Preface: A Data-driven Volumetric Prior for Few-shot Ultra High-resolution Face Synthesis – Supplementary Material

Marcel C. Bühler[1,2] Kripasindhu Sarkar[2] Tanmay Shah[2] Gengyan Li[1,2] Daoye Wang[2]
Leonhard Helminger[2] Sergio Orts-Escolano[2] Dmitry Lagun[2]
Otmar Hilliges[1] Thabo Beeler[2] Abhimitra Meka[2]
[1]ETH Zurich [2]Google

This supplementary document provides more details about our experimental setting in Sec. 1 and supplementary results and ablations in Sec. 2. For videos and ultra-high resolution results up to 4K, please see the supplementary HTML page.

1. Detailed Experimental Setting

1.1. Architecture and Hyperparameters

In the following, we describe the architecture of the prior and the finetuned model in detail and list the hyperparameters we used for training and finetuning our models.

1.1.1 Prior Model

Following Mip-NeRF [1], the prior model consists of two MLPs. The first MLP is the *proposal* network that only predicts density. The second MLP a neural radiance field (*NeRF*) that predicts both density and colour. The proposal MLP has 4 linear layers with $(256+512)\times 256$ parameters: 256 neurons for the features from the previous branch and 512 neurons for the concatenated latent code. The NeRF MLP has 8 linear layers with $(1024+512)\times 1024$ parameters: 1024 neurons for the features from the previous branch and 512 neurons for the concatenated latent code. The total parameter count of our prior model including all latent codes is 14.6 Mio.

During training and inference, we use three hierarchical sampling steps [10]. The first step uses 256 proposal samples, the second step 256 refined proposal samples, and the third step 128 NeRF samples.

We use the same number of positional encoding frequencies for both the proposal and the NeRF MLPs. The integrated positional encoding for the trunk networks $\hat{\gamma}_{\mathbf{x}}(\cdot)$ has 12 levels; the positional encoding $\gamma_{\mathbf{v}}(\cdot)$ for the view direction has 4 levels, and it appends the view direction without positional encoding. The view branch of the NeRF MLP has a bottleneck with width 256. The positionally-encoded view direction is concatenated to the bottleneck features and processed by a linear layer of width $(256 + 27) \times 128$ before being projected to RGB (256 bottleneck features and 27 features from the positional encoding of the view direction).

We optimise the prior model as an auto-decoder [2], where each identity has a latent code with 512 dimensions. Each training step samples 128 random rays from 8 views of 64 identities, which yields a batch size of $65,536$. We train our prior model for 1 Mio. steps, which takes 144 hours (approximately 6 days) on 36 TPUs. We optimise our model using Adam [7] with $\beta_1 = 0.9, \beta_2 = 0.999$. The learning rate starts at 0.002 and exponentially decays to 0.00002. We clip gradients with norms larger than 0.001.

1.1.2 Inversion

We perform inversion on the prior model to find a good initialisation for the finetuning. In each step, we sample 8 random patches of size 32×32 from all available views. We initialise the new latent code with zeros. The optimisation uses Adam with $\beta_1 = 0.9, \beta_2 = 0.999$ and a fixed learning rate of 0.001. We optimise for $1,500$ steps on 4 TPUs, which takes 10 minutes.

1.1.3 Finetuned Model

The architecture of the finetuned model is the same as the prior model, except for an additional linear layer that maps the features from the trunk to 3-d normal vectors.

We create batches of $8,912$ rays by sampling random pixels from all available views. We start with a learning rate of 0.001 and exponentially decay to 0.00002. The number of optimisation step depends on the resolution. For low-resolution (256×256), we optimise for $25,000$ steps. We increase the number of optimisation steps for higher resolutions: $50,000$ steps for 512×512; $100,000$ steps for 1024×1024; $200,000$ steps for 2048×2048; and $300,000$

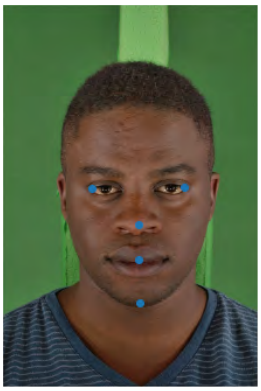

Figure 1. Visualisation of the five keypoints used for aligning captured subjects to a canonical pose.

steps for 4096×4096. We always optimise on four TPUs. The model finetuning takes 4 hours for $25,000$ steps and linearly increases for more training steps.

1.1.4 Camera Alignment

A crucial preprocessing step is to align all cameras to a canonical pose. As described in the main paper, we estimate five 3D keypoints on the outer eye corners, nose, mouth, and chin and calculate a similarity transform the the same five keypoints in a canonical space using Procrustes analysis. The canonical keypoints are computed as the median keypoint location across the first 260 subjects in our training set. Fig. 1 shows an example.

1.2. Studio Dataset

Our studio dataset consists of 1450 volunteers who were prompted to optionally self-report various characteristics like age, gender, skin colour and hair colour. We report the statistics here and in Fig. 2. 60% of the participants were male, 38% female, 0.2% non-binary and the rest preferred not to state. The age of the participants was heavily centred in the 24-50 age group. We also note the bias in appearance characteristics.

The participants were also given the option to wear or remove their glasses, hence a very small percentage $\sim$1% wore glasses during capture. The capture was performed over a period of many months. Initial captures contain a black background and was later changed to green screen to allow for better foreground segmentation if required. We do not mask out the background during prior model training. During finetuning, we estimate a foreground mask with a robust pretrained estimator [15]. Hence, our method works without any constraints on the background, as long as the camera poses are accurate.

$\mathcal{L}_v$	$\mathcal{L}_{\text{normal}}$	PSNR ↑	SSIM ↑	LPIPS ↓
×	×	23.91	0.7787	0.2233
×	✓	24.79	0.7839	0.2066
✓	×	25.53	0.7996	0.1963
✓	✓	**25.69**	**0.8040**	**0.1905**

Table 1. Ablation on regularisation when finetuning the model. The scores have been computed on models trained on two views with resolution 1024×1024 and averaged across six views of three holdout subjects. Please refer to Fig. 7 for visuals.

2. Supplementary Results and Analysis

This section supplements the results in the main paper with more visuals and detailed metrics. We provide supplementary results for comparisons related works in Sec. 2.1, more visuals for one- and few-shot synthesis in the studio setting and in-the-wild in Sec. 2.2, and a detailed analysis of our ablations in Sec. 2.3.

2.1. Supplementary Comparisons

This section supplements the comparison from the main paper with detailed metrics and visuals for individual holdout subjects.

2.1.1 Comparisons on Our Studio Dataset

This section provides supplementary results on our multiview studio dataset described in the main paper and in Sec. 1.2. Note that our goal is novel view synthesis so we refrain from comparing with methods that explicitly target geometry reconstruction [13, 14, 17, 22, 24].

