# Neural Face Rigging for Animating and Retargeting Facial Meshes in the Wild


Dafei Qin
qindafei@connect.hku.hk
The University of Hong Kong
Hong Kong

Jun Saito
jsaito@adobe.com
Adobe Research
Seattle, USA

Noam Aigerman
aigerman@adobe.com
Adobe Research
San Francisco, USA

Thibault Groueix
groueix@adobe.com
Adobe Research
San Francisco, USA

Taku Komura*
taku@cs.hku.hk
The University of Hong Kong
Hong Kong


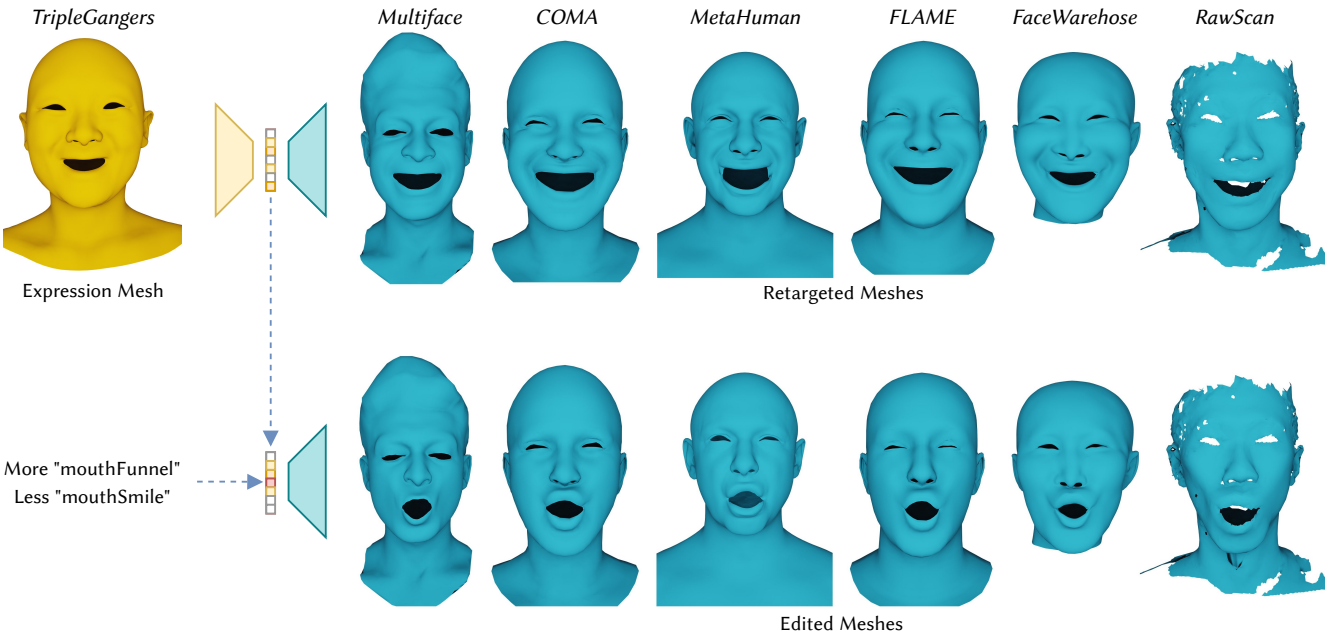


**Figure 1: We present NFR, an end-to-end deep-learning approach for automatic rigging and retargeting of 3D models of human faces in the wild. Top: Given an unrigged facial mesh with an unknown expression and identity in an arbitrary triangulation (yellow), NFR can transfer the expression to unrigged facial meshes with arbitrary triangulations (cyan). Bottom: NFR provides an interpretable latent space for user-friendly editing of the retargeted meshes.**


## ABSTRACT

We propose an end-to-end deep-learning approach for automatic rigging and retargeting of 3D models of human faces in the wild. Our approach, called Neural Face Rigging (NFR), holds three key properties: (i) NFR's expression space maintains human-interpretable editing parameters for artistic controls; (ii) NFR is readily applicable to arbitrary facial meshes with different connectivity and expressions; (iii) NFR can encode and produce fine-grained details of complex expressions performed by arbitrary subjects. To the best of our knowledge, NFR is the first approach to provide realistic and controllable deformations of in-the-wild facial meshes, without the manual creation of blendshapes or correspondence. We design a deformation autoencoder and train it through a multi-dataset training scheme, which benefits from the unique advantages of two data sources: a linear 3DMM with interpretable control parameters as in FACS and 4D captures of real faces with fine-grained details. Through various experiments, we show NFR's ability to automatically produce realistic and accurate facial deformations across a wide range of existing datasets and noisy facial scans in-the-wild, while providing artist-controlled, editable parameters.



*Corresponding author




*SIGGRAPH '23 Conference Proceedings, August 6–10, 2023, Los Angeles, CA, USA*



ACM ISBN 979-8-4007-0159-7/23/08…$15.00
https://doi.org/10.1145/3588432.3591556

## CCS CONCEPTS

• **Computing methodologies** → **Animation**; **Machine learning**; **Mesh models**.

## KEYWORDS

Facial Modeling, Facial Animation, Data-Driven Animation, Retargeting

**ACM Reference Format:**

Dafei Qin, Jun Saito, Noam Aigerman, Thibault Groueix, and Taku Komura. 2023. Neural Face Rigging for Animating and Retargeting Facial Meshes in the Wild. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings (SIGGRAPH '23 Conference Proceedings), August 6–10, 2023, Los Angeles, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3588432.3591556

## 1 INTRODUCTION

This paper is concerned with leveraging deep learning for automatic rigging and retargeting of 3D meshes of human faces in the wild, supporting both raw noisy scans of real faces, as well as clean, artist-authored meshes, while exposing interpretable expression parameters which an artist can use to intuitively bring any facial mesh to a desired expression.

Faces are a critical component of any human-centric tasks, with facial movements being of significant interest in many fields of computer science including graphics, computer vision, and HCI. Many computational representations of 3D human faces have been proposed, with early formulations such as *Facial Action Coding System* (FACS) [Ekman and Friesen 1978] and PCA-based learned linear *3D Morphable Models* (3DMM) [Blanz and Vetter 1999]. However, these rigged models can only operate on faces with known geometry and triangulation (where a consistent correspondence is provided), and cannot be automatically extended to novel face models with arbitrary triangulations.

Thus, the goal of this paper is to provide a method to understand and control real-world facial data with the following properties:

- The deformation can be controlled via *interpretable* parameters, enabling users to intuitively control expressions.
- The method should be applicable to arbitrary human faces of unknown subjects and expressions. It should generalize to arbitrary triangulations, and be robust to both noise and missing areas on the 3D face.
- The method should produce highly-accurate expressions. It should accurately encode and decode the fine-grained details of nonlinear deformations on real human faces.

