RigAnyFace: Scaling Neural Facial Mesh Auto-Rigging with Unlabeled Data

Wenchao Ma^{1*†} Dario Kneubuehler^{2*} Maurice Chu² Ian Sachs²
Haomiao Jiang² Sharon X. Huang¹

¹Penn State University ²Roblox

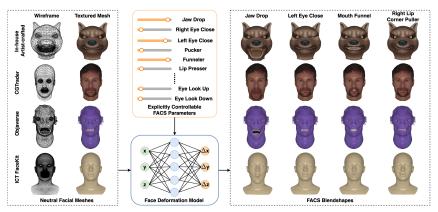


Figure 1: We present RigAnyFace (RAF), an auto-rigging framework that supports facial meshes of diverse topologies with multiple disconnected components such as eyeballs. These meshes are drawn from diverse sources and cover both humanoid and non-humanoid heads. Given only a neutral facial mesh and explicitly controllable FACS parameters specifying activated action units, RAF accurately deforms the input mesh into corresponding FACS poses, creating an expressive blendshape rig.

Abstract

In this paper, we present RigAnyFace (RAF), a scalable neural auto-rigging framework for facial meshes of diverse topologies, including those with multiple disconnected components. RAF deforms a static neutral facial mesh into industry-standard FACS poses to form an expressive blendshape rig. Deformations are predicted by a triangulation-agnostic surface learning network augmented with our tailored architecture design to condition on FACS parameters and efficiently process disconnected components. For training, we curated a dataset of facial meshes, with a subset meticulously rigged by professional artists to serve as accurate 3D ground truth for deformation supervision. Due to the high cost of manual rigging, this subset is limited in size, constraining the generalization ability of models trained exclusively on it. To address this, we design a 2D supervision strategy for unlabeled neutral meshes without rigs. This strategy increases data diversity and allows for scaled training, thereby enhancing the generalization ability of models trained on this augmented data. Extensive experiments demonstrate that RAF is able to rig meshes of diverse topologies on not only our artist-crafted assets but also in-the-wild samples, outperforming previous works in accuracy and generalizability. Moreover, our method advances beyond prior work by supporting multiple disconnected components, such as eyeballs, for more detailed expression animation. Project page: https://wenchao-m.github.io/RigAnyFace.github.io

^{*}Equal Contribution.

[†]Work partially completed during an internship at Roblox.

1 Introduction

Facial rigging aims to make a static neutral facial mesh animatable by defining a set of controllable deformations, typically represented either as blendshape rigs driven by activated action units in FACS-based systems [49, 31, 32, 54, 8, 7, 50] or as skeletal rigs driven by joint positions [64, 27, 48]. This is an essential step for creative AI, bringing digital avatars to life by enabling expressive and realistic facial movements across a wide range of applications. However, creating a rig for facial animation is laborious and expensive, often requiring skilled artists tens of hours to complete a single asset. In this paper, we propose a fully automated and generalizable facial rigging framework that alleviates the reliance on manual labor while achieving high-quality facial rigging.

Typically, facial auto-rigging methods transfer a complete set of blendshapes from a predefined template mesh to a neutral target facial mesh, often necessitating dense correspondences [47, 58, 33, 11] or a fixed mesh topology between the template and the target [34, 6]. Recent approaches [65, 12] utilize per-face VQ-VAEs [65] to build transferable latent spaces between faces or triangulation-agnostic networks [12] to bypass these limitations. However, a template blendshape rig is still required, which can compromise accuracy when the template and target shapes differ substantially. NFR [50] is currently the only approach capable of directly rigging facial meshes from explicitly controllable Facial Action Coding System (FACS) [49] parameters without relying on a template, although it has so far been demonstrated primarily on humanoid heads. Furthermore, existing approaches, including NFR, have yet to accommodate meshes with multiple disconnected components, such as eyeballs or teeth, limiting their ability to animate highly expressive avatars; for example, an "eye lookdown" pose is difficult to reproduce if the mesh lacks eyeballs.

To address the above challenges, we aim to build a facial auto-rigging framework with the following advantages: (i) it eliminates the reliance on predefined template blendshapes, removing the constraint that target facial meshes must rigorously resemble a predefined template; (ii) it is capable of animating in-the-wild facial meshes with varying topologies and shapes, including humanoid and non-humanoid samples as shown in Fig 1; and (iii) it supports facial meshes with multiple disconnected components to enable realistic and expressive 3D face animations.

We present RigAnyFace (RAF), a scalable and generalizable framework for facial auto-rigging. RAF employs a facial mesh deformation network built on DiffusionNet [55], a triangulation-agnostic backbone for meshes of different topologies. Guided by explicitly controllable FACS parameters, this network deforms a neutral facial mesh into a predefined set of FACS poses to form a blendshape rig. Compared to the original DiffusionNet, we introduce two key modifications: (i) a conditional diffusion block that extends the original diffusion block to incorporate FACS parameters as additional conditional inputs, and (ii) a global encoder designed to capture holistic mesh characteristics, enabling effective handling of multiple disconnected components. For network training, we curated a comprehensive dataset of facial meshes encompassing a wide variety of shapes with detailed disconnected components such as eyeballs and teeth. A subset of these meshes was meticulously rigged by professional artists to provide accurate ground-truth for 3D deformations.

Relying solely on rigged heads for training limits the model's generalizability in practice, given the scarcity of rigged samples due to the high cost of manual rigging. This motivates us to employ 2D supervision, which offers better accessibility and broader scalability compared to 3D supervision. We developed a 2D supervision strategy for 3D facial mesh deformation models, integrating appearance guidance from RGB images for prominent facial expressions and motion guidance from optical flow-like 2D displacement field for subtle micro-expressions. Supported by a generative 2D face animation model that synthesizes posed images from the renderings of a neutral mesh, along with an optical flow estimator that predicts the 2D displacement between neutral and posed images as 2D supervisions, we expand the training dataset using unlabeled neutral meshes without rigs. This enables the network to effectively distill rigging knowledge across diverse facial shapes, resulting in more accurate and generalizable 3D facial animations even with limited labeled training data.

Experiments show that our method outperforms prior work across assets from diverse sources, including our artist-crafted meshes and in-the-wild models from ICT FaceKit [36], Objaverse [15], and CGTrader [9]. In addition, we demonstrate several downstream applications of our auto-rigging system in user-controlled animation, retargeting human expressions from videos, and rigging generated facial meshes from a text-to-3D model.

2 Related Works

Auto-rigging. Auto-rigging facilitates efficient and realistic animation of 3D models by automatically generating hierarchical control systems. For full-body character auto-rigging [3, 25, 28, 35, 40, 42, 44, 68, 23, 57, 39, 70, 18], most approaches follow a two-step pipeline: skeleton construction and skinning to generate the Linear Blend Skinning (LBS) rig. In contrast, facial character rigs are often anatomically-inspired, typically based on the Facial Action Coding System (FACS) [49], a standardized framework that describes facial movements as combinations of muscle activations and is primarily implemented using blendshapes [31, 32, 54, 8].

