FACEGID: FACIAL MESH EXPRESSION RETARGETING WITH GEOMETRY-IMAGE DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

3D facial expression retargeting remains challenging due to the irregular structure of mesh data, the scarcity of large-scale 3D facial datasets, and the limitations of relying on fixed templates or reference meshes. In this work, we formulate this task as a 2D conditional generation problem using geometry images as the 3D representation. This allows us to fine-tune a pretrained 2D image generative model on geometry images and leverage priors learned from large-scale 2D data for 3D generation. Given a source facial mesh and a single reference image, our method transfers the reference expression while preserving the source identity and capturing expression-related appearance changes. We further introduce a training-free guidance strategy that enables controllable expression strength at inference time. Experiments demonstrate effective expression retargeting and good generalization across different facial meshes and reference images.

1 INTRODUCTION

Human facial expression retargeting aims to transfer the expression of a reference image to a source 3D face while preserving the identity-specific characteristics of the source subject. It is an essential topic in computer vision and graphics, with broad applications in virtual avatars, gaming and film production. Although recent generative models have substantially improved the realism and controllability of 2D facial expression retargeting (Ding et al., 2023), extending the advances to 3D remains difficult. Unlike 2D images, 3D meshes involve higher-dimensional data and more complex structural information. Moreover, due to privacy concerns and the practical difficulty of 3D facial data acquisition, publicly available facial mesh datasets are much smaller in scale than 2D face image datasets, making it difficult to support data-hungry learning-based approaches.

A common strategy for 3D facial expression retargeting relies on a predefined facial template. Given a raw 3D facial scan, these methods typically fit a 3D morphable model (3DMM) (Blanz & Vetter, 1999) to the input geometry and then manipulate the fitted representation using the expression blendshapes, provided by the 3DMM or predefined expression bases, such as those derived from the Facial Action Coding System (FACS) (Ekman & Friesen, 1978). By performing expression manipulation in a shared canonical space, template-based methods largely bypass the difficulties caused by heterogeneous mesh connectivity, resolution, and geometric complexity. However, this convenience comes at several costs. Mapping a high-resolution facial scan into a predefined parametric space can inevitably introduce approximation errors during registration and fitting, potentially limiting the preservation of fine-scale facial geometry and identity-specific details. Although modern parametric face models have substantially improved their ability to represent detailed geometry and texture (Wang et al., 2022), their outputs remain constrained by the adopted representation space. Besides, these methods rely on predefined expression representations, and therefore, their expressive capacity is inherently bounded by the adopted expression space, making it difficult to reproduce expressions that are poorly represented or absent from the predefined basis. Moreover, these methods often provide limited control over the intensity of expression changes. More recent approaches perform expression retargeting by directly deforming the source surface and substantially relax the dependence on fixed triangulations (Qin et al., 2023; Ma et al., 2025). However, these methods are primarily designed for geometric deformation or rigging with 3D or FACS-style controls, rather than transferring expression directly from a reference RGB image, and they generally do not synthesize corresponding expression-dependent textures.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To address these challenges and limitations, we investigate whether 3D facial expression retargeting can instead be formulated as conditional generation over 2D image-like representation. Our key insight is that a geometry image (Gu et al., 2002) can serve as a bridge between complex meshes and 2D image generative models. It represents surface geometry and texture in a 2D pixel space, while retaining a mapping to the original 3D surface. This formulation allows the model to exploit the generative representations learned by large image diffusion models, reducing its dependence on large-scale 3D training. Moreover, compared with 3DMM-based representations, geometry images can preserve finer geometric details without being constrained to a fixed mesh triangulation.

Based on this formulation, we propose a reference-driven framework FaceGID that jointly retargets facial geometry and synthesized expression-consistent texture while preserving source identity. There are three contributions. First, we reformulate facial mesh expression retargeting as conditional image generation over geometry images, providing a structured representation that connects heterogeneous facial meshes with pretrained image diffusion models. Second, we introduce a finetuning strategy that adapts a pretrained latent diffusion model to geometry-image data. Third, we propose a training-free guidance strategy that enhances expression consistency and provides flexible control over the intensity of the transferred expression. Our proposed method has three main properties. First, it provides flexibility by eliminating the need for input facial meshes to conform to a fixed triangulation, thereby supporting different vertex counts and connectivity patterns. The target expression can also be specified using a single reference image, without requiring a 3D facial mesh. Second, the expression retargeting intensity is controllable through a tunable guidance parameter. Finally, FaceGID achieves high-quality retargeting by preserving fine geometric details and generating textures that are consistent with the retargeted facial expressions. We evaluate our method on both topologically consistent unified meshes and raw facial scans. Experimental results show that our approach outperforms the compared methods in both expression consistency and identity preservation.

2 Related Work

Facial Action Coding System and 3DMM Early facial animation systems (Cosker et al., 2011) commonly relied on manually defined semantic controls, particularly the Facial Action Coding System (FACS), which decomposes facial motion into a set of Action Units (AUs). Subsequent approaches learned statistical parametric representations of facial identity and expression from 3D face data (Cao et al., 2014; Jiang et al., 2019; Yang et al., 2020; Li et al., 2017). By embedding facial variations into a shared low-dimensional space, these methods enable identity and expression to be modeled and manipulated separately. However, constructing such representations generally requires dense correspondence among training meshes, and their outputs are consequently constrained to a predefined vertex count and triangulation.

Deformation-based Methods Another line of research directly models deformation on facial surfaces. Classical deformation-transfer methods (Sumner & Popović, 2004) transfer local transformations between source and target meshes, while recent neural methods learn deformation operators directly from data. Qin et al. (2023) introduces a deformation autoencoder that supports facial meshes with varying connectivity and does not require manually constructed blendshapes or explicit correspondence. Ma et al. (2025) further employs a triangulation-agnostic surface network and extends neural facial rigging to meshes with diverse triangulations and multiple disconnected components. Although these approaches substantially relax the dependence on a fixed source template, their expression inputs are still provided through a 3D facial mesh rather than an image, and they primarily focus on geometric deformation without synthesizing corresponding expression-dependent textures.

Image-driven Methods These methods estimate or disentangle identity and expression information directly from monocular facial images. CPEM (Mo et al., 2022) explicitly introduces identity-consistent and expression-exclusive constraints for facial motion retargeting from 2D images to 3D faces. DECA (Feng et al., 2021) disentangles person-specific facial details from expression-dependent wrinkles to enable animatable detailed face reconstruction, while EMOCA (Daněček et al., 2022) further improves the reconstruction of emotionally consistent expressions from in-the-wild images. More recently, SMIRK (Retsinas et al., 2024) adopts an analysis-by-neural-synthesis framework to better capture subtle, asymmetric, and extreme expressions. Although these meth-

ods allow expression information to be inferred from a single image, the resulting geometry is still represented within a predefined parametric face model, which restricts the output to its fixed vertex correspondence and triangulation.

Geometry image Geometry image (Gu et al., 2002) is an efficient 3D representation that compresses 3D meshes into 2D images with low quality loss while greatly reducing storage requirements. Recently, geometry images have been increasingly adopted in 3D generation tasks (Alhaija et al., 2022; Elizarov et al., 2025; Yan et al., 2025; Zhang et al., 2026). These methods use geometry images as the 3D representation and train or fine-tune existing image generative models for 3D generation. However, most of them are conditioned on semantic categories or textual descriptions rather than reference images. Moreover, they do not address facial expression retargeting or other similar 3D editing tasks.

3 METHODOLOGY

The objective of the 3D facial retargeting problem is as follows: given a source 3D facial mesh $M_{i_s}^{e_s}$ with identity i_s and expression e_s, a reference image $I_{i_r}^{e_r}$ with identity i_r and expression e_r, we want to generate a facial mesh $M_{i_s}^{e_r}$ with source identity i_s and reference expression e_r,

$$(M_{i_s}^{e_s}, I_{i_r}^{e_r}) \rightarrow M_{i_s}^{e_r}.$$

In this section, we elaborate on the whole pipeline of FaceGID, including geometry image generation, fine-tuning the conditional image generative model, and deploying the training-free guidance during inference.

