3DAvatarGAN: Bridging Domains for Personalized Editable Avatars

Rameen Abdal[†1] Hsin-Ying Lee[2] Peihao Zhu[†1] Menglei Chai[2] Aliaksandr Siarohin[2]

Peter Wonka[1] Sergey Tulyakov[2]

[1]KAUST [2]Snap Inc.

1. Limitations

Our method also has some limitations. Overall, the visual quality is limited by the quality of StyleGAN2 pretraining. While we found the quality to be very high for the datasets shown in the paper, it relies on hundreds of images in the target domain to be available. Would be interesting to do few- shot domain adaptation in the future. Further, edits are largely limited to semantic edits of EG3D and global space deformations by *TPS*. Our method does not enable fine-grained geometric edits. Finally, a large part of our method is face specific. We justify this specialization by the importance of human models and the specific target domain of editable 3D avatars. We nevertheless believe that domain adaptation of general 3D-GANs will be an interesting avenue of future work.

2. Ethical Concerns

Deep learning-based image and video processing is a tool for image/video understanding, animation, and artistic expression. Similar to most software in this domain, our work could be used to produce offensive results. An application of concern would be if a user would generate offensive caricatures and cartoons of other people without consent. This can be used to insinuate biases against people or can have a detrimental effect on a person's autonomy, dignity, and/or privacy. The images used in this work are taken or derived from the FFHQ [3] dataset which has appropriate licenses for non-commercial research use and to the best of our knowledge, is committed to protecting the privacy of the individuals who do not wish to be included.

3. Training Details

We train our models on 4 V100 GPUs with a batch size of 8. Similar to EG3D, we start training from the neural rendering resolution of 64^2 which is increased during the training. Then we do fine-tuning on 128^2 resolution to produce the final 512^2 outputs. We sample $100k$ samples from each dataset. We train Caricatures on $\sim$ 880 $kimgs$, Pixar, and Cartoons on $\sim$ 500 $kimgs$. We fine-tune these models on 128^2 neural rendering resolution for an additional $80 - 160$ $kimgs$. Other hyperparameters of learning rates are the same as the EG3D. We train the *TPS* module on $\sim$2000 $kimgs$. We set the weight for the regularization term $\mathrm{R}(\Delta s)$ as 0.001, and $\mathrm{R}(D)$ as 0.005. For the *TPS* training we use the weights for α, β, σ and $\mathrm{R}(\mathrm{T}_2)$ as 150, 1, 3 and 1 respectively. For inversion, we perform 200 steps for the source domain inversion and 400 steps for the target domain to generate the final avatar.

4. Illustrations and Pipeline figures

We show the camera alignment illustration in Fig. 2. We also show the illustration of the inference pipeline of the *TPS* module in Fig. 3 and the final inference model with the *TPS* module added in Fig. 4.

5. Ablation Study

In order to validate the importance of the losses, and components of our design choices, in Table 1, we show an ablation study of these components with regularizers. Note that we evaluate these design choices on the caricature dataset. Notice adding each component improves the corresponding scores in FID, M_d, S_d, and ID as discussed in the main paper. Notice that by adding the TPS module, the FID is still comparable to $\mathrm{G_t}$. The slight drop is attributed to the stretching and squeezing of some parts of the texture (See Fig. 6 in the main paper). Nevertheless, by adding this module, we achieve better control over geometry and produce exaggerated features for a small drop in texture quality. In order to show that the TPS transformations are not random, we compute the FID scores with randomly perturbed front plane features which are derived from the perturbations of control points near the face *e.g.* taken at the early stages of training after it has stabilized a bit. This setup has less perturbation in the background. We

