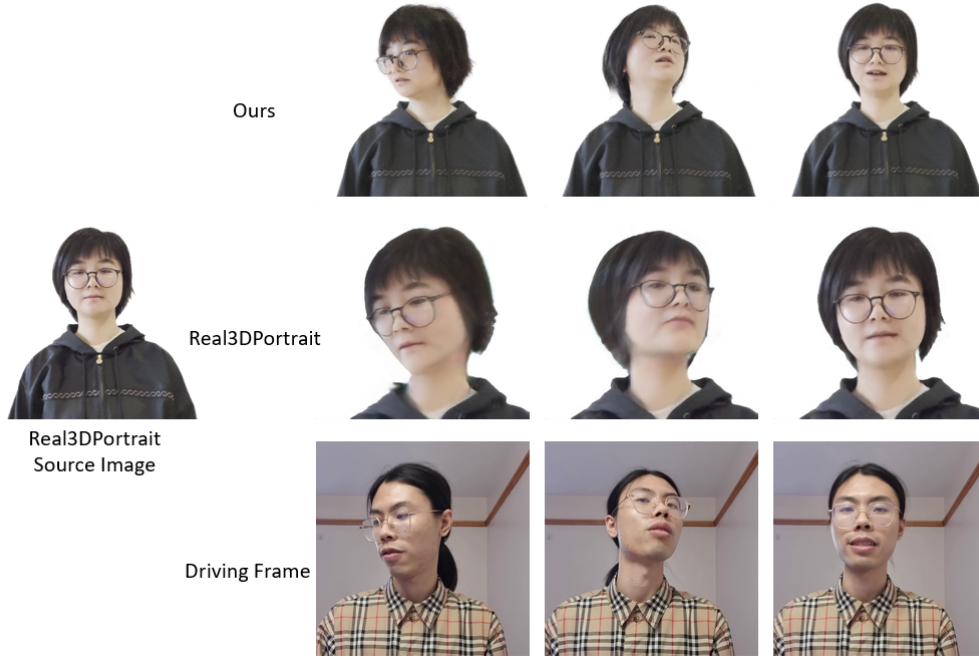


Figure 15: **Comparison with Real3DPortrait Ye et al. (2024) in Cross-Reenactment Task.** Although Real3DPortrait is trained with multi-identity datasets with rich facial prior extracted from the training, it fails to produce high-quality reenactment with extreme poses and cannot render shoulders and chest due to fixed tight framing in the training. Our method generates more faithful and accurate results in comparison.

vatar Wang et al. (2023). However, one key novelty of our method is the use of Anchor Gaussians as a constraint between 3D and 2D, and we can hence effectively utilize them to achieve a certain extent of novel view rendering. Specifically, we render the head Gaussians and the Anchor Gaussians at each novel view, reproject the Anchor Gaussians back to the image plane to obtain their 2D coordinates, and further compute a homography that minimizes the anchor constraint loss \mathcal{L}_{anchor} . This will ensure the body to move properly with the head and they always stay connected. Please note that similar to the existing neural avatar reconstruction method using monocular view, we can only render novel views with small displacement to the training views, as extrapolated views significantly degrade the results.

B.6 ADDITIONAL BASELINES

We include comparisons with additional baseline Real3DPortrait Ye et al. (2024); see Fig 15. StyleAvatar Wang et al. (2023) unfortunately degenerates and fails on our dataset; see Fig 16.

In Fig 17 and Table 7, we included comparison with Real3D-Portrait trained on single identity video. We trained the motion adapter for 100,000 steps on a single A100 GPU, which takes around 80 hours. We then trained the HTB-SR model for 80,000 steps, which takes around 30 hours. The inference speed of Real3D-Portrait is around 20 FPS on a single GPU. Note that in comparison, our method only requires less than 3 hours to train and can infer with around 130 FPS. It can be seen that our method is able to generate the head and cloth with much better quality. In Real3D-Portrait, a torso model is used to predict 2D warping from body keypoints to deform the latent image for fused body generation. While this approach can effectively learn to correctly connect the head to the body, without 3D-2D constraints from anchor Gaussian, it fails to learn sharp textures on the clothes. This result also matches our No Anchor Loss ablation in Fig 10.