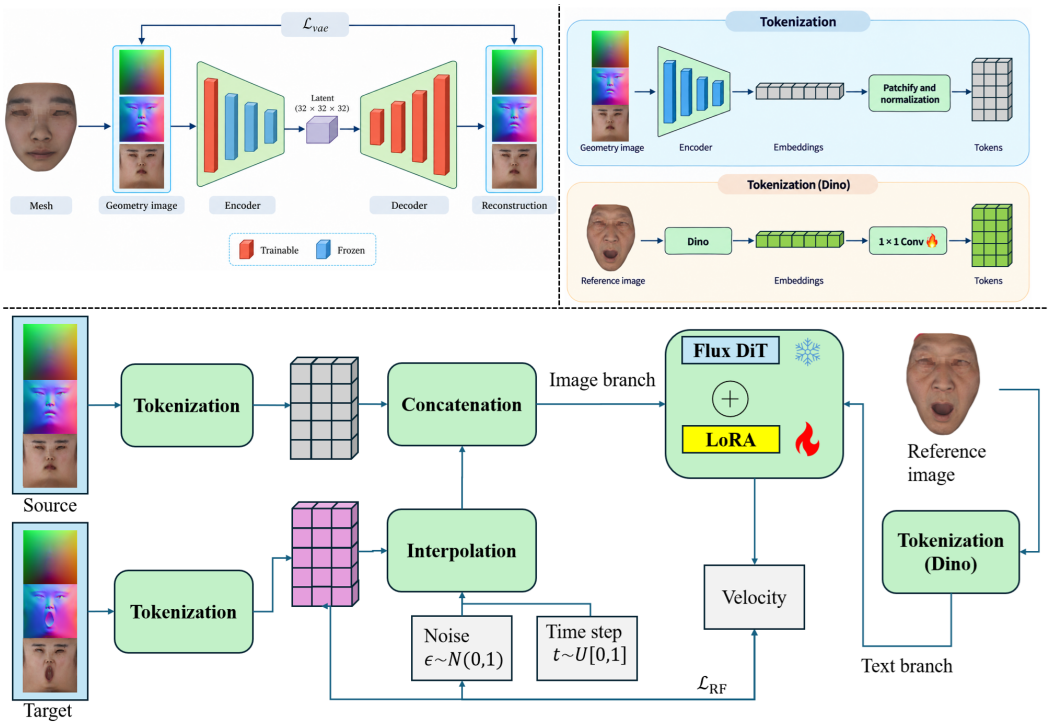

Figure 1: **Training pipeline**. Face meshes are converted into geometry images and encoded by the VAE. After patchification and normalization, each latent is represented as 256 tokens of dimension 128. Tokens from the same identity with different expressions are concatenated and fed into the Transformer image branch. Reference images are encoded by DINO, projected through a 1×1 convolution, and fed into the text branch. The Transformer is frozen during training, and only LoRA parameters and the projection layer are optimized.

3.1 Geometry Image Preparation and Reconstruction

To generate geometry images for a given mesh, we first compute the parameterization which maps the surface to a square domain in the UV space using Mean Value Coordinates (Floater, 2003). After obtaining the parameterization, we assign normalized Euclidean coordinates, the normal direction and the texture color to each vertex RGB respectively, then render the planar mesh in the UV domain to generate a position image, a normal image and a texture image each at a resolution of $3 \times 256 \times 256$ (Figure 2). These three images can represent the original 3D textured mesh and will be used for training a conditional generative model (Details can be found in Appendix A.4).

The generated geometry images can be directly converted back into 3D head models. By mapping the RGB values of each pixel in the position image to the Euclidean space, every pixel corresponds to a vertex of the reconstructed mesh. The connectivity is then recovered by linking the diagonals of the quadrilaterals formed by four neighboring pixels, yielding the edges and triangular faces of the mesh. To mitigate angle distortion, we connect the shorter diagonal for each quadrilateral. Since geometry images inherently encode surface parameterization, the reconstruction process also enables straightforward generation of textured meshes.

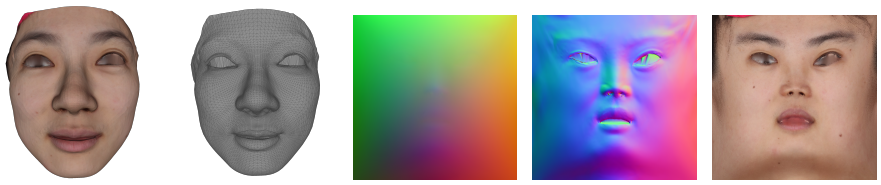

Figure 2: Visualization of a facial mesh and the corresponding geometry images.

3.2 Conditional Generation

Modern latent generative models commonly follow a two-stage paradigm, where a Variational Autoencoder (VAE) (Kingma & Welling, 2014) first compresses images into a compact latent space, and a diffusion- or flow-based model is subsequently trained to model the latent distribution. However, adapting such a pretrained framework to geometry images is non-trivial. General pretrained large image-generation models are designed for three-channel RGB images, whereas geometry images consist of nine channels spanning three different modalities. Moreover, the compression stage and the generative stage are tightly coupled through the latent representation. Due to the substantial domain gap between general images and geometry images, the pretrained VAE cannot faithfully reconstruct the geometric signals encoded in our inputs and therefore requires domain-specific adaptation (Please refer to Appendix A.5 for comparisons). In addition, the VAE cannot be fine-tuned solely for reconstruction quality. Since the second-stage Transformer is pretrained on the latent space produced by the original VAE, a substantial change in the latent distribution after VAE fine-tuning would create an additional domain gap in the latent space and weaken the benefit of the pretrained generative prior, making the subsequent adaptation of the Transformer substantially more difficult. In our work, we adopt FLUX.2 (Labs, 2025), a text-guided image generative model which employs a Transformer-based rectified-flow model (Liu et al., 2023), as our generative backbone (Please refer to Section A for details). To mitigate the above issues, we adapt the VAE to geometry images while constraining the shift of its latent representations from those produced by the pretrained encoder, then fine-tune the Transformer on the adapted geometry-image latent space. Details of model structure and implementation setup can be found in Appendix A.1- A.2.

3.2.1 Domain Adaptation for VAE

To adapt the pretrained VAE to our geometry-image representation, we modify its input and output layers from three channels to nine channels, corresponding to the position, normal, and texture maps, while preserving the internal architecture and the 32-channel latent space. Also, to minimize the distribution shift between geometry-image latent and the latent space learned by the pretrained model,

thereby facilitating subsequent transformer fine-tuning, we fine-tune only the input and output layers and the decoder, while freezing the remaining VAE parameters. The VAE is trained using a weighted combination of the L1 reconstruction loss and the KL divergence, as shown in Equation 1.

$$\mathcal{L}_{vae} = \lambda_1 \mathcal{L}_1^{pos} + \lambda_2 \mathcal{L}_1^{normal} + \lambda_3 \mathcal{L}_1^{texture} + \lambda_4 \mathcal{L}_{KL} \tag{1}$$

After finishing training, a geometry image of size $9 \times 256 \times 256$ is encoded by the VAE into a latent representation $\mathbf{z}$ of size $32 \times 32 \times 32$. The latent is then patchified using non-overlapping 2×2 windows, normalized with scaling factors given by FLUX.2, and reshaped into 256 tokens with a feature dimension of 128.

3.2.2 Transformer Fine-tuning

In FaceGID, the inputs are a pair of geometry image tokens $T_{i_s}^{e_s}$ and $T_{i_s}^{e_r}$, which share the same identity i_s but have different expressions e_s and e_r, and a reference human image $I_{i_r}^{e_r}$ that has the expression e_r and a different identity i_r from i_s. The latent tokens $T_{i_s}^{e_s}$ and $T_{i_s}^{e_r}$ are concatenated and fed into the transformer through the image branch. Meanwhile, the reference image $I_{i_r}^{e_r}$ is encoded by DINOv2 (Oquab et al., 2024) to extract 768-dimensional visual features, which are subsequently projected through a trainable 1×1 convolutional layer and injected into the transformer through the text branch as conditioning information.

We fine-tune the FLUX.2 transformer using Low-Rank Adaptation (LoRA) (Hu et al., 2022), where only the inserted low-rank adaptation parameters are optimized while the pretrained transformer weights remain frozen. The objective of rectified flow reads

$$\mathcal{L}_{\mathrm{RF}} = \mathbb{E}_{t,\boldsymbol{\epsilon}} \left[\left\| v_\theta(\mathbf{z}_t, t | I_{i_r}^{e_r}, \mathbf{z}_{i_s}^{e_s}) - (\boldsymbol{\epsilon} - \mathbf{z}_{i_s}^{e_r}) \right\|_2^2 \right] \tag{2}$$

$$\mathbf{z}_t = (1-t)\mathbf{z}_{i_s}^{e_r} + t\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{3}$$

The timestep t is uniformly sampled from the continuous interval $[0, 1]$, defining a linear noise schedule that interpolates between the clean target latent $\mathbf{z}_{i_s}^{e_r}$ and Gaussian noise.

