Supplementary Document for

MeGA: Hybrid Mesh-Gaussian Head Avatar for High-Fidelity Rendering and Head Editing

Cong Wang[1], Di Kang[2], Heyi Sun[1], Shenhan Qian[3], Zixuan Wang[4],
Linchao Bao[2], *Song-Hai Zhang[1]

[1]Tsinghua University, [2]Tencent, [3]Technical University of Munich, [4]Carnegie Mellon University

*Corresponding author: shz@tsinghua.edu.cn

Additional video results, including comparisons for novel view synthesis, novel expression synthesis, and cross-identity reenactment, are available on **our project page**.

In this document, we state our implementation details (Sec. A), baseline configurations (Sec. B), additional ablation studies on loss functions (Sec. C), and limitations (Sec. D). Detailed quantitative results for each subject are provided in Tab. 5 and Tab. 6. Comparisons of the average inference time, tested on an NVIDIA V100 GPU, are summarized in Table 3.

A. Network Structures & Implementation

Within our proposed framework, we train a Hybrid Mesh-Gaussian Head Avatar (MeGA) using multi-view videos as the supervision. Specifically, given a driving signal (i.e., FLAME parameters provided by GaussianAvatars [7]) and a view vector $\boldsymbol{d} \in \mathbb{R}^3$, three decoders are employed to predict the view texture map $\hat{\boldsymbol{T}}v \in \mathbb{R}^{1024\times1024\times3}$, the dynamic texture map $\hat{\boldsymbol{T}}dy \in \mathbb{R}^{1024\times1024\times3}$, and the UV displacement map $\hat{\boldsymbol{G}}_d \in \mathbb{R}^{256\times256\times3}$.

The view decoder $\mathcal{M}_v$ consists of 7 convolution layers and generates the view texture map $\hat{\boldsymbol{T}}_v$ from the tiled view vector (i.e., expanding the view vector $\boldsymbol{d}$ from $\mathbb{R}^3$ to $\mathbb{R}^{8\times8\times3}$). Both the dynamic decoder $\mathcal{M}_{dy}$ and the displacement decoder $\mathcal{M}_{disp}$ include one linear layer to map the FLAME expression ψ and pose ϕ parameters to a latent code $z \in \mathbb{R}^{256}$, followed by 7/5 convolution layers, respectively, to generate the dynamic texture map $\hat{\boldsymbol{T}}_{dy}$ and the UV displacement map $\hat{\boldsymbol{G}}_d$. As discussed in Sec. 3.1, $\hat{\boldsymbol{G}}_d$ is used to account for geometric details that cannot be represented within the FLAME space, and the three texture maps (i.e., $\hat{\boldsymbol{T}}_v$, $\hat{\boldsymbol{T}}_{dy}$, and $\hat{\boldsymbol{T}}_{di}$ which is a learnable latent map) are added up to generate the neural texture map. The neural textures are further decoded by our per-pixel decoder for the final RGB values. The structure of the per-pixel decoder are shown in Fig. 6.

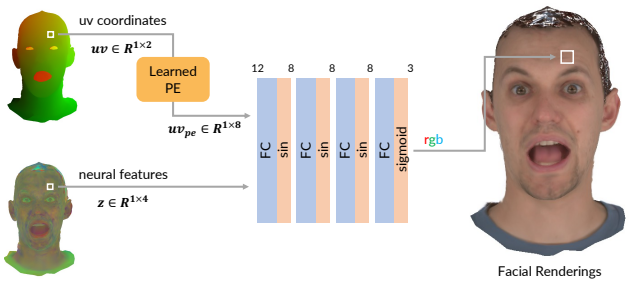

Figure 6. **The Lightweight Per-Pixel Decoder.** The learned positional encoding (Learned PE) is the same as PiCA [4].

For hair modeling, we initialize 3D Gaussian Splatting (3DGS) by sampling 50,000 on- and 100,000 off-surface points according to the scalp region of the tracked FLAME mesh. Using the densification and pruning strategies proposed by Kerbl et al. [1], we optimize the 3DGS from multi-view images of a selected training frame to produce a high-quality static canonical hair model (∼60,000 Gaussians). Our deformation field is represented by a two-layer MLP, with each layer comprising 256 neurons.

Loss weights in Sec. 4 are set to $\{\lambda_{pho} = 1.0, \lambda_{ssim} = 0.2, \lambda_d = 1.0, \lambda_n = 1.0, \lambda_{shr} = 1.0, \lambda_{lap} = 50, \lambda_{nc} = 0.1, \lambda_{el} = 100, \lambda_{silh} = 0.2, \lambda_{sol} = 0.1, \lambda_{aiap} = 1000\}$.