We address these challenges by developing an autoencoder for facial expressions. Specifically, we design an encoder to transform a given face mesh into a latent code representing its expression. We then employ recent advancements in deep learning of 3D deformations [Aigerman et al. 2022] to construct a decoder that takes an expression code and alters the face to the desired expression. To ensure the capture of all facial features and expression nuances, we create an encoder that combines image encoding of input face renderings with a state-of-the-art mesh encoder [Sharp et al. 2022]. Our encoder features two branches—one for input expression and one for facial identity—and incorporates a training setup that separates identity from expression.

Lastly, we aim to learn the *interpretable* latent space, i.e., that each entry in the code can be interpreted as a parameter corresponding to a specific, meaningful localized activation on the human face. This goal faces a significant challenge posed by the lack of data: existing datasets comprise real human scans which are not directly coupled with expression parameters, hence we need to devise an architecture and a training scheme that will disentangle interpretable parameters from human scans with unknown expressions.

Specifically, we use two different complementary datasets to achieve our goal: 1) *Multiface* [Wuu et al. 2022] which is rich in natural and identity-specific deformations, but lacks interpretable expressions; 2) *ICT FaceKit* [Li et al. 2020], which has interpretable parameters as in FACS, however, the expressions are synthetic and less realistic. We then use these two datasets in a training scheme of our expression autoencoder. We train the system such that for the ICT dataset the latent code matches the ICT deformations parameters, while for the *Multiface* dataset, the rich and expressive details of the faces are reconstructed. This joint training empirically leads to a latent space that directly matches the FACS parameters, while further representing the rich deformations in the *Multiface* dataset. Our framework thus couples the interpretable FACS parameters with the capability to produce realistic deformations as in the *Multiface* dataset.

We show through experiments that our framework is able to perform various tasks, such as deformation transfer from unknown faces to other faces, as well as user editing of facial expressions. We show our network carries its abilities across various models, from artist-authored artistic meshes up to noisy, partial face scans in the wild. Our system significantly improves the efficiency of the human facial animation pipeline, by-passing heavy data pre-processing such as facial alignment, remeshing, and manual rigging.

## 2 RELATED WORKS

In this section, we review techniques about facial deformation models and deformation transfer.

### 2.1 Facial Deformation Models

*Anatomy-inspired models.* Facial Action Coding System (FACS) [Ekman and Friesen 1978] defines facial movements as the combination of muscle activations, or *Action Units* (AUs). Variants of FACS have been adopted in graphics and animation for their intuitive artistic controls, typically implemented with blendshape deformers [Lewis et al. 2014]. Such FACS models can be viewed as 3D Morphable Models (3DMM) [Blanz and Vetter 1999] with handcrafted basis using the domain knowledge of the human anatomy. This in turn means FACS-based 3DMMs require manual sculpting of many shapes. The inverse rig problem (a.k.a. retargeting in the context of facial animation) which solves the optimal 3DMM parameters fitting to target shapes is also not trivial for conventional FACS-based models [Cetinaslan and Orvalho 2020a,b; Lewis and Anjyo 2010; Seol et al. 2011].

Several recent studies learn compact and sparse neural representations from FACS models. Bailey et al. [2020] replaces film-quality animation rigs with a learned deep model. Vesdapunt et al. [2020] propose a person-specific joint-based neural skinning model with highly compact and sparse latent space. Choi et al. [2022] design

an intuitive interface for controlling character expressions through curves drawn on the face along facial muscle structures. Although these models preserve intuitive deformation controls, they are subject-specific, i.e., building these models for a new character requires starting from scratch. Our goal is to maintain the intuition of FACS but learn subject-specific rigging and retargeting from data with minimal manual labor.

*Data-driven models.* To avoid the ad-hoc manual sculpting of 3DMM basis, learning linear 3DMMs from face scans is a popular choice [Blanz and Vetter 1999; Brunton et al. 2014; Choe and Ko 2006; Choe et al. 2001; Huber et al. 2016; Li et al. 2017; Paysan et al. 2009; Tewari et al. 2017; Wu et al. 2016]. The readers are referred to [Egger et al. 2020] for a thorough survey of 3DMM related methods. Limited by the linear nature, these models struggle to produce highly variant and complex facial expressions. Additionally, the PCA bases fail to provide interpretable and sparse controllers.

Mesh-based neural networks are proposed to learn more expressive facial representations than linear 3DMMs. Monti et al. [2017] propose a unified framework to operate on 3D mesh local geometries. Ranjan et al. [2018] apply spectral convolutions on 3D meshes and uses hierarchical sampling to capture local and global features. Verma et al. [2018] enable the model to dynamically learn the correspondence between the data during training. Gong et al. [2019] propose spiral convolution to retrieve information on triangle mesh neighborhood. Bouritsas et al. [2019] follow a similar approach to directly processing vertex offsets and achieves better results on face representation than 3DMM. Song et al. [2020] propose to learn the facial model in the differential subspace. Zhou et al. [2020] generalize the 3D mesh autoencoder to train on tetrahedra and non-manifold meshes. Because these methods depend on specific mesh templates, they fail to apply to meshes with different representations.

Several recent works target triangulation agnosticism. Chandran et al. [2022b] utilize a transformer architecture and positional encoding to align input meshes to a canonical space. While supporting different triangulations, correspondence is required if users want to apply their model to unseen facial meshes. Yang et al. [2022] propose a physically-based implicit model to control soft bodies like human faces, which supports different mesh resolutions of the same identity.

In summary, meshes from one face dataset/model cannot be repurposed for other datasets/models without going through intensive processing steps using non-rigid registration guided by manual landmarks to take the dense correspondence [Li et al. 2020]. Our method removes this burden by training on multiple datasets with different representations, enabling application to diverse facial meshes.

## 2.2 Deformation Transfer

*Deformation Transfer* is a retargeting technique that directly works on meshes. Sumner and Popović [2004] transfer the animation by mapping the deformation gradients from the source to the target. Li et al. [2010] propose to build a target blendshape model by combining existing rigging prior and a few target example expressions. These methods require dense correspondence of the source-target pair.

*Neural Deformation Transfer.* The investigation of object deformation through neural networks [Gao et al. 2018] have been extensively explored. A comprehensive overview can be found in the survey by [Roberts et al. 2021]. Tan et al. [2018] establish a mesh VAE for learning deformation spaces of a specifically given mesh. Gao et al. [2018] establish automatic deformation transfer between two mesh datasets, which does not require explicit correspondence between the pair of meshes. However, these methods assume the input mesh structures are consistent: Mesh VAEs cannot aggregate information from multiple datasets where the connectivities of the meshes are different. Retraining is required for transferring the deformation between new mesh pairs.

Certain facial models [Chandran et al. 2020, 2022a; Jiang et al. 2019] disentangle identity and expression spaces, simplifying deformation transfer within the identity space. Still, these models do not accommodate mesh templates divergent from the training set, precluding transfer to or from custom meshes. Moser et al. [2021] suggest transferring animations between rendered videos and 3D characters via an image-to-image model, but this approach is character-specific and sacrifices deformation interpretability with PCA bases. Notably, none of these methods addresses the primary objective of this paper: integrating the neural deformation space with interpretable parameters to facilitate fine-grained human control over deformation while accommodating in-the-wild facial meshes.