During inference, the latent representation of the source facial geometry image and randomly sampled Gaussian noise are fed into the transformer, conditioned on the reference facial image depicting the target expression. The transformer progressively transforms the noise into the predicted latent representation of the expression-retargeted face while preserving the source identity. The resulting latent is then decoded by the fine-tuned VAE decoder to obtain the nine-channel geometry image.

3.3 Training-free Expression Guidance

During transformer fine-tuning, we use paired geometry images together with DINO features extracted from the reference image as conditioning information. Since the source and target geometry images in each training pair share the same identity, the identity-consistent paired data implicitly encourages the generated result to preserve the identity of the source face. In contrast, the training objective does not impose fine-grained expression consistency between the **generated face** and the **reference image**. Although the target mesh and the reference image are assigned the same expression category or description during training, their actual expression intensities and local facial deformations can differ substantially, e.g., a broad smile versus a subtle smile.

This issue is further complicated by the use of DINO features as the reference-image condition. DINO is designed to extract general-purpose visual representations and does not explicitly disentangle facial identity and expression. Consequently, the conditioning features may not provide sufficiently precise supervision for reproducing the detailed expression of the reference image.

A straightforward solution would be to extract expression-specific features from both the reference image and the generated facial mesh and introduce an additional expression-consistency loss during training. However, such a strategy may substantially increase the computational cost and training complexity. Instead, we introduce a training-free expression guidance strategy that aligns the generated expression with the reference image during inference.

Let $\mathbf{z}_t$ denote the input latent at timestep t, and let $\mathbf{v}_t$ denote the velocity predicted by the frozen generative Transformer conditioned on the source geometry and the reference information:

$$\mathbf{v}_t = v_\theta(\mathbf{z}_t, t | I_{i_r}^{e_r}, \mathbf{z}_{i_s}^{e_s}). \tag{4}$$

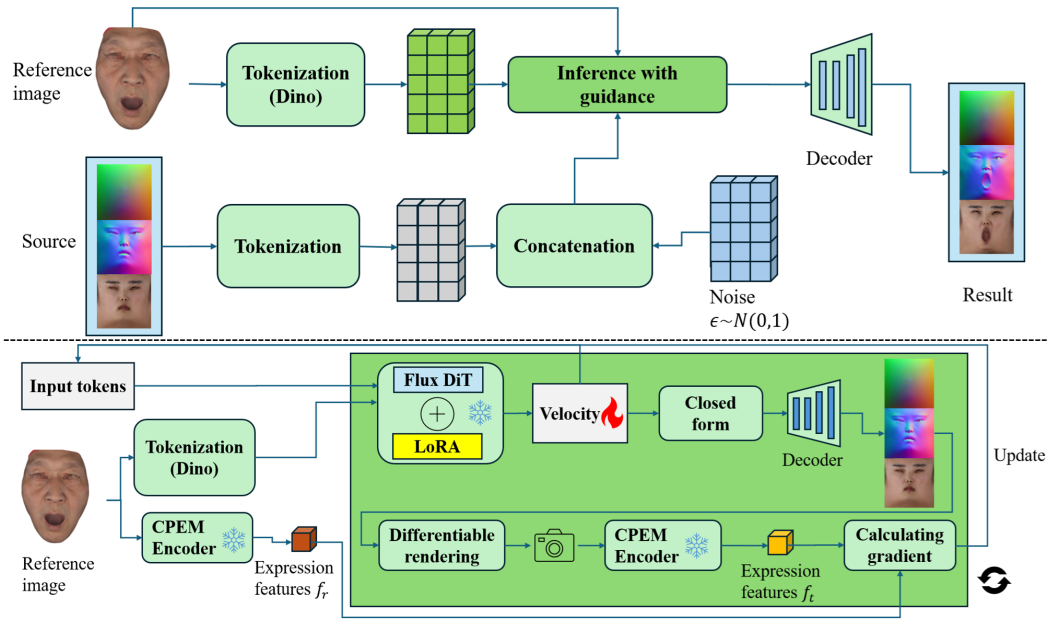

Figure 3: **Inference pipeline**. At each rectified-flow step, the velocity is treated as an optimizable variable, from which clean image tokens are reconstructed in closed form. These tokens are decoded into a geometry image, differentiably rendered into a face image, and passed through the CPEM encoder to extract expression features. The expression features are compared with those of the reference image, and the resulting gradient is used to update the velocity.

Following the rectified-flow formulation Eq. 3, we first estimate the corresponding clean latent from the current intermediate state as $\hat{\mathbf{z}}_t = \mathbf{z}_t - t\mathbf{v}_t$. The fine-tuned VAE decoder then reconstructs the corresponding geometry image $D(\hat{\mathbf{z}}_t)$, which is subsequently converted into an RGB facial image $R(D(\hat{\mathbf{z}}_t))$ using a differentiable renderer with the reconstructed texture and normal maps. To explicitly constrain facial expression, we employ the pre-trained expression encoder from CPEM (Mo et al., 2022), which disentangles facial attributes and extracts expression-specific representations from rendered facial images and reference images,

$$f_t = E_{CPEM}(R(D(\hat{\mathbf{z}}_t))),\ f_r = E_{CPEM}(I_{ir}^{er}). \tag{5}$$

We define the expression guidance objective as the squared Euclidean distance between the two expression representations:

$$\mathcal{L}_{exp} = \|f_t - f_r\|_2^2. \tag{6}$$

During inference, all model parameters remain frozen, and we compute and normalize the gradient of the expression objective only with respect to the predicted velocity to reduce variations in gradient magnitude and the modified velocity is

$$\hat{\mathbf{v}}_t = \mathbf{v}_t - \eta \frac{\nabla_{\mathbf{v}_t}\mathcal{L}_{exp}}{|\nabla_{\mathbf{v}_t}\mathcal{L}_{exp}| + \delta}, \tag{7}$$

where η controls the strength of expression guidance. The modified latent $\mathbf{z}_t - \Delta t\hat{\mathbf{v}}_t$ is subsequently used to continue the original rectified-flow sampling process to obtain the final generated result.

4 Experiments

4.1 Dataset

We selected Facescape (Yang et al., 2020), a large-scale 3D head dataset containing both raw scanned meshes reconstructed from a multi-view stereo system, and topology-unified meshes (TU model). The TU models include 847 identities, each with 20 different expressions. For each mesh

we extracted the facial region using masks provided by the dataset and closed holes around eyes and mouth to make the surface a topological disk. The processed TU models are split into training, validation, and test sets at the identity level with a ratio of 8:1:1, ensuring that the three splits are identity-disjoint. Within each split, for every identity, all possible pairs of different facial expressions are constructed. For each such intra-identity expression pair, another identity is then randomly sampled from the same split to form the corresponding cross-identity pairing. The raw scanned meshes are used as the second test set. Details of data preprocessing can be found in Appendix A.3.

4.2 Evaluation and Metric

We compare FaceGID with CPEM (Mo et al., 2022) and SMIRK (Retsinas et al., 2024). Both methods estimate disentangled identity and expression parameters from facial images and leverage 3DMMs to transfer the estimated expression parameters to a target identity. Although many other methods have been proposed for 3D facial expression retargeting, their problem settings and required inputs differ substantially from ours, making direct comparison difficult. Therefore, we restrict our comparison to methods that match our task setting as closely as possible in three key aspects: they generate expressive facial triangle meshes, use facial images as expression guidance, and support the generation of textures or rendered facial images.

