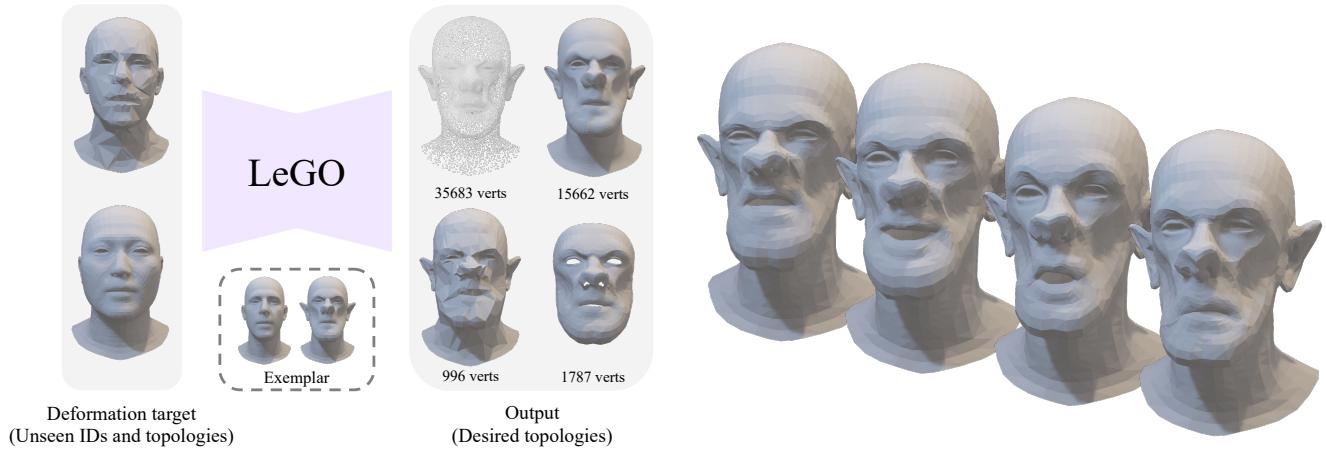
LeGO: Leveraging a Surface Deformation Network for Animatable Stylized Face Generation with One Example

Soyeon Yoon* Kwan Yun* Kwanggyoon Seo Sihun Cha Jung Eun Yoo Junyong Noh
KAIST, Visual Media Lab
{thoyeony, yunandy, skg1023, chacorp, jey920, junyongnoh}@kaist.ac.kr



(a) Diverse deformation targets and output of LeGO (b) Facial animation of stylized avatar

Figure 1. (a) The proposed method demonstrates robustness to unseen face identities and topologies and effectively generates stylized output faces with desired topologies. (b) Our stylized avatars can be animated using 3DMM blend shapes.

Abstract

Recent advances in 3D face stylization have made significant strides in few to zero-shot settings. However, the degree of stylization achieved by existing methods is often not sufficient for practical applications because they are mostly based on statistical 3D Morphable Models (3DMM) with limited variations. To this end, we propose a method that can produce a highly stylized 3D face model with desired topology. Our methods train a surface deformation network with 3DMM and translate its domain to the target style using a paired exemplar. The network achieves stylization of the 3D face mesh by mimicking the style of the target using a differentiable renderer and directional CLIP losses. Additionally, during the inference process, we utilize a Mesh Agnostic Encoder (MAGE) that takes deformation target, a mesh of diverse topologies as input to the stylization process and encodes its shape into our latent space. The resulting stylized face model can be animated by commonly used 3DMM blend shapes. A set of quantitative and qualitative evaluations demonstrate that our method can produce highly stylized face meshes according to a given style and output them in a desired topology. We also demonstrate example applications of our method including image-based stylized avatar generation, linear interpolation of geometric styles, and facial animation of stylized avatars.

1. Introduction

Crafting animatable stylized 3D avatars that encapsulate both personal identity and character style requires extensive efforts from skilled artists. When creating animated films, the artists design stylized 3D avatars whose facial appearance matches the theme of the entire content while putting careful effort into preserving the idiosyncrasy of the actors. Similarly, on social media, artists create numerous stylized presets so that the combinations of these presets can represent diverse identities.

To reduce the burden of manual crafting effort, generating stylized 3D faces has been a prominent area of research. Recent attempts include 3D-aware generative ad-

*These authors contributed equally to this work

versarial networks (GANs) and denoising diffusion models (DMs) [1, 12, 15, 18, 46], generating a stylized texture for the 3D morphable model (3DMM) [2, 47], text-based geometry deformation [7, 22, 24–26], and surface deformation methods [13, 43]. These methods have successfully demonstrated the possibility of producing high-quality and diverse stylized 3D faces, although each method has distinct challenges.

We identify three key elements for stylized avatar creation so that the results can be practically useful.

1. Avatar creation in a desired topology that is compatible with conventional CG pipelines.
2. Extending stylization capabilities beyond 3DMM.
3. Generating stylized avatars that are animatable using blend shapes.

Where each component meets the standards and demands of the dynamic entertainment landscape 1) allowing creators to reuse existing animation rigs and texture maps across different models.; 2) achieving greater diversity and flexibility in expressing unique and non-conventional characters.; 3) enabling animators to achieve coherent and natural movements across various facial features and expressions. While recent methods have made meaningful strides in one or two aspects of these key elements, there has not been a method that satisfies all three elements, as summarized in Figure 2.

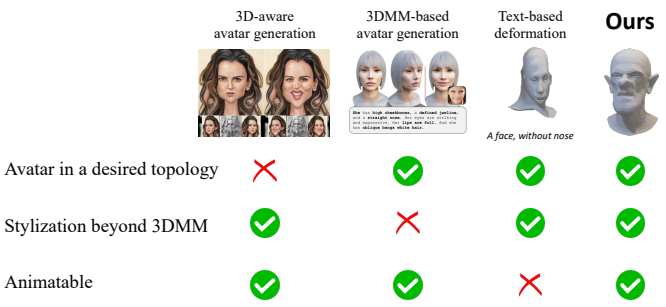


Figure 2. Comparison of different stylized 3D face generation methods and their limitations in meeting key elements. 3D-aware methods cannot generate 3D face in desired topologies. 3DMM-based methods have a limited stylization capability. Text-based deformation models are not directly animatable. The proposed method meets the goal of all three components.

To address this, we propose a novel method that can generate stylized 3D face meshes. This is achieved by translating the domain of a pre-trained surface deformation network based on one of the most widely used 3DMM model, FLAME [19], to a target style domain. We achieve this goal by first training the surface deformation network with the FLAME decoder to leverage its linear shape space combined with global expression space. During fine-tuning, we employ a directional CLIP-based domain adaptation method [16, 35, 48], widely used in 2D domain, to retain the face identity while reflecting the desired style.

In addition, to seamlessly integrate this 2D-based training method into the stylized 3D face generation task, we propose a hierarchical rendering scheme that captures local and global facial features, ensuring effective training and identity preservation. In the inference stage, we introduce a Mesh Agnostic Encoder (MAGE) to enable mesh agnostic stylization for an input, which we call a deformation target that has various mesh topologies. MAGE is composed of pre-trained encoders from Neural Face Rigging (NFR) [34] and latent mapping networks, which establish correspondences between shapes by encoding mesh representations into a topology-invariant latent space. This enhances the versatility and applicability of our approach in the context of 3D face stylization. As shown in Figure 1, our method can generate stylized 3D face models with varied mesh topologies that are equipped with the animation capability of 3DMM while ensuring consistency across diverse deformation target mesh representations.

2. Related Work

2.1. Stylized 3D face generation

With recent advances in the GANs and DMs that utilize neural fields for 3D-aware face generation [3, 10], the generation of faces in diverse styles has gained popularity [1, 12, 15, 18]. By leveraging 2D priors from generative models, which capture various patterns and variations observed in the extensive 2D training data, these methods can generate 3D faces with various styles. However, despite their success in producing consistent multi-view images through neural rendering, creating stylized faces in a desired topology is challenging, limiting their suitability for using existing graphics tools across different faces. On the other hand, 3DMM-based personalized 3D face generation methods with text-guidance [2, 47] have shown success in producing high-fidelity stylized textures for 3D face models. However, these methods confine the shape of the generated face models within the 3DMM shape space, constraining possibilities for geometric exaggerations or abstraction beyond the training data.

