I M Avatar: Implicit Morphable Head Avatars from Videos

Yufeng Zheng[1,3] Victoria Fernández Abrevaya[2] Marcel C. Bühler[1] Xu Chen[1,3]
Michael J. Black[2] Otmar Hilliges[1]
[1]ETH Zürich [2]Max Planck Institute for Intelligent Systems, Tübingen
[3]Max Planck ETH Center for Learning Systems

This supplementary document provides additional ablation studies and results in Sec. 1, implementation and training details in Sec. 2, and discussion on broader impact in Sec. 3. Please also watch the accompanying video to see animated results and hear an explanation of our proposed method.

1. Additional Ablations and Results

1.1. Comparison with Additional SOTAs

Tab. 1 lists comparisons with additional SOTA methods where code is publicly available. We run the pretrained models for Zhakarov et al. [17] and Buehler et al. [3] and train HyperNeRF [13] for all four real subjects. Since HyperNeRF is not 3DMM-controllable, we condition the warping and slicing networks on the FLAME expression and pose parameters instead of a learnable latent code ω_i.

Method	Expression ↓	L_1 ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Zhakarov et al.	17.107	0.13929	15.24	0.8900	0.07040
VariTex	3.704	0.09968	17.01	0.9233	0.04890
HyperNeRF	7.201	0.08143	18.94	0.9207	0.03953
Ours	**2.548**	**0.04878**	**23.91**	**0.9655**	**0.02085**

Table 1. **Additional SOTA baselines.** We evaluate SOTA baselines on our real dataset, and provide quantitative comparisons.

1.2. Experiment on MakeHuman Synthetic Dataset [8]

Metric	Female 1	Female 2	Male 1	Male 2
↑ Mask IoU	0.981	0.983	0.977	0.977
↓ RGB L1(Intersec)	0.030	0.023	0.018	0.038
↑ Normal(Intersec)	**0.958**	**0.958**	**0.947**	**0.952**
[8] on Normal(Intersec)	0.94	0.95	0.94	0.94

Table 2. **MakeHuman.** IMavatar is competitive with concurrent work [8] without test-time pose optimization.

We follow Grassal et al. [8] and train with the same frames for 4 subjects from MakeHuman. Tab. 2 lists IoU between the predicted and GT masks, image L1 over the intersection, and compares geometry with [8]. Our method achieves more accurate geometry *without* test-time pose optimization used in [8]. Qualitative results are shown in Fig. 1.

1.3. Ablation on FLAME Pseudo GT Supervision

Our method can also be trained *without* 3DMM supervision, using only mask and RGB losses ('Ours-' in Tab. 3 and Fig. 2). Expression error is higher without pseudo GT supervision (row 1 and 2). However, with TrainData+, which contains

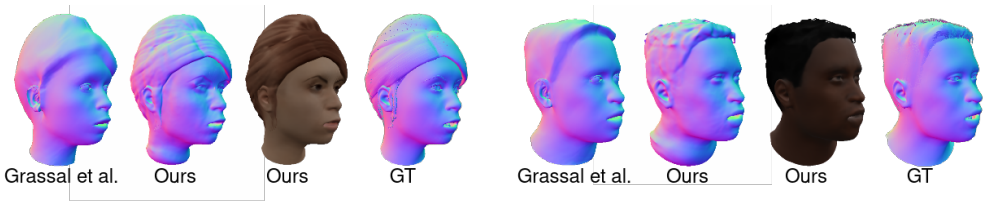

Figure 1. **MakeHuman.** IMavatar learns accurate and detailed deformable geometry from monocular RGB videos.

30% more frames and more expression variation than the original trainset, Ours- achieves comparable performance without leveraging pseudo GT (row 3 and 4).

Method	TrainData+	Expr. ↓	L_1 ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Ours-		3.337	0.0496	22.32	0.9532	0.0317
Ours		**2.973**	**0.0480**	**22.55**	**0.9572**	**0.0292**
Ours-	x	2.955	**0.0447**	21.98	0.9591	**0.0277**
Ours	x	**2.876**	0.0457	**22.07**	**0.9596**	0.0296

Table 3. **Pseudo GT supervision** improves metrics given limited data. With TrainData+, it is sufficient to learn unsupervisedly from images. Scores are calculated on one subject. See Fig. 2 for qualitative results.

