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

Recent work in Neural Fields (NFs) learn 3D representations from class-specific single view image collections. However, they are unable to reconstruct the input data preserving high-frequency details. Further, these methods do not disentangle appearance from geometry and hence are not suitable for tasks such as texture transfer and editing. In this work, we propose TEGLO (Textured EG3D-GLO) for learning 3D representations from single view in-the-wild image collections for a given class of objects. We accomplish this by training a conditional Neural Radiance Field (NeRF) without any explicit 3D supervision. We equip our method with editing capabilities by creating a dense correspondence mapping to a 2D canonical space. We demonstrate that such mapping enables texture transfer and texture editing without requiring meshes with shared topology. Our key insight is that by mapping the input image pixels onto the texture space we can achieve near perfect reconstruction ($\geq$ 74 dB PSNR at 1024^2 resolution). Our formulation allows for high quality 3D consistent novel view synthesis with high-frequency details even at megapixel image resolutions. Project Page: teglo-nerf.github.io

Figure 1. **Teaser** - Demonstrating TEGLO for high fidelity 3D reconstruction and multi-view consistent texture representation and texture editing from single-view image collections of objects.

1. Introduction

Reconstructing high-resolution and high-fidelity 3D consistent representations from single-view in-the-wild image collections is critical for applications in virtual reality, 3D content creation and telepresence systems. Recent work in Neural Radiance Fields (NeRFs) [6, 7, 17, 40] aim to address this by leveraging the inductive bias across a dataset of single-view images of class-specific objects for 3D consistent rendering. However, they are unable to preserve high frequency details while reconstructing the input data despite the use of SIREN [45] or positional encoding [34], in part due to the properties of MLPs they use [10]. For arbitrary resolution 3D reconstruction from single-view images, these methods face several challenges. These include image-space approximations that break multi-view consistency constraining the rendering resolution [6], requiring Pivotal Tuning Inversion (PTI) [42] or fine-tuning for reconstruction [6, 17, 46] and the inability to preserve high-frequency details [6, 17, 40, 46]. To address this, we propose TEGLO (Textured EG3D-GLO) that uses a tri-plane representation [6] and Generative Latent Optimization (GLO) [4] based training to enable efficient and high-fidelity 3D reconstruction and novel view synthesis at arbitrary image resolutions from single-view image collections of objects.

Recent works disentangle texture from geometry [10, 59] and enable challenging tasks such as texture editing and texture transfer. However, they depend on large-scale textured mesh data for high-fidelity 3D reconstruction which is laborious, expensive and time intensive to capture. Further, the use of a capture environment may cause a dataset-shift leading to generalization issues in downstream tasks, and the data use may require custom licensing. All of these fac-

*These authors contributed equally to this work.

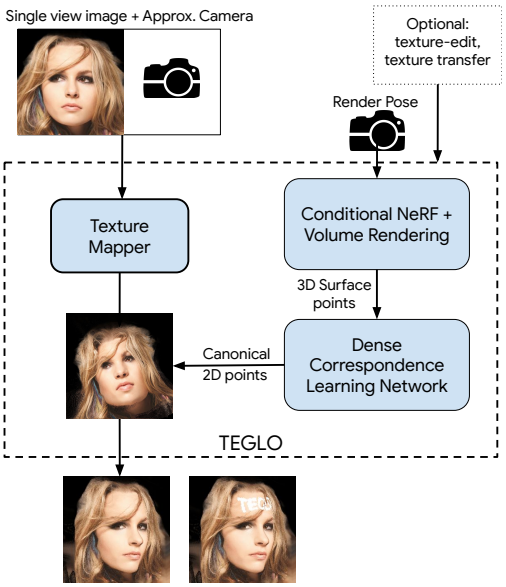

Figure 2. **Overview** - TEGLO enables 3D reconstruction and texture representation from single-view image collections of objects.

tors limit access from the broader research community. This motivates the need for a method to learn textured 3D representations from single-view in-the-wild images of objects. However, the task of disentangling texture and 3D geometry from in-the-wild image collections is a formidable challenge due to the presence of wide variations in poses, partial views, complex details in appearance, geometry, noise *etc*. in the given image collection. Inspired by surface fields [16], TEGLO leverages the 3D surface points of objects extracted from a NeRF to learn dense correspondences via a canonical coordinate space to enable texture transfer, texture editing and high-fidelity single-view 3D reconstruction.

Our key insight is that by disentangling texture and geometry using the 3D surface points of objects to learn a dense correspondence mapping via a 2D canonical coordinate space, we can extract a texture for each object. Then, by using the learned correspondences to map the pixels from the input image of the object onto the texture, we enable preserving high-frequency details. As expected, copying the input image pixels onto the texture accurately, allows near perfect reconstruction while preserving high frequency details with multi-view consistent representations. In this work, we present TEGLO, a tri-plane and GLO-based conditional NeRF, and a method to learn dense correspondences to enable challenging tasks such as texture transfer, texture editing and high-fidelity 3D reconstruction even at large megapixel resolutions. We also show that TEGLO enables single-view 3D reconstruction with no constraints on resolution by simply inverting the image into the latent table without any PTI [42] or fine-tuning. We present an overview of TEGLO in Fig.(2): TEGLO takes a single-view image and its approximate camera pose to map the pixels onto a texture. Then, to render the object from a different view, we extract the 3D surface points from the trained NeRF and use the dense correspondences to obtain the color for each pixel from the texture. Optionally, TEGLO allows texture edits and texture transfer across objects. In summary, our contributions are:

1. A framework for effectively mapping the pixels from an in-the-wild single-view image onto a texture to enable high-fidelity 3D consistent representations preserving high-frequency details.
2. A method for extracting canonical textures from single-view images enabling tasks such as texture editing and texture transfer for NeRFs.
3. Demonstrating effective mapping of single-view image pixels to a canonical texture space while preserving 3D consistency and achieving near perfect reconstruction ($\geq$ 74 dB PSNR at 1024^2 resolution).

2. Related Work

3D-aware generative models. Learning 3D representations from multi-view images with camera poses have been extensively studied since the explosion of Neural Radiance Fields (NeRFs) [2, 17, 34, 47, 62, 63]. However, these methods require several views and learn a radiance field for a single scene. RegNeRF [37] reduces the need from several views to only a handful, however, the results have several artifacts. Recently, several works learn 3D representations from single-view images [6, 7, 28, 40, 46, 64]. Further, [24, 48–50] enable multi-view consistent editing, however, they are limited by the rendering resolution. Recent work propose single image 3D consistent novel view synthesis [18, 29, 54, 60], however they are not yet suitable for texture representation. While point cloud based diffusion models [36, 61] enable learning 3D representations, they have limited applicability in textured 3D generation and high fidelity novel view synthesis. In this work, we show that TEGLO learns textured 3D representations from class-specific single-view image collections.