2.2. Learning-Based 3D Shape Network

Recently, learning implicit functions for 3D shapes has demonstrated remarkable performance in representing complex geometric structures [8, 23, 27, 30, 31]. In particular, DIF-Net [5] adopted MLPs for learning a standard signed distance function (SDF) and a volumetric deformation function, leading to comprehensive mapping between the produced SDFs. Most related to our work, DD3C [13] utilized template deformation for 3D caricature auto-decoder. They found that modeling each shape as a deformation of a fixed temaplate surface effective compared to absolute position. We advance a step further by training a surface deformation

network on 3DMM and transfer its domain into stylization.

Another line of research aims to transfer shape deformation [39]. Recently, attempts have been made to transfer deformation utilizing examples [40, 43, 45] or text [7, 24, 25]. These methods can deform a mesh into a desired style and identity using a given example. However these deformation methods may not be as effective when it comes to coherent control of animation.

2.3. Few-Shot Domain Adaptation

Domain adaptation refers to the process of utilizing a neural network that has been initially trained on a large dataset from a source domain, and subsequently fine-tuning it on a smaller dataset from a target domain. Few-shot domain adaptation is widely researched in the 2D domain, especially with generative priors [4, 12, 20, 28, 29, 41, 42, 44]. Attempts have been made in one-shot domain adaptation to utilize the semantic capabilities of vision-language networks like CLIP [35]. These networks can provide direction for distinguishing between identity and style [6, 16, 48]. We adopt this two-way directional guidance from CLIP and combine it with a differentiable renderer to effectively stylize the face mesh while preserving the original identity using a paired exemplar.

2.4. Mesh Agnostic Deformation

Recently, mesh agnostic networks [9, 11, 32, 33, 38] demonstrated a great potential for learning 3D information such as dense correspondences between different 3D shape representations without relying on consistent topology or vertex ordering. These networks operate on meshes and encode shapes into a topology-invariant latent space. Specifically, NFR [34] proposed an autoencoder framework for facial expression retargeting across different mesh topologies. The framework utilizes separate expression and identity encoders, both functioning in a topology-agnostic manner.

We integrate components of this approach into our method by mapping their embeddings into our latent space. This network translates the output of the pre-trained encoder from NFR to the latent space of our surface deformation network. This enables any geometric shape of the deformation target to be encoded and fed into our surface deformation network.

3. Method

Our research focuses on training and fine-tuning a surface deformation network to generate stylized 3D faces with diverse shapes and expressions. We start with the source face deformation network D_S that deforms the template face to a face with different identities and expressions. Thereafter, we fine-tune it into a target style face deformation network D_T using a paired exemplar. This process is outlined below.

1. We first train D_S using FLAME in a self-supervised manner, enabling the creation of versatile head meshes with different shapes and expressions.
2. For fine-tuning, we assemble a paired exemplar that consists of an identity exemplar mesh M_S and a style exemplar mesh M_T, sharing identity.
3. Both M_S and M_T jointly serve as example guidance to fine-tune D_T.
4. At inference, using MAGE and D_T, the deformation target of diverse topologies can be translated into a stylized face.

A detailed description of these processes will be provided in the following subsections.

3.1. Deformation Network as Parametric Model

D_S is a surface deformation network that deforms a template face with the given latent vectors. We train the network to generate diverse human faces with FLAME, in a self-supervising manner. FLAME can manipulate both the global head shape and local expressions through its shape parameters $\vec{\boldsymbol{\beta}} \in \mathbb{R}^{300}$, and expression parameters $\vec{\boldsymbol{\psi}} \in \mathbb{R}^{100}$, which are the components of the FLAME parameters Φ. This empowers D_S to generate diverse geometric face shapes and expressions.

To generate face meshes using these shape and expression components of Φ as input, we employ mapping networks, $\mathcal{M}_{shape}$ and $\mathcal{M}_{exp}$, which consist of Multi-Layer Perceptrons (MLP). Each mapping network separately transforms $\vec{\boldsymbol{\beta}}$ and $\vec{\boldsymbol{\psi}}$ into latent vectors, z_s and z_e, respectively.

$$z_s = \mathcal{M}_{shape}(\vec{\boldsymbol{\beta}}), \quad z_e = \mathcal{M}_{exp}(\vec{\boldsymbol{\psi}}) \tag{1}$$

These latent vectors are then fed into D_S, enabling the creation of diverse face meshes imbued with a wide range of expressions. The mapping networks $\mathcal{M}_{shape}$ and $\mathcal{M}_{exp}$ are trained jointly with D_S during the training process. The following losses are employed:

$$\mathrm{L}(\vec{\boldsymbol{\beta}}, \vec{\boldsymbol{\psi}}) = \left\| \mathrm{FLAME}\left(\vec{\boldsymbol{\beta}}, \vec{\boldsymbol{\psi}}\right) - D_S\left(\begin{bmatrix} \mathcal{M}_{shape}(\vec{\boldsymbol{\beta}}) \\ \mathcal{M}_{exp}(\vec{\boldsymbol{\psi}}) \end{bmatrix}\right) \right\|_2^2 \tag{2}$$

Surface-Intensive Mesh Sampling To enable D_S to represent various geometries, we introduce a surface-intensive mesh sampling (SIMS) strategy during training. SIMS is performed by randomly selecting points from the surfaces of the face mesh, $FLAME(\vec{\boldsymbol{\beta}}, \vec{\boldsymbol{\psi}})$, during training D_S. Specifically, we sample approximately 4 times more points from the surface than the number of vertices of the face mesh. Experiment results are reported in Sec. 4.2.

3.2. LeGO

The fine-tuning process utilizes a paired exemplar, M_S and M_T, to guide the adaptation of D_T for the generation of

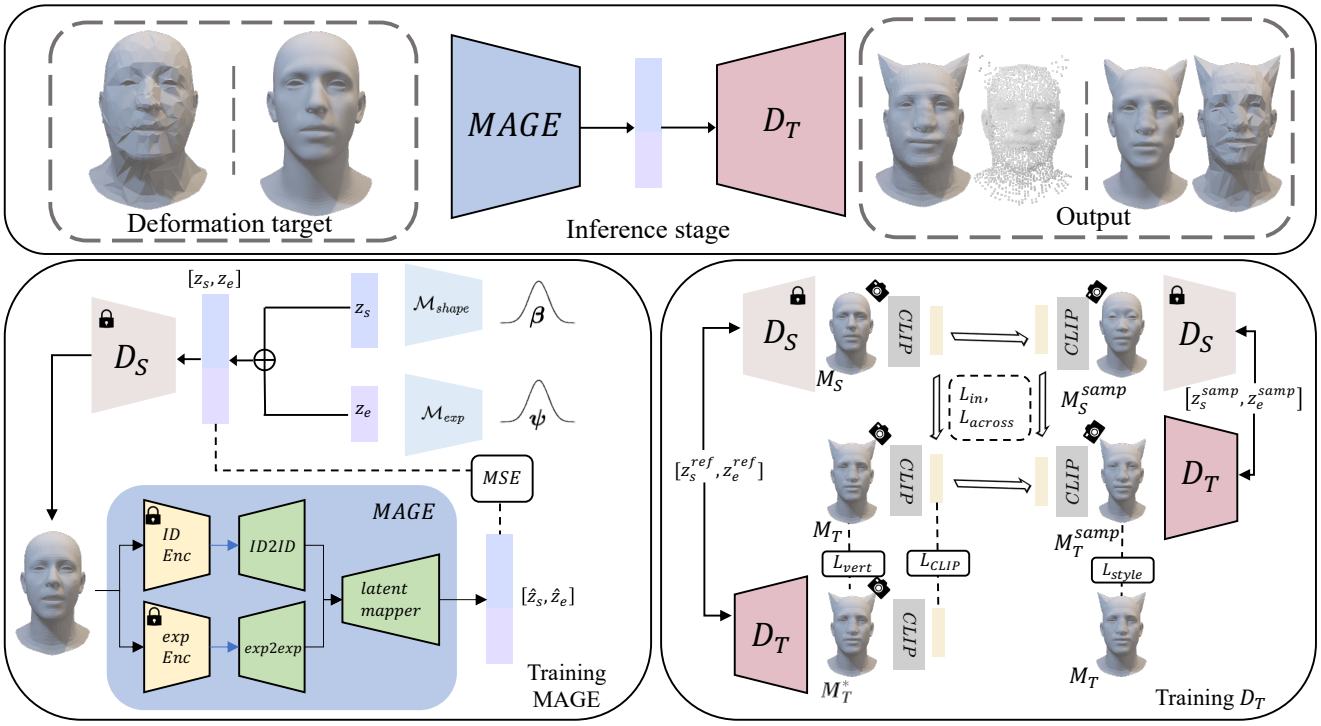


Figure 3. Overview of our method: The upper box illustrates the inference stage, where our method takes diverse deformation targets and generates stylized outputs. In the lower-left box, the training process of Mesh Agnostic Encoder (MAGE) is depicted. In the lower-right box, the fine-tuning process of D_T is illustrated.