Previous facial auto-rigging works [47, 58, 33, 11, 34, 6, 65, 12] are mostly based on a complete set of blendshapes from a predefined template mesh, transferring the template blendshapes to the target mesh. For example, Li et al.[34] proposed a CNN-based approach that predicts offsets between template and target blendshapes represented by 2D geometry images. Chandran et al. [12] use a transformer with positional encodings to map meshes into a canonical space from user-marked correspondences, enabling deformation transfer from template to target across different topologies. Several notable works [1, 10, 71, 5, 37, 52, 26] can directly generate animatable 3D faces based on 3D Morphable Models [4]. NFR [50] is able to deform a neutral facial mesh into target expressions by decoding FACS-aligned latent codes from a mesh auto-encoder, eliminating any template requirement. Its triangulation-agnostic backbone, trained on several face-animation datasets, generalizes to in-the-wild meshes with diverse topologies. Compared with NFR and other previous works, our method enables 2D supervision for scaled training and further improves accuracy and generalizability across a wider variety of facial meshes while natively supporting multiple disconnected components to allow finer-grained and more realistic expression animation.

Facial Animation Transfer from 2D. Facial animation transfer aims to retarget facial expressions from one character to another. Recent methods (e.g., [51, 2, 13, 30, 29, 45]) show impressive results in transferring expressions to 3D avatars from 2D images or videos. However, these methods focus on transferring expressions to avatars that already have a rig and are hence not directly comparable to our work, which focuses on automatically generating rigs for facial meshes.

Significant progress has also been made in transferring facial animation for both single-view and multi-view images and videos [22, 69, 66, 67, 19, 43, 63, 38, 16, 17, 62, 61]. Given a reference identity image, these methods can generate and manipulate facial expressions for the given identity using various control inputs, such as posed images of other identities or landmarks. Recent advancements in generative models [24, 72] and the availability of large-scale face video datasets [46, 73] have enabled those methods to achieve remarkable success in 2D facial expression animation. For instance, MegActor[69] utilizes a diffusion-based generative framework, incorporating a motion disentanglement module to separate identity and expression features, and a motion retargeting model to map expressions onto target portraits. In this work, we utilize 2D face animation models to generate 2D supervision for unrigged heads. Our proposed framework is agnostic to the choice of 2D face animation model, provided they deliver satisfactory animation results. In practice, we base our 2D supervision generation on MegActor [69], which is open-source and efficient to fine-tune.

3 Preliminary

3.1 DiffusionNet

DiffusionNet [55], proposed by Sharp et al., is a neural network that learns on 3D surfaces by mimicking the intrinsic heat diffusion process. It diffuses per-vertex features across the surface based on the Laplace–Beltrami operator, which captures the intrinsic geometry of the manifold. The resulting heat operator acts as a geometry-aware smoothing filter that blends nearby features over time. In discrete form, DiffusionNet approximates this process using the cotangent Laplacian L and mass matrix M, defined as

$$h_t(u_0) = (M + tL)^{-1} M u_0, (1)$$

where $h_t(u_0)$ represents the diffused features after time t, followed by a lightweight MLPs for non-linearity. Because diffusion depends only on surface intrinsic geometry, the same learned weights transfer across meshes with different resolutions or triangulation, making the model compact, discretization-agnostic, and effective for tasks such as classification and regression on geometric data.

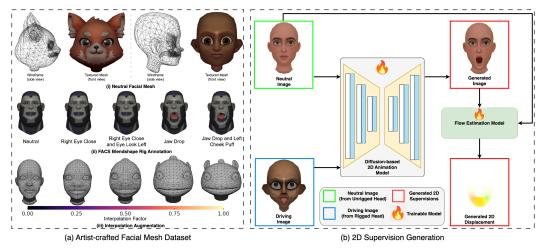


Figure 2: (a) Illustration of our artist-crafted facial mesh dataset. (i) Neutral head meshes from our dataset, each consisting of multiple disconnected components. (ii) A subset of neutral head meshes is meticulously annotated with blendshape rigs by professional artists. (iii) To augment the dataset, we develop a head interpolation strategy based on standardized UV layouts. (b) 2D Supervision Generation Pipeline: Given a posed image rendered from a rigged head and a neutral image from an unrigged head, the 2D animation model generates an image that replicates the expression in the posed image while preserving the identity of the neutral image. A flow estimation model is then applied to the neutral and generated posed images to predict the pixel offsets as 2D displacement.

3.2 Linear FACS Blendshape Rig

The linear FACS blendshape rig [31] models an animatable 3D face using a neutral mesh $M_0 = (V_0, F)$, where V_0 represents the vertex positions and F the mesh connectivity. It also defines a set of N blendshapes $\{M_i = (V_i, F)\}_{i=1}^N$, each obtained by adding a vertex offset d_i to the neutral mesh: $V_i = V_0 + d_i$. Each blendshape corresponds to an Action Unit (AU) from the Facial Action Coding System (FACS) [20], representing specific muscle movements such as "Right Eye Close." Complex facial expression animation, involving the activation of multiple action units, is achieved by assigning a weight $w_i \in [0,1]$ to each blendshape and computing the final mesh M=(V,F), where $V=V_0+\sum_{i=1}^N w_i d_i$.

4 Method

4.1 Data Collection

We collect a diverse set of artist-crafted facial meshes for model training and evaluation. As shown in Fig. 2 (a)(i), our dataset includes facial meshes with multiple disconnected components, such as separate eyeballs and features a variety of shapes, including both humanoid and non-humanoid heads.

Each dataset sample contains a neutral base mesh M_0 . For a select subset, artists manually annotated each mesh with its own complete blendshape rig $\{M_i = (V_i, F)\}_{i=1}^N$ across N FACS training poses, as described in Sec. 3.2 and illustrated in Fig. 2 (a)(ii). We set N=96, comprising 48 FACS poses and 48 corrective poses; further details are provided in the appendix. We also pair each blendshape with a one-hot-like FACS vector A_i as pose representation, where activated action entries are set to 1. Furthermore, those heads were also annotated with facial landmarks specified as vertex indices. For unlabeled heads, only a neutral head mesh $M_0 = (V_0, F)$ is included.

Creating head meshes with complex rigs for animation is an expensive process. In order to expand our dataset sufficiently for training a deep neural network, we developed a data augmentation strategy based on a standardized UV layout, enabling interpolate between different head geometries through linear blending to increases the size of our dataset, as illustrated in Fig. 2 (a)(iii).

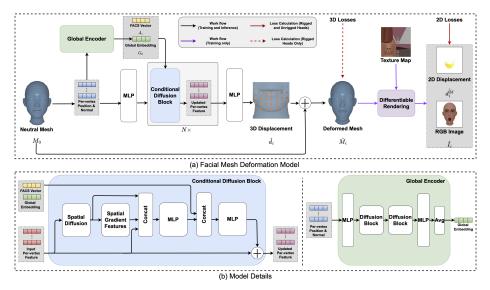


Figure 3: Model Architecture. (a) Given a neutral facial mesh, our deformation model predicts the 3D displacement needed to deform the mesh into different expressions based on the input FACS vector. During training, 2D supervision is utilized for both rigged and unrigged heads, while 3D supervision is exclusively applied to rigged heads. (b) We modify the original diffusion block in DiffusionNet to support the FACS vector as an additional conditional inputs (left). Additionally, we design a global encoder that processes vertex positions and normals of the neutral facial mesh to capture holistic information across disconnected components (right).

4.2 Deformation network

4.2.1 Network Architecture

As shown in Fig. 3 (a), our deformation network takes the neutral facial mesh $M_0 = (V_0, F)$ and a FACS pose vector A_i as inputs and predicts the displacement \hat{d}_i required to deform the neutral mesh into the corresponding posed mesh $\hat{M}_i = (\hat{V}_i, F)$, where $\hat{V}_i = V_0 + \hat{d}_i$. The posed meshes obtained for all FACS poses together form a linear FACS blendshape rig.