We train the competing methods [4, 9, 12, 19, 23] on our dataset and compare with our results in Tbl. 6.

In the following, we describe the experimental details for each competing method.

For KeypointNeRF, we use their publicly available code and their default training and network settings. We manually chose 13 keypoints that closely resemble the ones shown in their paper (Fig. 11) and compute the near and far planes from our own dataset. We made a considerable effort to train them at 1K resolution, but we found that their results at the resolution 256 is of much higher quality than their results at 1K. Therefore, we present their results at both 1K resolution (Tbl. 6) and at 256 resolution (Tbl. 5). For the lower resolution comparison, we compare with our lower-resolution prior model trained at resolution 256×384.

For the comparison with RegNeRF [11], we train their model with the default settings provided by the authors for the DTU dataset [6], except for adjusting the near / far planes and scene scaling. We also disable the loss from the appearance regulariser because the model is not available.

For FreeNeRF, we implement their frequency regularisation with a 90% schedule into our pipeline. We do not em-

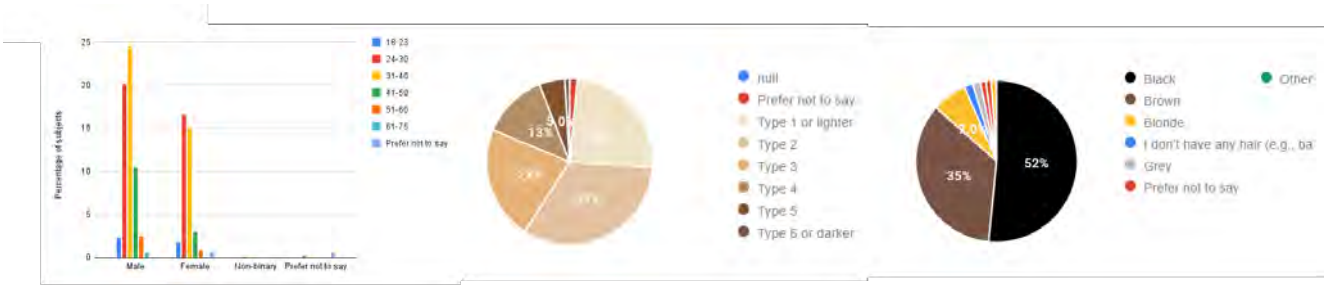

Figure 2. Distribution of characteristics in our dataset: we report the percentage distribution of our dataset by age, gender, skin colour and hair colour.

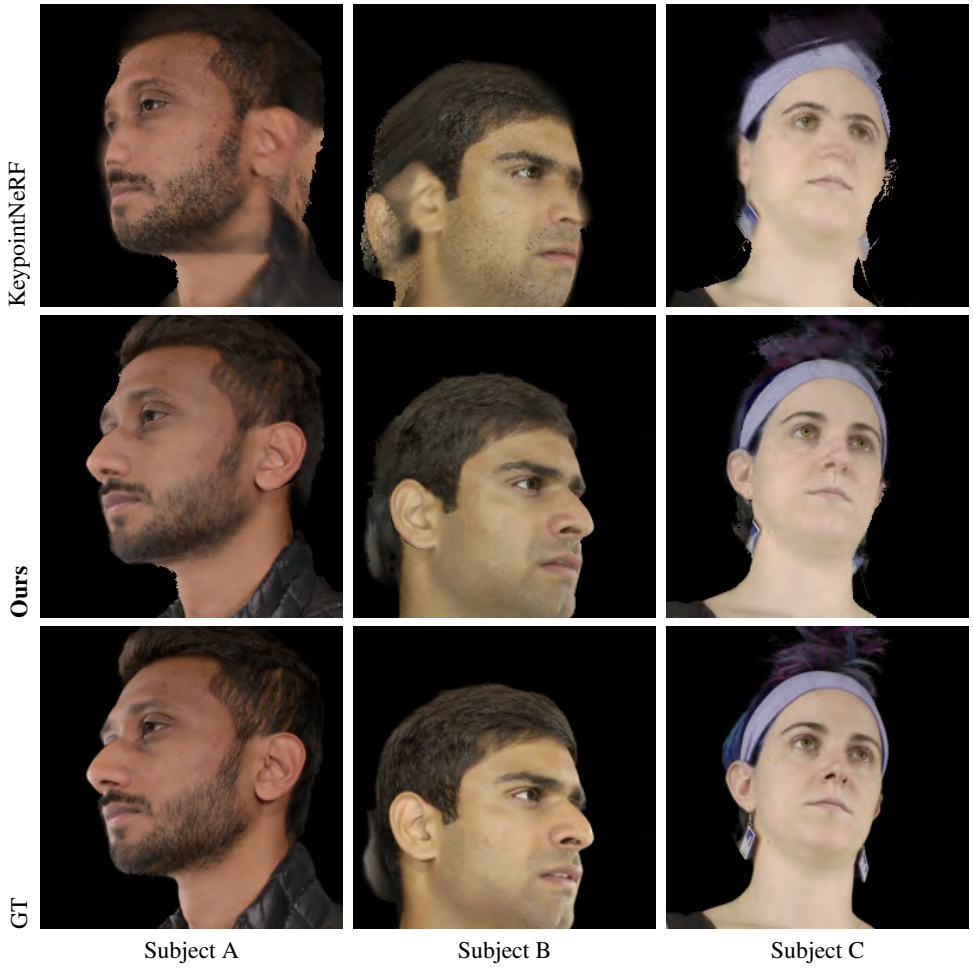

Figure 3. Visual comparison with KeypointNeRF [9] on low-resolution. Please see Tab. 5 for metrics.

ploy their occlusion regularisation because it causes transparent surfaces and floaters on our dataset.

For learnit [19], we adapt their publicly available notebook to work with our dataset. For training the meta model, we set the batch size to 4096, the number of inner steps to 64, the number of samples along the ray to 128, and train for 15,000 steps. We run the inference-time optimisation for the same number of steps as ours: 100,000 steps.

For the EG3D-based prior, we train a prior model with a triplane representation as proposed in Chan et al. [4]. The model is trained as an auto-decoder model similar to ours. We simultaneously optimize a per-identity latent code and the network weights to obtain an EG3D prior model that is finetuned to sparse views of a target subject for the same

Figure 4. Comparison with the state-of-the-art for novel view synthesis from sparse views on holdout identities from FaceScape [25]. For each identities, given four views as input, we show novel view reconstruction results and the L1 residue.

Objective	**PSNR ↑**			**SSIM ↑**			**LPIPS ↓**		
Subject	A	B	C	A	B	C	A	B	C
$\arg\min_{\theta_{\text{target}}}$	26.07	27.21	22.90	0.7949	**0.8000**	0.7998	**0.1823**	0.1651	0.2126
$\arg\min_{\theta_{\text{target}}, \mathbf{w}_{\text{target}}}$ (Ours)	**26.55**	**27.30**	**23.22**	**0.8113**	0.7996	**0.8009**	0.1962	**0.1650**	**0.2102**

Table 2. The model finetuning performs best when optimising both the model parameters Θ_{target} and the latent code $\mathbf{w_{target}}$. All metrics were computed after finetuning to two views at 1K resolution. Visually, the optimisation results look very similar, see Fig. 8.

number of steps as ours. We do not apply our additional regularisers when finetuning EG3D.