Texture representation. Template based methods [3, 11, 20, 39] deform a template mesh prior for 3D representations and are hence restricted in the topology they can represent. Texture Fields [38] enable predicting textured 3D models given an image and a 3D shape, but are unable to represent high-frequency details. While NeuTex [55] enables texture representation, it does not allow multi-view consistent texture editing at the desired locations due to a contorted UV mapping [59]. NeuMesh [59] learns mesh representations to enable texture transfer and texture editing using textured meshes. However, it performs mesh-guided texture transfer and requires spatial-aware fine-tuning for mesh-guided texture edits. While GET3D [15] learns textured 3D shapes by leveraging tri-plane based geometry and texture generators, it requires 2D silhouette supervision and is limited

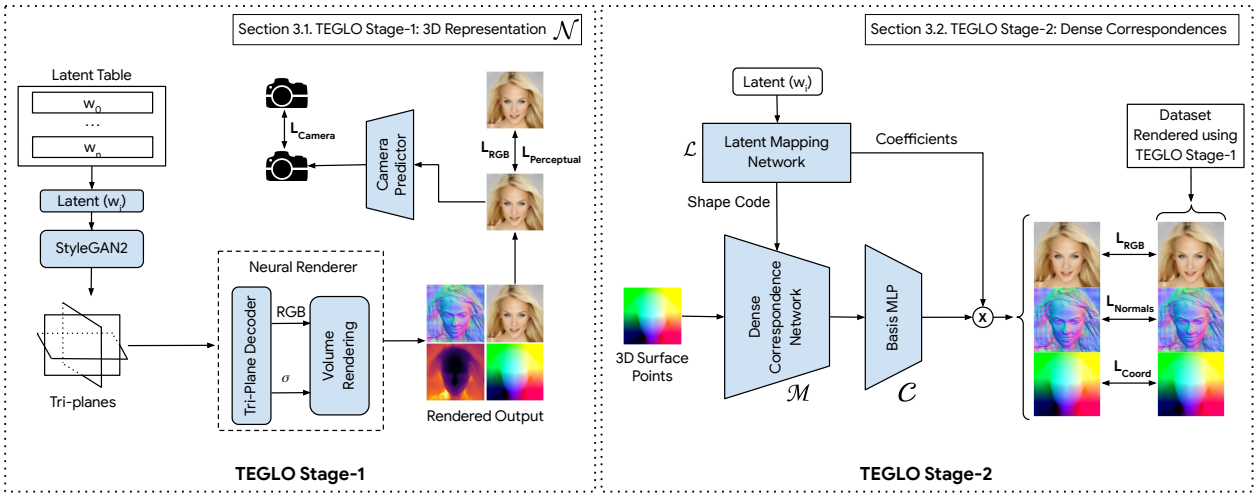

Figure 3. **Architecture** - TEGLO Stage-1 (left) uses a tri-plane and GLO based conditional NeRF to learn a per-object table of latents to reconstruct the single-view image collection. TEGLO Stage-2 (right) learns dense correspondences via a 2D canonical coordinate space.

to synthetic data. AUVNet [10] represents textures from textured meshes by learning an aligned UV mapping and demonstrates texture transfer. However, it depends on textured mesh data and requires multiple networks to enable single-view 3D reconstruction. In contrast, TEGLO learns textured 3D consistent representations from single-view images by inverting the image into the latent table.

Dense correspondences. Previous work in dense correspondence learning involve supervised [13, 27] or unsupervised [56, 58] learning methods. CoordGAN [35] learns dense correspondences by extracting each image as warped coordinate frames transformed from correspondence maps which is effective for 2D images. However, CoordGAN is unable to learn 3D correspondences. AUVNet [10] establishes dense correspondences across 3D meshes via a canonical UV mapping and asserts that methods that do not utilize color for dense correspondence learning [14,30] may have sub-par performance in texture representation.

3. Proposed Method

Given a collection of single-view in-the-wild images of objects and their approximate camera poses, TEGLO aims to learn a textured 3D representation of the data. TEGLO consists of two stages: 3D representation learning and dense correspondence learning. TEGLO Stage-1 consists of a conditional NeRF leveraging a Tri-Plane representation and an auto-decoder training regime based on generative latent optimization (GLO) [4] for 3D reconstruction of the image collection. To train TEGLO Stage-2, we use TEGLO Stage-1 to render a dataset of an object's geometry from five views using the optimized latent code. TEGLO Stage-2 uses the 3D surface points from the rendered dataset to learn dense pixel-level correspondences via a 2D canonical coordinate space. Then, the inference stage uses the learned dense correspondences to map the image pixels from the single-view input image onto a texture extracted from TEGLO-Stage 2. As a result, TEGLO effectively preserves high frequency details at an unprecedented level of accuracy even at large megapixel resolutions. TEGLO disentangles texture and geometry enabling texture transfer (Fig.(12)), texture editing (Fig.(9)) and single view 3D reconstruction without requiring fine-tuning or PTI (Fig.(8)).

3.1. TEGLO Stage 1: 3D representation

Formulation. We denote the single-view image collection ($\mathcal{I}$) with class specific objects as $\{o_0, o_1, ..., o_n\} \in \mathcal{I}$. To learn 3D representations, TEGLO uses a generative latent optimization (GLO) based auto-decoder framework, where the NeRF is conditioned on an image specific latent vector $\{w_0, w_1, ..., w_n\} \in \mathcal{R}^D$ to effectively reconstruct the image without requiring a discriminator.