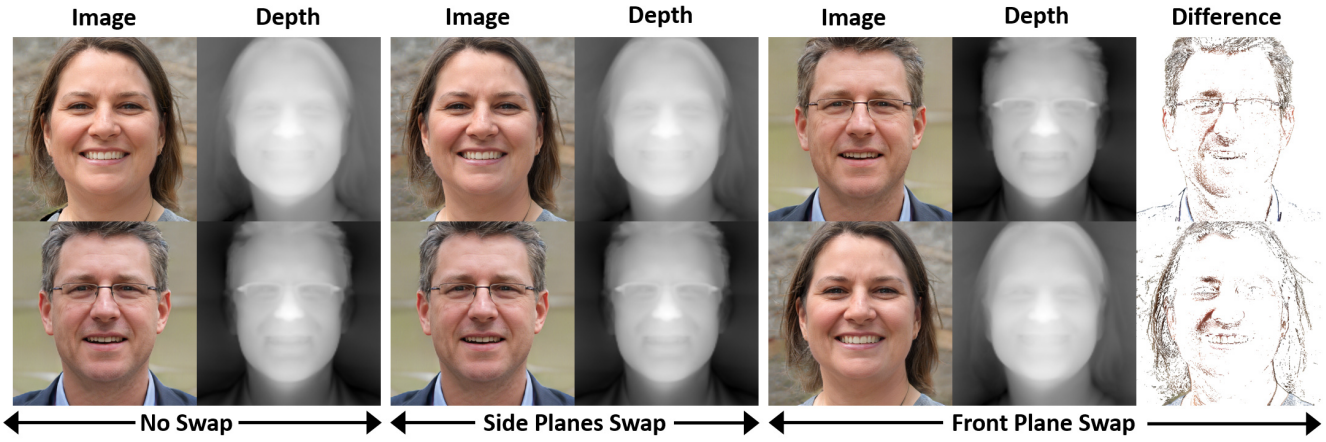

Figure 1. **Swaping tri-planes.** Validation of the information stored in the front tri-plane. Given two images and their tri-plane representations, there is almost no change in the final output if the side planes are swapped. While the output changes completely when the front tri-plane is swapped.

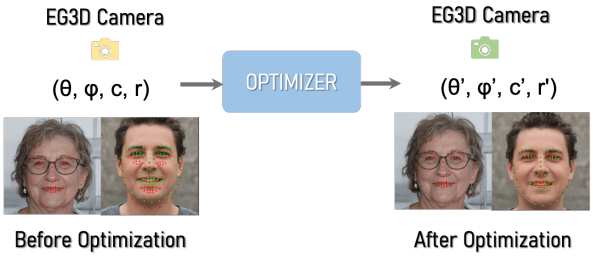

Figure 2. **Illustration of camera alignment (Sec 3.1) main paper.**

Table 1. **Ablation.** Ablation of the design choices made in Sec. 3 of the main paper. cam stands for the analysis in Sec. 3.1, Reg stands for the model after applying Eq. 2, DReg stands for the model after applying Eq. 4, and TPS stands for the model after applying Eq. 5, 6, and 7 in the main paper. This is the G_t used in the comparison. Note that by adding TPS the scores are affected as the geometry is exaggerated *e.g.* the identity is affected. This module can be added to do geometry editing. See the explanation in Ablation Study.

Method	FID	M_d	S_d	ID
G_{base} - cam	90.8	0.47	0.33	1.348
G_{base}	67.8	0.47	0.22	1.272
+ Reg	19.0	0.22	0.22	0.889
+ DReg	19.4	0.21	0.15	0.879
+ TPS	20.6	0.25	0.20	0.924

found that the FID score is worse *i.e.* 25.5 hence validating non-random transformations.

Table 2. **Ablation of TPS on metrics based on facial keypoints.**

Metric	with TPS	without TPS
Avg. Keypoint Distance	**5.7**	4.3
Avg. Keypoint Variation	**0.06**	0.04

Table 3. **FID comparison with SCG: StyleCariGAN, DSG: DualStyleGAN.**

Method	Cari. (SCG)	Cari. (DSG)	Pixar (DSG)	Cartoons (DSG)
2D-GANs	**51.68**	96.49	166.78	**105.57**
Ours	62.90	**89.73**	**162.06**	111.35

6. More ablation on *TPS* and what metric does it improve?

The purpose of adding the *TPS* module is to model the exaggerated geometries in the data and at the same time achieve geometric editing and animation capabilities (Caricatures in the main paper, and Project Page). We ablate (see Table 2) the average face Keypoint Distance between the paired FFHQ-generated images and corresponding avatars and the face Keypoint Variation (standard deviation of keypoints) with and without the *TPS* module. The results show that with *TPS* module, the deviations are large and match the exaggerations.

