Supplementary file for

TEGLO: High Fidelity Canonical Texture Mapping from Single-View Images

Vishal Vinod[1,2,*] Tanmay Shah[2,*] Dmitry Lagun[2]
[1]University of California, San Diego [2]Google Research
vvinod@ucsd.edu, {shaht, dlagun}@google.com

Abstract

This supplementary file contains additional qualitative results and more detail on experiments for TEGLO. We also detail the model architecture and training setup.

S1. Network architecture details

TEGLO Stage-1 denoted as $\mathcal{N}$ consists of a latent table, StyleGANv2 [10] generator layers, a 2-layer tri-plane decoder, a differentiable volume rendering module and a light weight camera predictor. For the experiments in this work, we use 512-dimensional latents in the latent table, and following [2], the output from the StyleGANv2 generator is of shape $256 \times 256 \times 96$ giving us $\mathrm{k} = 32$-channel tri-planes. Note that despite the use of a fixed size tri-plane, TEGLO enables arbitrary resolution synthesis as we employ a GLO-based auto-decoder training regime. As noted in the main paper, this also enables single-view 3D reconstruction at any resolution. Table.(S1), includes the details for the camera predictor network.

Table S1. Camera Predictor.

Layer	Kernel	Filters	Stride	Activation
Conv 2D	3	128	2	LeakyReLU
Conv 2D	3	64	2	LeakyReLU
Conv 2D	3	32	2	LeakyReLU
Conv 2D	3	16	2	LeakyReLU
Dense	-	25	-	Linear

TEGLO Stage-2 consists of a latent mapping network ($\mathcal{L}$), a dense correspondence network ($\mathcal{M}$) and a basis network ($\mathcal{C}$). We describe the network details in Fig.(S2). $\mathcal{M}$ is a LipMLP [13] with a Lipschitz regularizer at every layer to encourage Lipschitz continuity with respect to the inputs. Note that the same network implementation is used for all experiments - FFHQ [10], CelebA-HQ [8, 14], AFHQv2-Cats [6, 9] and SRN-Cars [3, 4].

*These authors contributed equally to this work.

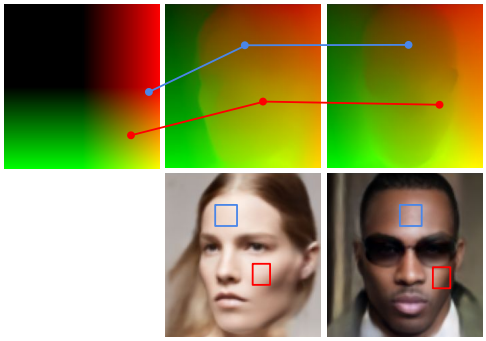

Figure S1. **Correspondence maps** - Establishing dense correspondence between 3D objects in the 2D canonical space.

S2. Training details

In TEGLO Stage-1, to train $\mathcal{N}$, we use the single-view image dataset and the approximate pose obtained from the shape-matching least-squares optimization described in Sec.(4) in the main paper following the procedure in [15]. We train $\mathcal{N}$ for 500K steps using the Adam optimizer [11] on 8 NVIDIA V100 GPUs (16 GB VRAM each) taking a total of 46 hours to complete training at 256^2 resolution. We also employ an exponential learning rate decay from $5e$-4 to $1e$-4. In each train step, we use latent w_i to condition the NeRF to reconstruct the object ($o_i \in \mathcal{I}$). Then, we use the loss $\mathcal{L}_{\mathcal{N}} = \mathcal{L}_{\text{RGB}} + \mathcal{L}_{\text{Perceptual}} + \mathcal{L}_{\text{Camera}}$ to train the network.

To train TEGLO Stage-2, we first render a dataset $\mathcal{D}$ with five camera views: $e_i = \{v_f, v_l, v_r, v_t, v_b\}$ for that object. In this work, we render 1000 identities for $\mathcal{D}$ giving us 5000 total views. We train TEGLO Stage-2 for 1 Million steps (1000 epochs) using the Adam [11] optimizer on 8 NVIDIA V100 GPUs (16 GB VRAM each) taking a total of 50 hours to complete training. In each train step, we use the latent w_i as input to $\mathcal{L}$, 3D surface points (p_i) and shape-code from $\mathcal{L}$ as input to $\mathcal{M}$, and the mapped canonical coordinate points from $\mathcal{M}$ as input to $\mathcal{C}$. We compute the $\mathcal{L}_{\text{Stage2}}$ loss using the output RGB (r_i), surface normals (s_i), and surface points (p_i) with the respective ground-truth $\{\{\hat{r}_i, \hat{s}_i, \hat{p}_i\} \in e_i\} \in \mathcal{D}$. The learned