Network architecture. The NeRF model $\mathcal{N}$ is represented by TEGLO Stage-1 in Fig.(3). The model $\mathcal{N}$ passes the input conditioning latent w_i to a set of CNN-based synthesis layers [23] whose output feature maps are used to construct a k-channel tri-plane. The sampled points on each ray are used to extract the tri-plane features and aggregate the k-channel features. Then the tri-plane decoder MLP outputs the scalar density σ and color which are alpha-composited by volume rendering to obtain the RGB image. Volume rendering along camera ray $r(t) = O + td$ is:

$$\mathcal{C}_{\text{NeRF}}(r, w) = \int_{b_n}^{b_f} T(t, w)\sigma(r(t), w)c(r(t), d, w)dt, \quad (1)$$
$$\text{where } T(t, w) = \exp\left(-\int_{b_n}^{b_f} \sigma(r(s), w)\right) ds,$$

Here, the radiance values can be replaced with the depth $d(x)$ or pixel opacity to obtain the surface depth. During inference, the surface depth map and 2D pixel coordinates

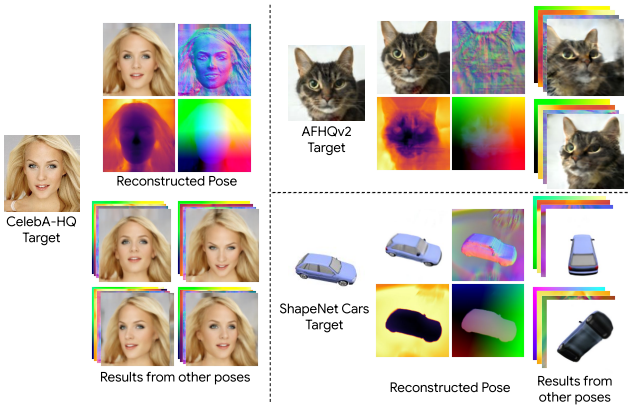

Figure 4. **Rendering the dataset for TEGLO Stage-2** - Rendering multiple views of images, surface normals, depth maps and 3D surface points from CelebA-HQ, AFHQv2-Cats and ShapeNet-Cars for learning dense correspondences in TEGLO Stage-2.

are used to obtain the 3D surface points via back-projection. The surface normals can be computed as the first derivative of the density σ with respect to the input as follows:

$$\widehat{n}(r,w) = -\int_{b_n}^{b_f} T(t,w)\ \sigma(r(t),w)\ \nabla_{r(t)}(\sigma(r(t),w))dt,$$

$$n(r,w) = \frac{\widehat{n}(r,w)}{||\ \widehat{n}(r,w)\ ||_2}, \tag{2}$$

Thus from an inference step, an RGB image, surface depth map, 3D surface points and the surface normals of the object instance can be obtained. In Fig.(4), we show the sample reconstruction results for $\mathcal{N}$ on the CelebA-HQ, AFHQv2 and ShapeNet-Cars datasets. In Fig.(5) we show qualitative results for novel view synthesis with $\mathcal{N}$ trained on SRN-Cars and evaluated on a held-out set of views. Since SRN-Cars is a multi-view dataset, we compare the rendered novel views with their corresponding ground-truth views.

Losses. $\mathcal{N}$ is trained by reconstructing the image and simultaneously optimizing a latent (w_i). As noted in [40], this allows the training loss to be enforced on individual pixels enabling training and inference at arbitrary image resolutions. For TEGLO Stage-1 (Fig.(3)), three losses are minimized to train $\mathcal{N}$: $\mathcal{L}_{\text{RGB}}$, is an $\mathcal{L}_1$ reconstruction loss between the rendered image and the ground truth image for o_i. The $\mathcal{L}_{\text{Perceptual}}$ loss is a LPIPS (Learned Perceptual Image Patch Similarity) loss between rendered image and the ground truth image. The $\mathcal{L}_{\text{Camera}}$ is the camera prediction $\mathcal{L}_1$ loss between the output of the camera encoder and the ground-truth camera parameters for the camera pose to learn 3D consistent representations of the object ($o_i \in \mathcal{I}$).

$$\mathcal{L}_{\mathcal{N}} = \mathcal{L}_{\text{RGB}} + \mathcal{L}_{\text{Perceptual}} + \mathcal{L}_{\text{Camera}}, \tag{3}$$

To train $\mathcal{N}$, we use the single-view image dataset and the approximate pose for each $o_i \in \mathcal{I}$ (Sec.(4)). We train the model for 500K steps using the Adam optimizer [25] on 8 NVIDIA V100 (16 GB) taking 36 hours to complete.

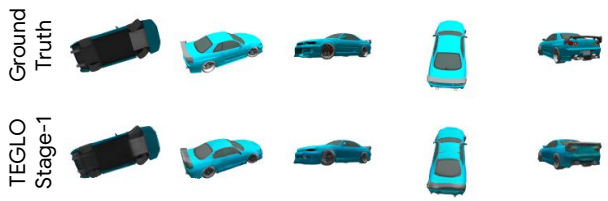

Figure 5. **Novel view synthesis** - Results for ShapeNet-Cars data.

Design choices. As noted in Sec.(1), EG3D [6] shows medium resolution (512^2) capacity while using image-space approximations in the super-resolution module which negatively affects the geometric fidelity [46]. While EpiGRAF [46] uses a patch-based discriminator for pure 3D generation, it is still prone to issues in scaling and training with multi-resolution data. Moreover, adversarial training using discriminators leads to training instability. Different from EG3D and EpiGRAF that use an adversarial training paradigm, $\mathcal{N}$ uses a GLO-based auto-decoder training paradigm which jointly optimizes a latent representation and reconstructs the image enabling arbitrary resolution synthesis - even at large megapixel resolutions - without the constraints of a discriminator. Hence, $\mathcal{N}$ enables 3D representations with geometric fidelity while also benefiting from an efficient tri-plane based representation.

EG3D [6] requires camera pose conditioning for the generator and discriminator to establish multi-view consistency. The limitation of a pose-conditioned generator is that it does not completely disentangle the pose from appearance which leads to artifacts such as degenerate solutions (2D billboards), or expressions such as the eye or smile following the camera. Since $\mathcal{N}$ optimizes a latent representation of an object and reconstructs it, we observe that the generator does not require camera pose conditioning and simply using a light-weight camera predictor network and training with a camera prediction loss ($\mathcal{L}_{\text{Camera}}$) is sufficient to learn 3D consistent representations.