7. Results on Real Images, Algorithmic Description of Projection, and User Study

Let x be the source image, let pixel-wise MSE loss be represented as $L_{mse}(x, w, G) = MSE(x, \mathrm{G}(w, \mathrm{M}(\theta', \phi', c', r')))$, and let $LPIPS$ loss be represented as $L_{lpips}(x, w, G) = LPIPS(x, \mathrm{G}(w, \mathrm{M}(\theta', \phi', c', r')))$ where camera parameters are determined by Sec. 3.1 in the main paper. Let $L_d(w)$ be the depth regularizer (Sec. 3.2) and $A_t(x, w, G)$ be the attribute classifier loss. We define the algorithm of projection of a single source image to 3D avatars in Algo-

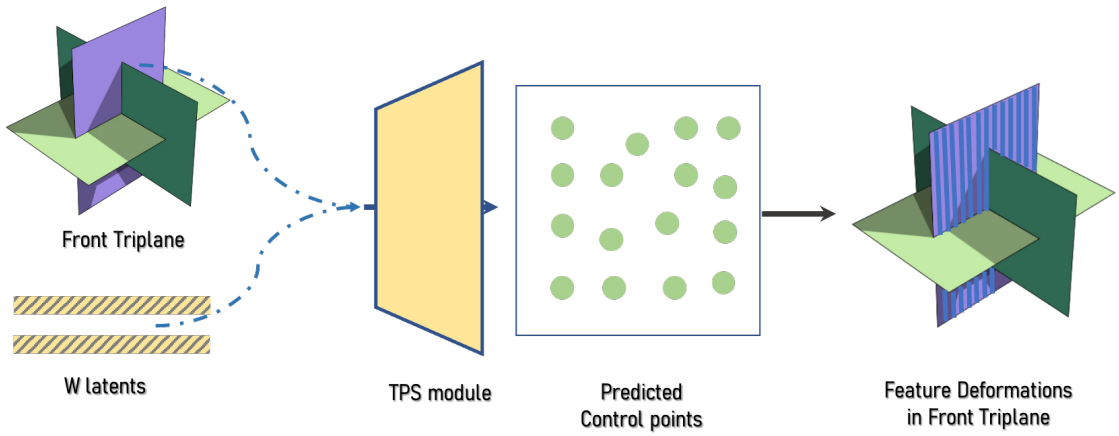

Figure 3. **Illustration of *TPS* module (Sec 3.4) main paper.**

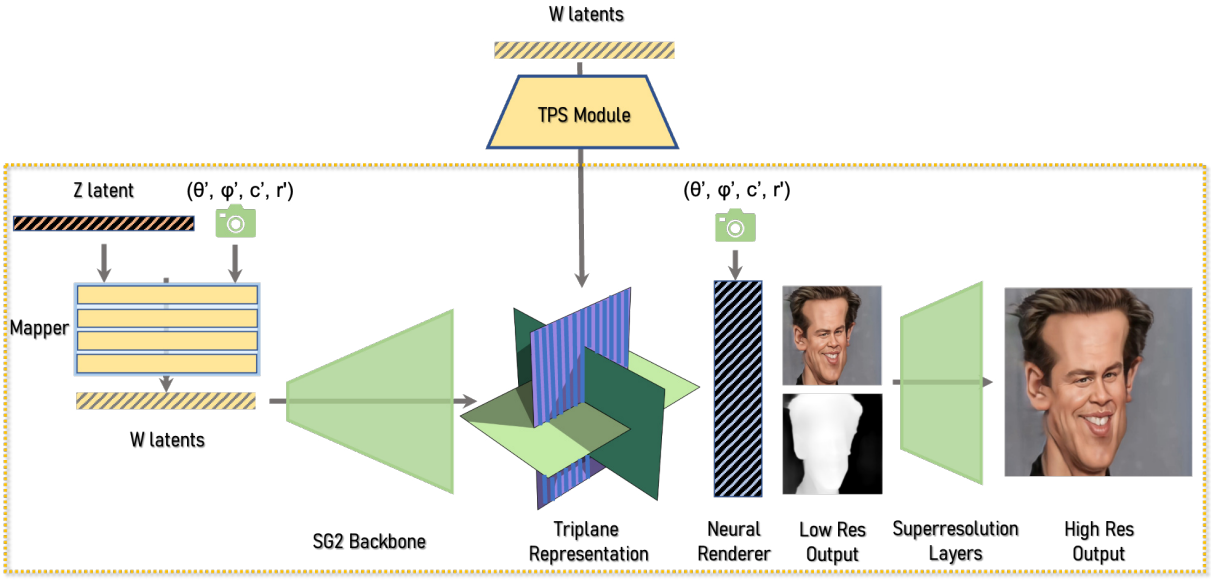

Figure 4. **Fine tuned EG3D (G_t) pipeline with *TPS* module.**

rithm 1. For additional results on real images please refer to Fig. 5. We conducted a user study using 50 real images (25 caricatures, and 25 Pixar) on identity preservation and 3D consistency versus the baseline method. We asked 21 unique workers where each triplet was reviewed by 10 workers and our avatars were chosen **92%** of the time.

8. Comparison to 2D-GANs

The quality drop is expected when the final model is compared with 2D-GANs as we derive our datasets from these 2D-GANs fine-tuned on avatar datasets. In Table 3, we compute the FID scores between the datasets (size ∼200 images for DualStyleGAN dataset) used to train these 2D-GANs and our corresponding avatar generators (size ∼10k images). We used the *3DCaricShop* [4] dataset for Caricatures as the authors of *WebCaricature* did not reply with the download link. The scores are comparable, even better in the case of Caricatures and Pixar, probably due to our regularizers including 3D view consistency.

9. Importance of Front Tri-plane Features

As discussed in Sec. 3.4 of the main paper, the front tri-plane of the EG3D architecture encodes most of the texture and depth information in the output. In Fig. 1, we show two images with their front and other side plane swapped. Then we show the corresponding effect on the output image. Notice that the results are consistent with the analysis in Sec. 3.4 where the front tri-plane dominates the information for output texture and depth.