stylized 3D faces. An identity exemplar mesh, M_S, is generated from the FLAME decoder using Φ_{ref}, and a style exemplar mesh, M_T, is manually crafted from M_S. During each iteration of fine-tuning, Φ_{sample} is sampled randomly to generate face mesh M_S^{samp} from D_S and corresponding stylized 3D face M_T^{samp} from D_T. In the fine-tuning process, quantities of M_S, M_T, D_S, and D_T are all singular, representing one pair of meshes and one pair of networks.

We introduce a hierarchical rendering scheme designed to preserve semantically important features, such as the shape and facial components, in the face mesh. With a differentiable renderer, this approach captures significant facial features from both local and global perspectives, enhancing stylization fidelity and identity preservation. To further enhance this process, we incorporate 2D-based losses, including CLIP reconstruction loss L_{CLIP}, CLIP in-domain loss L_{in}, CLIP across-domain loss L_{across}, and 3D-based losses such as vertex reconstruction loss L_{vert} and style loss L_{style}. Details of each loss are elaborated in Section 3.3.

3.3. Loss Functions

Vertex Reconstruction Loss The vertex reconstruction loss L_{vert} guides D_T in learning to deform a style exemplar mesh M_T from $[z_s^{ref}, z_e^{ref}]$, which is mapped from Φ_{ref}. The loss utilizes Mean Squared Error (MSE) to ensure that the vertices of the predicted mesh $M_T{}^*$, generated by D_T, closely match M_T. The loss can be written as follows:

$$L_{vert} = \|M_T - M_T{}^*\|_2^2 \tag{3}$$

CLIP Reconstruction Loss The CLIP reconstruction loss L_{CLIP} serves to maintain semantic consistency between the deformed mesh $M_T{}^*$ and M_T in the CLIP space. This is crucial because even minor displacements in 3D space can result in surface irregularities or undesirable shading variations. The CLIP reconstruction loss can be written as follows:

$$L_{\text{CLIP}} = \sum_{l \in L} \sum_{v \in V_l} \|E_C\left(R_{l,v}\left(M_T\right)\right) - E_C\left(R_{l,v}\left(M_T{}^*\right)\right)\|_2^2 \tag{4}$$

$R_{l,v}$ represents the hierarchical rendering of a differentiable renderer from level l and view direction v. Each view is anchored at a predefined position on a face mesh, including the front and both sides of the face. Each level corresponds to the distance from the face to the camera. At the highest level, where the face is captured in close-up, additional images are rendered from significant facial features such as the nose, eyes, and lips. E_C encodes rendered images into CLIP embeddings.

CLIP Directional Loss Inspired from one-shot stylization methods in the 2D domain [48], we incorporate the CLIP in-domain loss L_{in} and the CLIP across-domain loss

L_{across}. L_{in} ensures that the direction of two distinct faces in the source domain remains consistent in two distinct stylized faces of the target style domain. In contrast, L_{across} ensures that the direction of a face from the source domain and its corresponding stylized face in the target domain is preserved across different faces and their corresponding stylized faces. This can be shown in the lower right of Figure 3. Therefore our losses for CLIP in-domain and across-domain can be written as follows:

$$L_{\text{in}} = \sum_{l \in L} \sum_{v \in V_l} \|(E_C(R_{l,v}(M_S^{samp})) - E_C(R_{l,v}(M_S))) - (E_C(R_{l,v}(M_T^{samp})) - E_C(R_{l,v}(M_T)))\|_2^2 \tag{5}$$

$$L_{\text{across}} = \sum_{l \in L} \sum_{v \in V_l} \|(E_C(R_{l,v}(M_T)) - (E_C(R_{l,v}(M_S))) - (E_C(R_{l,v}(M_T^{samp})) - E_C(R_{l,v}(M_S^{samp})))\|_2^2 \tag{6}$$

where M_S^{samp} and M_T^{samp} refer to $D_S[z_s^{samp}; z_e^{samp}]$ and $D_T[z_s^{samp}; z_e^{samp}]$, respectively, where $[z_s^{samp}; z_e^{samp}]$ is the latent vectors mapped from Φ_{sample}.

Style Loss We introduce a novel style loss L_{style} to capture the style of M_T by utilizing surface normals. Style loss compares surface normal, which are strongly correlated with the semantic information of the mesh [5]. A straightforward approach would involve comparing M_T with ${M_T}^{samp}$. However, this will constrain ${M_T}^{samp}$ to have same expression as M_T, which could degrade the animatability. To address this, we construct a pseudo pair for normal calculations. This is done by generating a face mesh through the concatenation of z_e^{ref} from M_T and z_s^{samp}. This approach is utilized to compute the style loss, L_{style} without degrading the animatability while enhancing style adherence. In short, L_{style} ensures alignment of the surface normals from $D_T([z_s^{samp}; z_e^{ref}])$ with M_T. The formulation of the style loss can be written as follows:

$$L_{\text{style}} = \sum_{f \in S} \left(1 - \frac{n_f \cdot {n'}_f}{|n_f||{n'}_f|}\right) \tag{7}$$

where n_f refers to the surface normal from M_T while ${n'}_f$ refers to the corresponding normal from $D_T([z_s^{samp}; z_e^{ref}])$.

The final objective function can be written as follows:

$$L_{total} = \lambda_{vert}L_{vert} + \lambda_{\text{CLIP}}L_{\text{CLIP}} + \lambda_{in}L_{in} + \lambda_{across}L_{across} + \lambda_{style}L_{style} \tag{8}$$

where λ_{vert}, λ_{CLIP}, λ_{in}, λ_{across}, and λ_{style} are the weight for each loss term.

3.4. Mesh Agnostic Encoder

NFR [34] employs DiffusionNet [38] for encoding and decoding face meshes, facilitating face retargeting across different topologies. Given NFR's proficiency in extracting identity and expression details from faces with varying topologies, we extend its encoders to create a MAGE, as depicted in Figure 3. MAGE consists of ID2ID and exp2exp, both being MLPs that receive embeddings from each ID encoder and expression encoder pre-trained in NFR. Finally, given intermediate vectors from both ID2ID and exp2exp, the latent mapper outputs the latent vectors for D_S.

MAGE is trained in a self-supervised manner by randomly sampling β and ψ and passing them through the mapping networks of D_S to obtain $[z_s; z_e]$. Training involves comparing $[\hat{z_s}; \hat{z_e}]$, predicted from MAGE, with their corresponding ground truth latent vectors $[z_s; z_e]$. The objective function of the encoder uses a MSE loss as follows:

$$L_{\text{enc}} = \|[z_s; z_e] - [\hat{z_s}; \hat{z_e}]\|_2^2 \tag{9}$$

where $[\hat{z_s}; \hat{z_e}]$ is MAGE $[D_S([z_s; z_e])]$. With this encoder, we can project a face mesh with diverse topologies into the latent space of D_S.

3.5. Inference

After training, D_T is capable of generating a stylized 3D face mesh with a desired topology from a deformation target mesh that has an arbitrary topology. This is accomplished by first projecting the deformation target into a latent vector, using MAGE. This latent vector is then fed as input to D_T, allowing it to deform the template face of the desired topology into a stylized 3D face mesh. The resulting stylized face preserves both the identity of the deformation target and the desired style while providing animation control.