# 3 METHOD

In this section, we first give an overview of the NFR framework. Then we explain how to leverage existing face datasets and 3DMMs to train a network that is both interpretable and generalizable to meshes with various shapes and triangulations.

## 3.1 Architecture

In essence, the architecture of NFR can be described as that of an autoencoder (Fig. 2). Specifically, given a neutral identity mesh $M_i$ to be deformed to an expression, and an expression mesh $M_e$ representing the desired expressions, the expression encoder first maps $M_e$'s deformation into a FACS-like latent expression code $z_e$. Similarly, the identity encoder maps $M_i$ into an identity code $z_i$. Based on $z_e$ and $z_i$, the decoder deforms $M_i$ to $M_e^*$ to approximate the expression mesh as best as possible $M_e^* \approx M_e$. The trained deformation decoder thus acts as a high-fidelity facial rig applicable to any facial mesh, with human-friendly controls (its FACS-like latent space). The end-to-end pipeline allows NFR to automatically transfer facial expressions to different identities while maintaining interpretability. To accomplish this, we leverage the recent advances in triangulation-agnostic neural geometry learning. Namely, we use a combination of image CNN's applied to renderings, along with *DiffusionNet* (DN) [Sharp et al. 2022] to build encoders to extract 3D shape features, and use *Neural Jacobian Fields* (NJF) [Aigerman et al. 2022] as a decoder to produce deformations of the facial meshes.

*Identity Encoder.* We first feed a front-view rendering of $M_i$ through $CNN$, a plain 2D CNN, to receive a code $c_i \in \mathbb{R}^{128}$. This $c_i$ is then fed into $DN_i$, a DiffusionNet-based encoder. DiffusionNet is applied on *Per-vertex features*, which is the local shape features of $M_i$, namely the concatenation of the vertex coordinates $v_i$ and

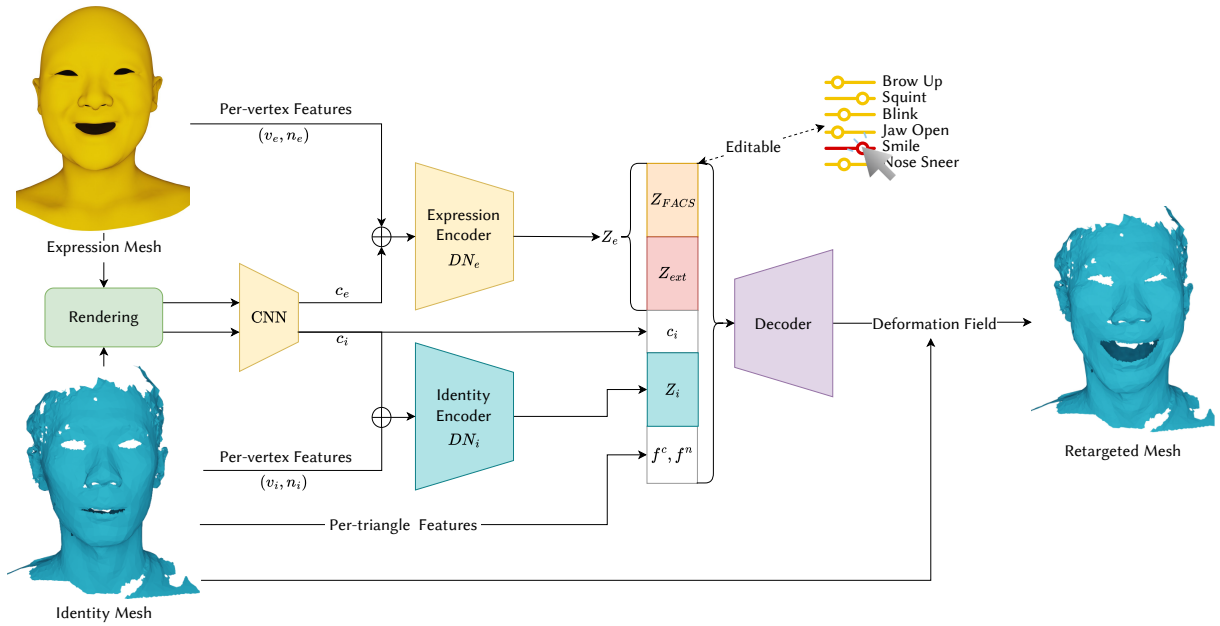


**Figure 2: Overview. Given an unrigged facial mesh with an unknown expression and identity in an arbitrary triangulation (yellow) and another unrigged *neutral* facial mesh with the target identity in an arbitrary triangulation (cyan), NFR extracts an expression code $z_e$ from the first mesh, and an identity code $z_i$ in the second mesh, and transfer the expression to the target identity with non-linear deformations as in real human faces. A part of $z_e$, denoted as $z_{FACS}$, follows the interpretable rigging parameters, allowing NFR to behave as an artist-friendly auto-rigger and retargeter.**

vertex normals $n_i$. Intuitively, combining the 2D CNN and the Per-vertex features enables $DN_i$ to capture subtle facial features. $DN_i$ receives the neutral identity as inputs, and outputs an identity latent code $z_i \in \mathbb{R}^{100}$, which is the weighted means of the per-vertex output.

*Expression Encoder.* Similarly, for the expression encoder, the rendering of $M_e$ is fed into $CNN$ to produce a code $c_e \in \mathbb{R}^{128}$, which is concantenated with the *Per-vertex features*, $(v_e, n_e)$ to be fed to a DiffusionNet-based encoder $DN_e$. In contrast to $DN_i$, $DN_e$ receives a face mesh with an expression, and outputs a predicted expression code $z_e \in \mathbb{R}^{128}$ whose elements are trained to be control parameters that are interpretable. During training, the first 53 dimensions of $z_e$ are regularized to mimic the ARKit-compatible FACS. We denote them as $z_{FACS}$. The remaining part of $z_e$, denoted as $z_{ext}$, is learned from the data. This explicit decoupling is critical for NFR to learn realistic expressions from real scans while maintaining maximum interpretability for editing.

*Decoder.* We adopt NJF as the decoder, where the core is a multi-layer perception (MLP). The MLP takes *Per-triangle features*, i.e. triangle centers $f_i^c$ and normals $f_i^n$ of the identity mesh $M_i$, the expression code $z_e$, the identity code $z_i$ and the shape code of the identity mesh $c_i$ as the input, and outputs the deformation Jacobian $g^* \in \mathbb{R}^{3\times3}$ for each triangle.

## 3.2 Datasets

The crux of our approach lies in the simultaneous training of both actual facial expressions from scanned faces and computer-generated expressions created through the manipulation of parametric controls on rigged meshes. As a result of this strategy, our neural method is capable of generating highly realistic expressions for these parameterized controls. Hence, we leverage two types of data: synthetic data and real scans.