3.2. TEGLO Stage 2: Dense correspondences

Formulation. We render a multi-view dataset ($\mathcal{D}$) using $\mathcal{N}$ trained on single-view image collections for the task of texture representation. We denote each object $e_i \in \mathcal{D}$ comprising of five views: $e_i = \{v_f, v_l, v_r, v_t, v_b\}$ where v denotes the view, and the sub-scripts (j for all v_j) denote frontal, left, right, top and bottom poses respectively (refer Fig.(4)). In $\mathcal{D}$, each view $v_j \in e_i$ includes the depth map ($\widehat{d}_j$), RGB image ($\widehat{r}_j$), surface normals ($\widehat{s}_j$), 3D surface points ($\widehat{p}_j$), and the optimized latent, w_i, which is identical for views of e_i as it is independent of camera pose (Fig.(4)). For TEGLO Stage 2, we use $\{\{\widehat{r}_j, \widehat{s}_j, \widehat{p}_j\} \in v_j, w_i\} \in e_i\}$.

Learning dense pixel-level correspondences across multiple views of an object is the task of locating the same 3D coordinate point in a canonical coordinate space. Inspired

by surface fields [16], we aim to learn dense correspondences using the 3D surface points extracted from $\mathcal{N}$ by back-projecting the depth ($\widehat{d}_j$) and pixel coordinates. Inspired by CoordGAN [35] and AUVNet [10], we propose a dense correspondence learning network in TEGLO Stage-2 trained in an unsupervised manner learning an aligned canonical coordinate space to locate the same 3D surface point across different views (v_j) of the same object (e_i).

Network architecture. TEGLO Stage-2 (Fig.(3)) consists of a latent mapping network ($\mathcal{L}$), a dense correspondence network ($\mathcal{M}$) and a basis network ($\mathcal{C}$) - all of which are MLP networks. The 3D surface points ($\widehat{p}_j$) from $v_j \in e_i$) are mapped to a 2D canonical coordinate space conditioned on a shape code mapped from the optimized latent w_i for e_i. We use a Lipschitz regularization [31] for each MLP layer in the dense correspondence network ($\mathcal{M}$). The latent mapping network ($\mathcal{L}$) is a set of MLP layers that takes the w_i-latent for e_i as input and predicts a shape-code for conditioning $\mathcal{M}$, and coefficients for the deformed basis. Previous work [10, 52] show that if the input is allowed to be represented as a weighted sum of basis images, *i.e.* to obtain a deformed basis before decomposition, then the 2D canonical coordinate space will be aligned. The basis network ($\mathcal{C}$) is similar to [10] and uses the predicted coefficients to decompose the deformed coordinate points. Thus, $\mathcal{M}$ maps the 3D surface points to an aligned 2D canonical coordinate space, enabling the network to learn dense correspondences using $p_j \in \mathcal{S}$ extracted from $\mathcal{N}$. Next, the basis network takes the 2D canonical coordinates as input to predict the deformed basis $\mathcal{B}$. Then, $\mathcal{B}$ is weighted with the predicted coefficients to decompose the basis into the 3D surface points (p_j), surface normals (s_j) and color (r_j).

Losses. TEGLO Stage-2 is trained using three $\mathcal{L}_2$ reconstruction losses: the $\mathcal{L}_{\text{RGB}}$ loss between the rendered RGB image $\widehat{r}_j$ and the predicted RGB image r_j; the $\mathcal{L}_{\text{Normals}}$ loss between the rendered surface normals $\widehat{s}_j$ and the predicted surface normals s_j; $\mathcal{L}_{\text{Coord}}$ loss between the extracted 3D surface points $\widehat{p}_j$ and the predicted 3D surface points p_j.

$$\mathcal{L}_{\text{Stage2}} = \mathcal{L}_{\text{RGB}} + \mathcal{L}_{\text{Normals}} + \mathcal{L}_{\text{Coord}}, \tag{4}$$

To train TEGLO Stage-2, we use the rendered dataset $\mathcal{D}$ consisting of 1000 objects with five views per object and the optimized latent for each identity. The networks are trained using $\mathcal{L}_{\text{Stage2}}$ loss for 1000 epochs using the Adam [25] optimizer to learn dense correspondences across $e_i \in \mathcal{D}$.

Design choices. We use the optimized w-latent from $\mathcal{N}$ for learning the shape code and coefficients for TEGLO Stage-2 because it represents the 3D geometry and appearance information for object (e_i) independent of camera pose. We observe that using a Lipschitz regularization for every MLP layer in $\mathcal{M}$ suitably regularizes the network to deform the input surface points $\widehat{s}_j$. Interestingly, our experiments show that simply reconstructing the 3D surface points instead of the color, surface points and surface normals also leads to learning reasonable dense pixel-level correspondences. We show qualitative results for TEGLO Stage-2 trained using only $\mathcal{L}_{\text{Coord}}$ loss in Fig.(7) as TEGLO-3DP.

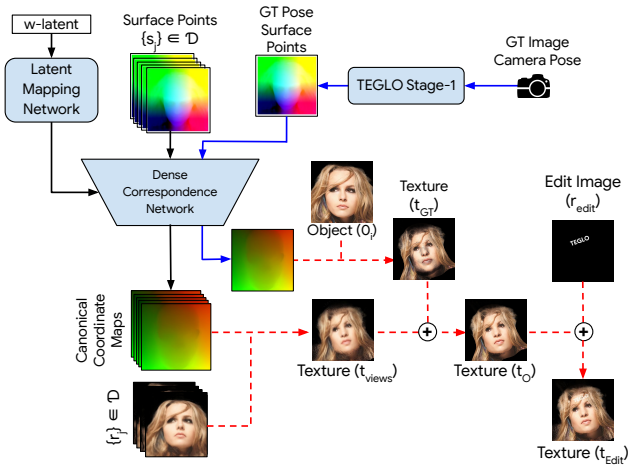

Figure 6. **Inference** - TEGLO texture extraction for texture transfer and editing. Red arrows indicate the use of a K-d tree to store the texture. Blue arrows indicate the use of input image pixels.

3.3. Inference.

Extracting the texture. We use the learned dense correspondences from TEGLO Stage-2 to extract a texture map for each object $o_i \in \mathcal{I}$. We use the pose of the target image o_i to extract the 3D surface points from $\mathcal{N}$ and use it to map the image pixels to the 2D canonical coordinate space. We denote this as texture t_{GT}. Similarly, we use $\mathcal{M}$ to map the respective RGB values from $\{v_f, v_l, v_r, v_t, v_b\} \in e_i$ using the corresponding 3D surface points (s_j) from five views to the 2D canonical space and denote it as t_{views}. Thus, textures t_{GT} and t_{views} store a mapping *i.e.* the canonical coordinate point and the corresponding RGB values. The procedure is represented in Fig.(6) and textures are depicted in Fig.(9) and Fig.(10). In Fig.(6) t_O represents the texture obtained by combining t_{GT} and t_{views}. We store this mapping in a K-d tree which enables us to index into the textures using accurate floating point indices to obtain the RGB values. The K-d tree allows querying with canonical coordinates to extract multiple neighbors making TEGLO robust to sparse "holes" in the texture. Refer Fig.(S6) in the supplementary.