Figure 5. **Real to avatar results.**

Algorithm 1: Projection of single image into 3D Avatar.

Input: source image $x \in \mathbb{R}^{n \times n \times 3}$; $\mathrm{G_s}$, $\mathrm{G_t}$, gradient-based optimizer F' and F''.
Output: the embedded code w' for $\mathrm{G_t}$

Initialize() the code $w = w_{avg}$;
while *not converged* **do**
  $L \leftarrow L_{mse}(x, w, \mathrm{G_s}) + L_{lpips}(x, w, \mathrm{G_s}) + L_d(w) + A_t(x, w, \mathrm{G_s})$;
  $w \leftarrow w - \eta F'(\nabla_w L, w)$;
end
Initialize() the code $w' = w$;
while *not converged* **do**
  $L' \leftarrow L_{mse}(x, w', \mathrm{G_t}) + L_{lpips}(x, w', \mathrm{G_t}) + L_d(w') + A_t(x, w', \mathrm{G_t})$;
  $w' \leftarrow w' - \zeta F''(\nabla_{w'} L', w')$;
end

10. Stylization

To validate that our chosen layers in Sec.3.2 are responsible for geometrical and texture changes, we resort to a stylization technique. For stylization given an arbitrary reference image *e.g.* painting, we use the *Style Loss* [1] to update the layers of G_s or G_t. Essentially, we use the same parameters used in Sec. 3.1. A critical technique to achieve multi-view consistency and circumvent the ghosting face artifact due to single image overfitting is to rotate the camera to cover the θ' and ϕ' ranges in Sec.3.1 in the main paper uniformly during the optimization. In Fig. 6, we show some results using only the layers used in *Texture Regularization* (Sec.3.2). Note the high-quality texture change in the images. In Fig. 7, we show results by adding layers of *Geometry Regularization* (Sec. 3.2). Note that the geometry is changed in the examples when we use this module. Note that in this example, the geometry is not expected to match as there is no such loss in the optimization. This example results in some arbitrary geometry change that is not flat. This validates our choice of geometry and texture layers used in this paper.