*Synthetic data. ICT FaceKit (ICT)* [Li et al. 2020] is a linear 3DMM for facial expressions. It has 153 control parameters: a 100-dimensional PCA-based identity space, and a 53-dimensional FACS-inspired hand-crafted expression rig. This FACS model is compatible with the Apple ARKit blendshape model [1] which allows us to collect data with plausible AU activations with ARKit face tracking. These AUs have pre-defined semantics, *e.g*, the first parameter, which is called 'browInnerUp_L', corresponds to lifting the left inner eyebrow. We adopt a template of ICT that has 3,694 vertices and 7,007 triangles after the mesh standardization (Sec. 3.3).

We generate two synthetic datasets with *ICT FaceKit*

- *ICT-Random-AU.* We sample eight random identities and 5,819 random AUs per identity. *ICT-Random-AU* thus contains 41,322 samples. We select the first six identities as the training set, one as validation, and one as testing.

[1] https://developer.apple.com/documentation/arkit/arfaceanchor/blendshapelocation

- *ICT-Real-AU.* We sample 22 random identities from ICT and 1,630 frames from ARKit face tracking of our own performances to capture plausible AU activations. We perform an 8:1:1 split on the frames. *ICT-Real-AU* thus contains 32,600 training frames, 3260 validation frames, and 3260 test frames.

*Real Scans. Multiface* [Wuu et al. 2022] is a large facial dataset with fine-grained details from 4D scans. It contains 13 different identities. Each subject is captured by a multi-camera setup at 30 FPS while reading a script designed to cover over a hundred facial expressions. We use the same train/test split as in [Wuu et al. 2022] while omitting expressions that are too close to the neutral face from the training set. The training split contains 28,991 meshes and the testing split contains 13,610 meshes. Each mesh has 5,385 vertices and 10,581 faces after applying the mesh standardization.

### 3.3 Data Augmentation and Standardization

*Augmentation.* In-the-wild facial meshes vary in many ways: some have no necks, some have no back of the head, and some have bad geometry regions and holes. We conduct two types of data augmentation to adjust to these scenarios. (i) randomly shift the input off the center and scale independently along the $x$, $y$, and $z$ axis, and apply masks that remove parts of the face. (ii) randomly cut holes on the input mesh to provide geometrical variance. We will show in Sec.4.7 this data augmentation is crucial for generalizing to in-the-wild inputs.

*Standardization.* Facial datasets have different ways of modeling the internal structures of human faces, most notably eye sockets and the oral cavity. E.g. *ICT* has realistic eye sockets, whereas *Multiface* covers up eye openings with meshes. Therefore, we cut out the eye and mouth internals from mesh templates to standardize.

### 3.4 Multi-dataset training

We introduce a multi-dataset training scheme to aggregate information from the synthetic data and real scans. We first train on data generated from *ICT FaceKit* to initialize our model to imitate a linear 3DMM, then we further train on a mix of real scans from the *Multiface* dataset and synthetic data.

*Decoder Training.* Throughout the training, we use the same loss on the decoder, which contains a vertex $L_2$ loss: $L_v = ||v_e - v_{e^*}||^2$, a Jacobian loss $L_g = ||g - g^*||^2$ and a normal vector loss $L_n = ||n_e - n_e^*||^2$. Here $g$ is the ground truth deformation Jacobian from $M_i$ to $M_e$. $v_e^*$, $n_e^*$, $g^*$ are the vertex positions, vertex normals, and deformation Jacobians of the output deformed mesh $M_e^*$. Given the weights of three loss terms $\lambda_v$, $\lambda_g$ and $\lambda_n$, the total loss of the decoder is defined as:

$$L_{dec} = \lambda_v L_v + \lambda_g L_g + \lambda_n L_n. \tag{1}$$

*Training on ICT FaceKit.* We first train the model with the two *ICT* datasets and explicitly supervise $z_{FACS}$ with an $L_2$ loss to reproduce the same expression code as *ICT*. The remaining latent expression code, $z_{ext}$, is forced to be zero. We warm up the decoder by feeding the ground truth $z_e$ to the MLP. The model is thus initialized to imitate a linear 3DMM. We define the output expression code as $z_e^* = [z_{FACS}^*, z_{ext}^*]$. The encoder loss is defined as:

$$L_{enc} = ||z_{FACS} - z_{FACS}^*||^2 + ||z_{ext}^*||^2. \tag{2}$$

The total loss is a weighted sum of $L_{enc}$ and $L_{dec}$: $L = \lambda_e L_{enc} + L_{dec}$.

*Training on Multiface and ICT FaceKit.* Second, we train with *Multiface* to strengthen the network's ability to represent fine-grained expressions that out-performs linear 3DMM. The challenge is that *Multiface* does not have ground truth latent expression parameters. To maintain an interpretable latent space, we rely on the fact that NFR is correctly initialized to form a FACS-like latent space. We keep a part of the batch from *ICT-Real-AU* with direct latent supervision to maintain this interpretability. *ICT-Random-AU* is not used here since this randomly generated dataset may contain unrealistic expressions. On *Multiface* we simply regularize the latent space if the parameters go out of the range $[0, 1]$ to follow the convention of FACS AUs:

$$L_r(x) = \begin{cases} -x, & x < 0 \\ 0, & 0 \le x \le 1 \\ x - 1. & x > 1 \end{cases} \tag{3}$$

The loss of the encoder at this stage becomes:

$$L_{enc} = \begin{cases} ||z_{FACS} - z_{FACS}^*||^2 + ||z_{ext}^*||^2, & \textit{ICT-Real-AU} \\ L_r(z_e^*). & \textit{Multiface} \end{cases} \tag{4}$$

After these two stages of training, our model can retarget realistic expressions to in-the-wild meshes and preserves an interpretable latent expression space for manipulation.

## 4 EXPERIMENTS

In this section, a series of experiments are conducted to showcase the capabilities of NFR. Sec. 4.1 evaluates the encoder's effectiveness by comparing inverse-rigging outcomes for *ICT-Real-AU* against *Seol* [Seol et al. 2011]. Sec. 4.2 benchmarks the expression quality of NFR against template-specific mesh reconstruction methods. Sec. 4.3 confirms the triangulation-agnostic property through inverse rigging experiments on *ICT-Real-AU* with a different mesh template. Sec. 4.4 demonstrates NFR's practicality by retargeting expressions to in-the-wild meshes of varying connectivity and resolutions. Sec. 4.5 further illustrates the model's practicality, plotting the activations of each position in $z_{FACS}$ to reveal a semantically related latent expression space. The ease of use is demonstrated through two sequences of editing processes applied to in-the-wild expressions. Sec. 4.6 qualitatively exhibits that sampling in the learned expression space produces plausible expressions, surpassing a basic 3DMM. Lastly, Sec. 4.7 presents various ablation studies to substantiate the key design choices of the model and training scheme. The readers are referred to the supplementary video for the details.

We use the per-vertex Euclidean distance as the error metric and additionally report the 90% percentile value to emphasize the bad cases.