We build our deformation network upon DiffusionNet [55] to take advantage of its triangulationagonistic property. However, DiffusionNet struggles to handle multiple disconnected components as its diffusion mechanism cannot propagate information between them. Furthermore, it is limited to processing a single mesh without additional input. In our task, we aim to deform facial meshes with multiple disconnected components conditioned on an additional input: the FACS vector. To this end, we introduce two key modifications to the original DiffusionNet: (i) Global Encoder to capture holistic mesh characteristics across multiple disconnected components. As shown in the right of Fig. 3 (b), this branch consists of a smaller 2-layer DiffusionNet that processes the input neutral mesh. Global average pooling is applied to the final layer's per-vertex features, producing a single vector encoding G_0 that compresses information about the mesh into a global feature vector. (ii) **FACS** Conditioning: We modify the original diffusion block in DiffusionNet to integrate a FACS pose vector as a conditional input, guiding the network's generation of facial expressions. This allows the network to learn the relationship between FACS values and corresponding mesh deformations. As shown on the left of Fig. 3 (b), the FACS pose vector A_i is concatenated with the global feature vector G_0 to create a latent representation. This latent representation is then injected into each conditional diffusion block of the main network. Within each block, the latent vector is replicated across the vertex dimension and fused with the block's output features. This fused information is then processed by a small MLP to refine the mesh's latent features.

4.2.2 2D Supervisions for 3D Deformation Model

Relying solely on fully rigged heads limits the training dataset size due to the scarcity of high-quality 3D ground truth, which hampers generalization to unseen facial meshes. In contrast, 2D supervision is more readily available thanks to advancements in 2D generation models, enabling the inclusion

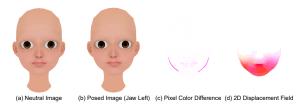


Figure 4: Illustration of our 2D displacement supervision (d), which provides denser feedback for the subtle pose differences between (a) and (b) than the appearance-level supervision (c). Subfigure (c) visualizes per-pixel color-difference magnitudes between (a) and (b), whereas subfigure (d) shows the corresponding pixel offsets using the standard optical-flow color map.

of unrigged heads to scale up the training dataset to enhance generalization. Thus, we introduce 2D supervision for the face auto-rigging network in terms of appearance and motion variation. Specifically, for appearance, we use the front-view image and binary segmentation mask of the posed head as supervision. We render the RGB image \hat{I}_i and binary mask \hat{B}_i of the predicted mesh \hat{M}_i onto the 2D image plane using differentiable rendering [41, 53]. The image loss \mathcal{L}_{img} and mask loss \mathcal{L}_{mask} are defined as the l1 distances between \hat{I}_i with the ground-truth image I_i and between \hat{B}_i with the ground truth mask B_i , respectively.

Using appearance-level supervisions like image and mask losses, provides a straightforward way to optimize the 3D deformation network using 2D supervision. These losses offer strong supervisory signals for poses that result in significant changes in pixel's color value. However, many target FACS poses involve subtle expressions, where changes are less visually apparent. For instance, as shown in Fig. 4, comparing the neutral image in Fig. 4 (a) with the jaw-left pose image in Fig. 4 (b), the differences are barely noticeable to the human eye. Similarly, as illustrated in Fig. 4 (c), the pixel error map on RGB value between these two images highlights that only a small portion of pixels contribute meaningful supervisory feedback for these subtle deformations. In other words, the magnitude of the loss remains minimal—even if the deformation model leaves all vertices fixed in the neutral expression.

To address this challenge, we introduce another 2D supervision for the 3D deformation model based on pixel motions. Specifically, we define the 2D displacement d_i^{2d} as the offset of each pixel on the image plane between the neutral and posed images, analogous to optical flow. This 2D displacement is computed from the 3D displacement d_i in a fully differentiable manner with differentiable rendering [41, 53] (see appendix for implementation). As shown in Fig. 4 (d), the 2D displacement is more distinguishable for subtle facial expressions because it explicitly represents the motion of each pixel in 2D, rather than relying on RGB value changes. This is particularly beneficial in areas with uniform texture, such as cheek, where RGB value changes may be unnoticeable. We define the 2D displacement loss \mathcal{L}_{dis-2d} as the l2 distance between the ground truth 2D displacement d_i^{2d} and predicted 2D displacement d_i^{2d} .

4.2.3 2D Supervision Generation

For all rigged heads, we can obtain the above 2D supervisions by rendering from 3D ground truth. However, for unrigged heads, this is not feasible due to the absence of complete 3D ground truth deformations. To this end, we leverage recent advancements in 2D generation models to generate 2D supervision for unrigged heads. These models effectively distill appearance and motion priors from large-scale 2D image and video datasets, generalizing well across diverse scenarios.

We implement a 2D face animation diffusion model based on Megactor [69]. As illustrated in Fig.2 (b), this model takes a neutral reference image rendered from an unrigged head and a driving posed image rendered from a rigged head, animating the neutral image to replicate the expression in the posed image while preserving its identity. The generated images serve as image-based ground truth for unrigged heads during the training of the 3D deformation model. In practice, we select one rigged head, render all its FACS poses images, and use them as driving images to generate corresponding posed images for all unrigged heads. Ground truth masks are obtained using a traditional image segmentation model[59], as all generated images are with a clean white background.

For the 2D displacement, we use the optical flow estimation model RAFT [60] to predict pixel offsets between the neutral image and the generated posed image of unrigged heads. These offsets serve as the ground truth 2D displacement for training the 3D deformation model.

To enhance the performance of the 2D face animation and flow estimation models on stylized faces in our artist-crafted dataset, we fine-tune their pre-trained weights using the ground truth renderings from a small set of rigged heads, improving effectiveness.

4.2.4 Network Training and Inference

We train the network in a two-stage, coarse-to-fine manner. In the first stage, the 3D deformation network is trained on a large-scale dataset comprising both rigged and unrigged heads, using only 2D supervision. We use a combination of photometric loss and 2D displacement loss, along with a l_2 regularization loss, \mathcal{L}_{reg} on the predicted 3D displacement. This regularization loss helps to improve model convergence speed and prevent "flying points" for non-line-of-sight vertices. The total training loss for the first stage is defined as:

$$\mathcal{L}_{s1} = \alpha_1 \mathcal{L}_{imq} + \alpha_2 \mathcal{L}_{mask} + \alpha_3 \mathcal{L}_{dis-2d}, +\alpha_4 \mathcal{L}_{req}$$
 (2)

where α are weighting parameters for different loss terms.

In the second stage, we fine-tune the pretrained model from the first stage using only rigged heads, incorporating both 2D and 3D supervision to achieve high-precision deformation predictions. Since the 3D ground truth deformed mesh $M_i = (V_i, F)$ for FACS pose i is available for rigged heads, we incorporate 3D supervision by applying the MSE loss \mathcal{L}_{mse-3d} in 3D space between the ground truth and predicted mesh vertices V_i and \hat{V}_i .

For 2D supervision, in addition to the image loss and mask loss, we added two loss terms, landmark loss \mathcal{L}_{lmk} and eye close loss \mathcal{L}_{ec} , as in [21] to provide supervision for specific facial landmarks and poses. We omit the 2D displacement loss in this stage since the 3D displacement ground truth is available. The total training loss for the second stage is defined as:

$$\mathcal{L}_{s2} = \alpha_1 \mathcal{L}_{img} + \alpha_2 \mathcal{L}_{mask} + \alpha_3 \mathcal{L}_{mse-3d} + \alpha_4 \mathcal{L}_{lmk} + \alpha_5 \mathcal{L}_{ec}. \tag{3}$$

The proposed model only consists of 5.4M parameters. Training runs on an instance with 8 NVIDIA A100 GPUs and takes about 2 days. For inference, it takes on average 8.72s on an Apple M2 Max CPU and 3.1s on an Nvidia T4 GPU to generate a FACS blendshape rig on the test set.