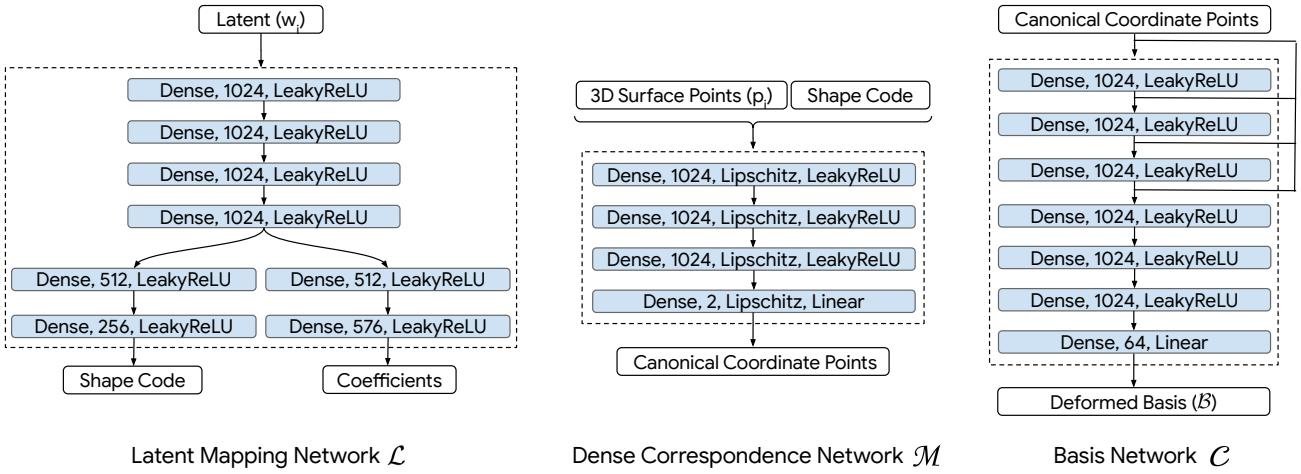

Figure S2. **TEGLO Stage-2 Network Architecture Details.**

dense correspondences are then used for TEGLO inference to extract the object texture. We show qualitative results for the correspondence maps in Fig.(S1) where canonical coordinate points from different posed images are mapped to the same location in the canonical coordinate map (canonical coordinate points are output from $\mathcal{M}$). Further, the quantitative results in Table.(1) and Table.(2) show a significant improvement in PSNR and LPIPS for reconstruction - demonstrating the effectiveness of the dense correspondences learned by TEGLO Stage-2 for texture representation and tasks such as texture transfer and texture editing.

S3. Testing details

Test data inference and single view 3D reconstruction: To train TEGLO Stage-1 for faces, we use the CelebA-HQ dataset [8, 14] with 26K training images and optimize the latent table. We use the remaining 4K images as the test set. For single view 3D reconstruction and inference, we randomly sample 1K images from the training set and render five views to train TEGLO Stage-2. To evaluate on the test data, we invert the image by optimizing its latent for 200 steps while keeping the network parameters frozen. Then, we render five camera views and back-project to obtain the surface points. We then map the surface points to the canonical coordinate space to register the predicted pixels for those surface points. Similarly, we also map the 3D surface points from the GT camera pose to the canonical space and register GT pixels for those surface points.

Our quantitative results demonstrate the effectiveness of the dense correspondences learned by TEGLO Stage-2 in mapping the 3D surface points to the canonical coordinate space. Since we map the pixels from the original image onto the texture space, we accurately preserve high frequency details with multi-view consistency. Hence, for quantitative metrics such as PSNR, which is computed on the normalized pixel values, we observe significantly higher values for TEGLO compared to previous methods. We draw the reader's attention to Table.(5) in the main paper where we report the 3D consistency metric. As described above, the effectiveness of our formulation enables TEGLO to significantly outperform the baseline methods. We include qualitative results for test data inference and single view 3D reconstruction in Fig.(S10, S11, S13, S15).

S4. Ablations

Using $\mathcal{L}_{\text{Camera}}$ loss. EG3D [2] conditions the generator and discriminator with the camera pose to enable 3D consistent novel view synthesis. As noted in the main paper, the pose-conditioned generator does not completely disentangle the camera pose from appearance leading to artifacts such as facial expressions/eyes following the camera. In TEGLO Stage-1, we use the $\mathcal{L}_{\text{Camera}}, \mathcal{L}_{\text{RGB}}$ and $\mathcal{L}_{\text{Perceptual}}$ losses to train $\mathcal{N}$. An ablation experiment without the camera prediction loss led to 2D banner artifacts. This is qualitatively represented in Fig.(S3) for "Views without $\mathcal{L}_{\text{Camera}}$" with flat and inconsistent geometries for different camera angles. However, the results for training $\mathcal{N}$ using $\mathcal{L}_{\text{Camera}}$ show multi-view consistent representations demonstrating the effectiveness of using the simple camera prediction loss. Furthermore, we show that the rendered orbits do not have expressions/eyes following the camera in '.gif' files in the supplementary folder. Qualitative results for novel view synthesis are also presented in Fig.(S3, S4, S5, S10, S9, S12, S13, S14, S15).

TEGLO Stage-2 with $\mathcal{L}_{\text{Coord}}$ loss only. Previous work AUV-Net [5] states that methods [7, 12] that do not use color for learning dense correspondences may learn sub-par texture representations. To verify, we train TEGLO Stage-2 with only $\mathcal{L}_{\text{Coord}}$ reconstruction loss instead of $\mathcal{L}_{\text{Stage2}} = \mathcal{L}_{\text{Coord}} + \mathcal{L}_{\text{Coord}} + \mathcal{L}_{\text{Coord}}$ reconstruction losses. The qualita-

Figure S3. **Ablation** - $\mathcal{L}_{\text{Camera}}$ for TEGLO Stage-1 training.

tive results are presented in Fig.(S14) comparing TEGLO-3DP with TEGLO and other baseline results. Of particular interest is Fig.(S4) with qualitative results for TEGLO and TEGLO-3DP including the texture image. We note that the reconstruction and novel view synthesis results are nearly identical. However, we also observe TEGLO-3DP including a wayward texture representation near the hair region. While the dense correspondences map the surface points to the appropriate RGB image pixels, there is a scope for null pixel artifacts around the hair region when using NNI. While the 3D reconstruction and novel view synthesis for TEGLO-3DP and TEGLO do not differ, we note the potential for black pixels to be obtained in novel view synthesis leading to lowered qualitative and quantitative results.