### 4.1 Inverse Rigging

*Inverse rigging* is a task to find optimal rigging parameters fitting the resulting deformation to a given geometry. Our encoder behaves

**Table 1: Inverse Rigging on *ICT-Real-AU*. Our encoder, evaluated both via *ICT rig* and *NFR*, outperform *Seol* [Seol et al. 2011] by a large margin.**

| Method | Mean (*mm*) | Median (*mm*) | 90% (*mm*) |
|---|---|---|---|
| Seol [Seol et al. 2011] | 0.688 ± 0.979 | 0.369 | 1.640 |
| Ours (ICT rig) | **0.378 ± 0.467** | **0.225** | **0.860** |
| Ours (NFR) | 0.443 ± 0.454 | 0.307 | 0.920 |

**Figure 3: Inverse Rigging. Top: The input *Multiface* expressions. Bottom: Applying the latent $z_{e,FACS}$ to the ICT rig. All deformations on the eyebrows, eyes, and mouth are solved correctly.**

as a triangulation-agnostic inverse rig predictor extracting FACS AUs from a facial mesh with unknown, entangled AU activations.

Table. 1 compares the inverse rigging results of our method against *Seol* [Seol et al. 2011], an iterative optimization-based method that relies on the ground truth rigging model. Here, *Ours (ICT rig)* represents the deformation from ICT's linear 3DMM driven by our inverse-rigged parameters. *Ours (NFR)* represents the deformation output of our decoder. In this complex setting where many AUs are activated simultaneously, our method outperforms *Seol* by a large margin, where the performance of *Ours (ICT rig)* validates the effectiveness of the expression encoder, and that of *Ours (NFR)* indicates that the rigging space of our model is close to the ICT rig.

In Fig. 3 we apply $z_{FACS}$ of *Multiface* on the ICT rig. Though limited by the expression ability of the linear rig, the expressions on the ICT rig are still semantically close to that of *Multiface*. This validates the inverse rigging ability of NFR on in-the-wild expressions. Note that while *Ours (NFR)* performs slightly worse on the synthetic dataset *ICT-Real-AU* than *Ours (ICT rig)*, NFR's major advantage is to work on in-the-wild data with arbitrary triangulation and high-frequency details.

## 4.2 Expression Quality

In this section, we assess the ability of our model to generate precise geometric details. The expression encoder is employed to map target deformations into an interpretable expression space, subsequently reconstructing the input deformation through the decoder. Our model is compared to three template-specific competitors: COMA [Ranjan et al. 2018], N3DMM [Bouritsas et al. 2019], and SpiralNet++ [Gong et al. 2019]. All models are trained on the same *Multiface* training split, with our model also receiving training on synthetic *ICT* datasets. NFR is designed to maintain an interpretable latent space and handle meshes in a triangulation-agnostic manner, whereas competitor models are constrained by a single mesh template, limiting their capacity to learn from diverse datasets and resulting in less interpretability. latent spaces. A baseline method, *NFR (Multiface)*, trained solely on the *Multiface* dataset, demonstrates the advantages of multi-dataset training.

**Table 2: Quantitative reconstruction results on *Multiface*. NFR outperforms COMA, Neural3DMM, and SpiralNet++ on all metrics by a large margin. See Fig. 4 and Fig. 5 for a qualitative comparison.**

| Method | Mean (*mm*) | Median (*mm*) | 90% (*mm*) |
|---|---|---|---|
| COMA | 2.324 ± 1.724 | 1.858 | 4.380 |
| Neural3DMM | 1.254 ± 1.092 | 0.951 | 2.406 |
| SpiralNet++ | 1.256 ± 1.105 | 0.944 | 2.438 |
| NFR (*Multiface*) | 1.005 ± 0.824 | 0.772 | 1.895 |
| NFR | **0.879 ± 0.727** | **0.678** | **1.651** |

Table. 2 reveals that NFR significantly outperforms its competitors, achieving a 32.3% reduction in the 90% percentile value compared to the best-performing competitor. Training on multiple datasets establishes a sparse and semantic latent space and enhances expression reconstruction. In Fig. 4, we visualize the per-vertex Euclidean error on *Multiface* samples and provide a magnified comparison in Fig. 5. NFR exhibits fewer coarse surface artifacts than other baselines and more effectively preserves the shape around facial features such as the eyes, nose, and mouth.

## 4.3 Triangulation Invariance

Our model is composed of multiple triangulation-agnostic components: The $CNN$ takes fixed-size rendered images as inputs; $DN_e$ and $DN_i$ are agnostic to the mesh templates [Sharp et al. 2022]; the MLP has shared parameters across all the input triangles. Thus, our model naturally supports meshes with different resolutions and connectivity. To quantitatively evaluate this property, we apply our model on *ICT-Real-AU* with a high-resolution mesh template and different identities. The template has 10,089 vertices and 19,758 triangles. Table. 3 quantitatively validate that NFR has minimal performance degradation on different triangulations, despite the fact that it is only trained on the original mesh template with less than half of vertices and triangles. Sec. 4.4 shows qualitatively that our model can transfer expression to in-the-wild meshes, even raw scans with a significant amount of noise.

## 4.4 Retargeting in the Wild

Given the target mesh of an arbitrary subject with unknown expressions, *retargeting* is a task to interpret the target facial expression

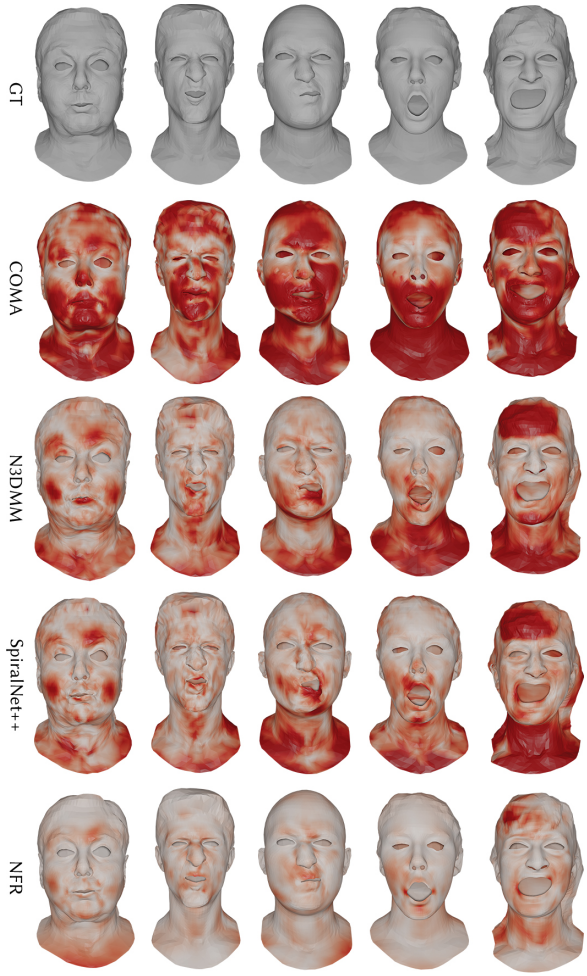


**Figure 4: Qualitative reconstruction results on *Multiface*. We color the meshes by their per-vertex Euclidean error. We compare COMA, SpiralNet++, N3DMM, and NFR. The color map clearly shows that NFR outperforms the competing approaches.**

**Table 3: Triangulation Invariance. We compare the inverse-rigging performance of NFR on the trained mesh template (Original) and a retriangulated version (High-resolution) of *ICT-Real-AU*. NFR performs equally well on both datasets, even though it is not trained with the retriangulated template.**

| Method | Templates | Mean (*mm*) | Median (*mm*) | 90% (*mm*) |
|---|---|---|---|---|
| Ours (ICT rig) | Original | 0.378 ± 0.467 | 0.225 | 0.860 |
| | High-resolution | 0.343 ± 0.416 | 0.212 | 0.774 |
| Ours (NFR) | Original | 0.443 ± 0.454 | 0.307 | 0.920 |
| | High-resolution | 0.444 ± 0.436 | 0.312 | 0.930 |

and transfer it to another subject. Note that this task is more challenging than a typical deformation transfer task where the neutral target mesh is known.