We train the EG3D prior on low-resolution images at resolution 256×256 that are super-resolved to resolution 1024×1024. The triplane resolution is 256×256 and the per-identity latent codes have dimensionality 512. Since the EG3D model requires rendering the full image, we reduce the number of initial samples per ray to 64 and the number

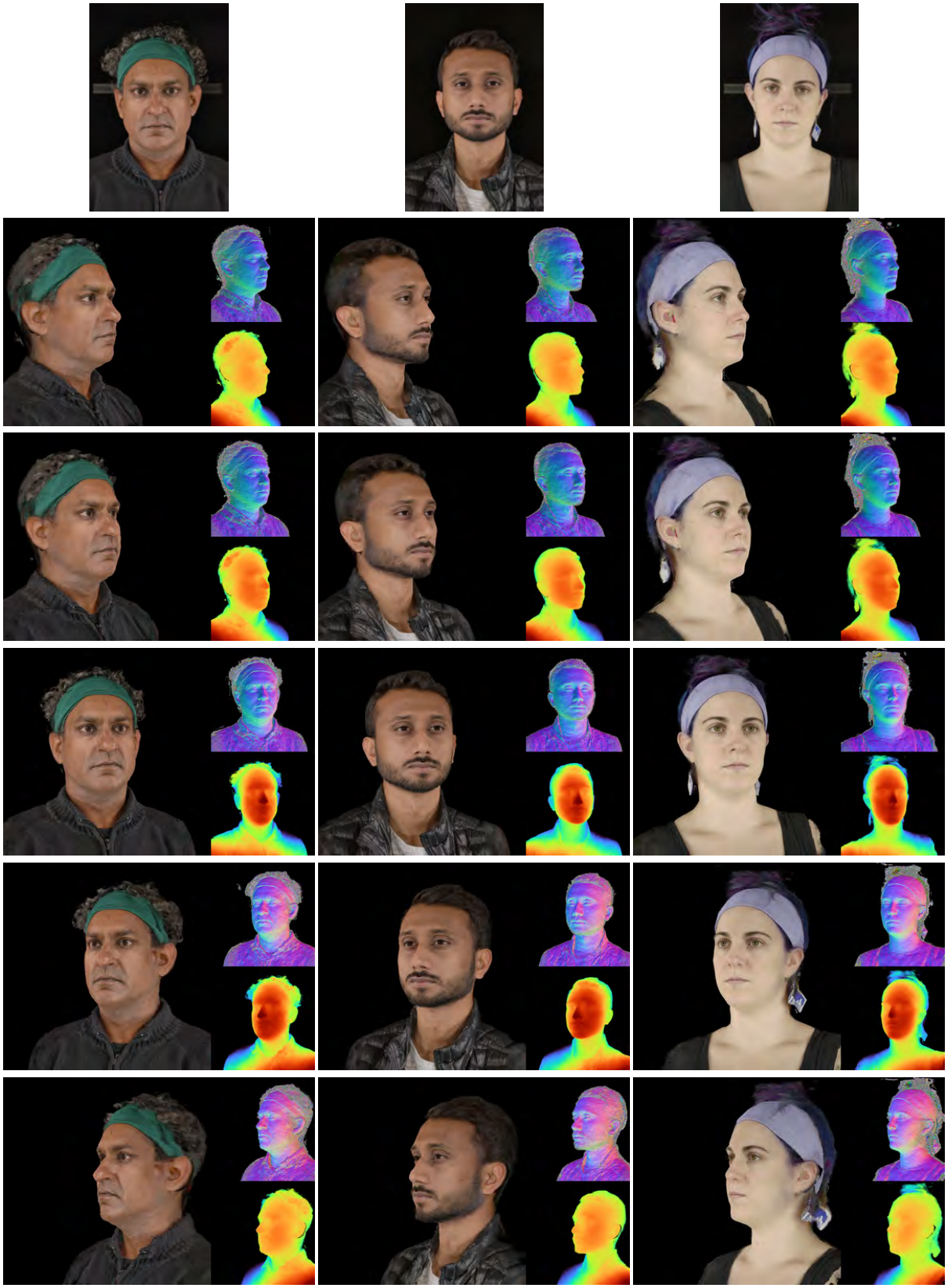

Figure 5. Single image reconstruction results from the main paper at higher resolution. The top row shows the input image captured in a studio setup. The rows below show synthesised views around the subject face using the image in the top row for model fitting. The inlays show the normals (top) and depth (bottom).