4. Experiments

4.1. Qualitative Evaluation

We evaluate our method based on three key elements for stylized 3D head avatar creation: Avatar creation in a desired topology, stylization beyond 3DMM, and animation capability with blend shapes. Also, implementation details are provided in supplementary material.

Avatar in a Desired Topology To evaluate the capability of our method in producing a mesh with a desired topology from the deformation target of arbitrary topologies, we first apply a re-meshing technique that includes Loop subdivision [21] and mesh simplification [37] to meshes from FLAME. We also use original meshes from the CoMA [36] and ICT-FaceKit [17] directly as additional diverse meshes. To visually confirm that our method can align different mesh topologies with a desired template face, we enhance the resulting meshes with a checkerboard texture. The visual results, as shown in Figure 4, demonstrate that our method consistently produces stylized 3D faces with the desired topology, regardless of variations in the topology of the deformation target.

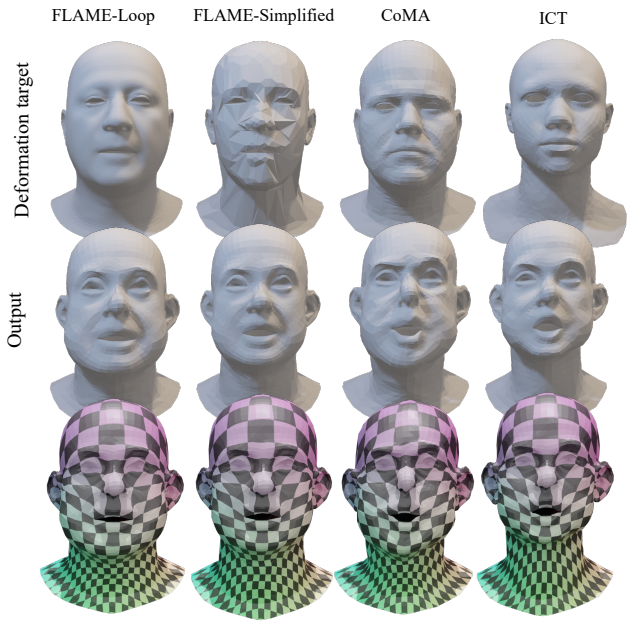


Figure 4. Demonstration of stylized 3D faces with desired topology, regardless of deformation target variations.

Stylization Capability We evaluate our method across various styles to assess its stylization capability. As shown in Figure 5, our method can generate a broad spectrum of styles encompassing human-like and non-human-like faces while preserving the original identities of the given face mesh. Our method also demonstrates the capability to achieve stylization across various geometries, including face masks and point clouds.

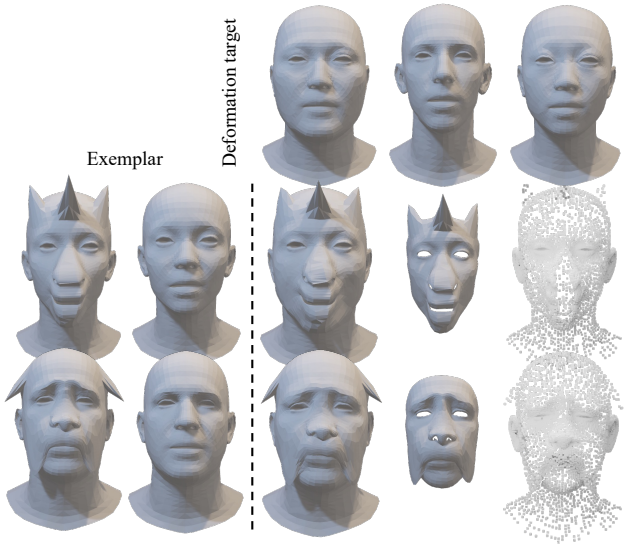


Figure 5. Stylization results across diverse styles and identities. Our approach generates varied styles while preserving the deformation target identity and generalizing to diverse geometric representations like masks and point clouds.

Animation Capability Using parameters from 3DMM, our method can animate facial expressions in the resulting stylized 3D faces. As illustrated in Figure 6, our method can generate 3D stylized faces with various expressions. This allows applications such as video-driven stylized talking heads, which are elaborated in supplementary material.

Figure 6. Visualization of dynamic expressions in stylized 3D faces.

4.2. Quantitative Evaluation

For a quantitative evaluation, we measured the average CLIP style preservation (CLIP-SP) and CLIP identity preservation (CLIP-IP) scores. These metrics reflect the trade-off between preserving style and identity, making their averages crucial for validating stylization. For both metrics, we calculated the cosine similarity of CLIP embeddings from rendered meshes. Specifically, for CLIP-SP, we compared the embeddings of the generated stylized face mesh and style exemplar mesh. For CLIP-IP, we compared the embeddings of the generated stylized face mesh and deformation target mesh. For dataset, we sampled 8 different face mesh from FLAME without expression and manually crafted stylized mesh corresponding to each identities. Also, for the quantitative evaluation, we randomly sampled another 10 different identities without expression to be used as deformation targets.

Comparison with Baselines on Mesh Stylization We compared our stylized face generation results with those produced by baselines that can deform a mesh into different styles: Deformation Transfer [39], TextDeformer [7], and X-mesh [22]. In case of Deformation Transfer [39], we identified the correspondences between the identity exemplar mesh and the deformation target using 68 facial landmarks along with 9 additional points on the forehead in order to apply learned deformation. TextDeformer and X-mesh employ a text-guided deformation that operates on an input mesh. For both methods, the input face is deformed based on the same descriptive text to generate stylized outputs. All these baselines were evaluated on three different

Table 1. Quantitative results are presented for comparison with baselines and ablations, featuring averages of CLIP-SP and CLIP-IP. The highest scores are denoted in bold, while the second highest scores are underlined.

Mesh Type	FLAME			Loop			Simplified			Overall-Average
Methods	CLIP-SP	CLIP-IP	Average	CLIP-SP	CLIP-IP	Average	CLIP-SP	CLIP-IP	Average	Average
Ours	0.9867	0.9347	**0.9607**	0.9833	0.9350	<u>0.9592</u>	0.9559	0.8987	<u>0.9273</u>	**0.9491**
Ours w/o L_{style}	0.9809	0.9374	0.9591	0.9768	0.9339	0.9553	0.9542	0.8987	0.9265	0.9470
Ours w direct L_{style}	0.9865	0.9348	**0.9607**	0.9833	0.9356	**0.9595**	0.9550	0.8973	0.9262	<u>0.9488</u>
w/o hierarchical rendering	0.9846	0.9349	0.9597	0.9824	0.9334	0.9579	0.9569	0.8985	**0.9277**	0.9484
Text deformer	0.8641	0.8363	0.8502	0.8630	0.8425	0.8528	0.8534	0.8173	0.8353	0.8461
X-mesh	0.8626	0.8602	0.8614	0.8817	0.8950	0.8884	0.8635	0.8448	0.8541	0.8680
Deformation transfer	0.9768	0.9376	0.9572	0.9598	0.9576	0.9587	0.9254	0.9288	0.9271	0.9477

mesh topologies: FLAME and FLAME with Loop subdivision and mesh simplification.

We present qualitative comparison results on stylized 3D face generation in Figure 7. Deformation Transfer is capable of generating faces that follow the style; however, it exhibits artifacts, including anomalies and concavities on the surface due to the calculation of displacement and its direct transfer. In contrast, text-based methods fall short in following styles. Our method, on the other hand, can generate stylized 3D faces that adhere to both the input identity and style example without any artifacts.

In quantitative evaluation, shown in Table 1, our method obtained the highest average scores computed from CLIP-SP and CLIP-IP. This clearly demonstrates that our approach achieves better performance in mesh stylization while preserving input identity compared to deformation baselines. More details and examples are provided in the supplementary material.