Directly evaluating identity and expression similarity on facial meshes with different triangulations is challenging. Therefore, we perform the evaluation in the rendered image domain. Specifically, we evaluate: (1) identity preservation by measuring the identity similarity between the rendered generated result $\hat{I}_{i_s}^{e_r}$ and the source face $I_{i_s}^{e_s}$, and (2) expression consistency by measuring the expression similarity between the rendered generated result $\hat{I}_{i_s}^{e_r}$ and the reference image $I_{i_r}^{e_r}$. For identity similarity, we employ StyleID (Yun et al., 2026), a state-of-the-art stylization-agnostic facial identity recognition method, to extract identity embeddings and compute the averaged cosine distance between the generated and source faces. For expression consistency, we calculate Action Units (AUs) (Ekman & Friesen, 1978) by the implementation in OpenFace 2.0 (Baltrusaitis et al., 2018; Baltrušaitis et al., 2015) to represent facial expressions and compute averaged ℓ_2 distance between the AU features of the generated face and the reference image. Since raw AU activations can be affected by identity-specific facial characteristics, we normalize the AU representation for each identity with respect to its neutral expression. Specifically, for an expressive face, we subtract the AU features of the corresponding neutral face of the same identity, and compute the ℓ_2 distance between these identity-normalized AU representations.

$$D_{id} = 1 - \cos \frac{E_{StyleId}(\hat{I}_{i_s}^{e_r}) \cdot E_{StyleId}(I_{i_s}^{e_s})}{|E_{StyleId}(\hat{I}_{i_s}^{e_r})| \cdot |E_{StyleId}(I_{i_s}^{e_s})|} \tag{8}$$

$$D_{exp} = \|(AU(\hat{I}_{i_s}^{e_r}) - AU(I_{i_s}^{neutral})) - (AU(I_{i_r}^{e_r}) - AU(I_{i_r}^{neutral}))\|_2 \tag{9}$$

We render the facial meshes with and without texture and compute the two metrics separately to evaluate the expression retargeting performance in terms of geometry and texture, respectively. Since SMIRK generates a facial mesh and a corresponding portrait image rather than a texture map, we use the generated portrait image to evaluate its texture-level expression retargeting performance.

4.3 Results

The evaluation result on TU models of Facescape dataset is shown in Table 1. Our method achieves the lowest distance for both expression and identity. Figure 4 presents some visualization results. Compared with CPEM and SMIRK, our method can better capture and reproduce changes in facial expressions while preserving identity features, effectively handling both pronounced expression changes and subtle micro-expression variations.

Our method does not require the source facial mesh to conform to a fixed triangulation, allowing inputs with varying vertex counts and mesh connectivity. To demonstrate this flexibility, we apply the trained model directly to raw scanned facial meshes in Facescape dataset with varying topologies and to real facial photographs, rather than the rendered images of meshes (Figure 5). The raw scan meshes are aligned to the TU coordinates using the provided camera parameters, and we extract the facial regions using the same protocol. The inference images are real world human photos from 3DRFE dataset (Stratou et al., 2011; Ma et al., 2007).

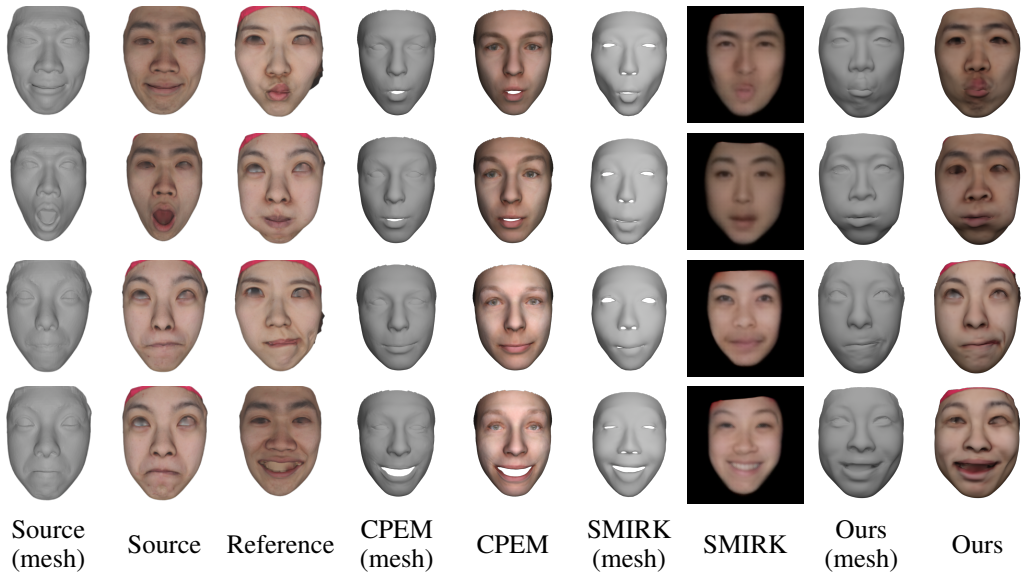

Figure 4: Retargeting results of different methods on the TU models of Facescape dataset. The source face is provided by a triangle mesh with texture and the target face is provided by a reference portrait.

Method	$D_{exp} \downarrow$	$D_{exp}\ (mesh) \downarrow$	$D_{id} \downarrow$	$D_{id}\ (mesh) \downarrow$
CPEM (Mo et al., 2022)	4.008	5.056	0.6795	0.5247
SMIRK (Retsinas et al., 2024)	4.193	6.443	0.5841	0.4524
Ours	**3.442**	**4.818**	**0.4217**	**0.3591**

Table 1: Evaluation results on TU models and rendered facial images.

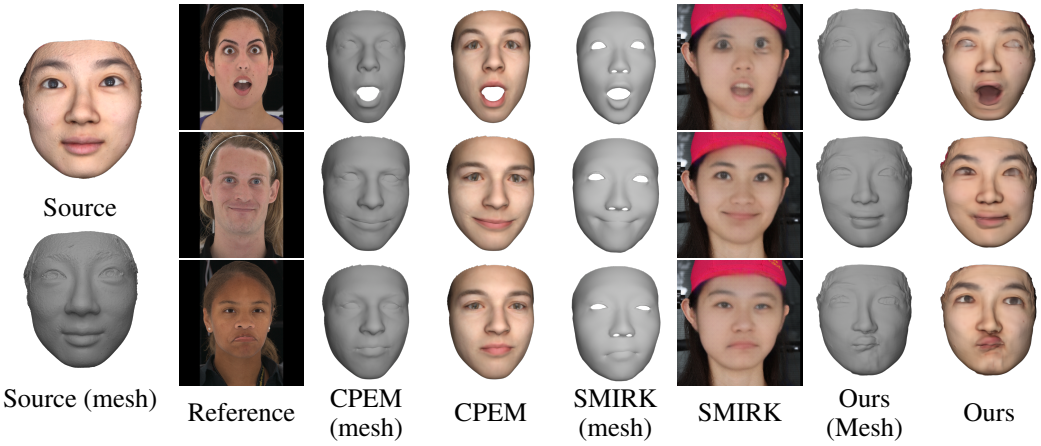

Figure 5: Retargeting results of different methods on the raw scans of Facescape dataset. The source face is provided by a triangle mesh with texture and the target face is provided by a reference portrait.

Method	$D_{exp} \downarrow$	$D_{exp}\ (mesh) \downarrow$	$D_{id} \downarrow$	$D_{id}\ (mesh) \downarrow$
CPEM (Mo et al., 2022)	5.511	6.051	0.7979	0.6425
SMIRK (Retsinas et al., 2024)	5.408	6.308	0.7445	0.5309
Ours	**5.203**	**5.856**	**0.4503**	**0.3310**

Table 2: Evaluation results on raw scanned meshes and real facial images.



Figure 6: Visualization of retargeting results with different guidance strength η.

η	0	0.2	0.4	0.6	0.8	1
$D_{exp} \downarrow$	3.954	3.607	**3.442**	3.465	3.503	3.525
$D_{exp}\ (mesh) \downarrow$	5.009	4.887	4.818	**4.789**	4.792	4.794
$D_{id} \downarrow$	**0.3659**	0.3899	0.4217	0.4435	0.4575	0.4684
$D_{id}\ (mesh) \downarrow$	**0.3507**	0.3528	0.3591	0.3627	0.3660	0.3676

Table 3: Impact of different guidance strength η.

As shown in Table 2, our method achieves the lowest distance for both expression and identity. Thanks to the expressive capability of geometry images in representing 3D surfaces, our method can still achieve high-quality facial expression transfer on raw scans, despite their substantially higher triangle counts and more complex geometric structures.