Limitations. As mentioned in the main paper, training TEGLO involves a computational overhead. Further, TEGLO is only able to map target image pixels spanning the target image and hence there may be artifacts for camera views with minimal mapped target image pixels. For example, the texture (T_{GT}) in row-4 in Fig.(S15) includes missing pixels near the lower jaw region. In the last column in row-4, the novel rendered view does not include any target image pixels for the ear. On a similar note, in row-2, column-7 in Fig.(S12), the novel view shows a slight twist in the nose geometry partially due to the thin veil on the face which could not be accounted for in Stage-1.

K-d tree and NNI for textures. In the main paper, we discuss the necessity of the K-d tree and Natural Neighbor Interpolation (NNI) [16] to prevent aliasing artifacts and enabling unambiguous texture image indexing to obtain the RGB values for each surface point. As noted in the main paper, our formulation uses the K-d tree and NNI to interpolate and index into the texture space with sparse "holes". In Fig.(S6), we depict the process to interpolate sparse holes. Each cell in the 5x6 grid represents a discrete pixel in the texture space and the red dot represents a canonical coordinate point. We include the three issues for better context:

1. The canonical coordinate points may not be aligned to the pixel centers and storing them in the discretized texture space may lead to imprecision.
2. There may be multiple canonical coordinates mapped to a discrete integral pixel wherein some coordinates may need to be dropped for an unambiguous texture indexing - leading to loss of information.
3. Some pixels may not be mapped to by any canonical coordinates, creating a "hole" in discretized space. This is represented by "**X**" in the grid in Fig.(S6).

The K-d tree allows extracting multiple neighbors by querying with canonical coordinate points and also enables indexing the texture using floating point values. Hence, using a K-d tree to store the texture helps address (1) and (2). Further, using a K-d tree in conjunction with Natural Neighbor Interpolation (NNI) effectively addresses (3). Natural Neighbor Interpolation (NNI) is formulated as follows:

$$\text{NNI}(x) = \sum_{i=0}^{n} w_i(x) \times f(x_i) \tag{1}$$

$$w_i(x) = \frac{\frac{1}{d_i(x)}}{\sum_{j=0}^{n} \frac{1}{d_j(x)}} \tag{2}$$

Where x is the query point, $w_i(x)$ is the simplified Laplace weight based on inverse distances to n neighbors corresponding to the polygon potentially encroached by the query point in the Voronoi tessellation plot, and $f(x_i)$ represents the extracted texture pixels. Storing the texture in the K-d tree and using Natural Neighbor Interpolation enables accurate and unambiguous (property of NNI using tessellation plot) floating point indexing into the texture to obtain the RGB color. We also note in the main paper about K-d tree + NNI enabling robustness to "holes" in the texture. To verify the utility of K-d tree + NNI we design an experiment where the texture is stored as an image and indexed using bi-linear interpolation. The qualitative results are presented in Fig.(S5) where t_{IMG} is the texture with mapped ground truth pixels stored as an image, and t_{GT} is the TEGLO texture stored as a K-d tree. We observe that the novel view synthesis using t_{IMG} which includes clipping the query indices into the texture image size range and bi-linear interpolation for indexing leading to several aliasing artifacts. This is because the canonical coordinate points may not be aligned to the pixel centers and storing them in the discretized texture space may lead to imprecision which manifests as aliasing in novel view rendering. We further note that the quantitative results for test set reconstruction using t_{IMG} is 30.828 dB PSNR and 0.0823 for LPIPS at 256^2 resolution for CelebA-HQ data. In contrast, novel view synthesis using t_{GT} leads to a reduction in aliasing artifacts with quantitative results of 86.2 dB PSNR and 7.4e-7 LPIPS for test set reconstruction of CelebA-HQ images at 256^2 resolution (refer Fig.(S5) and the accompanying '.gif' files).

S5. Qualitative results

We present qualitative results for texture transfer in Fig.(S7, S8). We present qualitative results for texture edits

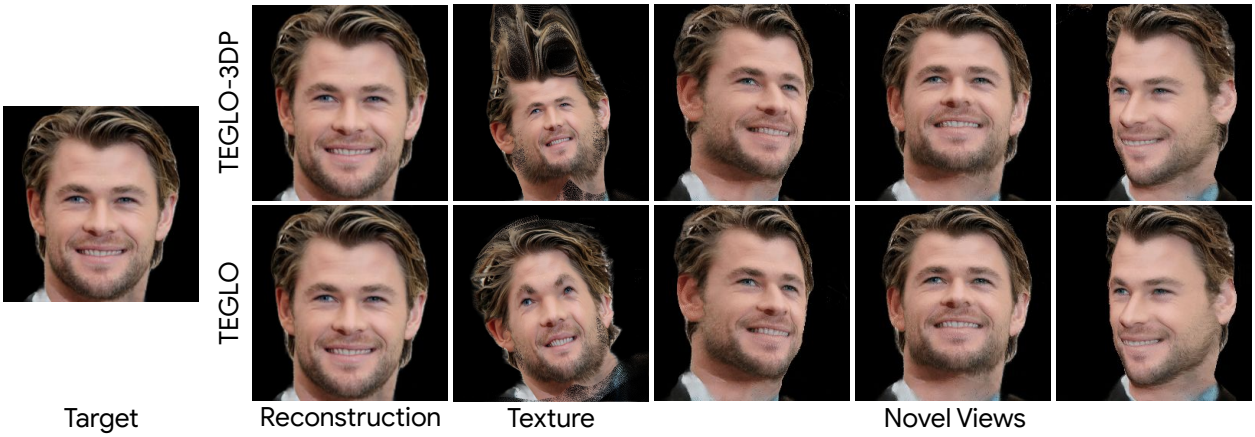

Figure S4. **Ablation** - Qualitative comparison with TEGLO-3DP (TEGLO Stage-2 with 3D surface point reconstruction only)