Novel view synthesis. For rendering novel views of o_i, we extract the 3D surface points for the pose from $\mathcal{N}$ and obtain the canonical coordinates from $\mathcal{M}$. For each 2D canonical coordinate point c_k, we query the K-d tree for three natural neighbors and obtain indices for the neighbors which are used to obtain the RGB values. Natural Neighbor Interpolation (NNI) [44] enables fast and robust reconstruction of a surface based on a Dirichlet tesselation - unique for every set of query points - to provide an unambiguous interpolation result. We simplify the natural neighbor interpola-

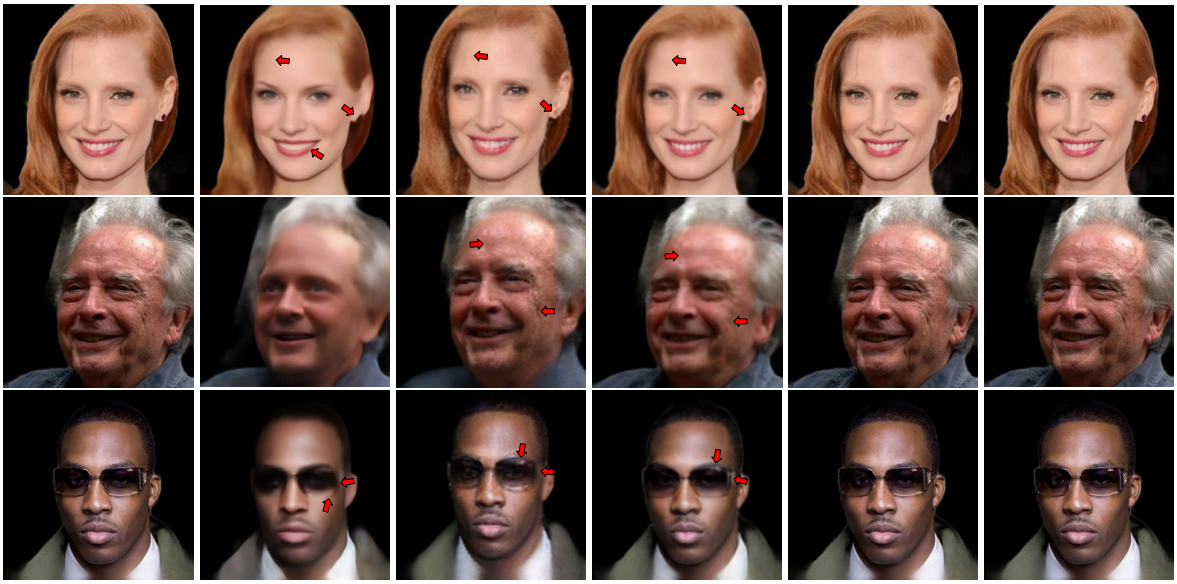

Figure 7. **Qualitative results** - Comparison with relevant 3D-aware generative baseline methods at 256^2 resolution for CelebA-HQ.

tion (NNI) based only on the distances of the points c_k in the 2D canonical coordinate space to obtain the RGB values from the stored texture. The robust and unambiguous interpolation enables TEGLO to effectively map the ground-truth image pixels from the input dataset $\mathcal{I}$ onto the geometry for novel view synthesis. To extract the Surface Field $\mathcal{S}$, we render e_i from five camera poses which may potentially cause camera pose biases leading to sparse "holes" in the texture. Our formulation uses the K-d tree and NNI to interpolate and index into the textures with sparse "holes" (Refer Fig.(S6)). There are three issues that may arise:

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.

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). We include more details in the supplementary material.

Texture editing. Texture with edits are represented as t_{Edit} in Fig.(6). We create the edits on a blank image the same size as t_{O} and denote it as r_{edit}. The edit image r_{edit} is considered to be in the canonical space and is directly indexed into the K-d tree to be overlay on t_O. Note that we do not constrain the texture space and it may be visually aligned to a canonical pose as in Fig.(9) and Fig.(10). The texture with an edit (t_{Edit}) is created by overlaying r_{edit} on t_O. Qualitative results are in Fig.(1) and Fig.(9).

Table 1. **Reconstruction of train images** - Quantitative comparison on training data reconstruction at 128^2 resolution.

Method	PSNR (↑)	LPIPS (↓)
π-GAN [7] (CelebA)	23.5	0.226
LoLNeRF [40] (FFHQ)	29.0	0.199
LoLNeRF [40] (CelebA-HQ)	29.1	0.197
ABC [41] (CelebA-HQ)	26.3	-
TEGLO Stage 1 (FFHQ)	29.0	0.294
TEGLO Stage 1 (CelebA-HQ)	28.9	0.317
TEGLO (CelebA-HQ)	**89.5**	**2.3e-7**

4. Experiments and Results

Datasets. We train TEGLO with single-view image datasets such as FFHQ [23], CelebA-HQ [21, 32] and AFHQv2-Cats [12, 22]. To obtain the approximate camera pose, we follow [40] by first using an off-the-shelf face landmark predictor MediaPipe Face Mesh [1] to extract landmarks appearing at consistent locations. Then, we use a shape-matching least-squares optimization to align the landmarks with 3D canonical landmarks to obtain the approximate pose. We also use a multi-view image dataset - ShapeNet-Cars [8, 9] with results in Fig.(1) and Table.(4).

3D reconstruction. We evaluate TEGLO on the task of reconstructing the input image in the same pose and compare with baseline methods. We report quantitative results for train data reconstruction in Table.(1) measuring the PSNR (Peak Signal to Noise Ratio) and LPIPS (Learned Perceptual Image Patch Similarity) metrics for CelebA-HQ and FFHQ. We observe similar results for LoLNeRF and

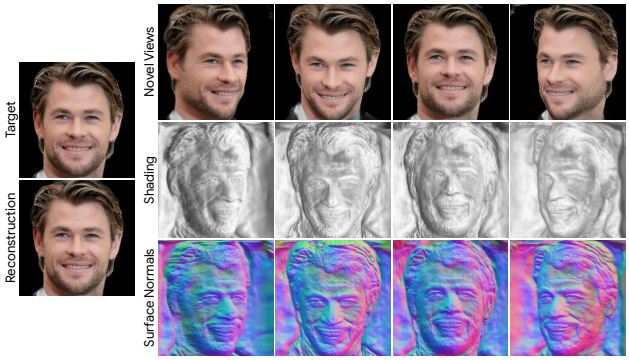

Figure 8. **Single view 3D reconstruction** - 3D reconstruction of test image from CelebA-HQ. Compare with Fig.(25) in [26]).