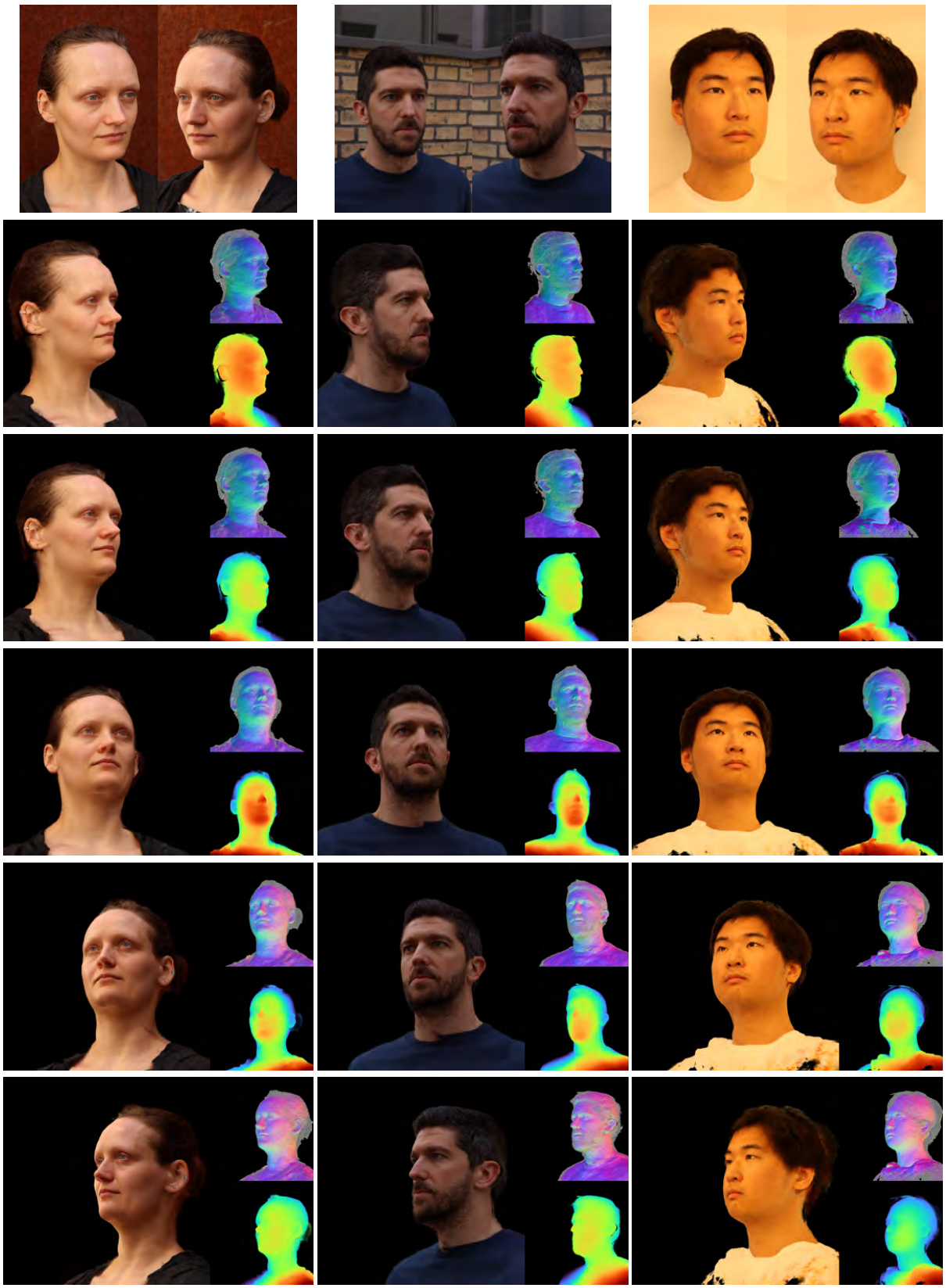

Figure 6. In-the-wild Results at Higher Resolution. We reconstruct a target identity from two images acquired with a consumer camera (left). Note how the novel views can extrapolate from the input camera angles. The inlays show the normals (top) and depth (bottom). The hair density is low, thus the grey normal colour in that region. We encourage the reader to see the supp. mat. for the high-resolution results and videos.

Figure 7. Visual results when applying regularisers. Training without regularisers ($\mathcal{L}_{recon} + \mathcal{L}_{prop}$, first column) leads to strong colour distortions for unseen views. Adding a regularisation loss on the model weights that process the view direction mitigates the colour distortions but yields fuzzy surfaces ($\mathcal{L}_{v}$, second column). Our final model employs an additional regulariser on predicted normals [21] to obtain well-defined surfaces ($\mathcal{L}_{\text{normal}}$, last column).

Initialisation	**PSNR** ↑			**SSIM** ↑			**LPIPS** ↓		
Subject	A	B	C	A	B	C	A	B	C
Mean	25.39	26.44	22.00	0.7963	0.7913	0.7927	0.1917	0.1749	0.2210
Noise	25.21	26.32	22.44	0.7993	0.7911	0.7966	0.206	0.1766	0.2169
Zeros	25.32	26.37	22.25	0.7956	0.7927	0.7939	0.1917	0.1732	0.2183
Furthest	24.07	25.57	22.09	0.7884	0.7829	0.7915	0.1997	0.1875	0.2250
Nearest	25.49	25.68	22.05	0.7934	0.7818	0.7948	**0.1915**	0.1852	0.2240
Inversion (**Ours**)	**26.55**	**27.30**	**23.22**	**0.8113**	**0.7996**	**0.8009**	0.1962	**0.1650**	**0.2102**

Table 3. Ablation on initialisation strategies for $\mathbf{w}_{\text{target}}$ for finetuning. This table lists metrics computed on face crops of 6 holdout views at resolution 1024×1024. *Furthest* (*nearest*) indicate initialising the latent code with the least (most) similar training subject. Figure 9 shows visuals examples.

of importance samples to 8.

For all methods, we perform the same inference-time bounding box based culling as we did for our method. Table 6 lists metrics for experiments on 2, 3, 5, and 7 views and Fig. 12, 13, and 14 show visual examples. Our method consistently outperforms related works.

We do not compare with DINER [16], Sparse NeRF [5], and SPARF [20] on our dataset because their training code is not publicly available at the time of submission.

2.1.2 Comparison on FaceScape

Figure 4 adds more examples for the comparison with Facescape [25], and Tbl. 7 lists metrics.

For the comparison on FaceScape [25], we obtain the outputs directly from the authors of DINER [16]. For each

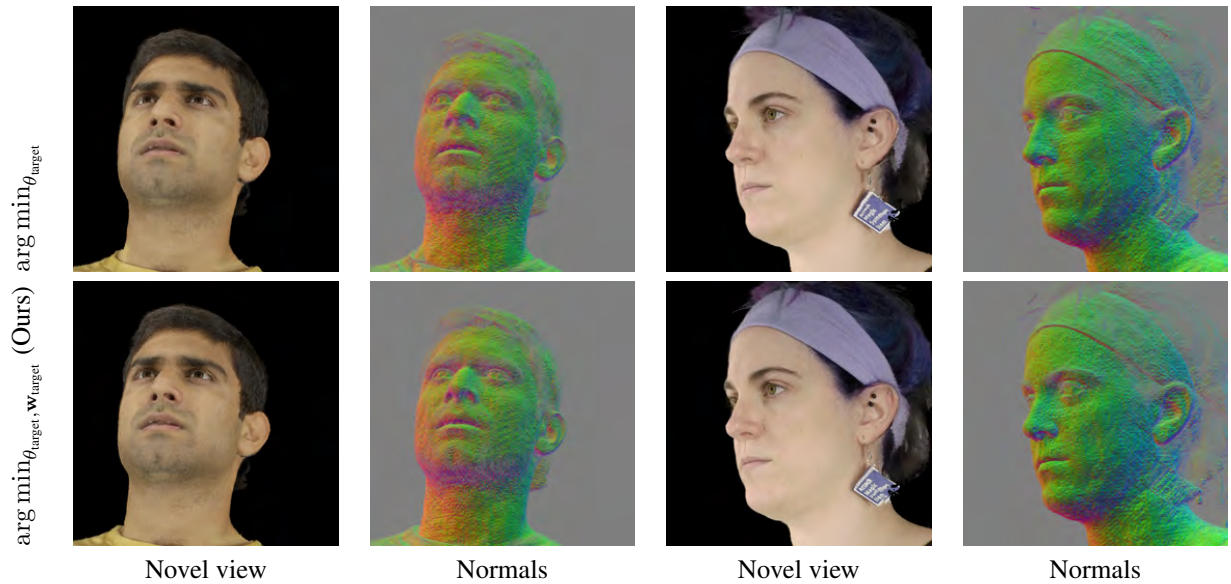

Figure 8. Effect of optimising only the model parameters θ_{target} (top row) and optimising both the model parameters and the latent code $\mathbf{w}_{\text{target}}$ (bottom row, Ours). The visual results are very similar. Tbl. 2 lists quantitative metrics.