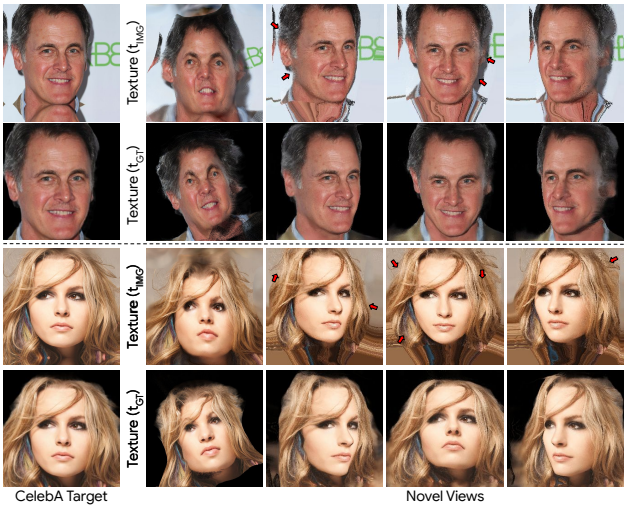

Figure S5. **Texture ablation** - Qualitative comparison with texture stored as an image indexed with bi-linear interpolation.

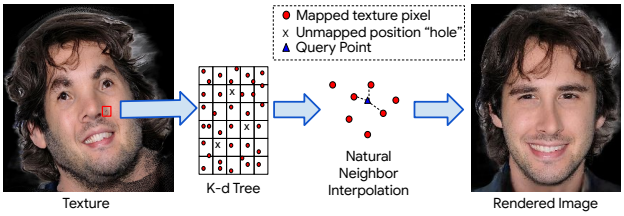

Figure S6. **Interpolating textures with sparse "holes"** - Depicting the KD-Tree and Natural Neighbor Interpolation (NNI) to interpolate "holes" (if any) in the texture for novel view synthesis.

and textured synthesis for AFHQv2-Cats in Fig.(S10), for SRN-Cars in Fig.(S11) and for CelebA-HQ in Fig.(S15). We also present single-view textured 3D reconstruction results for TEGLO trained on FFHQ data and evaluated on CelebA-HQ image targets - focus on complex details such eyeglasses and make-up - in Fig.(S13). Further, we also present comparative qualitative results with baselines and TEGLO-3DP (TEGLO Stage-2 trained with only $\mathcal{L}_{\text{Coord}}$ loss) in Fig.(S14). Lastly, we present qualitative results for high-resolution rendering at 1024^2 resolution in Fig.(S16) and show novel view synthesis with a specific focus on high-frequency details such as freckles (pigments under the skin) in row-1, make-up and jewelry in row-2, hair and beard in row-3, fine skin details in row-4 and wrinkles in row-5.

S6. Ethical considerations

One of the motivating goals for TEGLO stems from the need for photorealistic 3D reconstruction of objects from single-view image collections. As an example, [5] use Tripleganger heads [1] - a dataset containing 515 3D meshes faces at a high cost-per-scan requiring a custom commercial license for use. Similarly, [17] is a dataset of 938 textured meshes of heads made available at no cost. However, the authors allude to the demographic bias in compiling the data, the 68 DSLR camera setup, and the 6 month effort involved in dataset capture - all of which do not scale and has a high potential for bias and privacy issues. TEGLO enables high-fidelity 3D reconstruction and novel view synthesis from single-view image collections which alleviates these issues and also improves access to high quality data to the broader research community. Hence, TEGLO enables rendering a dataset of diverse objects (improving fairness and mitigating bias) and also reduces the need for large scale data collection (alleviating privacy issues). Further, we acknowledge the potential misuse of TEGLO and hence only make available the code for reproducibility purposes. In alignment with the motivating goals for TEGLO, we will not make available any trained weights for our method.

S7. Efficient dataset rendering

For 3D high-fidelity data rendering from single-view image collections of objects, TEGLO enables arbitrary resolution synthesis. Since the dense correspondences are learned point-wise (*i.e.* using p_j), there is no spatial constraint in

querying $\mathcal{M}$ for the canonical coordinate point. Hence, we render images of any size by first dividing the image pixels into 4 tiles, then obtain the surface points from TEGLO Stage-1, map the surface points to the canonical coordinate space using $\mathcal{M}$ and then index into the texture to obtain the RGB color value. Then, after all the tiles are computed, we can combine the divided computations into a single image of high-resolution. The orbit gif files with 60-120 frames each and the high-resolution frames at 1024^2 resolution in Fig.(S16) are rendered using this approach.