4.4 Ablation Study

We evaluate the effectiveness of training-free guidance. Table 3 reports the impact of different guidance strengths η during inference and Figure 6 presents the visualization. As the guidance strength increases, the expression distance first decreases and then increases, while the identity distance increases monotonically. Nevertheless, the identity distance remains lower than those of the competing methods reported in Table 1.

5 Limitation

Despite the effectiveness of the method, there are some limitations to be addressed. First, the geometry images used in the proposed pipeline require that the facial mesh should be a surface with zero genus and one boundary. This is determined by surface topology since a topological square is only homeomorphic to such manifold. If the input mesh has a different topology, such as a torus, it must be partitioned into multiple patches and represented as a multi-chart geometry image, which is inherently more challenging to train a generative model than a single-chart geometry image. Second, although FaceGID can take facial meshes with arbitrary triangulations as input, they may still need to be registered to the FaceScape TU models beforehand to ensure semantic consistency. Third, because the Facescape training set does not provide geometry or texture for the inner eye and oral regions, our method is unable to generate these components.

6 Conclusion

In this paper, we present FaceGID, a geometry-image-based framework that reformulates facial mesh expression retargeting as conditional generation and enables pretrained 2D generative models to operate on 3D facial data. We further introduce model adaptation and training-free expression guidance to improve expression consistency while preserving source identity and supporting meshes with different triangulations. Experiments on topology-unified meshes and raw scans demonstrate the effectiveness and flexibility of the proposed framework. Future work will focus on extending the proposed framework to more general 3D mesh editing tasks.

AI use statement

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools to provide feedback on research methodology or experiments and to search for related work. We have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, implementing methods, cleaning and reformatting dataset, supporting qualitative and thematic data analysis, and interpreting results. We reviewed all AI-assisted work. The retrieved papers were carefully examined. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Hassan Abu Alhaija, Alara Dirik, André Knörig, Sanja Fidler, and Maria Shugrina. Xdgan: Multi-modal 3d shape generation in 2d space. In *British Machine Vision Conference (BMVC 2022)*, 2022.

Tadas Baltrusaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency. Openface 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 59–66. IEEE Press, 2018. doi: 10.1109/FG.2018.00019.

Tadas Baltrušaitis, Marwa Mahmoud, and Peter Robinson. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 06, pp. 1–6, 2015. doi: 10.1109/FG.2015.7284869.

Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pp. 187–194, USA, 1999. ACM Press/Addison-Wesley Publishing Co. ISBN 0201485605. doi: 10.1145/311535.311556. URL https://doi.org/10.1145/311535.311556.

Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2014. doi: 10.1109/TVCG.2013.249.

Darren Cosker, Eva Krumhuber, and Adrian Hilton. A facs valid 3d dynamic action unit database with applications to 3d dynamic morphable facial modeling. In *2011 international conference on computer vision*, pp. 2296–2303. IEEE, 2011.

Radek Daněček, Michael J Black, and Timo Bolkart. Emoca: Emotion driven monocular face capture and animation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20311–20322, 2022.

Zheng Ding, Xuaner Zhang, Zhihao Xia, Lars Jebe, Zhuowen Tu, and Xiuming Zhang. Diffusionrig: Learning personalized priors for facial appearance editing. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12736–12746. IEEE, 2023.

Paul Ekman and Wallace V. Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press, Palo Alto, CA, 1978.

Slava Elizarov, Ciara Rowles, and Simon Donné. Geometry image diffusion: Fast and data-efficient text-to-3d with image-based surface representation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yao Feng, Haiwen Feng, Michael J Black, and Timo Bolkart. Learning an animatable detailed 3d face model from in-the-wild images. *ACM Transactions on Graphics (ToG)*, 40(4):1–13, 2021.

Michael S. Floater. Mean value coordinates. *Computer Aided Geometric Design*, 20(1):19–27, 2003. ISSN 0167-8396. doi: https://doi.org/10.1016/S0167-8396(03)00002-5. URL https://www.sciencedirect.com/science/article/pii/S0167839603000025.

Xianfeng Gu, Steven J Gortler, and Hugues Hoppe. Geometry images. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pp. 355–361, 2002.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Zi-Hang Jiang, Qianyi Wu, Keyu Chen, and Juyong Zhang. Disentangled representation learning for 3d face shape. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11949–11958. IEEE, 2019.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. `https://bfl.ai/blog/flux-2`, 2025.

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.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Wan-Chun Ma, Tim Hawkins, Pieter Peers, Charles-Felix Chabert, Malte Weiss, and Paul Debevec. Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, pp. 183–194, Goslar, DEU, 2007. Eurographics Association. ISBN 9783905673524.

Wenchao Ma, Dario Kneubuehler, Maurice Chu, Ian Sachs, Haomiao Jiang, and Sharon X. Huang. Riganyface: Scaling neural facial mesh auto-rigging with unlabeled data. In *39th Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

Langyuan Mo, Haokun Li, Chaoyang Zou, Yubing Zhang, Ming Yang, Yihong Yang, and Mingkui Tan. Towards accurate facial motion retargeting with identity-consistent and expression-exclusive constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Dafei Qin, Jun Saito, Noam Aigerman, Thibault Groueix, and Taku Komura. Neural face rigging for animating and retargeting facial meshes in the wild. In *ACM SIGGRAPH 2023 conference proceedings*, pp. 1–11, 2023.

George Retsinas, Panagiotis P. Filntisis, Radek Danecek, Victoria F. Abrevaya, Anastasios Roussos, Timo Bolkart, and Petros Maragos. 3d facial expressions through analysis-by-neural-synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Giota Stratou, Abhijeet Ghosh, Paul Debevec, and Louis-Philippe Morency. Effect of illumination on automatic expression recognition: A novel 3d relightable facial database. In *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, pp. 611–618, 2011. doi: 10.1109/FG.2011.5771467.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063.

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

Lizhen Wang, Zhiyuan Chen, Tao Yu, Chenguang Ma, Liang Li, and Yebin Liu. Faceverse: a fine-grained and detail-controllable 3d face morphable model from a hybrid dataset. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20301–20310. IEEE, 2022.

Xingguang Yan, Han-Hung Lee, Ziyu Wan, and Angel X Chang. An object is worth 64× 64 pixels: Generating 3d object via image diffusion. In *2025 International Conference on 3D Vision (3DV)*, pp. 123–133. IEEE, 2025.

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

Kwan Yun, Changmin Lee, AYeong Jeong, Youngseo Kim, Seungmi Lee, and Junyong Noh. Styleid: A perception-aware dataset and metric for stylization-agnostic facial identity recognition. *ACM Trans. Graph.*, 45(4), July 2026. ISSN 0730-0301. doi: 10.1145/3811360. URL https://doi.org/10.1145/3811360.

Junyi Zhang, Yiming Wang, Yunhong Lu, Qichao Wang, Wenzhe Qian, Xiaoyin Xu, David Gu, and Min Zhang. Spherical geometry diffusion: generating high-quality 3d face geometry via sphere-anchored representations. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence and Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence and Sixteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'26/IAAI'26/EAAI'26. AAAI Press, 2026. ISBN 978-1-57735-906-7. doi: 10.1609/aaai.v40i15.38252. URL https://doi.org/10.1609/aaai.v40i15.38252.

A Appendix

A.1 FLUX.2

The FLUX.2 VAE follows a convolutional encoder–decoder architecture with a spatial compression factor of eight. The encoder comprises four resolution stages with channel dimensions of 128, 256, 512, and 512. Each stage contains two residual convolutional layers, where each residual layer employs Group Normalization, SiLU activation, and 3×3 convolutions. The first three stages are followed by convolutional downsampling, reducing the spatial resolution by a factor of two at each stage, while the final stage preserves the spatial resolution.

The FLUX.2 transformer operates on the latent tokens produced by the VAE and follows a multimodal transformer architecture. The selected model, FLUX.2-Klein-base-4B, has a hidden dimension of 3072 with 24 attention heads, each of dimension 128. It consists of 5 joint transformer blocks followed by 20 single-stream transformer blocks. The joint blocks jointly process image latent tokens and text-condition tokens through multimodal attention, while the subsequent single-stream blocks further refine the fused representation. Rotary positional embeddings (RoPE) (Su et al., 2024) are used to encode the spatial positions of image tokens.