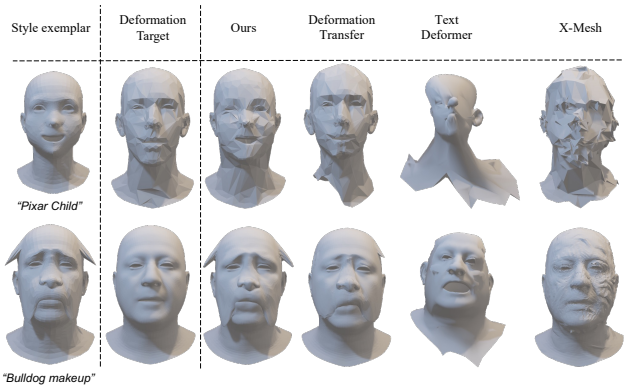


Figure 7. Qualitative comparison on stylization. Our method adheres to style and identity without artifacts unlike Deformation Transfer. Text-based approaches fail to match styles from text.

Surface-Intensive Mesh Sampling To test the proposed sampling method SIMS, we conducted a study on the reconstruction task comparing three different sampling variants for training D_S. SIMS is surface-intensive sampling, using approximately four times more points sampled from the mesh surface compared to mesh vertices. Hybrid sampling, originally proposed in DD3C [13], sampled points both vertices and faces in similar ratio ($\sim$1.1 times). Vertex-only used just the vertices of the mesh.

Table 2. Reconstruction comparison with mesh sampling methods

	Reconstruction loss↓ (all in e-5)				
Methods	original	simplified	loop-1	loop-2	average
SIMS(Ours)	**1.790**	**1.635**	**1.612**	**1.591**	**1.657**
Hybrid	2.300	2.019	2.138	2.118	2.144
Vertex	1.955	5.866	3.258	2.959	3.509

For the experiment, we calculated the reconstruction loss on four different experiment settings: (1) original FLAME, (2) a simplified FLAME mesh with 1/4 the vertex count, (3) FLAME with one loop subdivision, and (4) FLAME with two loop subdivisions. The reconstruction loss was compared to the ground truth position using mean squared error. As shown in Table 2, SIMS achieved the lowest error by a large margin on all experiments. Hybrid sampling generally performed better than vertex-only sampling.

Ablation Study To perform an ablation study, we trained our model D_T using different settings: 1) "Ours w/o L_{style}," where the style loss was removed; 2) "Ours w direct L_{style}," where the style loss was directly compared between $D_T([{z_s}^{samp}; {z_e}^{samp}])$ and M_T; 3) "Ours w/o hierarchical rendering," where a single canonical view was used to calculate the L_{in} and L_{across}. The results are displayed in Figure 8 and Table 1. The results produced by "Ours" and "Ours w direct L_{style}" do not show any artifacts unlike the results produced by w/o L_{style} and w/o hierarchical rendering, which exhibit surface artifacts. However, "Ours w. direct L_{style}" forces $D_T([{z_s}^{samp}; {z_e}^{samp}])$ and M_T, which have different expressions, to have the same normals; thus, it discards the expression. As a result, while "Ours w. direct L_{style}" does stylize effectively, it cannot animate. Conversely, "Ours" not only stylizes well but is also animatable. Additional ablation study results regarding animation are presented in the supplementary material.

User Study We conducted a user study with 34 participants to evaluate different stylized face generation methods on human perception. A total of eight different styles were used. Each participant was presented with 20 questions and asked to choose the result that best followed the style while

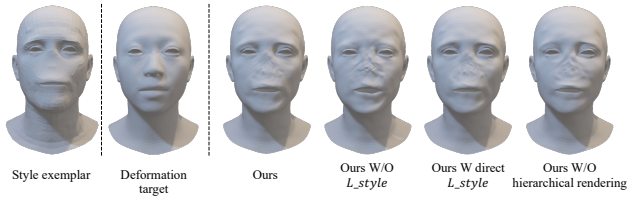


Figure 8. Visualization on ablation study.

preserving the identity. As shown in Table 3, our method received higher scores compared to the baseline methods. Ours outperformed the text-based stylization methods by a large margin and scored higher than the method requiring correspondence specification.

Table 3. Results from the user perceptual study given style exemplar and the deformation target. The percentage represents the selected frequency.

Method	User Score
Ours	**60.65%**
Deformation transfer	38.17%
Text deformer	0.44%
X-mesh	0.74%

5. Applications

5.1. Style Interpolation

LeGO enables linear style interpolation as illustrated in Figure 9, through weight blending. Diverse styles are seamlessly blended to create new stylized meshes by linearly interpolating weights from D_{T1} and D_{T2} using following equation:

$$W_{\text{new}} = \alpha W_{D_{T1}} + (1-\alpha) W_{D_{T2}} \tag{10}$$

Here, $W_{D_{T1}}$ and $W_{D_{T2}}$ denote the network weights for D_{T1} and D_{T2}, respectively, while α represents the blending weight controlling the interpolation.

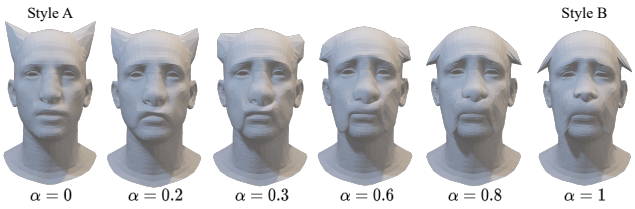


Figure 9. Linear interpolation of Style A and Style B

5.2. Image-based 3D Stylized Avatar Generation

A stylized 3D face can be generated from a single portrait by first using methods that reconstruct 3D faces from 2D portraits. Among these methods, we employed MICA [49] to reconstruct a 3D face from an image. This reconstructed shape was then fed into LeGO to create stylized faces. The results are visualized in Figure 10.

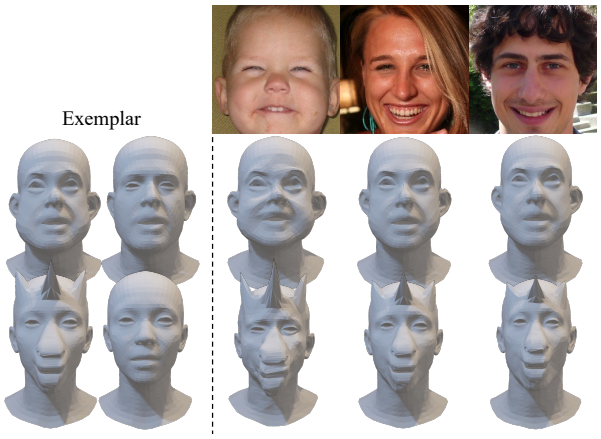


Figure 10. Visualization of generating stylized 3D faces from 2D portraits [14].

6. Limitation and Conclusion

We presented a novel approach for generating stylized 3D face meshes, considering three key elements. First, we proposed a surface deformation network that can generate a face in the desired topology using SIMS. Second, by the domain adaptation with hierarchical rendering, we achieved superior stylization capability. Lastly, Using 3DMM prior, we can generate a stylized face equipped with the animation capability.

In addition, we proposed MAGE for practical usage, which can take diverse mesh topologies as input, and a novel style loss that adheres to the style effectively while preserving animation ability. Comprehensive experimental results demonstrate that our method is capable of generating a stylized mesh with consistent topology given deformation target meshes exhibiting significant topological variation.

While our method shows promising results and enables significant advancements in practical avatar creation, it also has challenges to address. At inference, achieving a stylized output with the same topology as the input requires a two-stage process involving template replacement with the mean face mesh. Addressing these efficiency and practicality challenges is crucial for further enhancing our approach.

Acknowledgement This research was supported by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2023 (Project Name: Development of Universal Fashion Creation Platform Technology for Avatar Personality Expression, Project Number: RS-2023-00228331, Contribution Rate: 90%) and Korea Institute for Advancement of Technology(KIAT) grant funded by the Korea Government(MOTIE) (Project Name: The Competency Development Program for Industry Specialist, Project Number: P0012746, Contribution Rate: 10%).