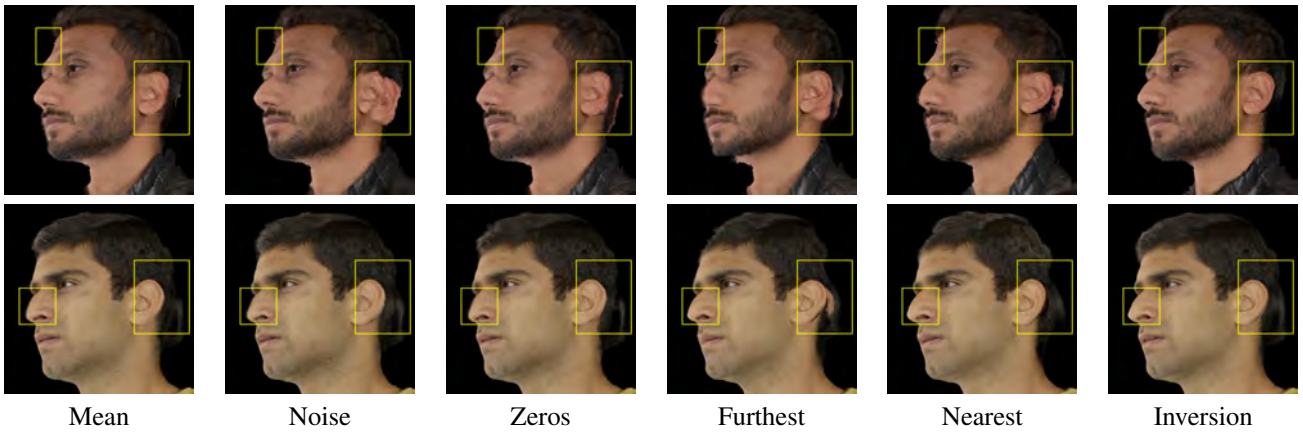

Figure 9. Visual comparison of different initialisation techniques. When the geometry is not initialised correctly at the start of finetuning, the final result can contain artifacts like a second ear, an unrealistic forehead, and a fuzzy surface. Starting from the inversion result mitigates these artifacts. Please see the text for an explanation of the different initialisation techniques and Tbl. 3 for metrics.

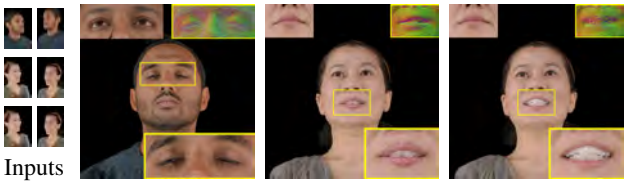

Figure 10. Out-of-distribution facial expressions. Our model was trained on neutral faces with a closed mouth. It can handle mild expressions but fails for strong expressions and teeth. We show a novel view with insets of the inversion result (top-left), normals (top-right) and a zoom-in patch (bottom-right).

# Views	PSNR ↑	SSIM ↑	LPIPS ↓
1	23.37	0.7658	0.2189
2	25.69	0.8040	0.1905
3	27.16	0.8275	0.1675
5	28.33	0.8445	0.1651
7	29.24	0.8600	0.1539

Table 4. Ablation on the performance for different number of views when finetuning the model. The scores are computed on models trained on images with resolution 1024×1024.

target identity, we perform model finetuning on two different subset of four views and average the scores. Since we develop our method on neutral faces, we filter out faces with non-neutral expressions.

For the comparison with RegNeRF [11], we follow the same protocol as described in Sec. 2.1.1. We follow

Method	PSNR ↑			SSIM ↑			LPIPS ↓		
Subject	A	B	C	A	B	C	A	B	C
KeypointNeRF [9]	24.47	23.42	20.33	0.7887	0.7736	0.7387	0.1866	0.1991	0.2462
Ours	**28.31**	**29.00**	**23.92**	**0.8703**	**0.8814**	**0.8321**	**0.1025**	**0.0937**	**0.1484**

Table 5. Comparison with KeypointNeRF [9] on our dataset. Despite considerable efforts, their implementation did not produce high-quality results at 1K resolution, hence, we compare on resolution 256×256. Please refer to Fig. 3 for visuals and Tbl. 6 for results at 1K resolution.

# Views	Method / Subject	PSNR ↑ A	PSNR ↑ B	PSNR ↑ C	SSIM ↑ A	SSIM ↑ B	SSIM ↑ C	LPIPS ↓ A	LPIPS ↓ B	LPIPS ↓ C
2	Learnit	22.07	21.18	16.86	0.7870	0.7765	0.7513	0.3068	0.3195	0.3635
	EG3D-based prior	20.25	20.60	18.24	0.7633	0.7575	0.7556	0.2678	0.2853	0.3159
	RegNeRF	20.63	19.93	20.63	0.7468	0.7361	0.7468	0.2791	0.2993	0.2791
	FreeNeRF	17.24	14.48	13.35	0.7091	0.6619	0.6675	0.2711	0.3140	0.3428
	KeypointNeRF	23.80	23.45	21.11	0.7964	0.7832	0.7838	0.2542	0.2628	0.2969
	Ours	**26.55**	**27.30**	**23.22**	**0.8113**	**0.7996**	**0.8009**	**0.1962**	**0.1650**	**0.2102**
3	Learnit	22.99	22.53	19.15	0.7939	0.7847	0.7775	0.2981	0.3031	0.3473
	EG3D-based prior	22.26	21.91	19.60	0.7902	0.7781	0.7823	0.2649	0.2819	0.3057
	RegNeRF	22.62	23.12	20.26	0.7794	0.7654	0.7714	0.2654	0.2768	0.3043
	FreeNeRF	24.71	21.74	21.52	0.7962	0.7582	0.7757	0.2150	0.2314	0.2622
	KeypointNeRF	24.62	24.52	22.19	0.8013	0.7904	0.7913	0.2364	0.2449	0.2751
	Ours	**27.89**	**28.86**	**24.72**	**0.8268**	**0.8305**	**0.8252**	**0.1633**	**0.1498**	**0.1893**
5	Learnit	23.03	23.01	18.54	0.7935	0.7874	0.7742	0.2991	0.3011	0.3494
	EG3D-based prior	20.16	21.32	19.13	0.7938	0.7832	0.7783	0.2694	0.2829	0.3137
	RegNeRF	24.85	23.56	20.93	0.7944	0.7787	0.7908	0.2611	0.2753	0.2919
	FreeNeRF	28.10	27.37	24.14	0.8291	0.8217	0.8274	0.1760	0.2022	0.2245
	KeypointNeRF	24.38	24.29	22.29	0.7969	0.7867	0.7864	0.2388	0.2434	0.2743
	Ours	**29.55**	**29.27**	**26.17**	**0.8466**	**0.8452**	**0.8417**	**0.1560**	**0.1483**	**0.1910**
7	Learnit	23.60	23.10	18.31	0.7984	0.7887	0.7659	0.2961	0.3000	0.3506
	EG3D-based prior	20.05	21.26	19.45	0.7991	0.7890	0.7890	0.2690	0.2815	0.3130
	RegNeRF	27.73	26.36	24.55	0.8229	0.8055	0.8225	0.2437	0.2589	0.2671
	FreeNeRF	28.09	25.03	20.03	0.8392	0.8027	0.7936	0.1704	0.2292	0.2458
	KeypointNeRF	23.84	23.97	22.11	0.7902	0.7811	0.7793	0.2430	0.2477	0.2829
	Ours	**29.54**	**30.42**	**27.76**	**0.8564**	**0.8639**	**0.8598**	**0.1510**	**0.1353**	**0.1755**