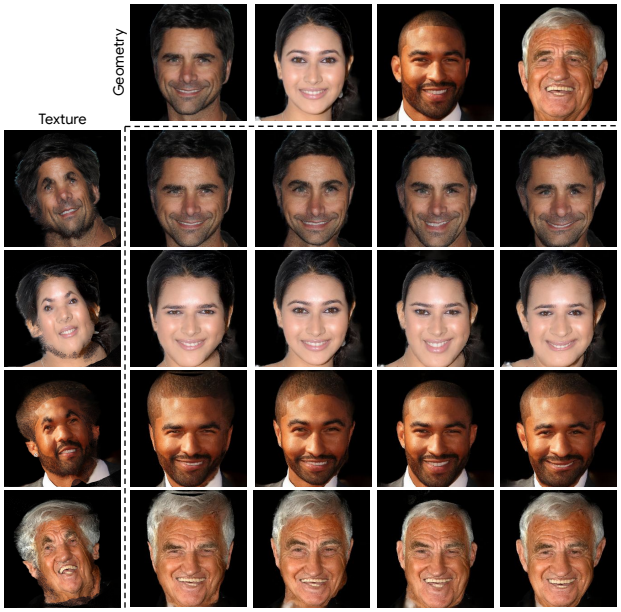

Figure S7. **Texture transfer** - Qualitative results for texture transfer with CelebA-HQ. (Top row shows CelebA-HQ image targets).

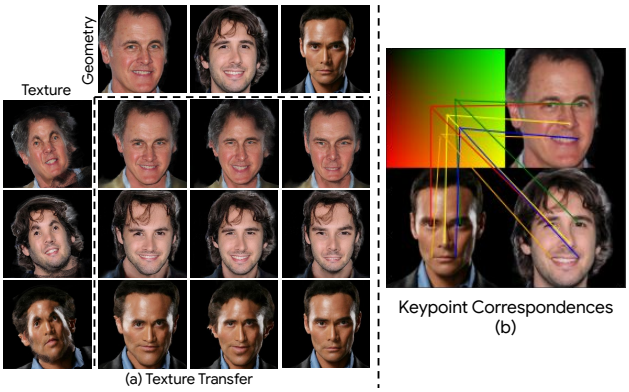

Figure S8. **Texture transfer** - (a) Qualitative results for texture transfer with CelebA-HQ. (Top row shows CelebA-HQ image targets). (b) Keypoint correspondences in the canonical space.

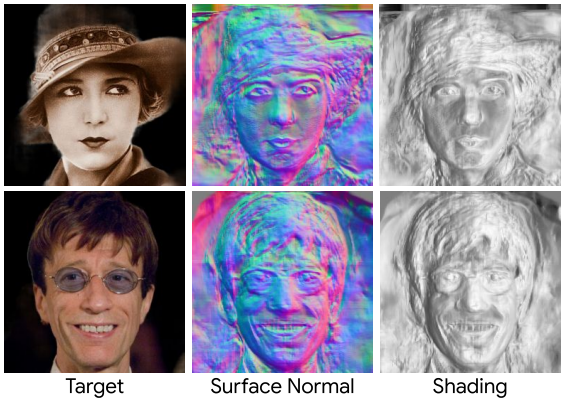

Figure S9. **Geometry results** - Target view reconstruction geometry for CelebA-HQ with hat and eyeglasses.

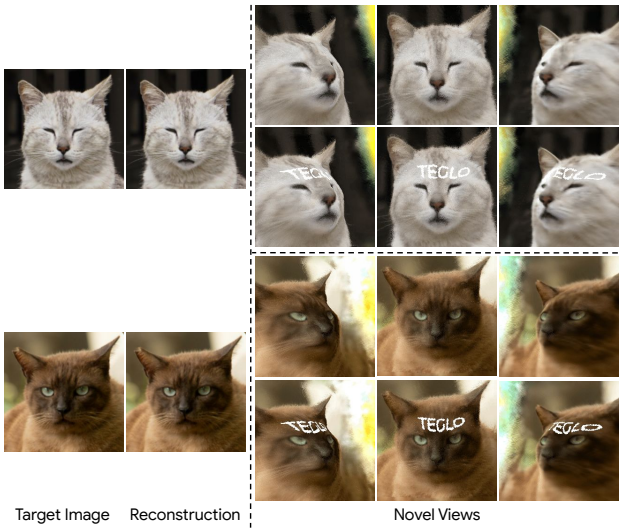

Figure S10. **Textured synthesis** - Target view reconstruction and novel view synthesis for AFHQv2-Cats.

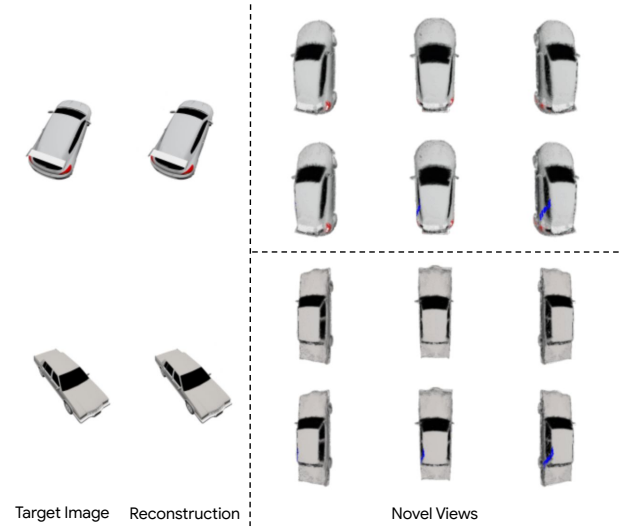

Figure S11. **Textured synthesis** - Target view reconstruction and novel view synthesis for SRN-Cars.