B. Baseline Configurations

All baselines are trained from scratch using the same train/test split as GaussianAvatars [7]. As outlined in Sec. 6, the quantitative metrics (i.e., PSNR, SSIM, and LPIPS) are calculated under the same masks for all baselines.

GaussianAvatars (GA). We use their public codes and identical settings to generate results for comparisons. All subject models are trained for 600,000 iterations to ensure

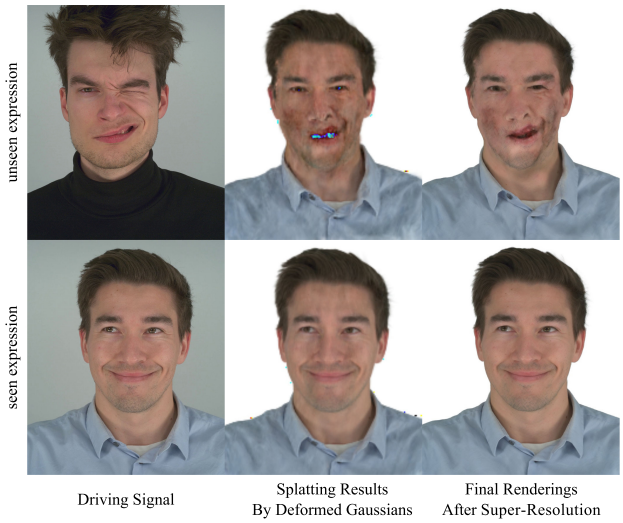

Figure 7. **Poor Generalization Performance of the Gaussian Head Avatar (GHA).** For unseen expressions/poses, the GHA cannot produce reasonable deformed Gaussians, resulting in the corrupted renderings.

convergence.

Gaussian Head Avatar (GHA). The original GHA paper [9] utilizes different subjects and expression sequences from the NeRSemble dataset [2] for training and evaluation. To enable comparisons, we download the raw NeRSemble video data and, following the instructions in their GitHub repository, process the same expression sequences as ours (i.e., EMO-1 to EMO-4, EXP-2 to EXP-5, and EXP-8 to EXP-9) for GHA's training/evaluating. All subject models are trained for 200 epochs to achieve convergence.

Note that the amount of training data in our experiments is significantly smaller than that used in the original GHA paper, which amplifies GHA's limitations in rendering novel expressions. Specifically, the deformation MLPs in GHA require a large amount of training data to achieve good generalization performance; otherwise, the GHA will produce rather bad renderings for unseen expressions. As shown in Fig. 7, when driven by unseen expression/pose parameters (row 1), the deformed head Gaussians result in corrupted splatting results (row 1, col 2) and poor final renderings, even after applying the super-resolution module (row 1, col 3). The renderings driven by seen expressions are shown in Fig. 7, row 2.

PointAvatar (PA). We use their publicly available codes for training, except for modifying their dataloader to support multi-view video data. Due to the limited memory of the NVIDIA V100 GPUs (32GB), we set the maximum number of points to 240,000. All subject models are trained for 63 epochs to achieve convergence.

DELTA. We make several modifications to the DELTA public codes to enable comparisons. First, we rewrite the dataloader to support multi-view video inputs. Next, since DELTA uses the SMPL-X [6] model instead of the FLAME model for their facial and body mesh, for fair comparisons, we write a Python script using PyTorch [5] to estimate SMPL-X parameters from our FLAME meshes by minimizing the point-to-point distances. The Python script will also be released later. All subject models are trained for 50 epochs to achieve convergence.

Note that we don't provide comparisons on alternating hairstyles with DELTA. The reason is that despite modifying their codes as described above, we were still unable to successfully generate hair transfer results. Additionally, since DELTA uses their own optimized camera parameters, which are estimated during the SMPL-X fitting procedure, we cannot freely control the viewpoint to facilitate dynamic video comparisons.

Table 3. **Comparisons on the averaged inference time.**

Methods	DELTA	PointAvatar	Gaussian Head Avatar	GaussianAvatars	MeGA (Ours)
Inference Time	214ms	293ms	59ms	**10ms**	37ms

Table 4. **Additional Ablation Studies on Subject 306.** We demonstrate the effectiveness of our introduced novel loss functions. (d) and (e) demonstrates that any loss functions introduced for the head and hair part presents positive effects on performance.