Table 6. Comparison with related works at 1K resolution on our studio dataset. We compare with Learnit [19], EG3D-based prior [4], RegNeRF [12], FreeNeRF [23], and KeypointNeRF [9] on different number of input views ranging from two to seven. Our method outperforms the related works by a clear margin. For a visual comparison, please refer to Figures 12,13, and 14.

Figure 11. Keypoints used for training KeypointNeRF [9] with our data.

the default settings provided by the authors for the DTU dataset [6], but adjust the near / far planes and scene scaling. Again, we disable the loss from the appearance regulariser.

For the EG3D-based prior [3], we train their model on Celeb-A [8] dataset at a 256 tri-plane and image resolution without the super-resolution module to ensure 3D consistent results. We note that their discretised volume representation leads to blurry results.

2.2. Few-shot Synthesis

Ultra High-res Our main setting is fitting to two or more views at a ultra-high resolution up to 4K. This goes far beyond the resolution of the prior model (512×768). Using at least two views provides the coverage from side angles such that the model can reconstruct intricate details like individual skin pores or a beard, which are not visible at lower resolutions. Please see the main paper and the supplementary HTML page for results.

Method	PSNR ↑				SSIM ↑				LPIPS ↓			
Subject	122	212	340	344	122	212	340	344	122	212	340	344
EG3D-based prior [3]	23.27	26.15	22.68	24.54	0.8678	0.9030	0.8862	0.8844	0.1504	0.1281	0.1228	0.1357
KeypointNeRF [9]	23.46	24.59	23.53	22.10	0.9171	0.9372	0.9187	0.9025	0.0940	0.0681	0.0743	0.0919
RegNeRF [12]	24.77	28.97	24.95	25.60	0.8903	0.9390	0.9129	0.8908	0.1334	0.0892	0.1001	0.1232
DINER [16]	25.79	29.78	26.27	**26.45**	0.9382	0.9597	0.9434	**0.9324**	0.0672	0.0672	0.0540	**0.0677**
Ours	**27.40**	**32.03**	**26.69**	25.51	**0.9359**	**0.9721**	**0.9489**	0.9135	**0.0671**	**0.0355**	**0.0533**	0.0761

Table 7. Comparison with the state-of-the-art for novel view synthesis from sparse views on Facescape [25]. This table supplements the main paper with individual metrics for each of the four test subjects. For a visual comparison, please refer to Fig. 4.

Single Image To showcase the robustness of our method, we show results for synthesising novel views from as little as a *single image* at the resolution of our prior model (512×768), see the main paper and Fig. 5.

In-the-wild Fig. 6 shows examples for in-the-wild captures with a mobile camera. The supplementary HTML page shows videos and adds high resolution results for in-the-wild captures with a smartphone camera.

2.3. Ablation

We perform extensive ablations on our prior model and on the finetuning algorithm. For the prior model, we ablate the impact of the number of training identities and the prior model resolution (Tbl. 8). For the finetuning algorithm, we ablate regularisation terms (Tbl. 1 and Fig. 7), number of views (Tbl. 4 and Fig. 12, 13, and 14), and initialisation techniques (Tbl. 3). We also ablate the effect of finetuning the full model including the latent codes vs. only finetuning the model parameters (Tbl. 2 and Fig. 8).

We provide all metrics cropped to the face region and evaluate on six holdout views to have comparable numbers across all ablations. All metrics are computed after finetuning for each of the three holdout subjects at resolution 1024×1024.

2.3.1 Prior Model

Table 8 ablates the effect of different variants of our prior model. We compare these variants of the prior model: lower resolution (256×384 instead of 512×768) and fewer training identities. The results show that a more diverse prior model performs better while a lower resolution prior model might not necessarily be required.

2.3.2 Model Finetuning

Initialisation This supplementary document complements the ablations in the main paper with metrics showing the benefits of the chosen regularisation (Tbl. 1 and Fig. 7) and visual examples for different initialisation techniques (Tbl. 1 and Fig. 9). For the *Nearest* (*Furthest*) Neighbour initialisation, we compute image embeddings using a pre-trained face recognition network [18]. We compute the similarity of the mean embedding of all target images with embeddings computed on a frontal rendering of all reconstructed training identities.

# Identities	Resolution	PSNR ↑	SSIM ↑	LPIPS ↓
15	512×768	24.25	0.7917	0.2187
350	512×768	24.62	0.7926	0.1985
750	512×768	25.43	0.7935	0.2035
1450	256×384	**25.99**	0.8034	**0.1810**
1450	512×768	25.69	**0.8040**	0.1905

Table 8. Ablation on the prior model. We train variants of our prior model at lower resolution and with fewer identities. The metrics are computed after finetuning to two views at resolution 1024×1024.

Number of Views We also provide a supplementary ablation on the performance when a different number of views are available in Tbl. 4. Figures 12, 13, and 14), and the supplementary HTML page shows visual results.

Frozen Latent Code Table 2 lists metrics and Fig. 8 shows the rendered images. We do not observe a strong difference in performance.

2.4. Limitations

Our model is trained on neutral faces with a closed mouth. It can handle mild expressions (e.g., closed eyes and a slightly open mouth) but fails for strong expressions and teeth, see Fig. 10.

While our results show robustness to in-the-wild settings, it is sensitive to correct camera calibration. In the reconstruction, this is particularly noticeable for thin structures like the eyes and eyelids. We also assume that the subject does not move during the capture.

Also, our prior model does not cover accessories like glasses or hats and reconstructions thereof are therefore not 3D consistent. Please see the supplementary HTML page for examples.

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022.

[2] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of generative networks. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2640–3498, 2018.

[3] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022.

[4] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *arXiv*, 2021.

[5] Guangcong, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. *Technical Report*, 2023.

[6] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014.

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[8] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[9] Marko Mihajlovic, Aayush Bansal, Michael Zollhoefer, Siyu Tang, and Shunsuke Saito. KeypointNeRF: Generalizing image-based volumetric avatars using relative spatial encoding of keypoints. In *European conference on computer vision*, 2022.