A.2 Implementation Details

In our work, FLUX.2 was fine-tuned on four NVIDIA H100 GPUs, each with 80 GB of memory. We optimize the VAE for 185 epochs using AdamW (Loshchilov & Hutter, 2019) with a learning rate of 1×10^{-4}, a batch size of 32 and a weight decay of 1×10^{-2}. The balancing weights for VAE training are $\lambda_1 = 2$, $\lambda_2 = 1$, $\lambda_3 = 1$ and $\lambda_4 = 5 \times 10^{-7}$, respectively. The Transformer is trained for 200,000 steps using the AdamW optimizer with a learning rate of 1×10^{-5}, a batch size of 88 and a weight decay of 0.01. To improve training efficiency and reduce memory consumption, we perform training using BF16 precision for the Transformer. Differentiable rendering during inference stage is implemented using Pytorch and the rendered images used for evaluation are rendered using Blender.

A.3 Data Preprocessing

For the TU models, we close the holes in the eye and mouth regions and extract facial regions using a facial mask with the same vertex indices (Figure 7). For the raw scans, we use the provided camera information to bake multi-view images to generate texture and project to the TU space. By finding the closest points of raw scans to the corresponding TU models, we obtain the face of the raw scans (Figure 8) .

The processed TU models are split into training set, validation set and testing set with a ratio of 8:1:1. The training, validation, and test splits are identity-disjoint, such that all scans from each identity appear in only one split. **According to the license of the FaceScape dataset[1], only 10 identities are permitted for visualization in publications. Among these identities, six have their facial regions blurred for privacy protection. Therefore, to better illustrate the qualitative results, we use the remaining four identities for result presentation.**



Figure 7: Preprocessing for TU models.

A.4 Geometry Image Generation

We use Mean Value Coordinates (Floater, 2003) to calculate parameterization for geometry image generation. For facial meshes with consistent triangulation, we use four pre-selected vertices with identical indices as the corner points and compute the MVC-based square parameterization accordingly, thereby preserving semantic consistency across the resulting geometry images. For meshes with different triangulations, a nonlinear registration can first be performed to establish dense correspondence, after which the corner points are determined by nearest-neighbor matching.

Four boundary vertices are selected as corner vertices and mapped to the four corners of the unit square $[0,1]^2$. These four corner vertices divide the boundary into four ordered segments. The non-corner boundary vertices on each segment are uniformly distributed along the corresponding side of the unit square.

After fixing the boundary coordinates, the interior vertices are parameterized using mean value coordinates. Let $\mathbf{u}_i \in \mathbb{R}^2$ denote the 2D parameter coordinate of an interior vertex $\mathbf{v}_i$, and let its one-ring neighbors be ordered as $\mathbf{v}_{j_1}, \mathbf{v}_{j_2}, \ldots, \mathbf{v}_{j_{d_i}}$, where d_i is the valence of $\mathbf{v}_i$. We define the angle between two consecutive one-ring edges as

$$\theta_{i,k} = \angle\left(\mathbf{v}_{j_k} - \mathbf{v}_i, \mathbf{v}_{j_{k+1}} - \mathbf{v}_i\right), \tag{10}$$

[1] https://nju-3dv.github.io/projects/FaceScape/

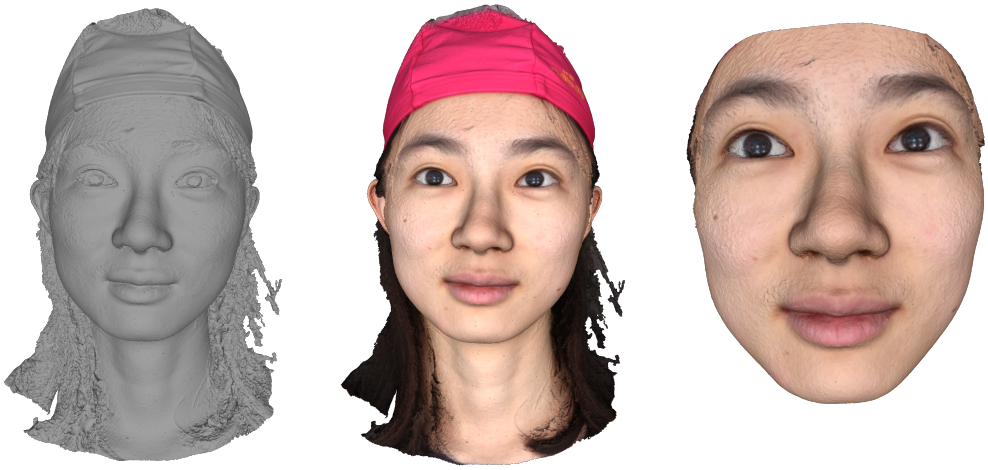

Figure 8: Preprocessing for raw scans.

where cyclic indexing is used, i.e., $j_{d_i+1} = j_1$. The mean value weight associated with neighbor $\mathbf{v}_{j_k}$ is defined as

$$w_{ij_k} = \frac{\tan\left(\theta_{i,k-1}/2\right) + \tan\left(\theta_{i,k}/2\right)}{\left\|\mathbf{v}_{j_k} - \mathbf{v}_i\right\|_2}, \tag{11}$$

Each interior vertex satisfies

$$\sum_{j \in \mathcal{N}(i)} w_{ij} \left(\mathbf{u}_i - \mathbf{u}_j\right) = 0. \tag{12}$$

Then we can build a sparse linear system with Dirichlet boundary condition and its solution is the 2D parameter coordinate of every vertex.

A.5 VAE Results

Figure 9 shows the 3D reconstruction results of the VAE before and after fine-tuning. In the second row, the original FLUX.2 VAE independently reconstructs each modality image, and the reconstructed modalities are subsequently combined to recover the final facial mesh. In contrast, the third row shows the results of our fine-tuned VAE, which directly takes the 9-channel geometry image as input and reconstructs all three modalities jointly, together with the corresponding reconstructed facial mesh. Although the differences are barely perceptible in the geometry images, the reconstructed meshes reveal that fine-tuning noticeably reduces geometric noise and surface distortions.

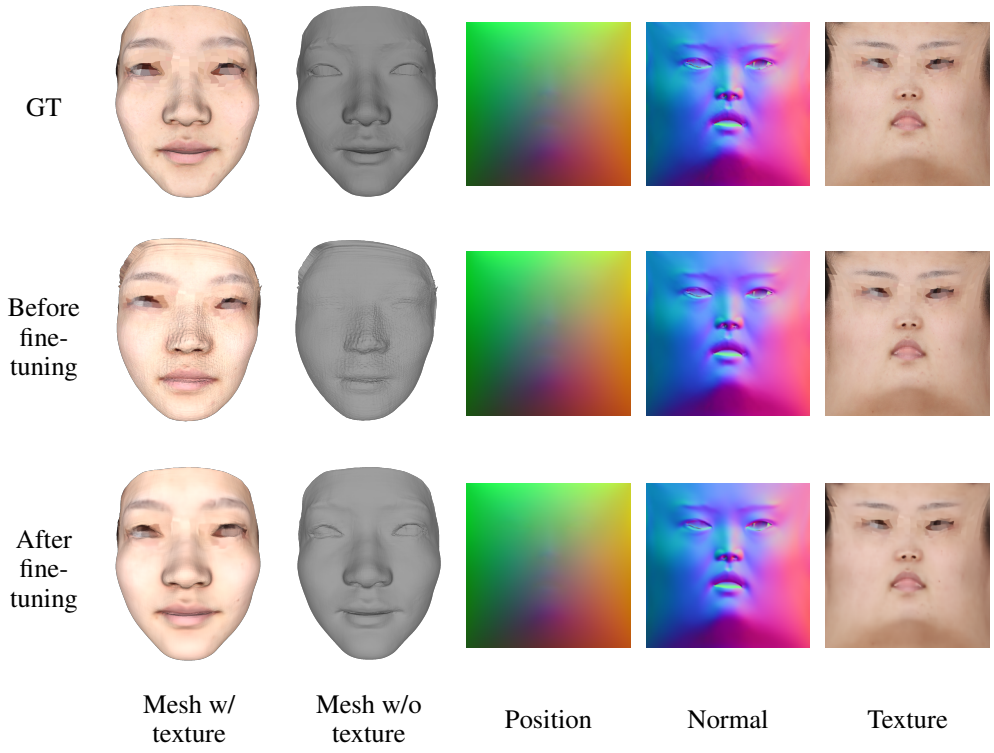

Figure 9: Geometry image reconstruction results before and after VAE fine-tuning.