References

[1] Rameen Abdal, Hsin-Ying Lee, Peihao Zhu, Menglei Chai, Aliaksandr Siarohin, Peter Wonka, and Sergey Tulyakov. 3davatargan: Bridging domains for personalized editable avatars. *arXiv preprint arXiv:2301.02700*, 2023. 2

[2] Shivangi Aneja, Justus Thies, Angela Dai, and Matthias Nießner. Clipface: Text-guided editing of textured 3d morphable models. *arXiv preprint arXiv:2212.01406*, 2022. 2

[3] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. 2

[4] Min Jin Chong and David Forsyth. Jojogan: One shot face stylization. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVI*, pages 128–152. Springer, 2022. 3

[5] 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, 5

[6] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022. 3

[7] William Gao, Noam Aigerman, Thibault Groueix, Vladimir G Kim, and Rana Hanocka. Textdeformer: Geometry manipulation using text guidance. *arXiv preprint arXiv:2304.13348*, 2023. 2, 3, 6

[8] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020. 2

[9] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018. 3

[10] 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. 2

[11] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: a network with an edge. *ACM Transactions on Graphics (ToG)*, 38(4):1–12, 2019. 3

[12] Wonjoon Jin, Nuri Ryu, Geonung Kim, Seung-Hwan Baek, and Sunghyun Cho. Dr. 3d: Adapting 3d gans to artistic drawings. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–8, 2022. 2, 3

[13] Yucheol Jung, Wonjong Jang, Soongjin Kim, Jiaolong Yang, Xin Tong, and Seungyong Lee. Deep deformable 3d caricatures with learned shape control. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 2, 7

[14] 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. 8

[15] Gwanghyun Kim and Se Young Chun. Datid-3d: Diversity-preserved domain adaptation using text-to-image diffusion for 3d generative model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14203–14213, 2023. 2

[16] Seongtae Kim, Kyoungkook Kang, Geonung Kim, Seung-Hwan Baek, and Sunghyun Cho. Dynagan: Dynamic few-shot adaptation of gans to multiple domains. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–8, 2022. 2, 3

[17] Ruilong Li, Karl Bladin, Yajie Zhao, Chinmay Chinara, Owen Ingraham, Pengda Xiang, Xinglei Ren, Pratusha Prasad, Bipin Kishore, Jun Xing, et al. Learning formation of physically-based face attributes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3410–3419, 2020. 5

[18] Shaoxu Li. Instruct-video2avatar: Video-to-avatar generation with instructions. *arXiv preprint arXiv:2306.02903*, 2023. 2

[19] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6):194–1, 2017. 2

[20] Yijun Li, Richard Zhang, Jingwan Lu, and Eli Shechtman. Few-shot image generation with elastic weight consolidation. *arXiv preprint arXiv:2012.02780*, 2020. 3

[21] Charles Loop. Smooth subdivision surfaces based on triangles. 1987. 5

[22] Yiwei Ma, Xiaoqing Zhang, Xiaoshuai Sun, Jiayi Ji, Haowei Wang, Guannan Jiang, Weilin Zhuang, and Rongrong Ji. X-mesh: Towards fast and accurate text-driven 3d stylization via dynamic textual guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2749–2760, 2023. 2, 6

[23] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 2

[24] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13492–13502, 2022. 2, 3

[25] Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Tiberiu Popa. Clip-mesh: Generating textured meshes from text using pretrained image-text models. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–8, 2022. 3

[26] Thu Nguyen-Phuoc, Gabriel Schwartz, Yuting Ye, Stephen Lombardi, and Lei Xiao. Alteredavatar: Stylizing dynamic 3d avatars with fast style adaptation. *arXiv preprint arXiv:2305.19245*, 2023. 2

[27] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5379–5389, 2019. 2

[28] Atsuhiro Noguchi and Tatsuya Harada. Image generation from small datasets via batch statistics adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2750–2758, 2019. 3

[29] Utkarsh Ojha, Yijun Li, Jingwan Lu, Alexei A Efros, Yong Jae Lee, Eli Shechtman, and Richard Zhang. Few-shot image generation via cross-domain correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10743–10752, 2021. 3

[30] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 2

[31] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer, 2020. 2

[32] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 3

[33] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 3

[34] Dafei Qin, Jun Saito, Noam Aigerman, Thibault Groueix, and Taku Komura. Neural face rigging for animating and retargeting facial meshes in the wild. *arXiv preprint arXiv:2305.08296*, 2023. 2, 3, 5

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 3

[36] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3d faces using convolutional mesh autoencoders. In *Proceedings of the European conference on computer vision (ECCV)*, pages 704–720, 2018. 5

[37] William J Schroeder, Jonathan A Zarge, and William E Lorensen. Decimation of triangle meshes. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 65–70, 1992. 5

[38] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. Diffusionnet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics (TOG)*, 41(3): 1–16, 2022. 3, 5

[39] Robert W Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM Transactions on graphics (TOG)*, 23(3):399–405, 2004. 3, 6

[40] Minhyuk Sung, Zhenyu Jiang, Panos Achlioptas, Niloy J Mitra, and Leonidas J Guibas. Deformsyncnet: Deformation transfer via synchronized shape deformation spaces. *arXiv preprint arXiv:2009.01456*, 2020. 3

[41] Yaxing Wang, Abel Gonzalez-Garcia, David Berga, Luis Herranz, Fahad Shahbaz Khan, and Joost van de Weijer. Minegan: effective knowledge transfer from gans to target domains with few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9332–9341, 2020. 3

[42] Jiayu Xiao, Liang Li, Chaofei Wang, Zheng-Jun Zha, and Qingming Huang. Few shot generative model adaption via relaxed spatial structural alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11204–11213, 2022. 3

[43] Xirui Yan, Zhenbo Yu, Bingbing Ni, and Hang Wang. Cross-species 3d face morphing via alignment-aware controller. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3018–3026, 2022. 2, 3

[44] Shuai Yang, Liming Jiang, Ziwei Liu, and Chen Change Loy. Pastiche master: Exemplar-based high-resolution portrait style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7693–7702, 2022. 3

[45] Wang Yifan, Noam Aigerman, Vladimir G Kim, Siddhartha Chaudhuri, and Olga Sorkine-Hornung. Neural cages for detail-preserving 3d deformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 75–83, 2020. 3

[46] Chi Zhang, Yiwen Chen, Yijun Fu, Zhenglin Zhou, Gang Yu, Billzb Wang, Bin Fu, Tao Chen, Guosheng Lin, and Chunhua Shen. Styleavatar3d: Leveraging image-text diffusion models for high-fidelity 3d avatar generation. *arXiv preprint arXiv:2305.19012*, 2023. 2

[47] Longwen Zhang, Qiwei Qiu, Hongyang Lin, Qixuan Zhang, Cheng Shi, Wei Yang, Ye Shi, Sibei Yang, Lan Xu, and Jingyi Yu. Dreamface: Progressive generation of animatable 3d faces under text guidance. *arXiv preprint arXiv:2304.03117*, 2023. 2

[48] Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Mind the gap: Domain gap control for single shot domain adaptation for generative adversarial networks. In *International Conference on Learning Representations*, 2022. 2, 3, 4

[49] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Towards metrical reconstruction of human faces. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIII*, pages 250–269. Springer, 2022. 8

LeGO: Leveraging a Surface Deformation Network for Animatable Stylized Face Generation with One Example

Supplementary Material

Overview

In this supplementary material, we present implementation details in Section A. Section B contains the details of the LeGO architecture. Section C covers additional experiments and their corresponding details. Section D is dedicated to visualizing example results of the applications of our method. Lastly, Section E presents additional results. For additional results, please visit the project page https://kwanyun.github.io/lego/.

A. Implementation Details

The implementation of LeGO including the surface deformation networks D_S and D_T, and MAGE on an Nvidia RTX 3090 GPU. The implementation details are sequentially presented, in the order of with D_S, D_T, MAGE, and the baselines for the experiments in the following paragraphs.

Training D_S To deform a template face to a person with a different identity and expression, we trained the source surface deformation network D_S using the FLAME [3] parameter Φ and its corresponding face mesh, as described in the main paper. We sampled 100k instances of Φ and their corresponding faces for self-supervised training, with Φ being sampled from a uniform distribution to learn diverse identities and expressions. We jointly trained $\mathcal{M}_{shape}$, $\mathcal{M}_{exp}$, and D_S for 400 epochs with a fixed learning rate of 1e-6. We adopted the SIMS approach to enable D_S to handle diverse topologies by training it using surface points instead of only sampling from mesh vertices.