[10] 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 *European Conference on Computer Vision*, pages 405–421. Springer, 2020.

[11] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022.

[12] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[13] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020.

[14] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021.

[15] Rohit Pandey, Sergio Orts Escolano, Chloe Legendre, Christian Haene, Sofien Bouaziz, Christoph Rhemann, Paul Debevec, and Sean Fanello. Total relighting: learning to relight portraits for background replacement. *ACM Transactions on Graphics (TOG)*, 40(4):1–21, 2021.

[16] Malte Prinzler, Otmar Hilliges, and Justus Thies. Diner: Depth-aware image-based neural radiance fields, 2022.

[17] Eduard Ramon, Gil Triginer, Janna Escur, Albert Pumarola, Jaime Garcia, Xavier Giro-i Nieto, and Francesc Moreno-Noguer. H3d-net: Few-shot high-fidelity 3d head reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5620–5629, 2021.

[18] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[19] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P Srinivasan, Jonathan T Barron, and Ren Ng. Learned initializations for optimizing coordinate-based neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2846–2855, 2021.

[20] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2023.

[21] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022.

[22] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.

[23] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[24] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021.

[25] Hao Zhu, Haotian Yang, Longwei Guo, Yidi Zhang, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. Facescape: 3d facial dataset and benchmark for single-view 3d face reconstruction. *arXiv preprint arXiv:2111.01082*, 2021.

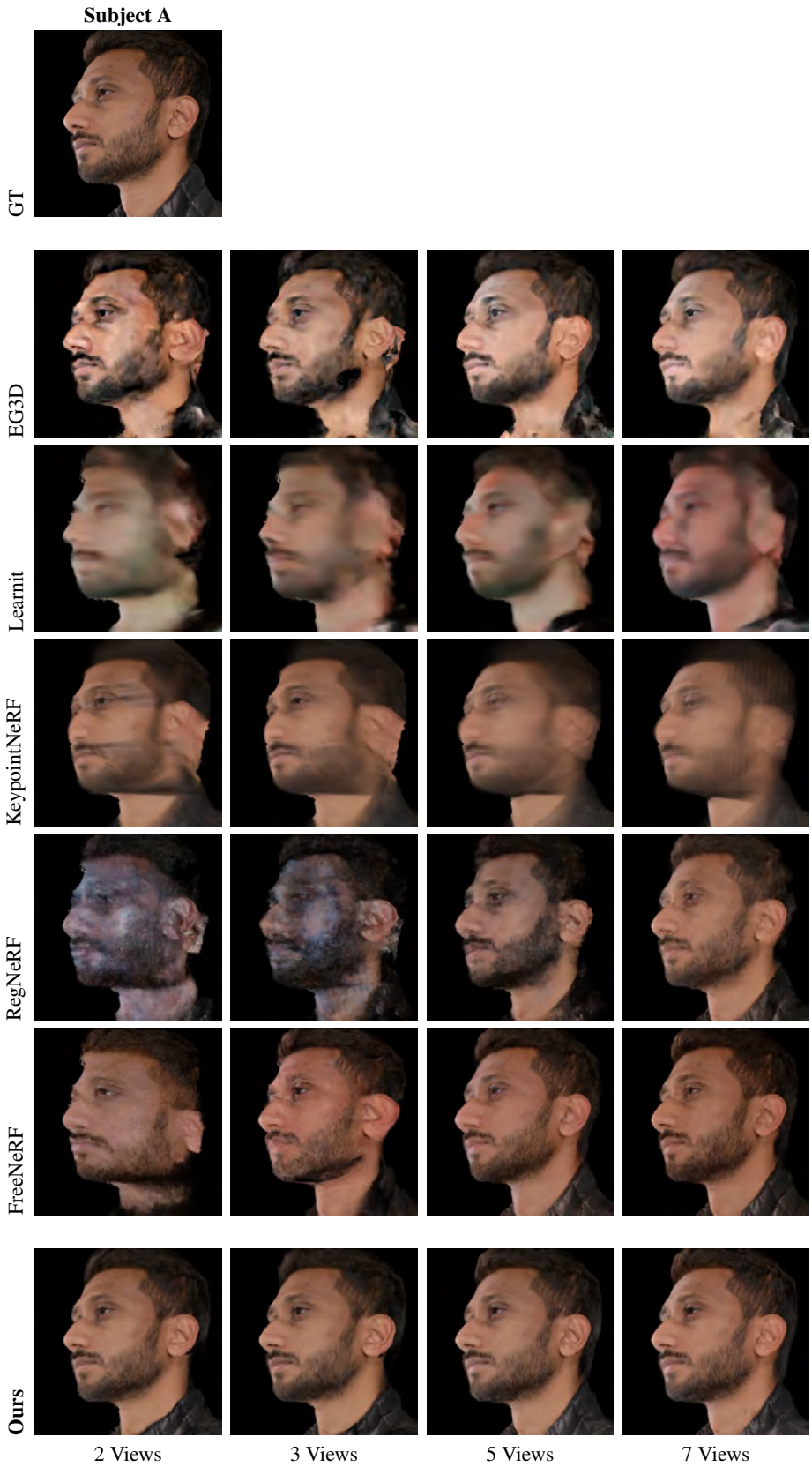

Figure 12. Comparison with related works at 1K resolution on our studio dataset. We compare with Learnit [19], EG3D [4], RegNeRF [12], FreeNeRF [23], and KeypointNeRF [9] on different number of input views ranging from two to seven. Please see Tbl. 6 for metrics.