Figure 2. Our method can be trained without FLAME pseudo ground-truth supervision. With more diversed training data, IMavatar learns more detailed expression and pose deformations. Neck geometry, however, is not guaranteed to be correct due to the lack of movement and ambiguity between head and neck rotation in the training data.

1.4. Mouth Interior Improvement

Figure 3. With more diverse data and pre-estimated semantic segmentations, IMavatar can learn better mouth interior geometry and texture.

Simply training on a dataset with more expressions (TrainData+) and setting blendshape supervision in the mouth interior region to 0 (with estimated semantic maps) faithfully reproduces teeth (Fig. 3)

1.5. Ablation on Pre-processing: Tracking and Segmentation

We experiment on Female 2 from MakeHuman.

3DMM tracking:. We add uniformly distributed noise to the fitted 3DMM global, neck and jaw poses, with a noise range from 0.025(1.4°) to 0.1(5.7°).

Foreground mask:. We randomly select a 61×61 square and set the mask value to True or False randomly. We degrade 10%, 50% and 100% of the masks. For both ablation experiments, the random degradation is a pre-processing step (not changed during training). See Tab. 4.

Metric	3DMM tracking			Baseline	Foreground Mask		
	0.1	0.05	0.025		10%	50%	100%
↑ Mask IoU	0.927	0.954	0.967	**0.983**	0.982	0.980	0.975
↑ Normal	0.905	0.938	0.959	0.958	**0.961**	0.958	0.932
↓ RGB L1	0.062	0.045	0.032	**0.023**	0.024	0.027	0.029

Table 4. **Ablation Pre-processing.** IMavatar relies on accurate 3DMM tracking, but it is reasonably robust to mis-segmentations.

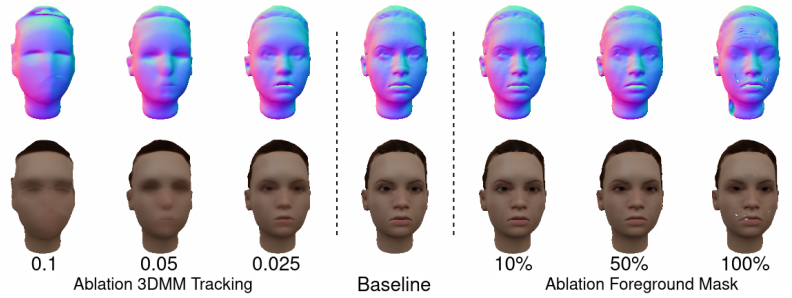

Figure 4. **Ablation Pre-processing.** Noisy 3DMM tracking is a major reason for blurry texture and geometry. Applying degradation to all masks leads to visible artifacts.

1.6. Jaw Pose Extrapolation

Fig. 4 in the main paper shows how the performance of baseline methods drops for stronger expressions. We extend this comparison by plotting the error with respect to the norm of the jaw pose parameter in Fig. 5. Our method achieves low errors even for strong jaw poses, while the error for baseline methods increases drastically.

1.7. Additional Interpolation Results

We show interpolations and extrapolations as animations in our supplementary video. For each expression, we interpolate the individual FLAME expression parameter from [-4, 4], and keep all other pose and expression parameters fixed as zero. We show the smiling (1st), upper lip lifting (2nd), lip side movement (3rd), and eyebrow raising (10th) expressions.

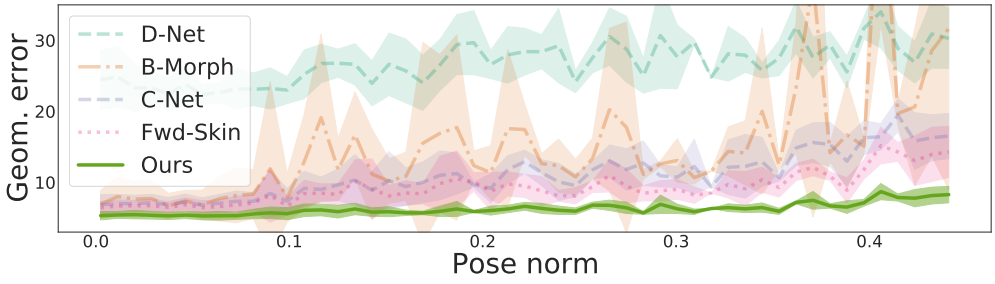

Figure 5. **Jaw pose extrapolation.** The x-axis denotes the norm of the jaw pose parameters in radians. The y-axis plots the angular error of the surface normals (lower is better).Performance of baseline methods worsen drastically as pose become more extreme.