Training D_T To modify the template face to incorporate styles while maintaining the same identity and expression as D_S, we trained the target surface deformation network D_T following the procedure outlined in the main paper. The training began with an initial learning rate of 3e-5, which gradually decreased to 1e-5 over 2,000 iterations. The balancing weights in Equation (9) in the main paper, λ_{vert},λ_{CLIP}, λ_{in}, λ_{across}, and λ_{style} were fixed at 80, 2e-3, 6e-3, 6e-3, and 4e-3, respectively.

During the training of D_T, we adopted a hierarchical rendering approach comprising three levels. The first level, featuring the most enlarged views, focused on rendering local facial parts such as the eyes, nose, and lips (illustrated in the blue box in Figure 1). The second level includes close-up views of faces from three directions, encompassing the front and sides of the face. The last level comprises full-face views from the same directions.

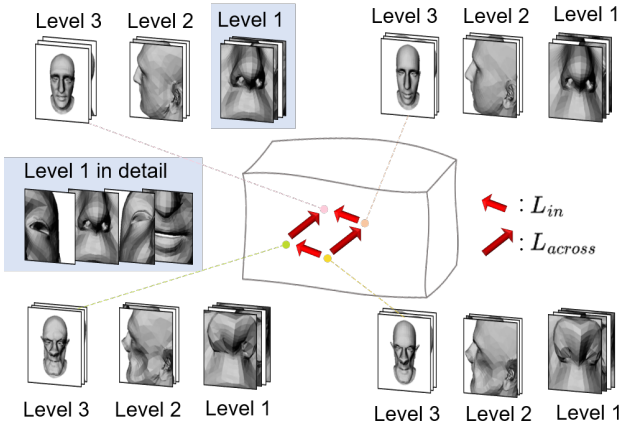


Figure 1. Detailed illustration of L_{in} and L_{across} with hierarchical rendering. The blue box shows an example of the predefined pivots of hierarchical rendering.

Training MAGE MAGE functions as an encoder that transforms faces with diverse topologies into the latent space of LeGO, specifically into the shared latent space of D_S and D_T. NFR [5] encoders were fixed during training, while ID2ID, exp2exp, and latent mapper were jointly trained. The model was trained with an initial learning rate of 3e-4, which gradually decreased to 5e-5 over 12,000 iterations.

Figure 2. First two rows are identity exemplar mesh and style exemplar mesh in our dataset, created for fine-tuning. Last row shows deformation target meshes for experiments

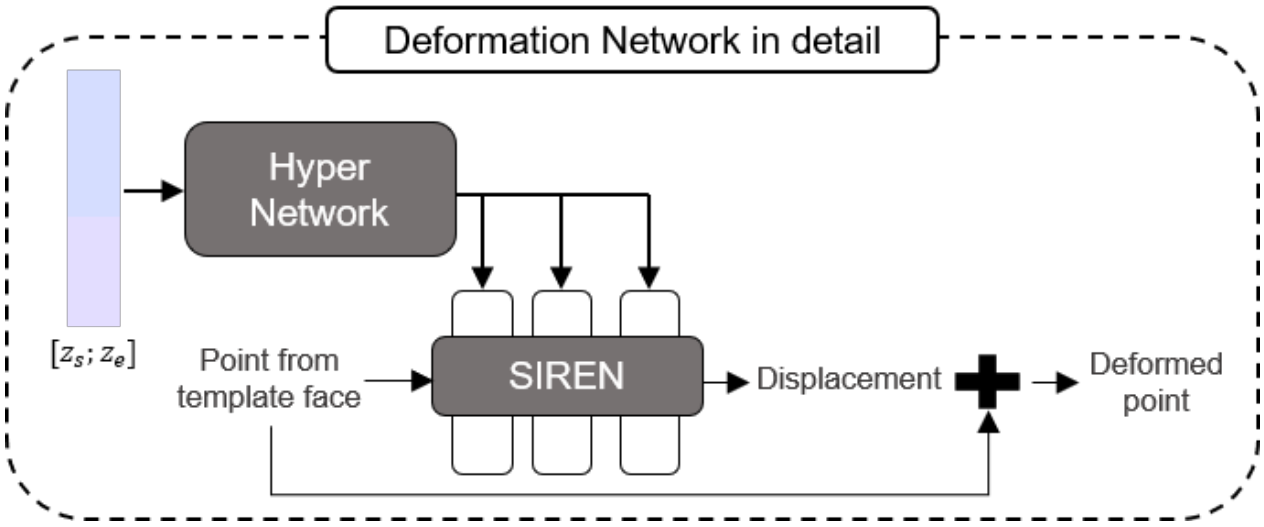


Figure 3. Visualization of deformation network in detail.

Baseline Methods All baseline methods were using their default settings as specified in their respective papers or the official code provided by the authors. As specified in the main paper, we utilized 8 different faces with corresponding manually crafted meshes as a dataset, which are shown in Figure 2. We also randomly sampled another 10 different identities without expressions from FLAME decoder as deformation targets for experiments. For the text-based methods [1, 4], we used the deformation target as the source template mesh and text to specify the target style. Eight text prompts that describe styles are as follows: "Bulldog makeup", "Disney Dwarf", "Exaggerated smile", "Musical Cats", "Orc", "Person with unicorn horn", "Person without nose", and "Pixar child".

B. LeGO Architecture

B.1 Deformation Network Architecure

The architectures of D_S and D_T are designed to compute the displacement of a point, either from a vertex or surface, and produce a deformed face output. The architectures are inspired by DD3C [2]. The architecture of the deformation network is illustrated in Figure 3. The latent code $[z_s;z_e]$ enters the hypernetwork, modifying the parameters of the SIREN MLP [6]. Subsequently, as the point from the template face traverses the network, the displacement is added to the point, determining the output position.

B.2 Rationale Behind Using the Deformation Network

The main reason to utilize a deformation network instead of simply adding an displacement lies in the inability of handling diverse inputs and outputs when the simple method is used. Another reason is to avoid severe artifacts that may occur when the identity of the deformation target and identity exemplar mesh are too different to directly transfer the displacement from one face to another. Examples of these artifacts are illustrated in Figure 4.

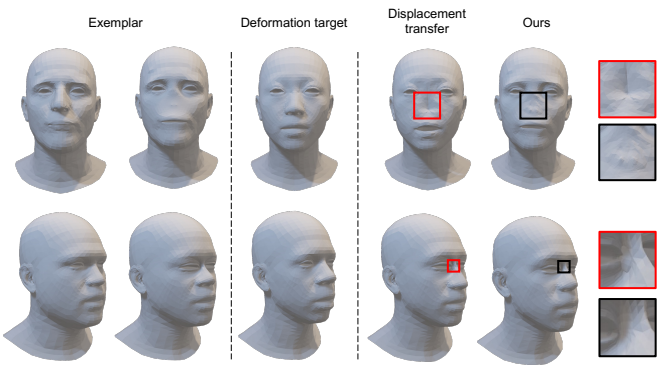


Figure 4. Artifacts occurred by simple displacement transfer. The red boxes show the close-up views. The boxes colored in red show the result of simple dispacement with artifacts(sunk nose and penetration on eye region) unlike Ours, colored in black.

C. Additional Experiments

C.1 Comparison with NFR

NFR [5] is a method that can transfer the expression of a target facial mesh to an unrigged identity mesh of arbitrary topology. Because NFR is specifically designed for expression transfer, it is difficult for the method to preserve both identity and style in the resulting mesh. However, because NFR is the backbone of MAGE, we compared it with our

Figure 5. Comparison with NFR and LEGO. Becaus NFR is designed for expression transfer, it's expression encoder cannot preserve identity.

method. As shown in Figure 5, although NFR could generate a stylized mesh, it failed to preserve the original identity, resulting in all similar outcomes that reflect the style exemplar.

C.2 Ablation on Direct Style Loss

As stated in the main paper, "Ours" and "Ours w. direct L_{style}" loss produced the least amount of surface artifacts. Here, the style loss that directly compared $D_T([{z_s}^{samp}; {z_e}^{samp}])$ and M_T, it forced the stylized face to have the same expression as M_T. In contrast, ours that compared $D_T([{z_s}^{samp}; {z_e}^{ref}])$ and M_T successfully maintained animatability. This is illustrated in Figure 6.

D. Applications

D.1 Visualization of Results Produced by the Applications.