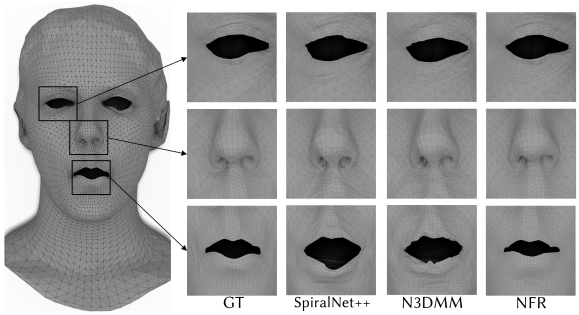


**Figure 5: Zoomed-in comparison on *Multiface*. NFR not only gets rid of the coarse surface artifacts of the other baselines but also preserves the shape better around the eyes, nose, and mouth. Left is the ground truth.**

In Fig. **??**, we test the generalizability of our model's retargeting with various identities, expressions, triangulations, and mesh quality from different face models and datasets. The retargeted results to various neutral face meshes successfully capture our high-level perception of the target expressions with natural deformations, though the ground-truth AUs of the target expressions are unknown. We retarget an ARKit face tracking sequence and a *Multiface* speaking sequence to these in-the-wild meshes. Please refer to the supplementary video for details.

## 4.5 Human-Friendly Editing

NFR behaves as a user-friendly rig to edit the expressions with the latent code activation intensities. Fig. **??** shows the series of such controllable manual edits, starting off from the latent code inferred by the inverse rigging, then increasing/decreasing the latent activations corresponding to FACS AUs. We show all the 53 FACS-like control codes of our latent space in Fig. **??**. Note that each activation follows the semantics of the pre-defined FACS.

## 4.6 Non-Linear Deformations

Along with the interpretable latent code learned from *ICT FaceKit*, our model also captures natural, non-linear deformations from *Multiface*. We show this in Fig. **??** where many AUs are activated simultaneously. Linear 3DMMs struggle with such complex activations as they simply add more offsets on top of offsets, resulting in implausible deformations. A usual workaround is to activate manually-sculpted corrective shapes of activation combinations to introduce non-linearity [Lewis et al. 2014]. Our model maintains natural, non-linear deformations without such ad-hoc sculpting.

## 4.7 Ablations

We perform a series of ablation studies to demonstrate the effectiveness of our key designs.

*Multi-dataset Training Scheme.* The training scheme is the core of our method for realistic expression generation and interpretable editing. Fig. **??** demonstrates the effectiveness of our user-friendly editing. In contrast, in Fig. **??** we show the activation of the first five individual AUs, without training on the ICT datasets. The deformed

**Table 4: Ablation showing the importance of real data (*Multiface*). We evaluate on *Multiface* test set with and without training on *Multiface* real scans.**

| Method | Mean (*mm*) | Median (*mm*) | 90% (*mm*) |
|---|---|---|---|
| With *Multiface* | **0.879 ± 0.727** | **0.678** | **1.651** |
| Without *Multiface* | 1.470 ± 1.319 | 1.088 | 2.860 |

**Table 5: Ablation study on the network structures. Without the CNN features, the model struggles to deal with the random shift and scale introduced in data augmentation. When replacing the DN layers with PN, the model fails to solve for the correct expression codes.**

| Dataset | Method | Mean (*mm*) | Median (*mm*) | 90% (*mm*) |
|---|---|---|---|---|
| ICT-Real-AU | Ours (w/o CNN) | 1.034 ± 0.985 | 0.748 | 2.053 |
| | Ours (PN) | 0.710 ± 0.756 | 0.483 | 1.502 |
| | Ours (DN) | **0.443 ± 0.454** | **0.307** | **0.920** |
| Multiface | Ours (w/o CNN) | 1.559 ± 1.278 | 1.184 | 3.011 |
| | Ours (PN) | 1.408 ± 1.207 | 1.050 | 2.758 |
| | Ours (DN) | **0.879 ± 0.727** | **0.678** | **1.651** |

vertices are highly entangled. Without the direct supervision of ICT AUs, making adjustments by tweaking the expression codes is impractical. We then show quantitatively in Table. 4 that our model has poor quality on the reconstructed expressions without training on the real world *Multiface* data.

*Data Augmentation.* We show in Fig. **??** NFR with/without data augmentation to retarget meshes from two different datasets: *FaceWarehouse* [Cao et al. 2013] and *Triplegangers*. Without training on the augmented datasets in Sec 3.3, the model fails to apply the correct deformations to these in-the-wild meshes.

*Extended latent space.* We expand the latent expression space to be 128-dimensional, where the first 53-dimensional vector, denoted as $z_{FACS}$ is supervised by the ICT expression codes. The remaining $z_{ext}$ is learned from data. Fig. **??** compares individual activations of $z_{FACS}$ between NFR and a similar model without $z_{ext}$. Both models supervise $z_{FACS}$ to follow the ICT rig. Since *Multiface* contains deformation patterns that are not covered by the ICT rig space, the model without $z_{ext}$ adapts some AUs, *e.g.* 'eyeLookUp_L', to capture these new patterns (*e.g.* neck deformation). With the additional expression space of $z_{ext}$, NFR can capture the *Multiface* deformations and maintain maximum interpretability.

*Network Structures.* The *CNN* serves to generalize NFR to the translation and scale introduced in data augmentation for better in-the-wild applicability. $\text{DN}_e$ and $\text{DN}_i$ are crucial for solving the correct expression codes and transferring them to in-the-wild meshes. Table. 5 compares a model without the CNN features and another one that replaces the two DNs by PointNets (PN) [Qi et al. 2017]. The performance drops significantly.

## 5 CONCLUSIONS

We have presented *Neural Face Rigging*, a novel learning approach to instantly rig and retarget 3D facial meshes in the wild with any reasonable shape variations and triangulation. The key technical contribution is the multi-stage training combining the advantages of different face datasets to learn interpretable and editable latent code over high-fidelity facial deformations. While our model is unique in its generalization to mesh triangulation, it learns better facial deformation than other methods that require fixed triangulation.

*Limitations and Future Work.* Although our model provides a triangulation-agnostic facial rigging and retargeting pipeline, users still need a standardization step by removing internal structures around the eyes and mouth. Segmenting the facial regions could automate the process.