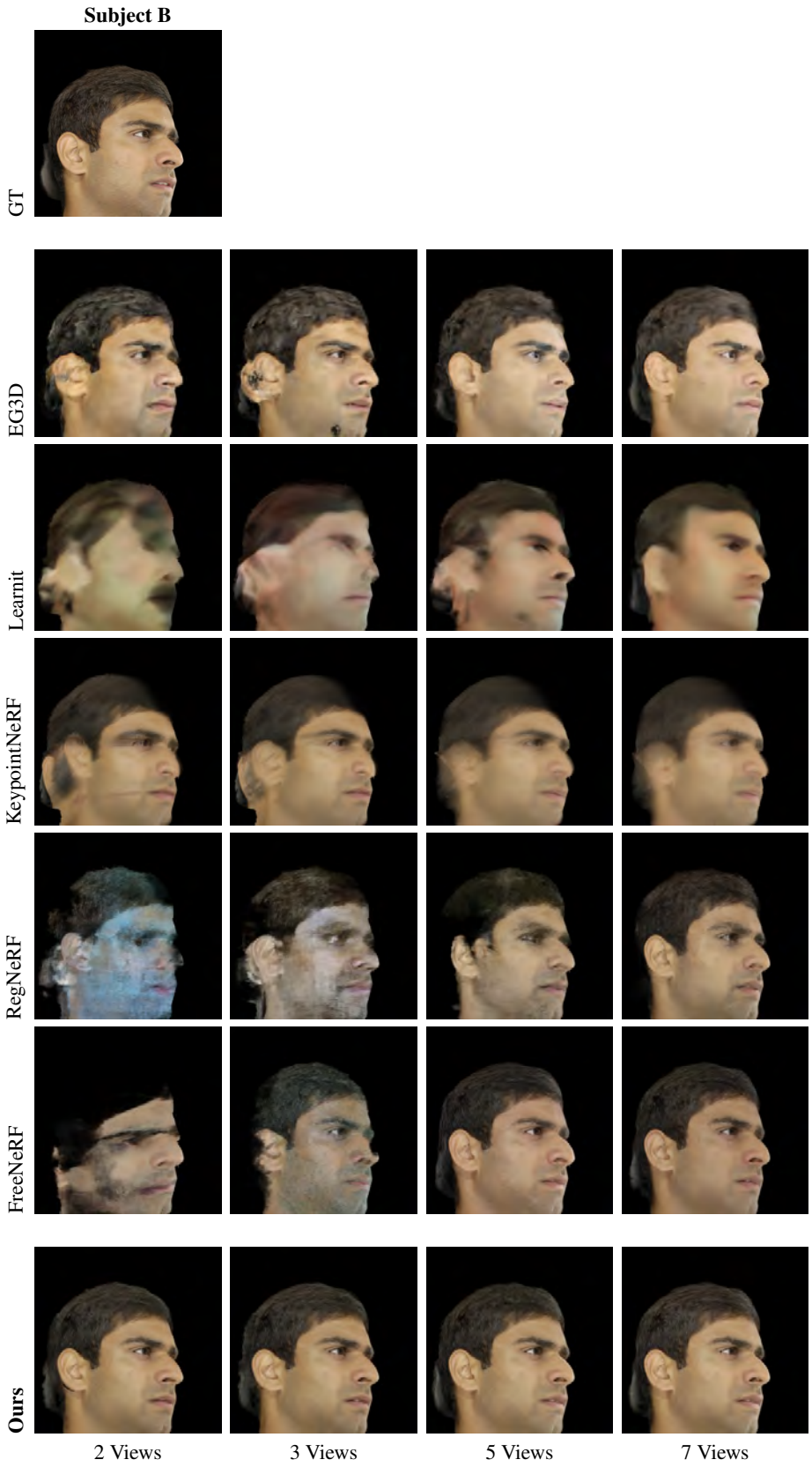

Figure 13. Comparison with related works at 1K resolution on our studio dataset. We compare with Learnit [19], EG3D [4], RegNeRF [12], FreeNeRF [23], and KeypointNeRF [9] on different number of input views ranging from two to seven. Please see Tbl. 6 for metrics.

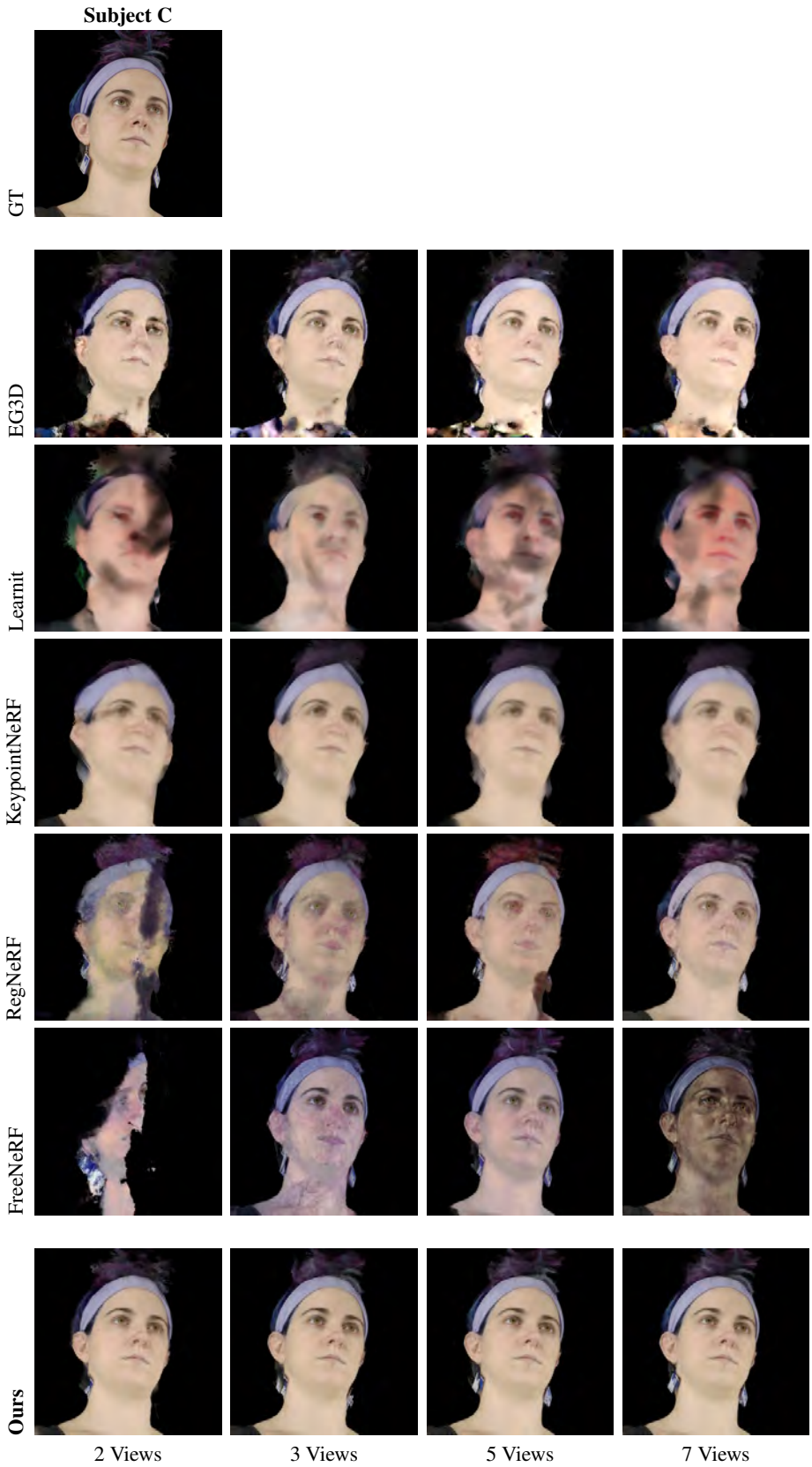

Figure 14. Comparison with related works at 1K resolution on our studio dataset. We compare with Learnit [19], EG3D [4], RegNeRF [12], FreeNeRF [23], and KeypointNeRF [9] on different number of input views ranging from two to seven. Please see Tbl. 6 for metrics.