We present additional results of style interpolation in Figure 7. These results demonstrate that our method can effectively construct the latent space for identities and styles, ensuring that even when mixing weights, the person's identity remains unchanged while the styles transition smoothly. This finding inspired the generation of new styles by blending existing ones. Additionally, we showcase further results on generating stylized 3D faces from 2D portraits, indicating that our method does not require a mesh as input; instead, any image can be used to create a stylized face with a specific identity, thereby broadening the practical application of our method.

D.2 Retargeting from Video

Retargeting is one of widely used applications in animation in which target follows the animation of the source. We performed an additional experiment on video driven stylized 3D face retargeting. Using a metrical photometric tracker [9], we can obtain the shape and expression parameters of FLAME from video, which can be directly adopted to LeGO. From these parameters, we achieved 3D stylized face retargeting as shown in Figure 8. Additional results are shown in the supplementary video.

E. Additional Results

We present additional stylization results produced by LeGO trained with a paired exemplar. Figures 9, 10 and 11 display the results of all eight styles and deformation targets with different topologies.

References

[1] William Gao, Noam Aigerman, Thibault Groueix, Vladimir G Kim, and Rana Hanocka. Textdeformer: Geometry manipulation using text guidance. *arXiv preprint arXiv:2304.13348*, 2023. 2

[2] Yucheol Jung, Wonjong Jang, Soongjin Kim, Jiaolong Yang, Xin Tong, and Seungyong Lee. Deep deformable 3d caricatures with learned shape control. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 2

[3] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6):194–1, 2017. 1

[4] Yiwei Ma, Xiaoqing Zhang, Xiaoshuai Sun, Jiayi Ji, Haowei Wang, Guannan Jiang, Weilin Zhuang, and Rongrong Ji. X-mesh: Towards fast and accurate text-driven 3d stylization via dynamic textual guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2749–2760, 2023. 2

[5] Dafei Qin, Jun Saito, Noam Aigerman, Thibault Groueix, and Taku Komura. Neural face rigging for animating and retargeting facial meshes in the wild. *arXiv preprint arXiv:2305.08296*, 2023. 1, 2

Figure 6. Comparison with Ours and Ours w. direct L_{style}. Both methods generated the style well while Ours followed the expression from the deformation target better compared to Ours w. direct L_{style}.

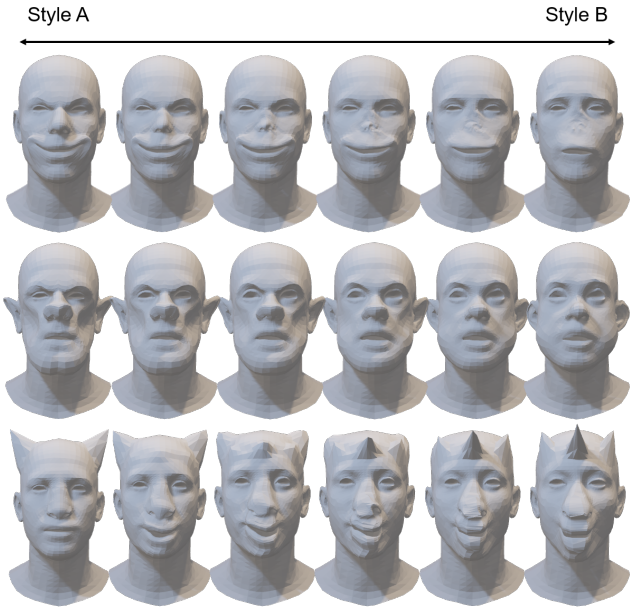


Figure 7. Additional results of style interpolation.

[6] 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. 2

[7] Kaisiyuan Wang, Qianyi Wu, Linsen Song, Zhuoqian Yang, Wayne Wu, Chen Qian, Ran He, Yu Qiao, and Chen Change Loy. Mead: A large-scale audio-visual dataset for emotional talking-face generation. In *European Conference on Computer Vision*, pages 700–717. Springer, 2020. 5

[8] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *CVPR*, 2021. 5

[9] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Towards metrical reconstruction of human faces. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIII*, pages 250–269. Springer, 2022. 3

Figure 8. Retargeting from video, two different styles are shown for each input. Left examples are from Talking-head-1KH [8] and right examples are from MEAD dataset [7].

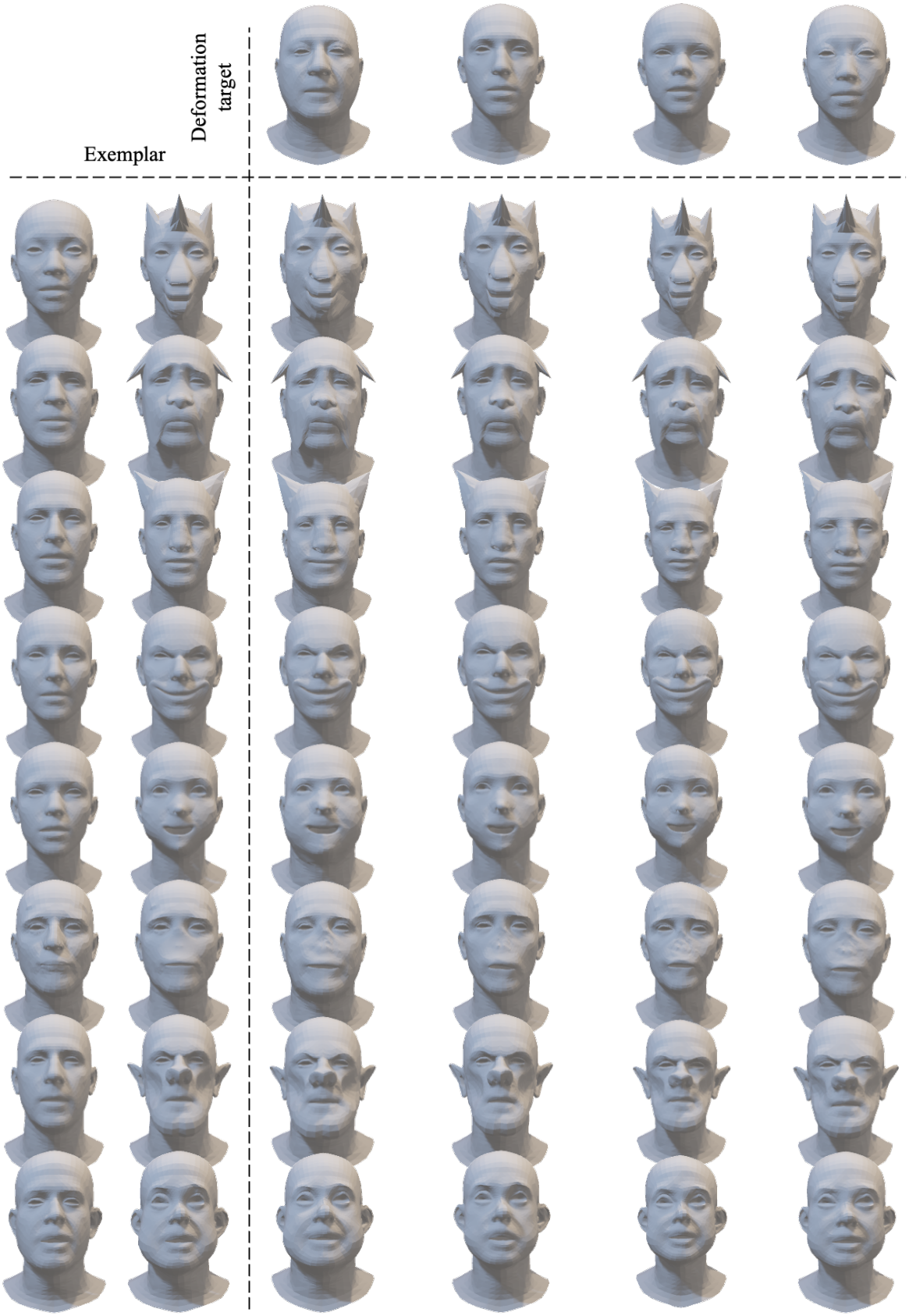


Figure 9. Additional results on all 8 different styles.

Figure 10. Additional results on all 8 different styles.

Figure 11. Additional results on all 8 different styles.