We chose to learn an ARKit-compatible FACS rig for its popularity and accessibility. In theory, NFR could learn any other arbitrary rig parameterizations such as *MetaHuman*, though this is not tested because of the limited access to such data.

With an interpretable, controllable latent space and triangulation invariance, our model can serve as a backbone for various facial animation tasks such as talking face generation. Including the appearance model for photo-realistic expression generation is also worth exploring.

Human faces are sensitive subjects. We must take precautions deploying our model, with in-depth studies on the bias to different human attributes, e.g. age, gender, and ethnicity. We hope this work can contribute to the community by minimizing such bias with its ability to combine multiple face datasets and perform instant rigging on in-the-wild meshes.

## ACKNOWLEDGMENTS

This research is supported by Innovation and Technology Commission (Ref:ITS/319/21FP) and Research Grant Council (Ref: 17210222), Hong Kong.

## REFERENCES

Noam Aigerman, Kunal Gupta, Vladimir G. Kim, Siddhartha Chaudhuri, Jun Saito, and Thibault Groueix. 2022. Neural Jacobian Fields: Learning Intrinsic Mappings of Arbitrary Meshes. *ACM Trans. Graph.* 41, 4, Article 109 (jul 2022), 17 pages. https://doi.org/10.1145/3528223.3530141

Stephen W Bailey, Dalton Omens, Paul Dilorenzo, and James F O'Brien. 2020. Fast and deep facial deformations. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 94–1.

Volker Blanz and Thomas Vetter. 1999. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. 187–194.

Giorgos Bouritsas, Sergiy Bokhnyak, Stylianos Ploumpis, Michael Bronstein, and Stefanos Zafeiriou. 2019. Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7213–7222.

Alan Brunton, Timo Bolkart, and Stefanie Wuhrer. 2014. Multilinear wavelets: A statistical shape space for human faces. In *European Conference on Computer Vision*. Springer, 297–312.

Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. 2013. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics* 20, 3 (2013), 413–425.

Ozan Cetinaslan and Verónica Orvalho. 2020a. Sketching Manipulators for Localized Blendshape Editing. *Graphical Models* 108 (2020), 101059.

Ozan Cetinaslan and Verónica Orvalho. 2020b. Stabilized blendshape editing using localized Jacobian transpose descent. *Graphical Models* 112 (2020), 101091.

Prashanth Chandran, Derek Bradley, Markus Gross, and Thabo Beeler. 2020. Semantic deep face models. In *2020 International Conference on 3D Vision (3DV)*. IEEE, 345–354.

Prashanth Chandran, Gaspard Zoss, Markus Gross, Paulo Gotardo, and Derek Bradley. 2022a. Facial Animation with Disentangled Identity and Motion using Transformers. *ACM/Eurographics Symposium on Computer Animation* (2022).

Prashanth Chandran, Gaspard Zoss, Markus Gross, Paulo Gotardo, and Derek Bradley. 2022b. Shape Transformers: Topology-Independent 3D Shape Models Using Transformers. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 195–207.

Byoungwon Choe and Hyeong-Seok Ko. 2006. Analysis and synthesis of facial expressions with hand-generated muscle actuation basis. In *ACM SIGGRAPH 2006 Courses*. 21–es.

Byoungwon Choe, Hanook Lee, and Hyeong-Seok Ko. 2001. Performance-driven muscle-based facial animation. *The Journal of Visualization and Computer Animation* 12, 2 (2001), 67–79.

Byungkuk Choi, Haekwang Eom, Benjamin Mouscadet, Stephen Cullingford, Kurt Ma, Stefanie Gassel, Suzi Kim, Andrew Moffat, Millicent Maier, Marco Revelant, Joe Letteri, and Karan Singh. 2022. Animatomy: An Animator-Centric, Anatomically Inspired System for 3D Facial Modeling, Animation and Transfer. In *SIGGRAPH Asia 2022 Conference Papers* (Daegu, Republic of Korea) *(SA '22)*. Association for Computing Machinery, New York, NY, USA, Article 16, 9 pages. https://doi.org/10.1145/3550469.3555398

Bernhard Egger, William A. P. Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, Christian Theobalt, Volker Blanz, and Thomas Vetter. 2020. 3D Morphable Face Models - Past, Present and Future. *ACM Transactions on Graphics* 39, 5 (August 2020). https://doi.org/10.1145/3395208

Paul Ekman and Wallace V. Friesen. 1978. Facial action coding system: a technique for the measurement of facial movement. In *Consulting Psychologists Press*.

Lin Gao, Jie Yang, Yi-Ling Qiao, Yu-Kun Lai, Paul L Rosin, Weiwei Xu, and Shihong Xia. 2018. Automatic unpaired shape deformation transfer. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–15.

Shunwang Gong, Lei Chen, Michael Bronstein, and Stefanos Zafeiriou. 2019. Spiralnet++: A fast and highly efficient mesh convolution operator. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 0–0.

P Huber, G Hu, R Tena, P Mortazavian, P Koppen, WJ Christmas, M Ratsch, and J Kittler. 2016. A Multiresolution 3D Morphable Face Model and Fitting Framework.

Zi-Hang Jiang, Qianyi Wu, Keyu Chen, and Juyong Zhang. 2019. Disentangled representation learning for 3d face shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11957–11966.

John P Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Frederic H Pighin, and Zhigang Deng. 2014. Practice and theory of blendshape facial models. *Eurographics (State of the Art Reports)* 1, 8 (2014), 2.

John P Lewis and Ken-ichi Anjyo. 2010. Direct manipulation blendshapes. *IEEE Computer Graphics and Applications* 30, 4 (2010), 42–50.

Hao Li, Thibaut Weise, and Mark Pauly. 2010. Example-based facial rigging. *Acm transactions on graphics (tog)* 29, 4 (2010), 1–6.

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

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

Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. 2017. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5115–5124.

Lucio Moser, Chinyu Chien, Mark Williams, Jose Serra, Darren Hendler, and Doug Roble. 2021. Semi-supervised video-driven facial animation transfer for production. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–18.

Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. 2009. A 3D Face Model for Pose and Illumination Invariant Face Recognition. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*. 296–301. https://doi.org/10.1109/AVSS.2009.58

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

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

Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. 2020. Accelerating 3D Deep Learning with PyTorch3D. *arXiv:2007.08501* (2020).

Richard A. Roberts, Rafael Kuffner dos Anjos, Akinobu Maejima, and Ken Anjyo. 2021. Deformation transfer survey. *Computers Graphics* (2021). https://doi.org/10.1016/j.cag.2020.10.004

Yeongho Seol, Jaewoo Seo, Paul Hyunjin Kim, John P Lewis, and Junyong Noh. 2011. Artist friendly facial animation retargeting. *ACM Transactions on Graphics (TOG)* 30, 6 (2011), 1–10.

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

Steven L. Song, Weiqi Shi, and Michael Reed. 2020. Accurate Face Rig Approximation with Deep Differential Subspace Reconstruction. *ACM Trans. Graph.* 39, 4, Article 34 (aug 2020), 12 pages. https://doi.org/10.1145/3386569.3392491

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

Qingyang Tan, Lin Gao, Yu-Kun Lai, and Shihong Xia. 2018. Variational Autoencoders for Deforming 3D Mesh Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ayush Tewari, Michael Zollhofer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. 2017. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 1274–1283.