5 Experiments

In this section, we evaluate RAF on both the artist-crafted and in-the-wild facial meshes and compare it with the prior art NFR [50] and a representative deformation-transfer method [58].

5.1 Evaluation on Artist-crafted Data

Table 1: Quantitative results on our artist-crafted dataset, validating each component of the model.

	Network	Supervision terms		Training dataset		Test Results (mm)		
	Global encoder	\mathcal{L}_{mse-3d}	\mathcal{L}_{img}	\mathcal{L}_{dis-2d}	Rigged	Unrigged	MAE↓	MAE Q95↓
w/o Global Encoder	×	/	×	×	/	х	2.14	6.64
w/o 2D Loss	/	/	×	×	/	×	2.08	5.84
w/o Unrigged Data	/	/	/	×	/	×	2.01	5.81
w/o 2D Displacement	/	/	/	×	/	/	1.95	5.89
Full Model	/	/	/	/	/	/	1.92	5.63

Table 2: Quantitative comparison with NFR and Deformation Transfer on 12 artist-annotated humanoid heads. (* additional inputs needed)

	MAE (mm) ↓	MAE Q95 (mm) ↓
Deformation Transfer [58]*	2.93	8.41
NFR [50] Ours	2.77 1.01	7.21 2.94

We evaluate our model both quantitatively and qualitatively on our artist-crafted dataset. The evaluation includes two test sets: one with rigged heads for detailed accuracy analysis, and another with unrigged heads featuring diverse species and shapes to assess generalization on out-of-distribution samples, simulating real-world applications.

For rigged heads with 3D ground-truth deformations, we compute the Mean Absolute vertex Error (MAE) and the 95th-percentile vertex error (MAE Q95) to capture challenging cases; both metrics are evaluated over the full set of 96 FACS poses. During evaluation, all facial meshes are normalized to fit within a unit sphere with a radius of 1 meter. Quantitative results are presented in Tab. 1, while

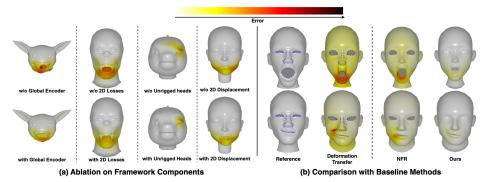


Figure 5: Visual comparisons. Meshes are colored by per-vertex absolute error. (a) Ablation on Framework Components. 1st Col.: without the global encoder, disconnected parts intersect; 2nd Col.: 2D image loss reduces errors; 3rd Col.: additional unrigged heads improve generalization, addressing challenging cases such as animal eye closure; 4th Col.: 2D-displacement loss further refines subtle poses such as "Jaw Left." (b) Comparison with Baseline Methods. Our method achieves more accurate and expressive animation results while handling multiple disconnected components. Reference mesh and corresponding points (marked as blue) are provided for Deformation Transfer.

qualitative results are shown in Fig. 5(a). Together, these results validate the effectiveness of each component in our model.

We also conduct additional ablations on the global encoder to demonstrate how a single feature vector from it enables our model to handle multiple disconnected components. We evaluate penetration between inner components (e.g., teeth) and the outer face surface by reporting the percentage of penetrating vertices with and without the global feature (Tab. 3). Furthermore, we perturb the disconnected components by randomly adding offsets or removing varying numbers of components from each sample. As shown in Fig. 6, the t-SNE visualization of global features from the perturbed samples forms separate clusters from the original ones. These results demonstrate that the global feature effectively encodes both the position and presence of disconnected components, avoiding penetration and achieving accurate deformation.

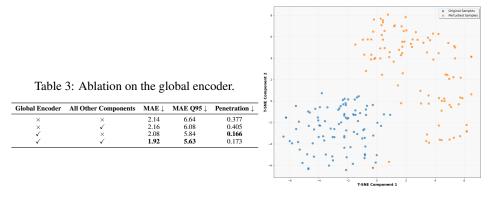


Figure 6: t-SNE visualization of features from the global encoder.

To ensure a fair comparison with NFR [50], each input mesh was preprocessed following the same procedure as in their original implementation, to retain only the largest connected component of the neutral mesh with auxiliary structures (e.g., eyeballs and the mouth socket) removed and the inner surfaces of the lips and eyelids trimmed. Since NFR is trained and evaluated only on humanoid faces, we limit the test cases to 12 humanoid heads. Another method that we compare to is the Deformation Transfer [58], which requires an exemplar expression mesh and user-annotated point correspondences as additional input. We choose one rigged head from the training set as the exemplar, deform it into all FACS poses, and provide artist-annotated landmarks as correspondence points. As reported in Tab. 2 and Fig. 5(b), our method outperforms both baselines by a wide margin. It also has the additional advantage of not requiring any additional input and being able to handle meshes with multiple disconnected components.

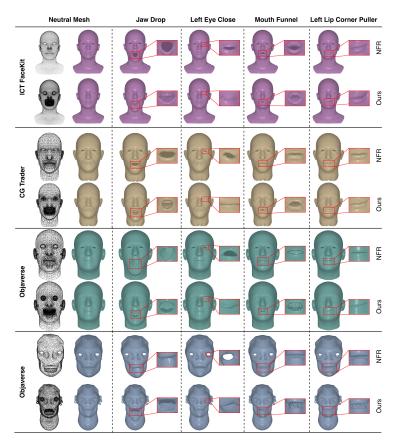


Figure 7: Auto-rigging results on in-the-wild facial meshes compared with NFR [50].

For the unrigged head test set, we provide qualitative results only, as 3D ground truth data is unavailable. Fig. 8 showcases qualitative examples, where our model delivers highly accurate and vivid auto-rigging results across facial meshes of various shapes and styles. Additional results for all FACS poses and more samples can be found in the supplementary materials.



Figure 8: Qualitative results on our artist-crafted unrigged heads.

5.2 Evaluation on In-the-wild Heads

Our method generalizes effectively to in-the-wild facial meshes with diverse topology and shape variations. To demonstrate this, we present qualitative results on samples from ICT FaceKit [36], Objaverse [15, 14], and CGTrader [9], and compare that with the results by NFR [50]. We do not provide a comparison with Deformation Transfer here, as correspondence point annotations are not available for these samples. We similarly preprocess the input meshes for the comparison with NFR. As shown in Fig. 7, our model consistently achieves better accuracy and generalizability. In particular, although NFR was trained on the ICTFaceKit dataset and ours was not, our results are comparable to those of NFR. For humanoid assets from Objaverse and CGTrader, neither our method

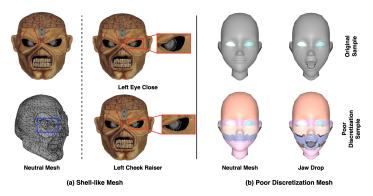


Figure 9: Failure cases of our RAF model.

nor NFR was trained on data from these sources, our model demonstrates superior performance. For the non-humanoid head (last column), NFR leaves it largely undeformed, whereas our model successfully generalizes to this challenging case.