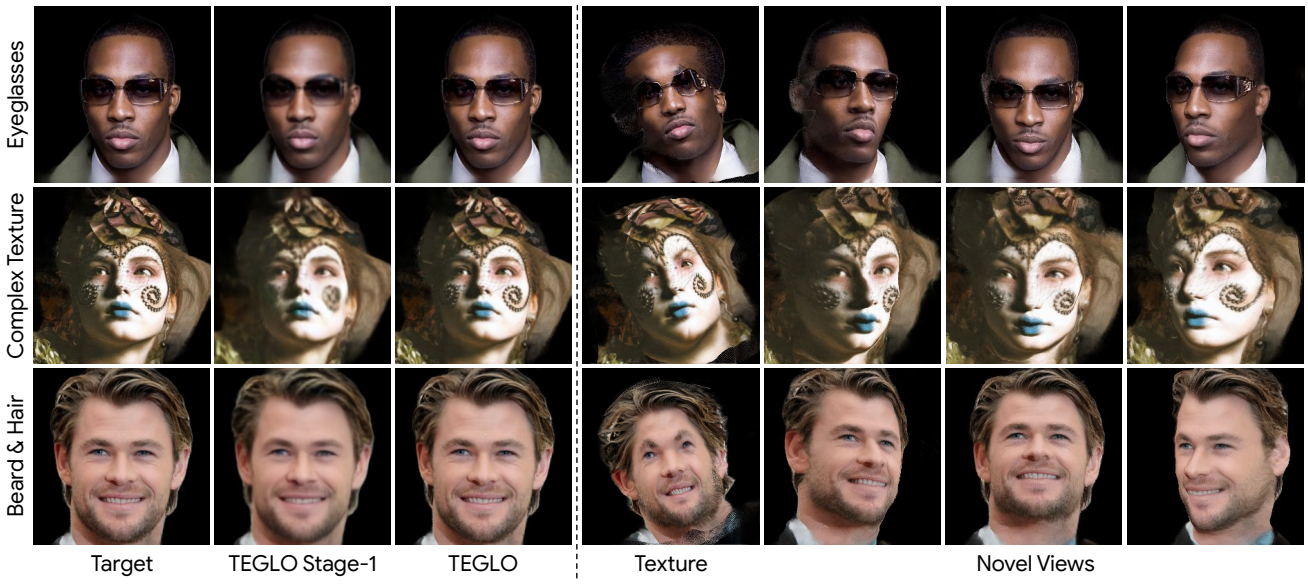

Figure S12. **Complex geometry and texture** - Qualitative results for textured novel view synthesis.