A.5.1 ZOOMED-IN FIGURES

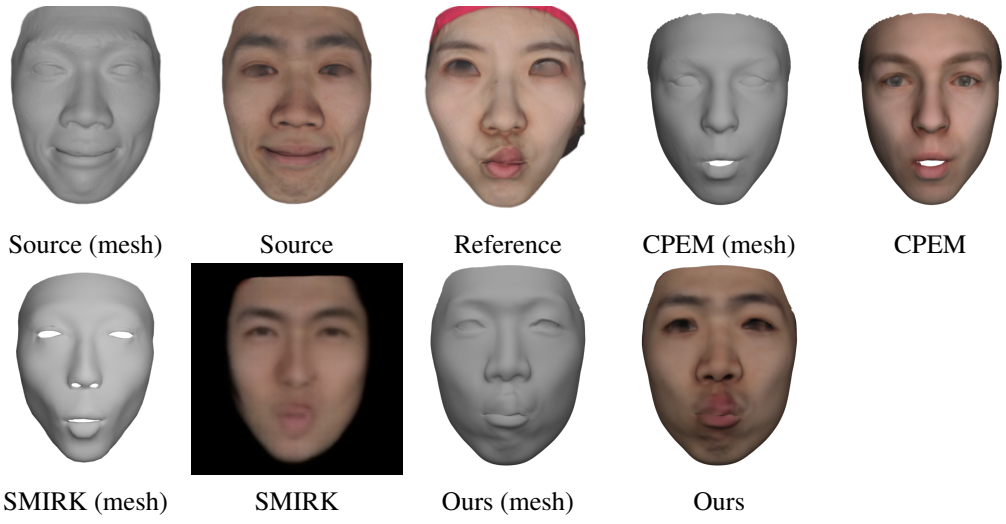

Figure 10: Enlarged views of the first row in Figure 4.

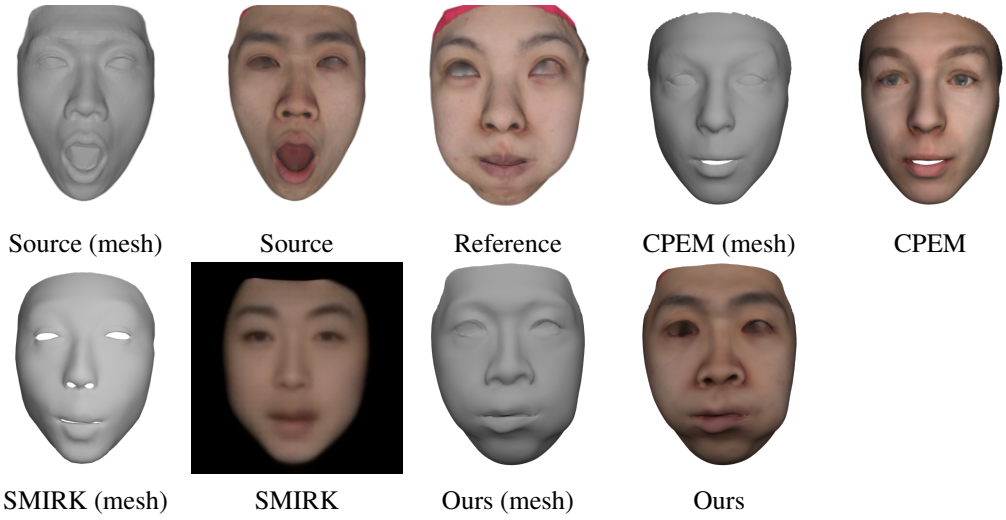

Figure 11: Enlarged views of the second row in Figure 4.

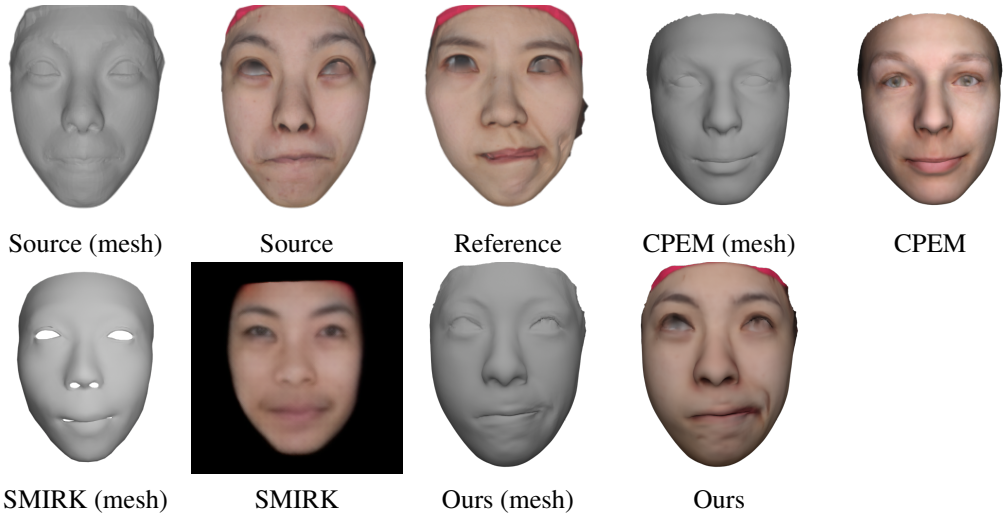

Figure 12: Enlarged views of the third row in Figure 4.

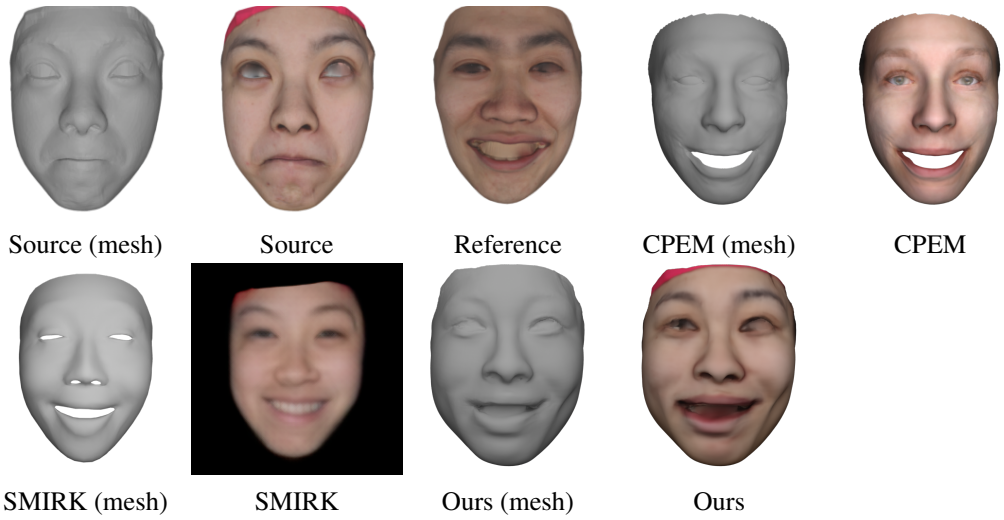

Figure 13: Enlarged views of the fourth row in Figure 4.

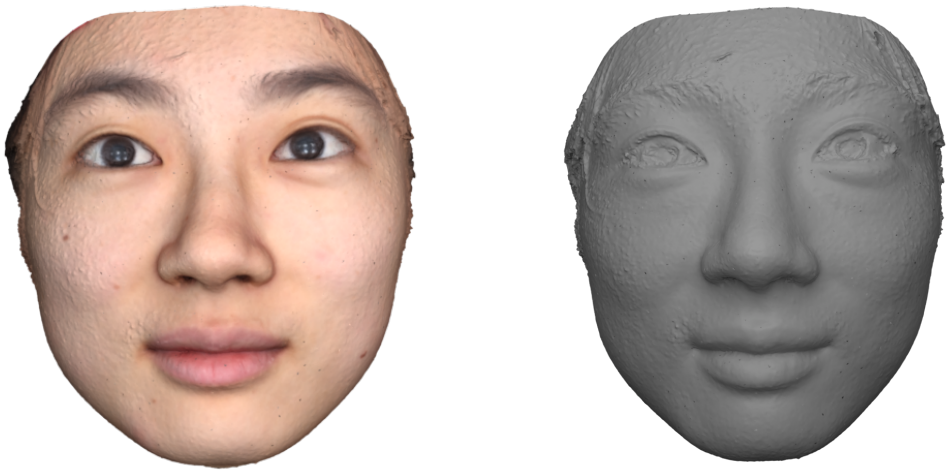

Figure 14: Enlarged views of the source mesh with and without texture in Figure 5.