5.3 Applications

We demonstrate three real-world applications of RAF: (i) user-controlled animation, where the predicted FACS rig allows users to pose a mesh by editing FACS parameters; (ii) video-to-mesh retargeting, which transfers expressions of a subject in the video via tracked FACS sequences to an unrigged mesh; (iii) animating a facial mesh generated from a text-to-3D model, turning it from a neutral facial mesh into a fully animatable avatar. Demos can be found on our project page.

6 Conclusion

We propose RAF, a framework for auto-rigging facial meshes. Powered by our tailored design for multiple disconnected components and FACS conditioning and scaled by training on unrigged heads with 2D supervision, RAF can animate meshes of diverse topologies with even multiple disconnected components, across both artist-crafted assets and in-the-wild samples.

Limitations and Future Work. Our model faces challenges in two scenarios: (i) When the input mesh structure deviates significantly from the training data, such as shell-like meshes that lack the fine-grained geometric details necessary for high-quality facial animation, the model's performance may decline (Fig.9 (a)). Expanding the dataset to include a broader range of mesh structures could enhance generalization in such cases. (ii) When the mesh has poor discretization that causes the main facial mesh to break into multiple disconnected components (shown in different colors), our model fails to maintain spatial coherence among these components after deformation (Fig.9 (b)). Incorporating a diffusion operator defined on a high-quality background triangulation [56] could enhance robustness in such cases.

7 Acknowledgement

We thank Hsueh-Ti Derek Liu, Chrystiano Araújo, and Jinseok Bae for proofreading the draft and providing helpful comments, and Jihyun Yoon for curating the dataset. Wenchao Ma is supported by a travel grant from the Penn State College of Information Sciences and Technology for attending the conference.

References

- [1] Victoria Fernández Abrevaya, Adnane Boukhayma, Stefanie Wuhrer, and Edmond Boyer. A decoupled 3d facial shape model by adversarial training. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9418–9427, 2019.
- [2] Deepali Aneja, Bindita Chaudhuri, Alex Colburn, Gary Faigin, Linda G. Shapiro, and Barbara Mones. Learning to generate 3d stylized character expressions from humans. In 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018, pages 160–169. IEEE Computer Society, 2018.
- [3] Ilya Baran and Jovan Popovic. Automatic rigging and animation of 3d characters. *ACM Trans. Graph.*, 26(3):72, 2007.
- [4] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In Warren N. Waggenspack, editor, Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1999, Los Angeles, CA, USA, August 8-13, 1999, pages 187–194. ACM, 1999.
- [5] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Trans. Vis. Comput. Graph.*, 20(3):413–425, 2014.
- [6] Emma Carrigan, Eduard Zell, Cédric Guiard, and Rachel McDonnell. Expression packing: As-few-as-possible training expressions for blendshape transfer. Comput. Graph. Forum, 39(2):219–233, 2020.
- [7] Ozan Cetinaslan and Verónica Orvalho. Sketching manipulators for localized blendshape editing. *Graph. Model.*, 108:101059, 2020.
- [8] Ozan Cetinaslan and Verónica Orvalho. Stabilized blendshape editing using localized jacobian transpose descent. *Graph. Model.*, 112:101091, 2020.
- [9] CGTrader. Cgtrader 3d models for professionals, 2025.
- [10] Prashanth Chandran, Derek Bradley, Markus Gross, and Thabo Beeler. Semantic deep face models. In International Conference on 3D Vision (3DV), pages 345–354. IEEE, 2020.
- [11] Prashanth Chandran, Loïc Ciccone, Markus Gross, and Derek Bradley. Local anatomically-constrained facial performance retargeting. *ACM Trans. Graph.*, 41(4):168:1–168:14, 2022.
- [12] Prashanth Chandran, Gaspard Zoss, Markus Gross, Paulo F. U. Gotardo, and Derek Bradley. Shape transformers: Topology-independent 3d shape models using transformers. *Comput. Graph. Forum*, 41(2):195–207, 2022.
- [13] Byungkuk Choi, Haekwang Eom, Benjamin Mouscadet, Stephen Cullingford, Wan-Duo Kurt Ma, Stefanie Gassel, Suzi Kim, Andrew Moffat, Millicent Maier, Marco Revelant, Joe Letteri, and Karan Singh. Animatomy: an animator-centric, anatomically inspired system for 3d facial modeling, animation and transfer. In SIGGRAPH Asia 2022 Conference Papers, SA, pages 16:1–16:9. ACM, 2022.
- [14] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2023.
- [15] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 13142–13153, 2023.
- [16] Yu Deng, Duomin Wang, Xiaohang Ren, Xingyu Chen, and Baoyuan Wang. Portrait4d: Learning one-shot 4d head avatar synthesis using synthetic data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7119–7130, 2024.
- [17] Yu Deng, Duomin Wang, and Baoyuan Wang. Portrait4d-v2: Pseudo multi-view data creates better 4d head synthesizer. In *European Conference on Computer Vision (ECCV)*, volume 15075 of *Lecture Notes in Computer Science*, pages 316–333. Springer, 2024.
- [18] Yufan Deng, Yuhao Zhang, Chen Geng, Shangzhe Wu, and Jiajun Wu. Anymate: A dataset and baselines for learning 3d object rigging. *arXiv preprint arXiv:2505.06227*, 2025.
- [19] Nikita Drobyshev, Antoni Bigata Casademunt, Konstantinos Vougioukas, Zoe Landgraf, Stavros Petridis, and Maja Pantic. Emoportraits: Emotion-enhanced multimodal one-shot head avatars. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8498–8507, 2024.