Table 2. **Reconstruction of test images** - Quantitative comparison on test data reconstruction at various rendering resolutions.

Method	Res.	PSNR (↑)	LPIPS (↓)
π-GAN [7] (CelebA)	256^2	21.8	0.412
LoLNeRF [40] (FFHQ)	512^2	25.3	0.491
LoLNeRF [40] (CelebA-HQ)	256^2	26.2	0.363
TEGLO Stage 1 (FFHQ)	256^2	27.3	0.334
TEGLO Stage 1 (CelebA-HQ)	256^2	27.5	0.260
TEGLO (FFHQ)	256^2	**84.9**	**2.1e-6**
TEGLO (CelebA-HQ)	256^2	**86.2**	**7.4e-7**
TEGLO (CelebA-HQ)	512^2	82.6	4.4e-6
TEGLO (CelebA-HQ)	1024^2	74.7	6.9e-5

Table 3. **Comparing with GLO baselines** - Quantitative results for test set reconstruction in PSNR at 256^2 resolution.

Dataset	PSNR (↑)		
	LoLNeRF [40]	TEGLO Stage-1	TEGLO
AFHQv2-Cats	24.94	29.26	**87.38**

TEGLO Stage-1 at 128^2 resolution. However, as expected, TEGLO attains 89.5 dB PSNR and $7.4e$-7 for LPIPS. We report quantitative results for test data reconstruction from a held-out set at 256^2 resolution for CelebA-HQ and FFHQ data in Table.(2) and for AFHQv2-Cats data in Table.(3).

We depict qualitative results for CelebA-HQ in Fig.(7) where the red arrows indicate missing details. For EG3D-Fit, we invert the image into the EG3D [6] latent space and perform Pivotal Tuning Inversion (PTI) [42] for the single-view image. We observe missing details in the results from LoLNeRF [40], EG3D-Fit [6] and TEGLO stage-1 in terms of jewelry, skin wrinkles, eyeglass opacity, eyeglass frame, hair strand etc. As expected, results from TEGLO and TEGLO-3DP (where TEGLO Stage-2 is trained with only surface point supervision) preserve high frequency details missed by baselines methods, demonstrating near perfect reconstruction. In Fig.(10), we show qualitative results with the texture (t_O) for complex appearance and geometry.

3D consistent novel view synthesis. To evaluate multi-view consistent synthesis, we report quantitative results for novel view reconstruction on the multi-view SRN-Cars data in Table.(4). We observe that TEGLO attains near-perfect reconstruction of test data with 67.5 dB PSNR whereas

Table 4. **Novel view reconstruction** - Quantitative results for novel view reconstruction on the SRN-Cars dataset [8] at 256^2 resolution to evaluate 3D consistent novel view synthesis. (LoLNeRF result is from the "Concatenation" baseline in ABC [41]).

Dataset	PSNR (↑)			
	LoLNeRF	ABC	TEGLO Stage-1	TEGLO
SRN-Cars	25.80	29.10	30.48	**67.52**

Table 5. **Comparing with 3D generative baselines** - Test data reconstruction with previous state-of-the-art methods.

Method	PSNR↑	LPIPS↓	SSIM↑	ID↑	3D Consistency↑
EG3D-PTI	26.64	0.323	0.879	0.465	21.20
RealTime-RF [51]	22.29*	0.269	0.665	0.542	-
IDE-3D [48]	26.45	0.273	0.878	0.671	20.69
HFGI3D [57]	29.43	0.172	0.918	0.744	21.69
TEGLO	**84.90**	**2.1e-6**	**0.999**	**0.883**	**33.47**

baselines achieve 30.4 dB PSNR. We evaluate the identity consistency across multiple synthesised views using the ID score metric by computing the mean of the MagFace [33] cosine similarity scores from a sampled camera pose. We compare the ID score for TEGLO with other recent 3D GANs and observe that TEGLO outperforms the baselines with a score of 0.883. We also use the 3D consistency metric from [17] to compare the multi-view consistent synthesis of TEGLO with 3D GAN baselines. In brief, we synthesize five novel views near an input camera pose and use IBRNet [53] to predict the input image and then compute the reconstruction PSNR. We report the 3D consistency metric in Table.(5) and observe that TEGLO outperforms the 3D GAN methods. [57] notes that "quantitative evaluation of 3D consistency is still an open question" and since 3D consistency in novel view synthesis is better viewed as videos, we urge the reader to refer to the supplementary videos.

Single-view 3D reconstruction. It is the task of representing an in-the-wild or out-of-distribution image using a trained network. Qualitative results for a held-out sample from the CelebA-HQ dataset for pre-trained TEGLO is in Fig.(8). Previous work such as AUVNet [10] require additional training of a ResNet-18 [19] for the image encoder and IM-Net [11] for the shape decoder followed by ray marching to obtain the mesh to represent the image while methods such as EG3D [6] require PTI (Pivotal Tuning Inversion [42]) fine-tuning to represent the image. For single-view textured 3D representation in TEGLO, we simply invert the image into the latent with no fine-tuning. Further details about obtaining the latent are in the supplementary.

Reconstructing single-view images at arbitrary resolutions while preserving 3D consistency is highly desirable for several applications. However, EG3D [6] is limited by its camera conditioned generator to possess a "baked-in" training resolution. TEGLO does not include any camera conditioning, and as a result, it allows single-view 3D reconstruction and novel view synthesis at arbitrary resolutions without any re-training for different resolutions.