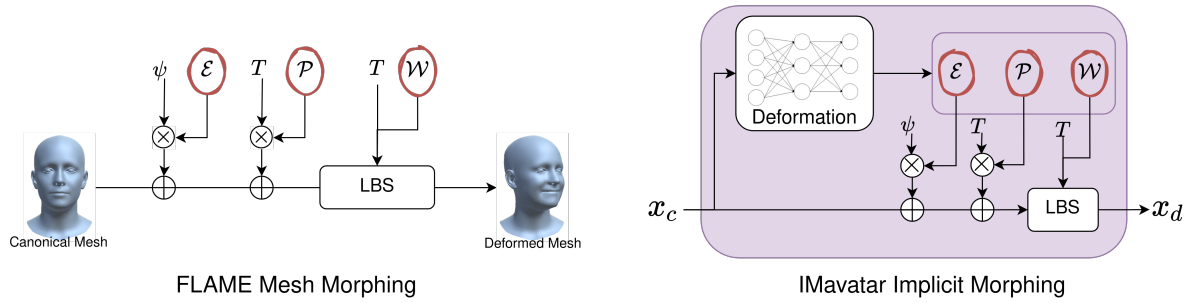

Figure 6. **FLAME morphing v.s. Implicit Morphing.**

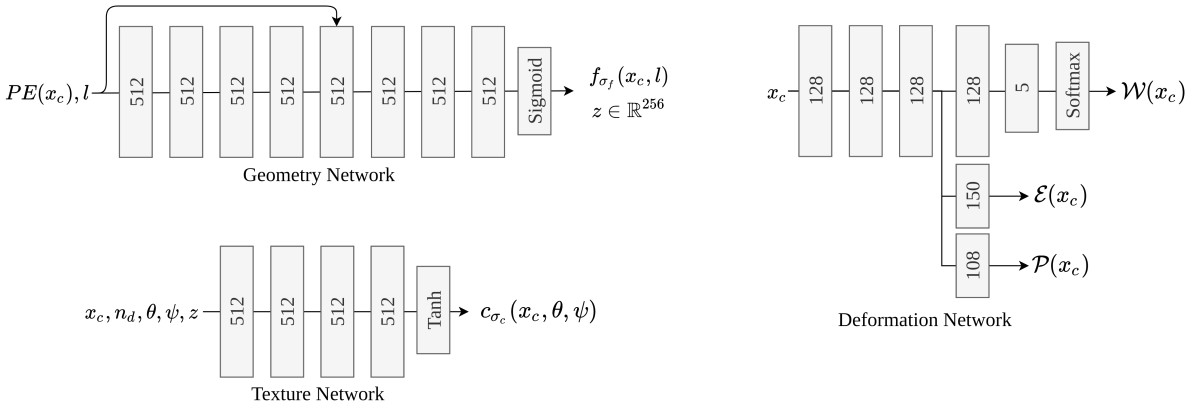

Figure 7. **Network Architecture.** Each block represents a linear layer with its output dimension specified in the inset, followed by a weight normalization layer [15] and an activation layer. We use Softplus [6] activation for the geometry and deformation network, and ReLU activation for the texture network. $z \in \mathbb{R}^{256}$ is the latent feature from the geometry network which is used as an input condition for the texture network.

2. Implementation Details

2.1. Visual Illustration of Mesh Morphing v.s. Implicit Morphing

2.2. Network Architecture

We implement our models in PyTorch [14]. The network architectures for the geometry-, texture-, and deformation-networks are illustrated in Fig. 7. We initialize the geometry network with geometric initialization [1] to produce a sphere at the origin. For the deformation network, we initialize the linear blend skinning weights to have uniform weights for all bones and the expression and pose blendshapes to be zero. For the geometry network, we use positional encoding [12] with 6 frequency components, and condition on a per-frame learnable latent code $\boldsymbol{l} \in \mathbb{R}^{32}$.