- [20] Paul Ekman and Wallace V Friesen. Facial action coding system. Environmental Psychology & Nonverbal Behavior, 1978.
- [21] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3d face model from in-the-wild images. *ACM Trans. Graph.*, 40(4):88:1–88:13, 2021.
- [22] Jianzhu Guo, Dingyun Zhang, Xiaoqiang Liu, Zhizhou Zhong, Yuan Zhang, Pengfei Wan, and Di Zhang. Liveportrait: Efficient portrait animation with stitching and retargeting control. CoRR, abs/2407.03168, 2024.
- [23] Zhiyang Guo, Jinxu Xiang, Kai Ma, Wengang Zhou, Houqiang Li, and Ran Zhang. Make-it-animatable: An efficient framework for authoring animation-ready 3d characters. *CoRR*, abs/2411.18197, 2024.
- [24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.
- [25] Alec Jacobson, Ilya Baran, Jovan Popovic, and Olga Sorkine. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.*, 30(4):78, 2011.
- [26] Zi-Hang Jiang, Qianyi Wu, Keyu Chen, and Juyong Zhang. Disentangled representation learning for 3d face shape. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11957–11966, 2019.
- [27] Ladislav Kavan, John Doublestein, Martin Prazák, Matthew Cioffi, and Doug Roble. Compressed skinning for facial blendshapes. In ACM SIGGRAPH 2024 Conference Papers, SIGGRAPH, page 47. ACM, 2024.
- [28] Ladislav Kavan and Olga Sorkine. Elasticity-inspired deformers for character articulation. *ACM Trans. Graph.*, 31(6):196:1–196:8, 2012.
- [29] Seonghyeon Kim, Sunjin Jung, Kwanggyoon Seo, Roger Blanco i Ribera, and Junyong Noh. Deep learning-based unsupervised human facial retargeting. *Comput. Graph. Forum*, 40(7):45–55, 2021.
- [30] Ariel Larey, Omri Asraf, Adam Kelder, Itzik Wilf, Ofer Kruzel, and Nati Daniel. Facial expression re-targeting from a single character. *CoRR*, abs/2306.12188, 2023.
- [31] John P. Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Frédéric H. Pighin, and Zhigang Deng. Practice and theory of blendshape facial models. In Sylvain Lefebvre and Michela Spagnuolo, editors, 35th Annual Conference of the European Association for Computer Graphics, Eurographics, pages 199–218. Eurographics Association, 2014.
- [32] John P. Lewis and Ken-ichi Anjyo. Direct manipulation blendshapes. *IEEE Computer Graphics and Applications*, 30(4):42–50, 2010.
- [33] Hao Li, Thibaut Weise, and Mark Pauly. Example-based facial rigging. *ACM Trans. Graph.*, 29(4):32:1–32:6, 2010.
- [34] Jiaman Li, Zhengfei Kuang, Yajie Zhao, Mingming He, Karl Bladin, and Hao Li. Dynamic facial asset and rig generation from a single scan. *ACM Trans. Graph.*, 39(6):215:1–215:18, 2020.
- [35] Peizhuo Li, Kfir Aberman, Rana Hanocka, Libin Liu, Olga Sorkine-Hornung, and Baoquan Chen. Learning skeletal articulations with neural blend shapes. *ACM Trans. Graph.*, 40(4):130:1–130:15, 2021.
- [36] Ruilong Li, Karl Bladin, Yajie Zhao, Chinmay Chinara, Owen Ingraham, Pengda Xiang, Xinglei Ren, Pratusha Prasad, Bipin Kishore, Jun Xing, and Hao Li. Learning formation of physically-based face attributes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3407–3416, 2020.
- [37] 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):194:1–194:17, 2017.
- [38] Xueting Li, Shalini De Mello, Sifei Liu, Koki Nagano, Umar Iqbal, and Jan Kautz. Generalizable one-shot 3d neural head avatar. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2023.
- [39] Isabella Liu, Zhan Xu, Wang Yifan, Hao Tan, Zexiang Xu, Xiaolong Wang, Hao Su, and Zifan Shi. Riganything: Template-free autoregressive rigging for diverse 3d assets. *CoRR*, abs/2502.09615, 2025.
- [40] Lijuan Liu, Youyi Zheng, Di Tang, Yi Yuan, Changjie Fan, and Kun Zhou. Neuroskinning: automatic skin binding for production characters with deep graph networks. ACM Trans. Graph., 38(4):114:1–114:12, 2019.

- [41] Shichen Liu, Weikai Chen, Tianye Li, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7707–7716. IEEE, 2019.
- [42] Jing Ma and Dongliang Zhang. Tarig: Adaptive template-aware neural rigging for humanoid characters. *Comput. Graph.*, 114:158–167, 2023.
- [43] Yue Ma, Hongyu Liu, Hongfa Wang, Heng Pan, Yingqing He, Junkun Yuan, Ailing Zeng, Chengfei Cai, Heung-Yeung Shum, Wei Liu, and Qifeng Chen. Follow-your-emoji: Fine-controllable and expressive freestyle portrait animation. In SIGGRAPH Asia 2024 Conference Papers, pages 110:1–110:12. ACM, 2024.
- [44] Albert Mosella-Montoro and Javier Ruiz Hidalgo. Skinningnet: Two-stream graph convolutional neural network for skinning prediction of synthetic characters. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022, pages 18572–18581. IEEE, 2022.
- [45] Lucio Moser, Chinyu Chien, Mark Williams, Jose Serra, Darren Hendler, and Doug Roble. Semi-supervised video-driven facial animation transfer for production. ACM Trans. Graph., 40(6):222:1–222:18, 2021.
- [46] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: A large-scale speaker identification dataset. In 18th Annual Conference of the International Speech Communication Association, Interspeech 2017, Stockholm, Sweden, August 20-24, 2017, pages 2616–2620. ISCA, 2017.
- [47] Jun-yong Noh and Ulrich Neumann. Expression cloning. In Lynn Pocock, editor, Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), pages 277–288. ACM, 2001.
- [48] Verónica Costa Teixeira Orvalho, Ernesto Zacur, and Antonio Susín. Transferring the rig and animations from a character to different face models. *Comput. Graph. Forum*, 27(8):1997–2012, 2008.
- [49] Ekman Paul and Wallace V Friesen. Facial action coding system: a technique for the measurement of facial movement. Consulting Psychologists, 4, 1978.
- [50] 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, SIGGRAPH, pages 68:1–68:11, 2023.
- [51] Feng Qiu, Wei Zhang, Chen Liu, Rudong An, Lincheng Li, Yu Ding, Changjie Fan, Zhipeng Hu, and Xin Yu. Freeavatar: Robust 3d facial animation transfer by learning an expression foundation model. In SIGGRAPH Asia 2024 Conference Papers, SA '24, page 1–11. ACM, December 2024.
- [52] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3d faces using convolutional mesh autoencoders. In *European Conference on Computer Vision (ECCV)*, volume 11207 of *Lecture Notes in Computer Science*, pages 725–741. Springer, 2018.
- [53] Nikhila Ravi, Jeremy Reizenstein, David Novotný, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. CoRR, abs/2007.08501, 2020.
- [54] Yeongho Seol, Jaewoo Seo, Paul Hyunjin Kim, John P. Lewis, and Junyong Noh. Artist friendly facial animation retargeting. ACM Trans. Graph., 30(6):162, 2011.
- [55] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. Diffusionnet: Discretization agnostic learning on surfaces. *ACM Trans. Graph.*, 41(3):27:1–27:16, 2022.
- [56] Nicholas Sharp, Yousuf Soliman, and Keenan Crane. Navigating intrinsic triangulations. *ACM Trans. Graph.*, 38(4):55:1–55:16, 2019.
- [57] Chaoyue Song, Jianfeng Zhang, Xiu Li, Fan Yang, Yiwen Chen, Zhongcong Xu, Jun Hao Liew, Xiaoyang Guo, Fayao Liu, Jiashi Feng, and Guosheng Lin. Magicarticulate: Make your 3d models articulation-ready. CoRR, abs/2502.12135, 2025.
- [58] Robert W. Sumner and Jovan Popovic. Deformation transfer for triangle meshes. ACM Trans. Graph., 23(3):399–405, 2004.
- [59] Satoshi Suzuki and Keiichi Abe. Topological structural analysis of digitized binary images by border following. *Comput. Vis. Graph. Image Process.*, 30(1):32–46, 1985.