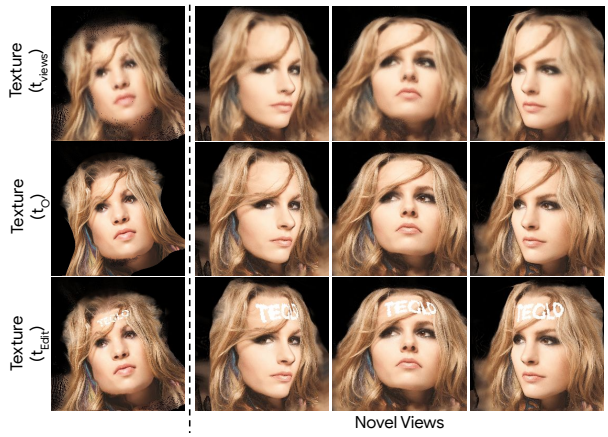

Figure 9. **Texture editing** - Qualitative results for texture edits.

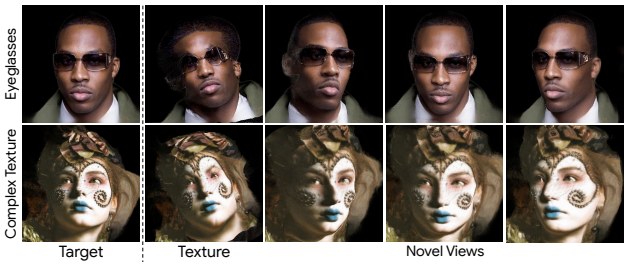

Figure 10. **Results for complex texture and geometry** - Qualitative results for texture representation and novel view synthesis with complex image samples. Compare with Fig.(24) in [5].

Texture editing. In Sec.(3.3), we describe the procedure to edit textures. Qualitative results with texture editing for CelebA-HQ is in Fig.(9) and for AFHQv2-Cats and ShapeNet-Cars in Fig.(1). Our edits are class-specific and target image agnostic because edits are performed in the canonical space. Previous work, NeuMesh [59] requires spatial-aware fine-tuning and mesh guided texture editing for precise transfer. However, TEGLO simply maps a texture edit image of the same size as the texture into the K-d tree with an overlay of the pixels (*i.e.* obtaining t_{Edit}) - precisely transferring the edit without requiring any optimization strategies. Further results are in the supplementary.

Texture transfer. As discussed in Sec.(3.3), the extracted textures are aligned in a canonical coordinate space allowing texture transfer across geometries. We demonstrate texture transfer in Fig.(12(a)). Here, row-1 represents the target image from CelebA-HQ for the geometry learned by TEGLO Stage-1, and column-1 represents the textures (stored in a K-d tree) extracted after TEGLO Stage-2. We observe realistic texture transfer despite arbitrary camera biases in rendering $\mathcal{D}$ which are mitigated by using the K-d tree and NNI. To test if TEGLO is restricted to the range of the five arbitrary views chosen for Stage-2, we show large angle view results for Stage-2 trained with just a single view instead of five in Fig.(11) to validate our hypothesis. Fig.(12(b)), shows the keypoint correspondences mapped to the canonical coordinate space across different face identities. Since the keypoints from different identities map to the same location in the canonical space, the effectiveness of the correspondences for texture transfer is demonstrated.

Figure 11. Result for TEGLO Stage-2 trained with 1 arbitrary view

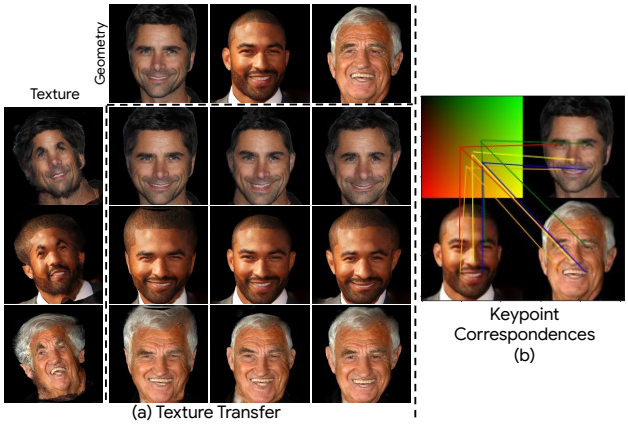

Figure 12. **Texture transfer** (a) Qualitative results for texture transfer with CelebA-HQ. (Top row shows CelebA-HQ image targets). (b) Keypoint correspondences in the canonical space.

5. Discussion

While TEGLO enables near perfect 3D reconstruction of objects from single-view image collections, it requires multi-stage training. We hope that future work can simplify the framework with an elegant end-to-end formulation. A potential next step would be to use StyleGANv2 [23] to generate high quality textures for texture transfer and editing. TEGLO could enable 3D full-body avatars from single views with high frequency details extending methods such as PIFu [43]. Future work could explore representing light stage data across different camera angles in an illumination invariant manner using 3D surface points. One limitation of our method is that the texture does not include ground truth pixels from the obstructed parts of the object. We hope future work can address this limitation.

6. Conclusion

In this work, we present TEGLO for high-fidelity canonical texture mapping from single-view images enabling textured 3D representations from class-specific single-view image collections. TEGLO consists of a conditional NeRF and a dense correspondence learning network that enable texture editing and texture transfer. We show that by effectively mapping the input image pixels onto the texture, we can achieve near perfect reconstruction ($\geq$ 74 dB PSNR at 1024^2 resolution). TEGLO also allows single-view 3D reconstruction by simply inverting the single-view image into the latent table without requiring any PTI or fine-tuning.

References

[1] Mediapipe face mesh. https://google.github.io/mediapipe/solutions/face_mesh.html. Online; Accessed: 2022-06-20. 6

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 2

[3] Anand Bhattad, Aysegul Dundar, Guilin Liu, Andrew Tao, and Bryan Catanzaro. View generalization for single image textured 3d models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6081–6090, 2021. 2

[4] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of generative networks. *arXiv preprint arXiv:1707.05776*, 2017. 1, 3

[5] Chen Cao, Tomas Simon, Jin Kyu Kim, Gabe Schwartz, Michael Zollhoefer, Shun-Suke Saito, Stephen Lombardi, Shih-En Wei, Danielle Belko, Shoou-I Yu, et al. Authentic volumetric avatars from a phone scan. *ACM Transactions on Graphics (TOG)*, 41(4):1–19, 2022. 8

[6] 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, 4, 7

[7] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5799–5809, 2021. 1, 2, 6, 7

[8] 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. 6, 7

[9] 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. 6

[10] 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. 1, 3, 5, 7

[11] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 2, 7

[12] 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. 6

[13] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. *Advances in neural information processing systems*, 29, 2016. 3