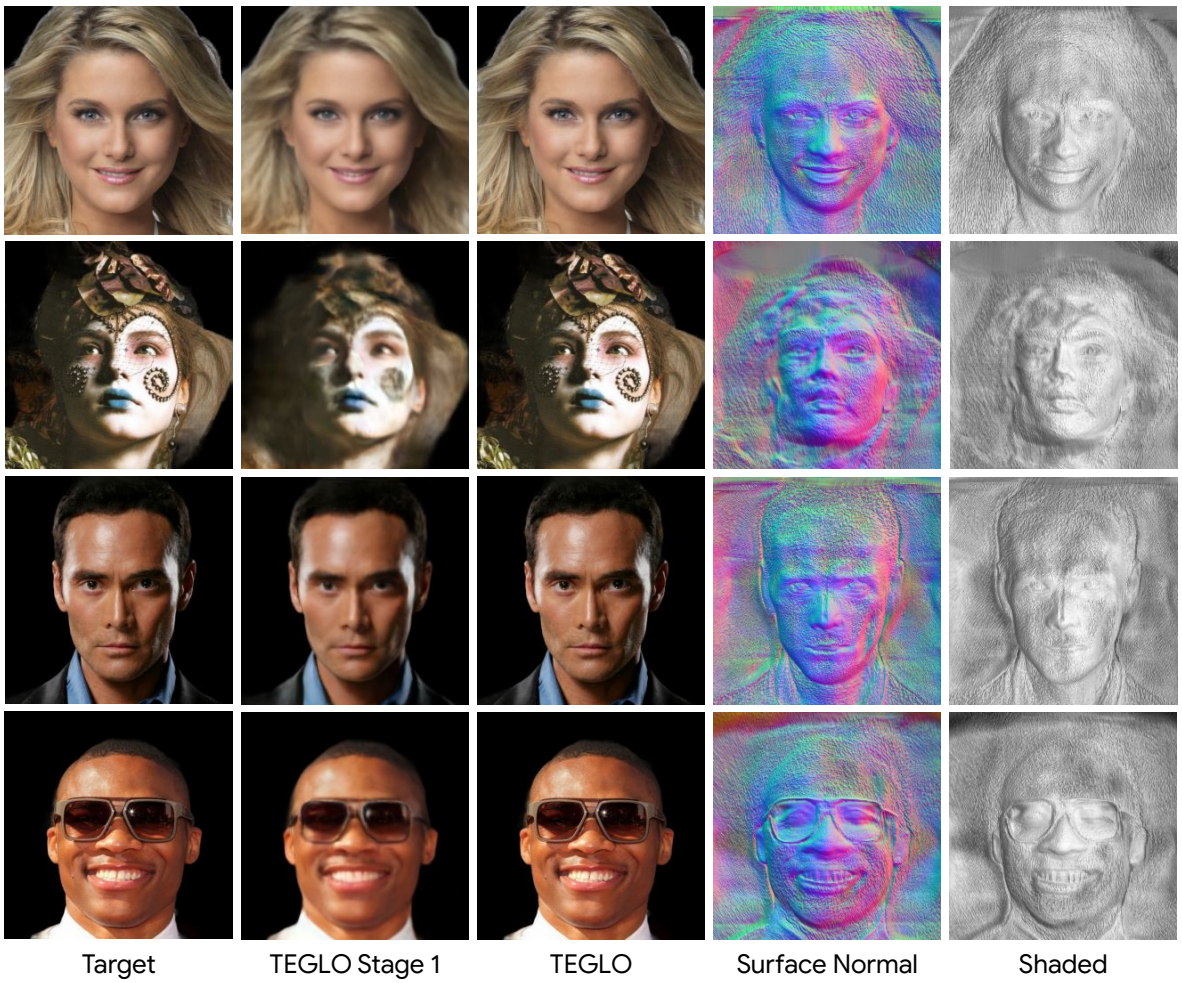

Figure S13. **Single view 3D reconstruction** - Results for TEGLO trained on FFHQ data and evaluated on CelebA-HQ image targets.

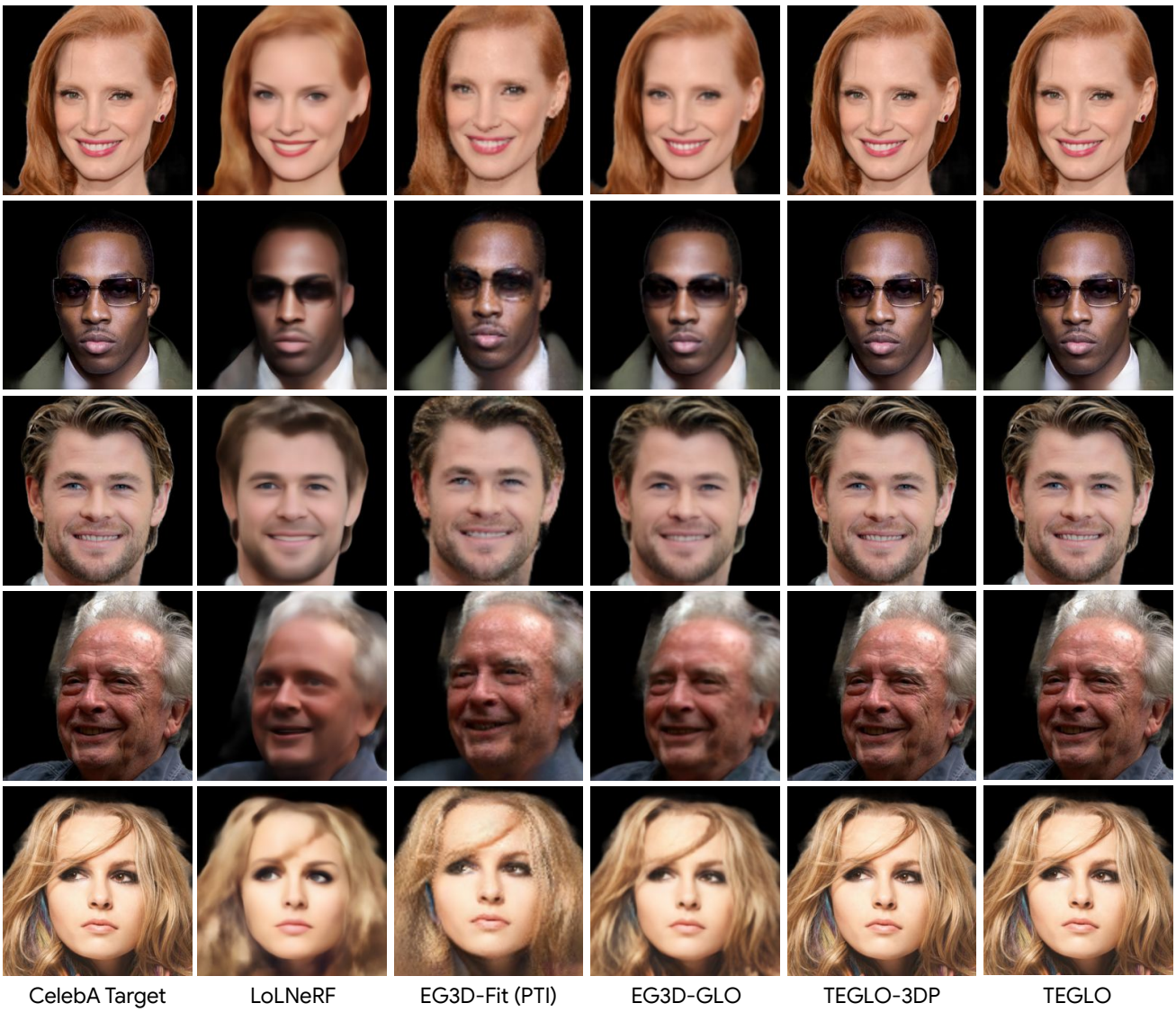

Figure S14. **Qualitative results** - Comparison with relevant 3D-aware generative baseline methods at 256^2 resolution for CelebA-HQ.