- [60] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, volume 12347, pages 402–419. Springer, 2020.
- [61] Kartik Teotia, Hyeongwoo Kim, Pablo Garrido, Marc Habermann, Mohamed Elgharib, and Christian Theobalt. Gaussianheads: End-to-end learning of drivable gaussian head avatars from coarse-to-fine representations. *ACM Trans. Graph.*, 43(6):264:1–264:12, 2024.
- [62] Phong Tran, Egor Zakharov, Long-Nhat Ho, Adilbek Karmanov, Ariana Bermudez Venegas, McLean Goldwhite, Aviral Agarwal, Liwen Hu, Anh Tran, and Hao Li. VOODOO XP: expressive one-shot head reenactment for VR telepresence. ACM Trans. Graph., 43(6):253:1–253:26, 2024.
- [63] Phong Tran, Egor Zakharov, Long-Nhat Ho, Anh Tuan Tran, Liwen Hu, and Hao Li. VOODOO 3d: Volumetric portrait disentanglement for one-shot 3d head reenactment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10336–10348. IEEE, 2024.
- [64] Noranart Vesdapunt, Mitch Rundle, HsiangTao Wu, and Baoyuan Wang. JNR: joint-based neural rig representation for compact 3d face modeling. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *European Conference on Computer Vision (ECCV)*, volume 12363, pages 389–405. Springer, 2020.
- [65] Jingying Wang, Yilin Qiu, Keyu Chen, Yu Ding, and Ye Pan. Fully automatic blendshape generation for stylized characters. In *IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pages 347–355, 2023
- [66] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10039–10049, 2021.
- [67] You Xie, Hongyi Xu, Guoxian Song, Chao Wang, Yichun Shi, and Linjie Luo. X-portrait: Expressive portrait animation with hierarchical motion attention. In ACM SIGGRAPH 2024 Conference Papers, page 115, 2024.
- [68] Zhan Xu, Yang Zhou, Evangelos Kalogerakis, Chris Landreth, and Karan Singh. Rignet: neural rigging for articulated characters. ACM Trans. Graph., 39(4):58, 2020.
- [69] Shurong Yang, Huadong Li, Juhao Wu, Minhao Jing, Linze Li, Renhe Ji, Jiajun Liang, and Haoqiang Fan. Megactor: Harness the power of raw video for vivid portrait animation. CoRR, abs/2405.20851, 2024.
- [70] Jia-Peng Zhang, Cheng-Feng Pu, Meng-Hao Guo, Yan-Pei Cao, and Shi-Min Hu. One model to rig them all: Diverse skeleton rigging with unirig. *arXiv* preprint arXiv:2504.12451, 2025.
- [71] Longwen Zhang, Qiwei Qiu, Hongyang Lin, Qixuan Zhang, Cheng Shi, Wei Yang, Ye Shi, Sibei Yang, Lan Xu, and Jingyi Yu. Dreamface: Progressive generation of animatable 3d faces under text guidance. *ACM Trans. Graph.*, 42(4):138:1–138:16, 2023.
- [72] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3813–3824, 2023.
- [73] Hao Zhu, Wayne Wu, Wentao Zhu, Liming Jiang, Siwei Tang, Li Zhang, Ziwei Liu, and Chen Change Loy. Celebv-hq: A large-scale video facial attributes dataset. In *European Conference on Computer Vision* (*ECCV*), volume 13667, pages 650–667. Springer, 2022.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: We outline the scope, key contributions, and experimental results of this work in the abstract and introduction.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions
 made in the paper and important assumptions and limitations. A No or NA answer to this
 question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discuss the limitations of this work in the paper.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how
 they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems
 of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers
 as grounds for rejection, a worse outcome might be that reviewers discover limitations that
 aren't acknowledged in the paper. The authors should use their best judgment and recognize
 that individual actions in favor of transparency play an important role in developing norms that
 preserve the integrity of the community. Reviewers will be specifically instructed to not penalize
 honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA].

Justification: This paper does not include theoretical results.

- The answer NA means that the paper does not include theoretical results.
- All theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: Yes

Justification: We clearly detail the algorithm and implementation.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions
 to provide some reasonable avenue for reproducibility, which may depend on the nature of the
 contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No].

Justification: Upon acceptance, we will release our inference code and model weights so that reported evaluation results can be reproduced. We are also working to make the full dataset and training code publicly available, pending internal approval and legal clearance.

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide enough details for understanding the results in the main paper and appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Not applicable for this task.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report
 a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is
 not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Yes, we include this in the paper and appendix.

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

• The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: We adhere to the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss societal impact in the Appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or
 why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used
 as intended and functioning correctly, harms that could arise when the technology is being used
 as intended but gives incorrect results, and harms following from (intentional or unintentional)
 misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies
 (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the
 efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper poses no such risks.

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require
 this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the code and datasets used and comply with their licenses.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA] .

Justification: This paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an
 anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA] .

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the
 paper involves human subjects, then as much detail as possible should be included in the main
 paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA].

Justification: This research does not involve LLMs as any important, original, or non-standard components.

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Training and Inference Details

In the first stage of training, the weights for the image loss, mask loss, 2D displacement loss, and regularization loss are set to 10.0, 1.0, 1.0, and 0.0001, respectively. In the second stage, the weights for the image loss, mask loss, 3D MSE loss, 2D landmark loss, and 2D eye closure loss are set to 10.0, 1.0, 100.0, 0.5, and 0.5, respectively. We train our model on an Nvidia A100 instance with 8 GPUs and a total batch size of 8 (i.e., effectively 1 sample per GPU if using distributed data parallel). The training proceeds in two stages. For the first stage, we train the deformation model on both rigged and unrigged head datasets (8,386 samples in total) using only 2D supervision for 15 epochs. This stage typically takes around 1.5 days to complete. For the second stage, we then finetune the model from the first stage on the rigged head dataset (2,929 samples), incorporating both 2D and 3D supervision for 20 epochs. This finetuning phase finishes in approximately 1 day. Throughout both stages, we use the Adam optimizer, initializing the learning rate at 0.0001. For learning rate scheduling, we employ CosineAnnealingWarmRestarts, allowing it to decay from 0.0001 to nearly 0 by the end of training. Additionally, we use a warm-up phase of 20,000 steps to stabilize early training.

For inference speed, our model runs a single forward pass to predict blendshapes offline, requiring only one run per input mesh. The outputs are converted into classical FACS blendshape rigs, enabling efficient animation by simply linear blending. The proposed model consists of 5.4M parameters and it takes on average 8.72s on an Apple M2 Max CPU and 3.1s on a Nvidia T4 GPU to generate a FACS blendshape rig on the test set (1,750 vertices, 3,362 faces on average).

B Details for 2D Displacement Calculation

In the following code sample, we demonstrate how to compute the 2D displacement of each pixel from mesh vertex deformations in a fully differentiable manner. This implementation leverages PyTorch3D's differentiable rendering functionality.

```
def render_displacement(vertices, deformed_vertices, faces, renderer, camera, res
    =(512,512)):
   Parameters
   vertices: torch.tensor (V, 3)
   deformed_vertices: torch.tensor (V, 3)
   faces: torch.tensor (F, 3)
   renderer: pytorch3d.renderer.MeshRenderer object
   camera: pytorch3d.renderer.cameras.CamerasBase object
   res: tuple
   Returns
   displacement_2D: torch.tensor (res[0], res[1], 2)
   verts_2d = camera.transform_points_screen(vertices, image_size=res)
   verts_2d_deformed = camera.transform_points_screen(deformed_vertices, image_size
       =res)
   verts_flow = (verts_2d_deformed - verts_2d)[:, :2] # Vx2
   verts_flow = verts_flow / res * 0.5 + 0.5 # 0~1
   flow_tex = torch.nn.functional.pad(verts_flow, pad=[0, 1]) # Vx3
   texture = TexturesVertex(verts_features=[flow_tex])
   meshes = pytorch3d.structures.Meshes(
       verts=[vertices], faces=[faces], textures=texture
   displacement_2D = renderer(meshes, cameras=camera)
   return displacement_2D[...,:2].squeeze()
```

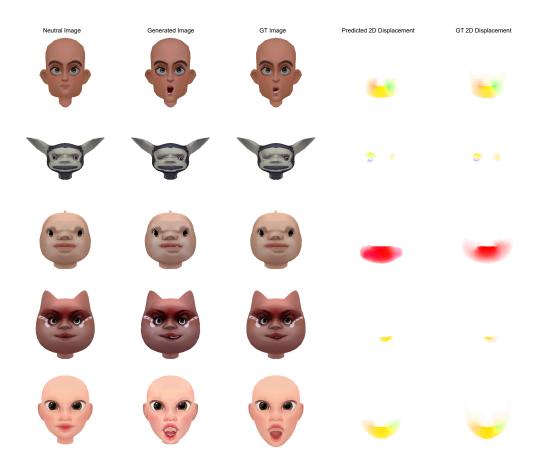


Figure 10: Example results of 2D generation pipeline.