[14] 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. 3

[15] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. In *Advances In Neural Information Processing Systems*, 2022. 2

[16] Lily Goli, Daniel Rebain, Sara Sabour, Animesh Garg, and Andrea Tagliasacchi. nerf2nerf: Pairwise registration of neural radiance fields. *arXiv preprint arXiv:2211.01600*, 2022. 2, 5

[17] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*, 2021. 1, 2, 7

[18] Jiatao Gu, Alex Trevithick, Kai-En Lin, Josh Susskind, Christian Theobalt, Lingjie Liu, and Ravi Ramamoorthi. Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. *arXiv preprint arXiv:2302.10109*, 2023. 2

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[20] Paul Henderson, Vagia Tsiminaki, and Christoph H Lampert. Leveraging 2d data to learn textured 3d mesh generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7498–7507, 2020. 2

[21] 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. 6

[22] 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. 6

[23] 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. 3, 6, 8

[24] Hyunsu Kim, Gayoung Lee, Yunjey Choi, Jin-Hwa Kim, and Jun-Yan Zhu. 3d-aware blending with generative nerfs. *arXiv preprint arXiv:2302.06608*, 2023. 2

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4, 5

[26] Jiaman Li, Zhengfei Kuang, Yajie Zhao, Mingming He, Karl Bladin, and Hao Li. Dynamic facial asset and rig generation from a single scan. *ACM Trans. Graph.*, 39(6):215–1, 2020. 7

[27] Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[28] Connor Z Lin, David B Lindell, Eric R Chan, and Gordon Wetzstein. 3d gan inversion for controllable portrait image animation. *arXiv preprint arXiv:2203.13441*, 2022. 2

[29] Kai-En Lin, Yen-Chen Lin, Wei-Sheng Lai, Tsung-Yi Lin, Yi-Chang Shih, and Ravi Ramamoorthi. Vision transformer for nerf-based view synthesis from a single input image. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 806–815, 2023. 2

[30] 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. 3

[31] 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. 5

[32] 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. 6

[33] Qiang Meng, Shichao Zhao, Zhida Huang, and Feng Zhou. Magface: A universal representation for face recognition and quality assessment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14225–14234, 2021. 7

[34] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2

[35] Jiteng Mu, Shalini De Mello, Zhiding Yu, Nuno Vasconcelos, Xiaolong Wang, Jan Kautz, and Sifei Liu. Coordgan: Self-supervised dense correspondences emerge from gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10011–10020, 2022. 3, 5

[36] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 2

[37] 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. 2

[38] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4531–4540, 2019. 2

[39] Dario Pavllo, Jonas Kohler, Thomas Hofmann, and Aurelien Lucchi. Learning generative models of textured 3d meshes from real-world images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13879–13889, 2021. 2

[40] 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, 2, 4, 6, 7

[41] Daniel Rebain, Mark J Matthews, Kwang Moo Yi, Gopal Sharma, Dmitry Lagun, and Andrea Tagliasacchi. Attention beats concatenation for conditioning neural fields. *arXiv preprint arXiv:2209.10684*, 2022. 6, 7

[42] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Transactions on Graphics (TOG)*, 42(1):1–13, 2022. 1, 2, 7

[43] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2304–2314, 2019. 8

[44] Robin Sibson. A brief description of natural neighbour interpolation. *Interpreting multivariate data*, pages 21–36, 1981. 5

[45] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020. 1

[46] Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. Epigraf: Rethinking training of 3d gans. *arXiv preprint arXiv:2206.10535*, 2022. 1, 2, 4

[47] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7495–7504, 2021. 2

[48] Jingxiang Sun, Xuan Wang, Yichun Shi, Lizhen Wang, Jue Wang, and Yebin Liu. Ide-3d: Interactive disentangled editing for high-resolution 3d-aware portrait synthesis. *arXiv preprint arXiv:2205.15517*, 2022. 2, 7

[49] Jingxiang Sun, Xuan Wang, Yong Zhang, Xiaoyu Li, Qi Zhang, Yebin Liu, and Jue Wang. Fenerf: Face editing in neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7672–7682, 2022. 2

[50] Junshu Tang, Bo Zhang, Binxin Yang, Ting Zhang, Dong Chen, Lizhuang Ma, and Fang Wen. Explicitly controllable 3d-aware portrait generation. *arXiv preprint arXiv:2209.05434*, 2022. 2

[51] Alex Trevithick, Matthew Chan, Michael Stengel, Eric R Chan, Chao Liu, Zhiding Yu, Sameh Khamis, Manmohan Chandraker, Ravi Ramamoorthi, and Koki Nagano. Real-

time radiance fields for single-image portrait view synthesis. *arXiv preprint arXiv:2305.02310*, 2023. 7

[52] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991. 5

[53] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021. 7

[54] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. *arXiv preprint arXiv:2210.04628*, 2022. 2

[55] Fanbo Xiang, Zexiang Xu, Milos Hasan, Yannick Hold-Geoffroy, Kalyan Sunkavalli, and Hao Su. Neutex: Neural texture mapping for volumetric neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7119–7128, 2021. 2

[56] Taihong Xiao, Sifei Liu, Shalini De Mello, Zhiding Yu, Jan Kautz, and Ming-Hsuan Yang. Learning contrastive representation for semantic correspondence. *International Journal of Computer Vision*, 130(5):1293–1309, 2022. 3

[57] Jiaxin Xie, Hao Ouyang, Jingtan Piao, Chenyang Lei, and Qifeng Chen. High-fidelity 3d gan inversion by pseudo-multi-view optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 321–331, 2023. 7

[58] Jiarui Xu and Xiaolong Wang. Rethinking self-supervised correspondence learning: A video frame-level similarity perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10075–10085, 2021. 3

[59] Bangbang Yang, Chong Bao, Junyi Zeng, Hujun Bao, Yinda Zhang, Zhaopeng Cui, and Guofeng Zhang. Neumesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing. *arXiv preprint arXiv:2207.11911*, 2022. 1, 2, 8

[60] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 2

[61] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. *arXiv preprint arXiv:2210.06978*, 2022. 2

[62] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2

[63] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (TOG)*, 40(6):1–18, 2021. 2

[64] Xiaoming Zhao, Fangchang Ma, David Güera, Zhile Ren, Alexander G Schwing, and Alex Colburn. Generative multiplane images: Making a 2d gan 3d-aware. *arXiv preprint arXiv:2207.10642*, 2022. 2