Label	Name	Losses	PSNR ↑	SSIM ↑	LPIPS ↓	Geo. MAE ↓
	MeGA (Ours)		**33.57**	**0.963**	**0.040**	2.25mm
(d.1)	MeGA-nodipho	w/o $\mathcal{L}^F_{di\text{-}pho}$	33.10	0.961	0.046	3.24mm
(d.2)	MeGA-noshrink	w/o L_{shr}	33.11	0.959	0.047	**2.17mm**
(d.3)	MeGA-noheadreg	w/o $\mathcal{L}^F_{reg}$	32.51	0.959	0.051	15.74mm
(e.1)	MeGA-nosolid	w/o L^H_{sol}	33.04	0.959	0.046	2.25mm
(e.2)	MeGA-noaiap	w/o $\mathcal{L}_{aiap}$	33.06	0.961	0.048	2.25mm

C. Ablation Studies on Loss Functions

Tab. 4 presents the quantitative results after removing certain loss functions. The removal of any loss function degrades the performance.

Specifically, (d.1) removes the diffuse loss $\mathcal{L}^F_{di\text{-}pho}$ during facial mesh optimization and joint optimization, resulting in worse mesh geometry (3.24mm vs. 2.25mm Geo. MAE) and final renderings (33.10 vs. 33.57 PSNR). More importantly, removing $\mathcal{L}^F_{di\text{-}pho}$ entangles the view- and expression-dependent effects into the diffuse texture map $\hat{\boldsymbol{T}}_{di}$, negatively impacting the subsequent texture editing functionality. (d.2) removes the shrink loss $\mathcal{L}_{shr}$ during facial mesh optimization. Without constraints applied to the scalp of the FLAME mesh, "MeGA-noshrink" achieves slightly better facial geometry (2.17mm vs. 2.25mm Geo. MAE). However, the scalp of the estimated FLAME mesh may become overly large and potentially cover the hair regions, leading to incorrect occlusion relationships and poor renderings (33.11 vs. 33.57 PSNR). (d.3) removes the head regularization term $\mathcal{L}^F_{reg}$ during facial mesh optimization, significantly harming the learned facial geometry (15.74mm vs. 2.25mm Geo. MAE) and resulting in worse render-

ings (32.51 vs. 33.57 PSNR). (e.1) removes $\mathcal{L}_{sol}^{H}$ during the optimization of the canonical Gaussian hair and joint optimization, leading to transparent Gaussian hair, impairing the disentanglement of the Gaussian hair and mesh head, and degrading final rendering performance (33.06 vs. 33.57 PSNR). (e.2) removes the as-isometric-as-possible regularization [8] during joint optimization. Without this rigid constraint, our MeGA tends to produce Gaussian floaters around the hair, particularly when rendering consecutive video frames.

D. Limitations & Discussions

While our approach effectively captures detailed skin textures (e.g., wrinkles), generates high-fidelity head renderings, and supports various editing functionalities, there are several limitations that require further exploration and improvement. (1) Due to the use of learning-based modules for facial appearance and geometry modeling, our approach typically depends on larger training datasets to achieve superior performance for novel expressions that significantly differ from those in the training set. Insufficient training data may lead to poor generalization performance. (2) Our network structures and supervision methods are specifically designed for multi-view videos with relatively dense viewpoints (e.g., larger than 16 views). For instance, the view decoder $\mathcal{M}_v$ requires dense view inputs to ensure optimal generalization performance, and our ground truth depth maps are derived from multi-view images using Multi-View Stereo (MVS), which also relies on dense views for sufficiently accurate results. We believe that addressing these limitations could potentially reduce the reliance on multi-view video setups, paving the way for monocular video setups and expanding the range of potential applications. (3) The current Gaussian hair in MeGA is rigid and cannot be used for further editing or physical simulation. We speculate that replacing the current Gaussian hair with some strand-based hair representations [3, 10] could produce better renderings and enable more versatile hair editing capabilities.

Table 5. **Comparisons with State-of-the-Art Methods on novel view synthesis.** MeGA achieves better LPIPS, SSIM, and PSNR (1dB higher than the 2nd best method on average). We bold (underline) the best (2nd best) results.

Subject	MeGA (Ours)			GaussianAvatars			Gaussian Head Avatar			PointAvatars			DELTA		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
074	**32**	0.931	**0.097**	31.88	**0.936**	**0.097**	27.81	0.904	0.124	25.67	0.905	0.172	23.56	0.842	0.173
104	**29.27**	0.923	**0.088**	29.20	**0.934**	**0.088**	27.52	0.883	0.094	23.62	0.891	0.116	21.69	0.836	0.142
218	**32.79**	**0.971**	0.052	31.79	0.964	**0.046**	26.54	0.881	0.082	27.99	0.932	0.073	24.94	0.901	0.144
253	**36.80**	**0.971**	**0.035**	35.68	0.966	0.037	30.82	0.908	0.065	27.48	0.929	0.073	25.71	0.881	0.117
264	**35.48**	0.971	**0.037**	34.29	**0.974**	0.042	30.64	0.900	0.086	27.64	0.940	0.077	25.01	0.873	0.126
302	**36.58**	**0.963**	**0.040**	35.52	0.954	0.055	31.42	0.913	0.071	28.57	0.925	0.078	24.97	0.869	0.152
304	30.19	0.901	**0.065**	**30.70**	**0.906**	**0.065**	29.30	0.822	0.099	24.99	0.862	0.114	22.45	0.837	0.186
306	**36.28**	**0.975**	**0.035**	35.51	0.953	0.039	29.69	0.906	0.080	28.36	0.931	0.065	26.47	0.890	0.108
460	**37.64**	**0.979**	**0.023**	37.27	0.974	0.027	31.59	0.927	0.052	29.42	0.949	0.049	26.81	0.908	0.097
avg.	**34.11**	**0.954**	**0.052**	33.54	0.951	0.055	29.48	0.894	0.084	27.08	0.918	0.091	24.62	0.871	0.138