Figure 16: **StyleAvatar Wang et al. (2023). Results.** We attempted to evaluate StyleAvatar on our dataset with the original framing. However, it seems that StyleAvatar quickly degenerates and fails after training for 10K iterations.

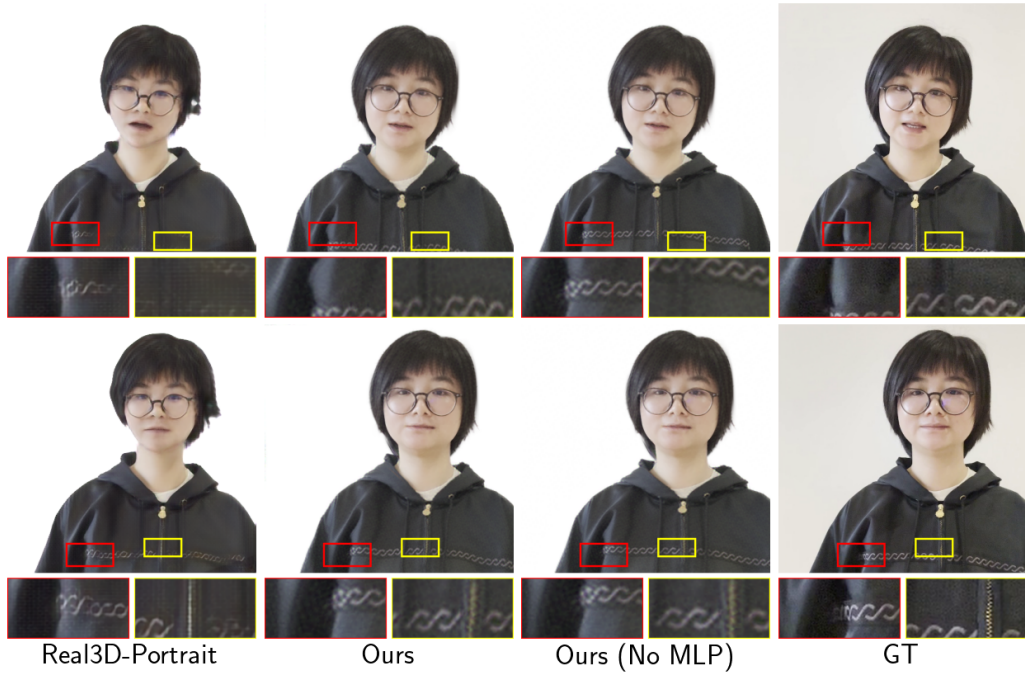


Figure 17: **Additional Comparison with Real3D-Portrait Ye et al. (2024).** We re-trained Real3D-Portrait on our single identity video to generate fair comparisons. We trained the motion adapter for 100,000 steps on a single A100 GPU, which takes around 80 hours. We then trained the HTB-SR model for 80,000 steps, which takes around 30 hours. The comparison shows that our method is able to reconstruct both the head and the cloth texture with much better quality.

	004		
	PSNR	SSIM	LPIPS
Real3D-Portrait	16.07	.739	.209
Ours	24.68	.839	.120
Ours No MLP	24.61	.837	.120
Real3D-Portrait (Head)	23.01	.895	.083
Ours (Head)	31.58	.957	.039
Ours No MLP (Head)	31.45	.956	.041

Table 7: **Quantitative Comparisons with Real3D-Protrait Hu et al. (2024a) in Self-Reenactment Task.**

B.7 LIMITATIONS

Although we propose a no MLP version that is able to render at novel poses with 130 FPS, as it completely relies on rigid homography transformation to map body texture to the view space, it is unable to model any non-rigid deformation in the body. In addition, for sequences with extreme head rotations, it might move the body in a way that is not exactly aligned with the ground truth, as shown in the supplementary videos. However, we observe that the results produced with this no MLP version still present a faithful rendering. For cases where the non-rigid body deformation is important, we recommend the use of the full version, whose rendering speed is around 70 FPS and can be further optimized by caching the fine texture only.

C ETHICS

We captured 4 human subjects with mobile phones for our experiments. All subjects have signed consent forms for using the captured video in this research and publication. We will release the data for subjects with permission.

Our method constructs faithful and animatable head avatars and can be used to generate videos of real people performing synthetic poses and expressions. We do not condone any misuse of our work to generate fake content of any person with the intent of spreading misinformation or tarnishing their reputation.