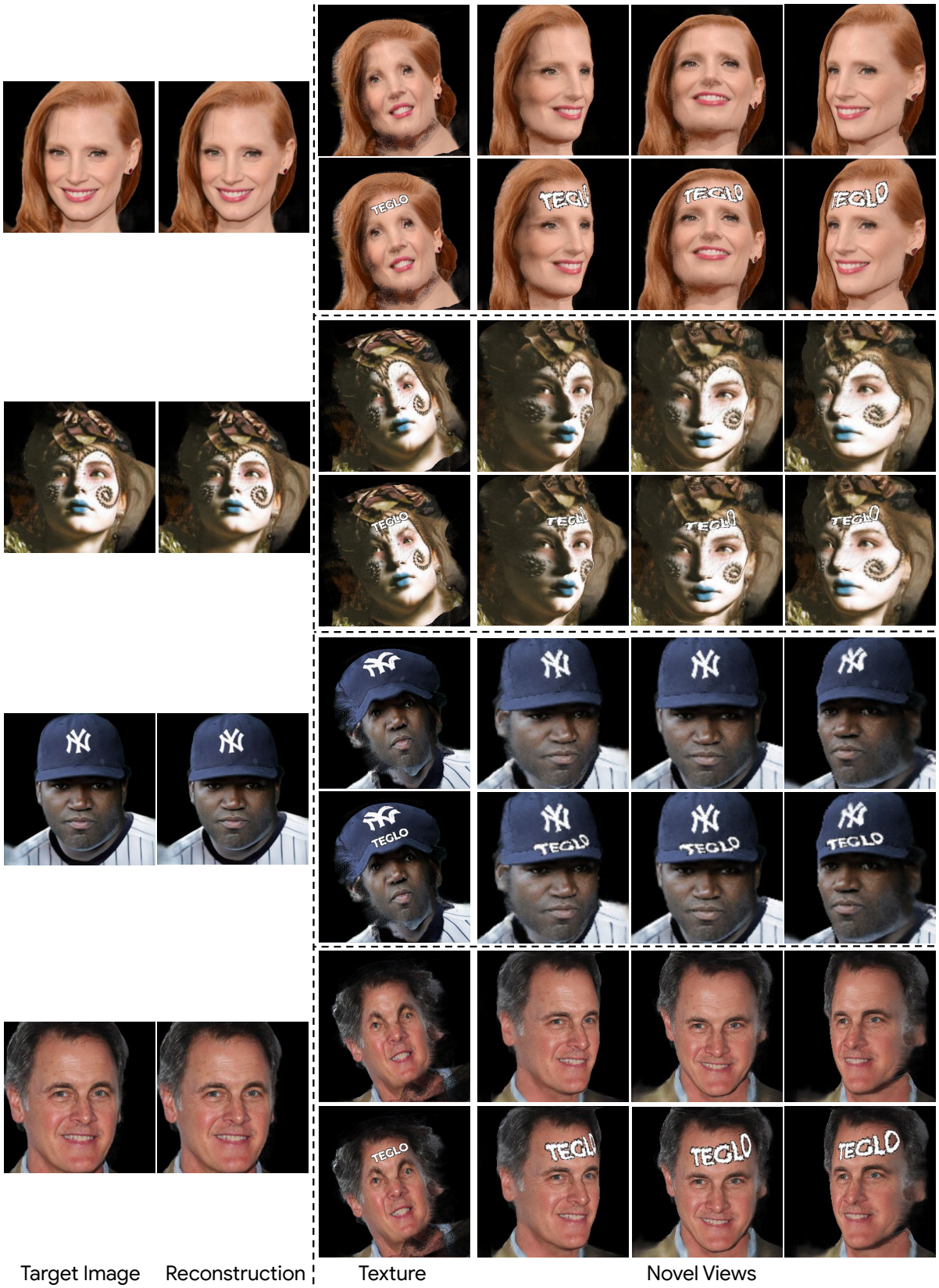

Figure S15. **Textured synthesis** - Target view reconstruction and novel view synthesis for CelebA-HQ.

Figure S16. **High resolution rendered views** - Qualitative results for novel views in high-resolution with high-frequency details such as freckles, jewelry, make-up, hair, fine details and wrinkles.

References

[1] Triplegangers heads. In *https://triplegangers.com/*, 2022. 4

[2] 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. 1, 2

[3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1

[4] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 1

[5] Zhiqin Chen, Kangxue Yin, and Sanja Fidler. Auv-net: Learning aligned uv maps for texture transfer and synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1465–1474, 2022. 2, 4

[6] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020. 1

[7] Yu Deng, Jiaolong Yang, and Xin Tong. Deformed implicit field: Modeling 3d shapes with learned dense correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10286–10296, 2021. 2

[8] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 1, 2

[9] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021. 1

[10] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 1

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[12] Feng Liu and Xiaoming Liu. Learning implicit functions for topology-varying dense 3d shape correspondence. *Advances in Neural Information Processing Systems*, 33:4823–4834, 2020. 2

[13] Hsueh-Ti Derek Liu, Francis Williams, Alec Jacobson, Sanja Fidler, and Or Litany. Learning smooth neural functions via lipschitz regularization. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–13, 2022. 1

[14] 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. 1, 2

[15] Daniel Rebain, Mark Matthews, Kwang Moo Yi, Dmitry Lagun, and Andrea Tagliasacchi. Lolnerf: Learn from one look. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1558–1567, 2022. 1

[16] Robin Sibson. A brief description of natural neighbour interpolation. *Interpreting multivariate data*, pages 21–36, 1981. 3

[17] Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. Facescape: a large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 601–610, 2020. 4