Nitika Verma, Edmond Boyer, and Jakob Verbeek. 2018. Feastnet: Feature-steered graph convolutions for 3d shape analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2598–2606.

Noranart Vesdapunt, Mitch Rundle, HsiangTao Wu, and Baoyuan Wang. 2020. JNR: Joint-based neural rig representation for compact 3D face modeling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer, 389–405.

Chenglei Wu, Derek Bradley, Markus Gross, and Thabo Beeler. 2016. An anatomically-constrained local deformation model for monocular face capture. *ACM transactions on graphics (TOG)* 35, 4 (2016), 1–12.

Cheng-hsin Wuu, Ningyuan Zheng, Scott Ardisson, Rohan Bali, Danielle Belko, Eric Brockmeyer, Lucas Evans, Timothy Godisart, Hyowon Ha, Alexander Hypes, Taylor Koska, Steven Krenn, Stephen Lombardi, Xiaomin Luo, Kevyn McPhail, Laura Millerschoen, Michal Perdoch, Mark Pitts, Alexander Richard, Jason Saragih, Junko Saragih, Takaaki Shiratori, Tomas Simon, Matt Stewart, Autumn Trimble, Xinshuo Weng, David Whitewolf, Chenglei Wu, Shoou-I Yu, and Yaser Sheikh. 2022. Multiface: A Dataset for Neural Face Rendering. In *arXiv*. https://doi.org/10.48550/ARXIV.2207.11243

Lingchen Yang, Byungsoo Kim, Gaspard Zoss, Baran Gözcü, Markus Gross, and Barbara Solenthaler. 2022. Implicit Neural Representation for Physics-Driven Actuated Soft Bodies. *ACM Trans. Graph.* 41, 4, Article 122 (jul 2022), 10 pages. https://doi.org/10.1145/3528223.3530156

Yi Zhou, Chenglei Wu, Zimo Li, Chen Cao, Yuting Ye, Jason Saragih, Hao Li, and Yaser Sheikh. 2020. Fully convolutional mesh autoencoder using efficient spatially varying kernels. *Advances in Neural Information Processing Systems* 33 (2020), 9251–9262.

## A IMPLEMENTATION DETAILS

We render the front views of $S$ and $T$ by Pytorch3D [Ravi et al. 2020] with a fixed point light and a gray Lambert shading. The model is not sensitive to these settings as long as the whole face is shown in the image.

The rendered 256×256 RGBD images feed into $CNN$ which has four 2D convolution layers followed by a fully-connected layer to output $c_S$ and $c_T$. $\text{DN}_e$ and $\text{DN}_i$ have four and two diffusion layers, respectively. $MLP_{dec}$ contains eight linear layers of 256 hidden dimensions, with ReLU activations in between.

We train the model with $\lambda_v = 10$, $\lambda_g = 1$, $\lambda_n = 1$ and $\lambda_e = 0.1$. During the first training stage, we warm up the MLP by providing the ground truth $z_{FACS}$ in the first 100 epochs. And then train another 200 epochs with $z^*_{FACS}$ from the expression encoder. In the second stage, we train on both *ICT-Real-AU* and *Multiface* until convergence. We set the initial learning rate as 1e-4 and decrease it by a factor of 0.75 for every 100 epochs.

## B DATA AUGMENTATION

Here we visualize the output meshes after data augmentation:

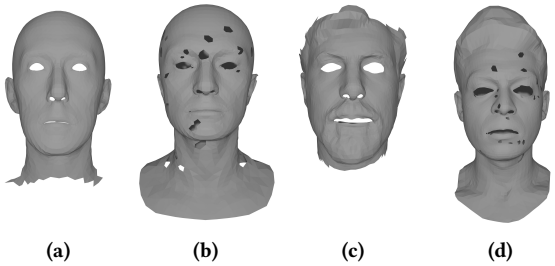


**Figure 6: Data augmentation scheme. (a) and (c) show shift, scale, and applying a front-facial mask. (b)and (d) show holes cutting. (a), (b) are identities from the *ICT* and (c), (d) are from *Multiface*.**

## C STANDARDIZATION

We give two examples of the data standardization treatment. Fig. 7 illustrates the process for an in-the-wild raw scan, while Fig. 8 shows the pipeline for an artist-created mesh.

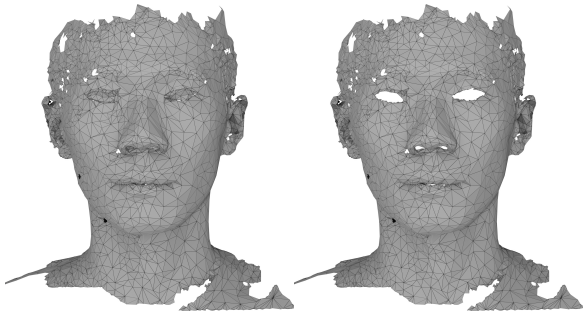

**Figure 7: Left: Raw Scan. Right: Raw Scan after standardization. For an in-the-wild mesh with connected eyes, nose, and mouth, we need to cut those areas to have the correct global solving of the deformation transfer process.**

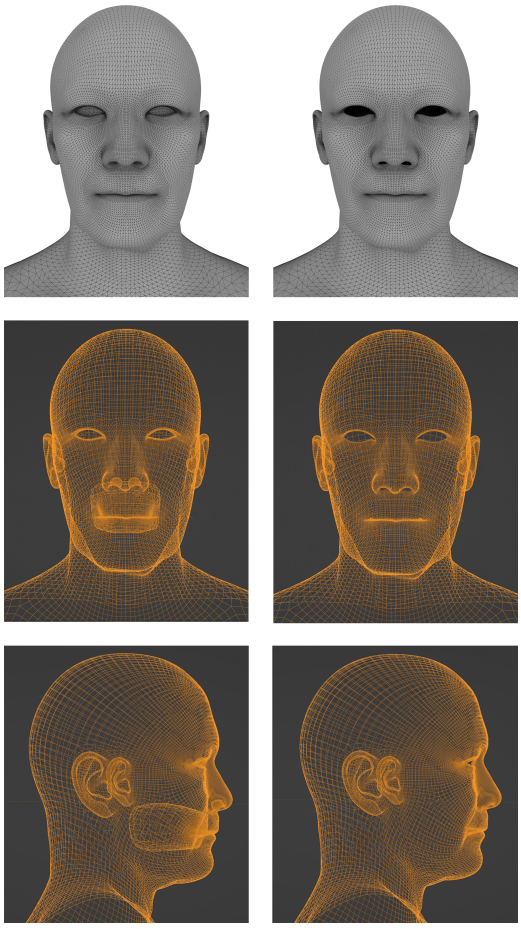

**Figure 8: Left: The original *MetaHuman* mesh. Right: The same mesh after standardization. We remove the inner mouth socket to make it consistent with other meshes during training.**