C Effectiveness of 2D Generation Pipeline

To validate the effectiveness of our 2D supervision generation pipeline, we exclude several rigged heads during the fine-tuning of the 2D face animation model and the flow estimation model. In Fig. 10, we present random sample results showcasing different faces and poses. The ground truth images and 2D displacements are rendered using the ground truth deformations of the rigged heads. The 2D face animation model generates pose images based on the neutral image input, while the flow estimation model takes the neutral and generated images as input to predict the 2D displacement. The 2D displacement is visualized according to the standard optical flow convention.

D Data Collection Details

FACS Poses For each rigged heads, our artist annotated 48 FACS poses and 48 corrective poses as blendshape rig. We show those 48 FACS poses in Tab. 4, and Fig 11. In addition to blendshapes for individual FACS poses, we generate corrective blendshapes by linearly combining certain poses and manually correcting artifacts. These corrective blendshapes account for the complex deformations resulting from pose interactions.

Semantic Annotation We provide a semantic annotation map for rigged heads, labeling different regions on the mesh (e.g., ears, mouth, eyes), along with facial landmark annotations specified as vertex indices. These annotations allow for the application of weighted losses or region-specific training objectives.

Head Interpolation First, we standardized the UV layout across all head meshes, ensuring that corresponding facial features like eyes and mouths occupy the same region in UV space. This consistent mapping enables the identification of 3D correspondences between vertices on different meshes. Using these correspondences, we can smoothly interpolate between different head geometries through linear blending to significantly increases the size of our dataset.

#	SHORT	FULL	#	SHORT	FULL	
1	neutral	neutral		1_EC	LeftEyeClosed	
2	c_COR	Corrugator		1_EULR	LeftEyeUpperLidRaiser	
3	c_CR	ChinRaiser		1_IBR	LeftInnerBrowRaiser	
4	c_CRUL	ChinRaiserUpperLip	28	1_LCD	LeftLipCornerDown	
5	c_ELD	EyesLookDown	29	1_LCP	LeftLipCornerPuller	
6	c_ELL	EyesLookLeft	30	1_LLD	LeftLowerLipDepressor	
7	c_ELR	EyesLookRight	31	1_LS	LeftLipStretcher	
8	c_ELU	EyesLookUp	32	1_NW	LeftNoseWrinkler	
9	c_FN	Funneler	33	1_OBR	LeftOuterBrowRaiser	
10	c_FP	FlatPucker	34	1_ULR	LeftUpperLipRaiser	
11	c_JD	JawDrop	35	r_BL	RightBrowLowerer	
12	c_JL	JawLeft	36	r_CHP	RightCheekPuff	
13	c_JR	JawRight	37	r_CHR	RightCheekRaiser	
14	c_LLS	LowerLipSuck	38	r_DM	RightDimpler	
15	c_LP	LipPresser	39	r_EC	RightEyeClosed	
16	c_LPT	LipsTogether	40	r_EULR	RightEyeUpperLidRaiser	
17	c_ML	MouthLeft	41	r_IBR	RightInnerBrowRaiser	
18	c_MR	MouthRight	42	r_LCD	RightLipCornerDown	
19	c_PK	Pucker	43	r_LCP	RightLipCornerPuller	
20	c_ULS	UpperLipSuck	44	r_LLD	RightLowerLipDepressor	
21	1_BL	LeftBrowLowerer	45	r_LS	RightLipStretcher	
22	1_CHP	LeftCheekPuff	46	r_NW	RightNoseWrinkler	
23	1_CHR	LeftCheekRaiser	47	r_OBR	RightOuterBrowRaiser	
24	l_DM	LeftDimpler	48	r_ULR	RightUpperLipRaiser	

Table 4: FACS Short and Full Name Mapping.

E Dataset Split

Our dataset includes 161 rigged heads and 175 unrigged heads. From these, a subset of 24 rigged heads with 3D ground-truth annotations forms the test set to for accurate absolute error evaluation. Additionally, we select 37 diverse unrigged heads as the test set, representing different species and shapes to evaluate the model's generalization on out-of-distribution (OOD) faces. For training, we augment the dataset using interpolations, manually filtering out poor interpolation results. Specifically, we interpolate the remaining 137 unrigged heads with a factor of 50, generating 5,457 samples, and interpolate the remaining 137 rigged heads with a factor of 25, producing 2,929 samples.

F Pre-processing for Baseline Method NFR

All NFR baseline results were obtained after applying the official preprocessing pipeline³: we keep only the largest connected component and remove the inner-lip and eyelid surfaces. These steps are crucial for NFR to generate reasonable deformations. Figure 12 shows that retaining multiple disconnected components causes self-penetration, while Fig 13 shows jarring artifacts when the inner-lip surfaces are not trimmed. In contrast, our method do not need such preprocessing.

³https://github.com/dafei-qin/NFR_pytorch

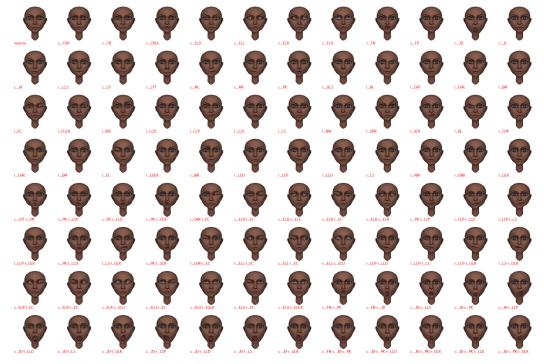


Figure 11: A sample of all the FACS and corrective poses used in this work.

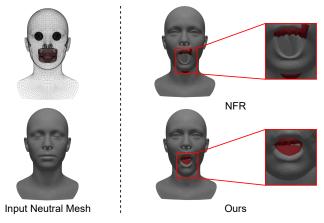


Figure 12: Compared to NFR during inference on meshes with multiple disconnected components from ICT Facekit Dataset. We highlight one of these components: "gums and tongue" in red. While animating a Jaw Drop pose, this component causes penetration issues for NFR.

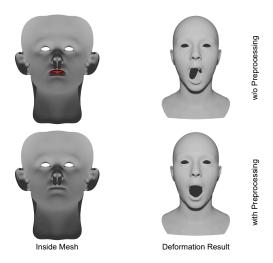


Figure 13: Illustration of the trimming preprocessing step for NFR. The inner-lip surfaces to be trimmed are highlighted in red in the top-left figure. Omitting this step results in implausible deformations produced by NFR.

G Border Impact

Our face-autorigging framework could broaden access to high-quality animation by letting small studios, educators, and assistive-tech developers create expressive avatars quickly, which benefits entertainment, remote communication, and certain medical visualization tasks. However, the same ease of use can lower the barrier for deepfake production, intensifying privacy concerns around emotion tracking and biometric profiling. Careful dataset curation, explicit usage licenses, and watermarking tools are essential to realize the creative upside while limiting misuse and inequitable impacts.