Table 6. **Comparisons with State-of-the-Art Methods on novel expression synthesis.** MeGA achieves better LPIPS, SSIM, and PSNR (1dB higher than the 2nd best method on average). We bold (underline) the best (2nd best) results.

Subject	MeGA (Ours)			GaussianAvatars			Gaussian Head Avatar			PointAvatars			DELTA		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
074	**29.32**	0.917	0.096	28.90	**0.919**	**0.094**	21.24	0.826	0.158	24.24	0.883	0.132	21.37	0.836	0.173
104	**27.80**	0.921	0.098	27.60	**0.937**	**0.094**	20.31	0.816	0.167	22.57	0.899	0.111	20.81	0.824	0.168
218	**32.76**	**0.968**	**0.042**	30.89	0.962	0.045	23.88	0.869	0.131	28.25	0.935	0.073	24.37	0.903	0.148
253	**36.06**	**0.968**	**0.038**	33.12	0.964	0.041	25.31	0.871	0.124	26.33	0.925	0.081	23.25	0.874	0.133
264	**34.05**	0.966	**0.044**	33.36	**0.971**	0.046	21.34	0.880	0.123	27.12	0.939	0.076	23.71	0.847	0.154
302	**33.50**	**0.954**	**0.050**	32.28	0.945	0.062	22.64	0.861	0.134	25.88	0.916	0.156	22.38	0.851	0.179
304	**29.71**	**0.917**	**0.076**	29.43	0.903	0.081	20.91	0.797	0.166	23.98	0.874	0.121	19.48	0.824	0.201
306	**33.57**	**0.963**	**0.040**	32.57	0.948	0.044	21.20	0.859	0.180	25.54	0.922	0.079	23.49	0.884	0.146
460	**36.56**	**0.975**	**0.027**	34.94	0.974	0.031	25.79	0.899	0.114	28.24	0.952	0.099	24.56	0.879	0.121
avg.	**32.59**	**0.949**	**0.057**	31.45	0.947	0.060	22.51	0.853	0.144	25.79	0.916	0.103	22.60	0.858	0.158

References

[1] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139:1–139:14, 2023. 1

[2] Tobias Kirschstein, Shenhan Qian, Simon Giebenhain, Tim Walter, and Matthias Nießner. Nersemble: Multi-view radiance field reconstruction of human heads. *ACM TOG*, 42(4): 161:1–161:14, 2023. 2

[3] Haimin Luo, Min Ouyang, Zijun Zhao, Suyi Jiang, Longwen Zhang, Qixuan Zhang, Wei Yang, Lan Xu, and Jingyi Yu. Gaussianhair: Hair modeling and rendering with light-aware gaussians. *CoRR*, abs/2402.10483, 2024. 3

[4] Shugao Ma, Tomas Simon, Jason M. Saragih, Dawei Wang, Yuecheng Li, Fernando De la Torre, and Yaser Sheikh. Pixel codec avatars. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 64–73. Computer Vision Foundation / IEEE, 2021. 1

[5] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035, 2019. 2

[6] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 10975–10985. Computer Vision Foundation / IEEE, 2019. 2

[7] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 20299–20309. IEEE, 2024. 1

[8] Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 5020–5030. IEEE, 2024. 3

[9] Yuelang Xu, Bengwang Chen, Zhe Li, Hongwen Zhang, Lizhen Wang, Zerong Zheng, and Yebin Liu. Gaussian head avatar: Ultra high-fidelity head avatar via dynamic gaussians. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 1931–1941. IEEE, 2024. 2

[10] Egor Zakharov, Vanessa Sklyarova, Michael Julian Black, Gi-Joon Nam, Justus Thies, and Otmar Hilliges. Human hair reconstruction with strand-aligned 3d gaussians. *CoRR*, abs/2409.14778, 2024. 3