Fig. 8 shows the modified geometry network for the C-Net, and the deformation network for the D-Net and B-Morph baselines (see Section 4.2 in the main paper for definitions). We initialize the ablated geometry and deformation networks in the same way as our method. The displacement output for D-Net is also initialized to be zero.

2.3. Ray Tracing

Our ray tracing algorithm is similar to IDR [16], except that we do not perform the sphere ray tracing with signed distance values (SDF). This is because SDFs are not guaranteed to be correct in value after non-rigid deformation, and might lead to over-shooting. For this reason, we also eliminated the Eikonal loss [9] in IDR [16], and reconstruct an occupancy field instead of an SDF field.

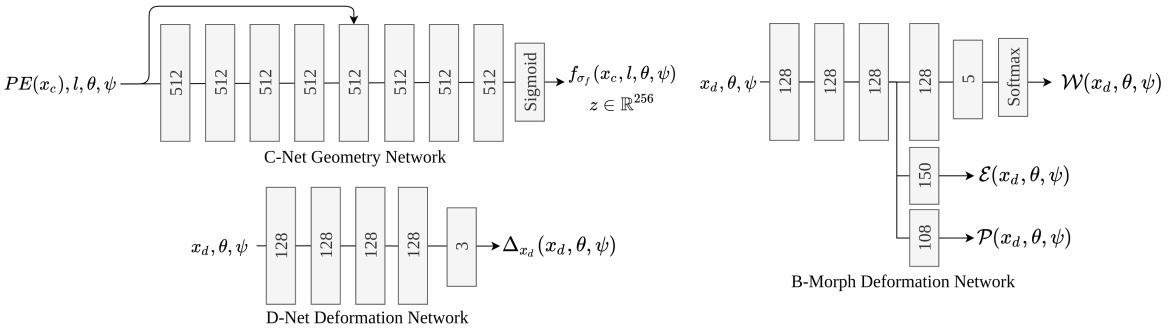

Figure 8. **Network Architecture for Baselines.** We show the modified geometry network for C-Net, which is additionally conditioned on the expression and pose parameters, ψ and θ. The deformation network for the B-Morph baseline is conditioned on the deformed point x_d and the expression and pose parameters. For D-Net, the input condition is the same as B-Morph, but the output is the displacement distance for the deformed location.

2.4. Correspondence Search

Following SNARF [5], for each deformed point x_d we initialize the canonical point x_c in multiple locations. More specifically, we inversely transform x_d with the transformation matrix of the head, jaw, and shoulder to ensure one of the initialized locations is close enough to the canonical correspondence. Then, we leverage Broyden's method [2] to find the root of $w_{\sigma_d}(x_c) = x_d$ in an iterative manner. We set the maximum number of update steps to 10 and the convergence threshold to $1e-5$. In the case of multiple converged canonical correspondences, the occupancy of the deformed point is defined as the minimum of all occupancy values.

2.5. Training Details

We train our network for 60 epochs with Adam optimizer [10] using a learning rate of $\eta = 1e^{-4}$, and $\beta = (0.9, 0.999)$. Learning rate is decayed by 0.5 after 40 epochs.

2.6. Real video dataset pre-processing

Our training and testing videos are all captured with one single fixed camera. For training, we record two videos: one head rotation video to capture the full facial appearance from different angles, and one talking video to capture common and mild expressions in a speech sequence. For testing, we ask the subjects to perform strong unseen expressions such as a big smile, jaw opening, pouting, and rising of the eyebrows.