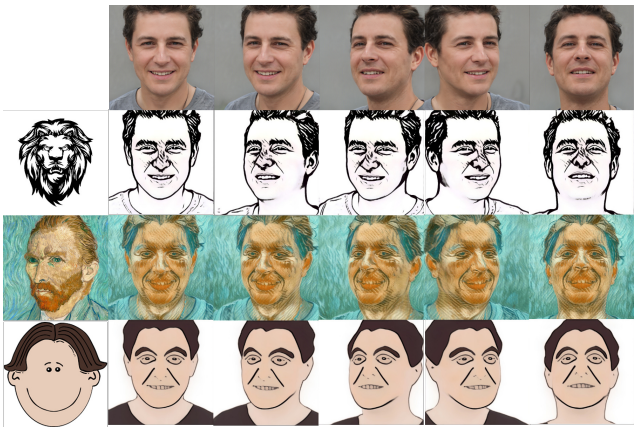

Figure 6. **Validation of texture regularization.** Style Transfer using the texture layers discussed in the main paper. Here the style of the image is changed using the texture layers.

Figure 7. **Validation of geometry regularization.** Geometry change (third row) using the geometry layers discussed in the main paper. Here the style (second row) is changed using the texture layers and the geometry (third row) is changed using the geometry layers. Note that the geometry is not correct as we apply *Style Loss* using a single image and is only used to demonstrate the usage of different layers.

11. Failure cases

Our method has some failure cases that stem from the samples of the 2D-GANs *i.e.* DualStyleGAN [5] and StyleCariGAN [2]. In the caricature domain, the random samples generated in the case of $\mathrm{G_t}$ trained on StyleCariGAN samples can have some severe artifacts (See Fig. 8). Although these do not appear often, such a sample can be improved

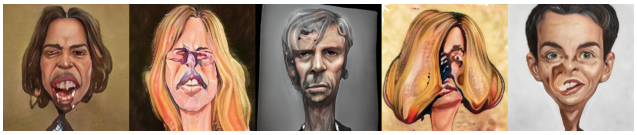

Figure 8. **Failure cases.** Failure cases in caricature generation that stems from the artifacts in the 2D-GAN from which the dataset is generated. Here the artifacts are the result of the generated results of StyleCariGAN [2]

by attribute classifier and depth regularization losses used on a single image as discussed in Sec. 4 of the main paper.

12. Depth Map visualization

In order to show some more samples and the corresponding depth maps for G_{base} and G_t, in Fig. 9, we show some samples from both the generators and the corresponding depth maps with pose changes. Notice the flat geometry in the case of G_{base} results. Next, in Fig. 10 and Fig. 11, we show some grid samples of our method with depth maps on the Caricature, Pixar toon, Cartoon, and Comic datasets.

13. Video Results

We also show our 3D avatar editing results in videos. We design a simple UI to show that the avatars can be edited in an interactive manner. Please refer to the webpage for editing videos and the interactive editing sessions.

References

[1] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 4

[2] Wonjong Jang, Gwangjin Ju, Yucheol Jung, Jiaolong Yang, Xin Tong, and Seungyong Lee. Stylecarigan: Caricature generation via stylegan feature map modulation. 40(4), 2021. 4, 5

[3] NVlabs. ffhq-dataset. https://github.com/NVlabs/ffhq-dataset. 1

[4] Yuda Qiu, Xiaojie Xu, Lingteng Qiu, Yan Pan, Yushuang Wu, Weikai Chen, and Xiaoguang Han. 3dcaricshop: A dataset and a baseline method for single-view 3d caricature face reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10236–10245, 2021. 3

[5] Shuai Yang, Liming Jiang, Ziwei Liu, and Chen Change Loy. Pastiche master: Exemplar-based high-resolution portrait style transfer. In *CVPR*, 2022. 4