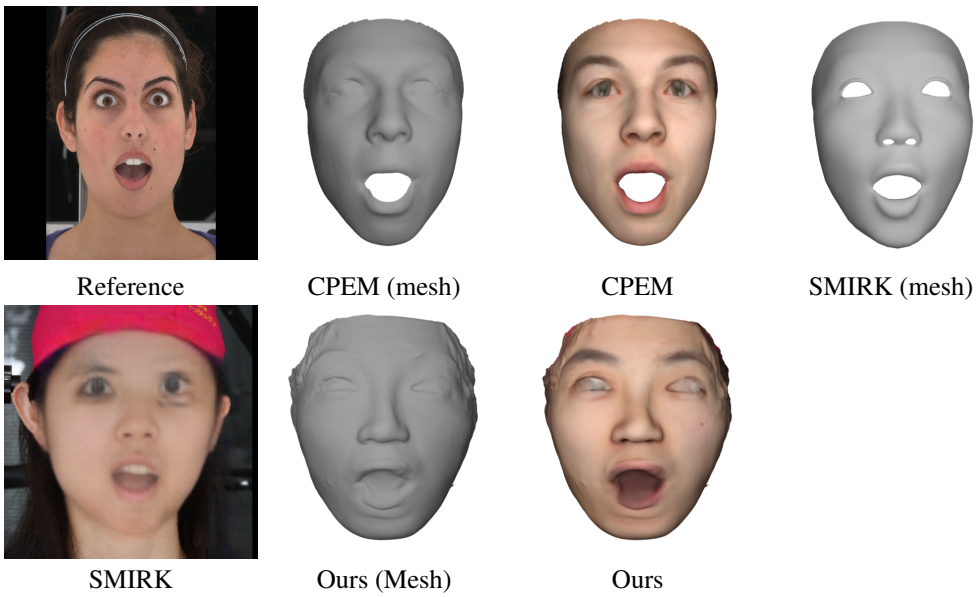

Figure 15: Enlarged views of the first row on the right hand side in Figure 5.

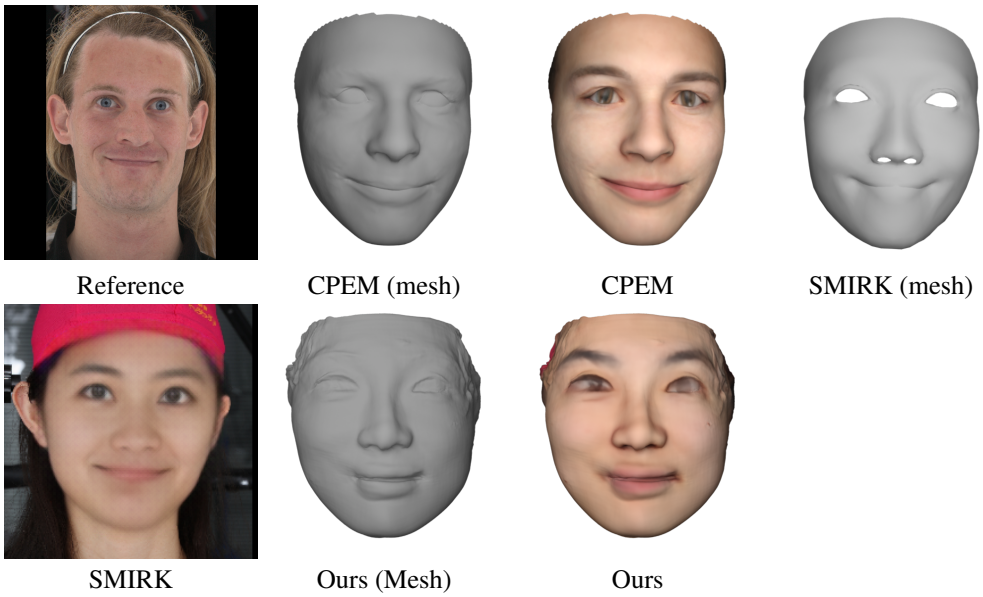

Figure 16: Enlarged views of the second row on the right hand side in Figure 5.

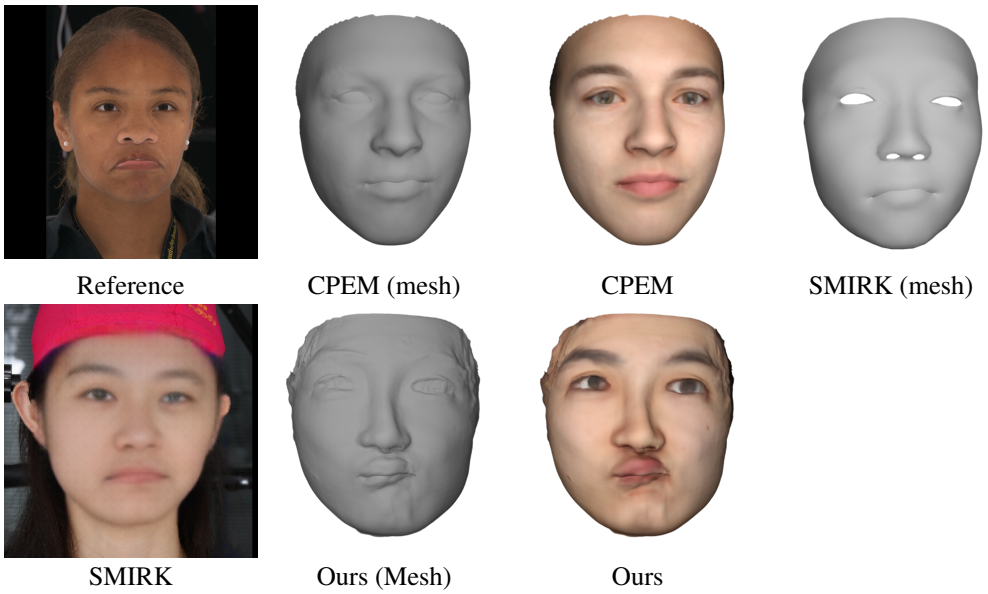

Figure 17: Enlarged views of the third row on the right hand side in Figure 5.

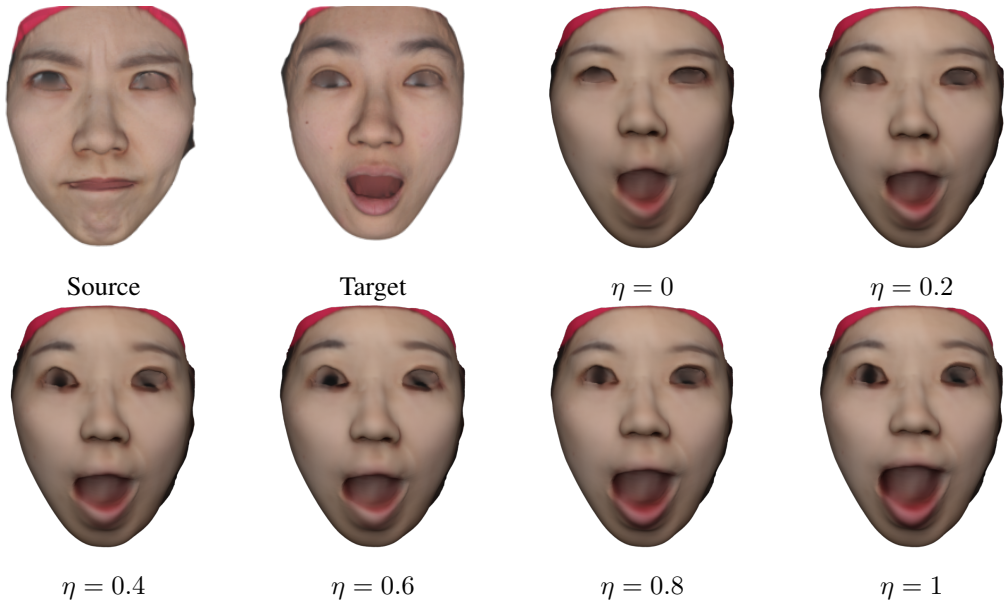

Figure 18: Enlarged views of Figure 6.