For both training and testing videos, we use DECA [7] to regress the initial FLAME [11] shape, expression, and pose parameters. Unfortunately, the eye poses (gaze directions) are not tracked in our pre-processing pipeline. To refine the regressed FLAME parameters, we estimate the facial keypoint with [4] and optimize the regressed parameters and global translation vectors jointly. The primary optimization objective is the keypoint error:

$$E_{kp} = \left\| K(\theta, \psi, \beta, t) - K^{target} \right\|_2 ,$$

where $K(\theta, \psi, \beta, t)$ are the predicted 2D keypoints from FLAME pose, expression and shape parameters θ, ψ, β and global translation t, and K^{target} are the optimization targets predicted by [4]. We use one single shape parameter β for each subject. During optimization, we regularize shape and expression by:

$$E_{reg} = \lambda_\beta \left\| \beta \right\|_2^2 + \frac{\lambda_\psi}{T} \sum_{\tau \in [0,T]} \left\| \psi_\tau \right\|_2^2 ,$$

where β and ψ are the shape and expression parameters, T is the number of frames, and λ_β and λ_ψ are objective weights, set to $1e-4$ and $2e-4$, respectively. We also leverage temporal consistency terms for expression, pose and global translations:

$$E_{temp} = \frac{1}{T-1} \sum_{\tau \in [0,T-1]} (\lambda_\psi^{temp} \left\| \psi_{\tau+1} - \psi_\tau \right\|_2^2 + \lambda_\theta^{temp} \left\| \theta_{\tau+1} - \theta_\tau \right\|_2^2 + \lambda_t^{temp} \left\| t_{\tau+1} - t_\tau \right\|_2^2), \tag{1}$$

where θ and t are the pose parameters and global translation vectors. λ_{ψ}^{temp}, λ_{θ}^{temp} and λ_{t}^{temp} are set to $1e-3$, $\frac{2}{3}$ and $\frac{10}{3}$, respectively. The final optimization objective can be represented as:

$$E = E_{kp} + E_{reg} + E_{temp}.$$

We will release the pre-processing pipeline for real videos.

3. Broader Impact

Our work reconstructs a high fidelity facial avatar from monocular videos, which can extrapolate to unseen expressions given a training video of only mild deformations. This takes an important step towards democratizing 3D acquisition devices, as it does not require the user to have access to expensive capture equipment in order to get an animatable 3D model of itself. Thanks to the extrapolation abilities, it does not impose overly restrictive constraints in terms of the capture process itself, greatly simplifying it without sacrificing in geometric quality.

There is nevertheless the danger of nefarious use of any technology that can generate plausible renderings of individuals under fine grained control of expressions and head pose. The foremost danger here is the use of so-called deep-fakes and dispossession of identity. We are aware of the potential for abuse of our technology – despite its intended use for positive causes such as connecting people via mixed reality videoconferencing. We argue that performing research on topics such as this are best performed in an open and transparent way, including full disclosure of the algorithmic details, data and models which we intend to release for research purposes. While we may, unfortunately, not be able to prevent the development of deep-fake technologies entirely, we may however inform the general understanding of the underlying technologies and we hope that our paper will therefore also be useful to inform counter-measures to nefarious uses.

References

[1] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 4

[2] Charles G Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of computation*, 19(92):577–593, 1965. 5

[3] Marcel C. Buehler, Abhimitra Meka, Gengyan Li, Thabo Beeler, and Otmar Hilliges. Varitex: Variational neural face textures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 1

[4] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017. 5

[5] Xu Chen, Yufeng Zheng, Michael J. Black, Otmar Hilliges, and Andreas Geiger. SNARF: Differentiable forward skinning for animating non-rigid neural implicit shapes. In *Proc. International Conference on Computer Vision (ICCV)*, pages 11594–11604, Oct. 2021. 5

[6] Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2001. 4

[7] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3D face model from in-the-wild images. *ACM Transactions on Graphics (ToG), Proc. SIGGRAPH*, 40(4):88:1–88:13, Aug. 2021. 5

[8] Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. Neural head avatars from monocular RGB videos. *CoRR*, abs/2112.01554, 2021. 1

[9] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Proceedings of Machine Learning and Systems 2020*, pages 3569–3579. 2020. 4

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[11] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017. 5

[12] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 4

[13] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021. 1

[14] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward 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 H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 4

[15] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. 4

[16] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33, 2020. 4

[17] Egor Zakharov, Aleksei Ivakhnenko, Aliaksandra Shysheya, and Victor Lempitsky. Fast bi-layer neural synthesis of one-shot realistic head avatars. In *European Conference on Computer Vision*, pages 524–540. Springer, 2020. 1