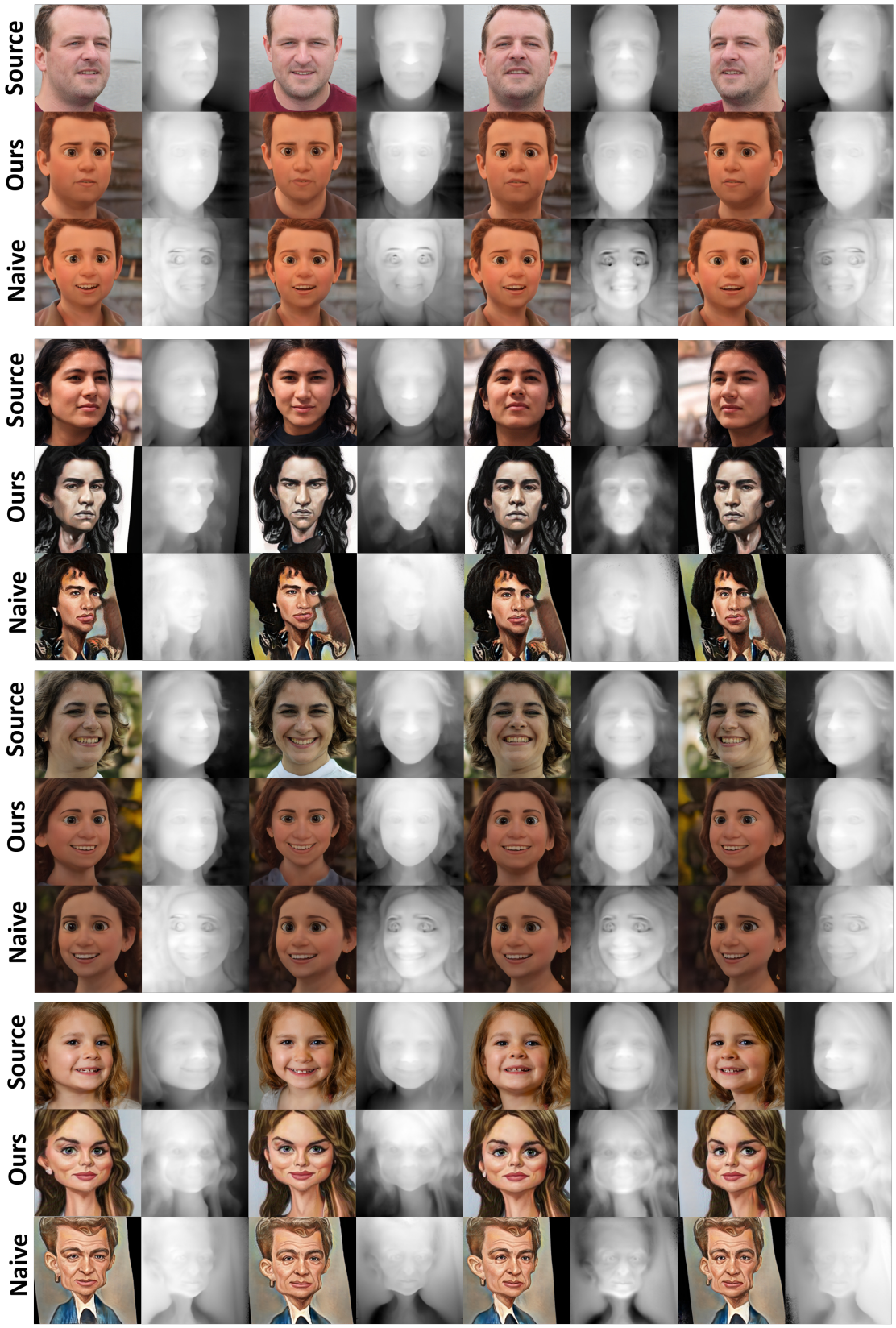

Figure 9. **Comparison with Naive method.** Results of the Caricatures and Pixar toons were viewed from different viewpoints and compared with the baseline method. Note that the depth maps are also visualized highlighting flat geometry. For more results refer to the videos on the Project Page.

Figure 10. **Grid samples.** Samples from the source domain and corresponding results in the target domain. Corresponding images and depth outputs of the Caricatures and Pixar Toons are shown.

Figure 11. **Grid samples.** Extension of Fig. 6. Corresponding images and depth outputs of the Comics and Cartoons are shown.