# *FaceLift*: Learning Generalizable Single Image 3D Face Reconstruction from Synthetic Heads

Supplementary Material

## 1. Overview

This supplementary material presents additional results to complement the main manuscript. We first provide a supplementary video showcasing additional visual results. We then provide further experiments in Sec. 3, including a comparison with DimensionX [55], additional visual results of *FaceLift* on in-the-wild images, additional ablation study results and an autoregressive generation pipeline to apply *FaceLift* on videos to achieve 4D rendering. We deliver more details on our method in Sec. 4 and illustrate experimental details in Sec. 5. Finally, we discuss the limitations of *FaceLift* in Sec. 6.

## 2. Supplementary Video

Please refer to our supplementary video for a more comprehensive visualization of the results. The video includes additional examples of single-image-to-3D head reconstruction, demonstrations in the interactive viewer, and results showcasing video-based input for 4D novel view synthesis.

## 3. Additional Experiments

### 3.1. Comparison with DimensionX

We provide additional comparison results on single image to 3D tasks with a state-of-the-art video diffusion model, DimensionX [55]. DimensionX is a framework designed to generate photorealistic 3D and 4D scenes from a single image with video diffusion. The results are shown in Fig. 13. As a video diffusion model, DimensionX struggles to produce multi-view consistent results and lacks a clear spatial understanding of head shapes. As a result, it often generates eyes gazing in the wrong direction and ears positioned incorrectly, along with inaccurate shoulder shapes. In contrast, *FaceLift* generates highly realistic 3D human heads while also producing more visually striking hair.

### 3.2. Comparison with Mesh-based Reconstruction Methods

We provide comparison results with mesh-based reconstruction methods InstantMesh [70], Unique3D [67], and TRELLIS [69] on the Cafca dataset [8]. Quantitative results are shown in Tab. 3, and quantitative comparisons are shown in Fig. 14. Results show that mesh-based reconstruction methods fail to provide realistic hair texture and detailed skin wrinkles. Meanwhile, thanks to the input view reconstruction strategy, *FaceLift* achieves superior identity preservation.

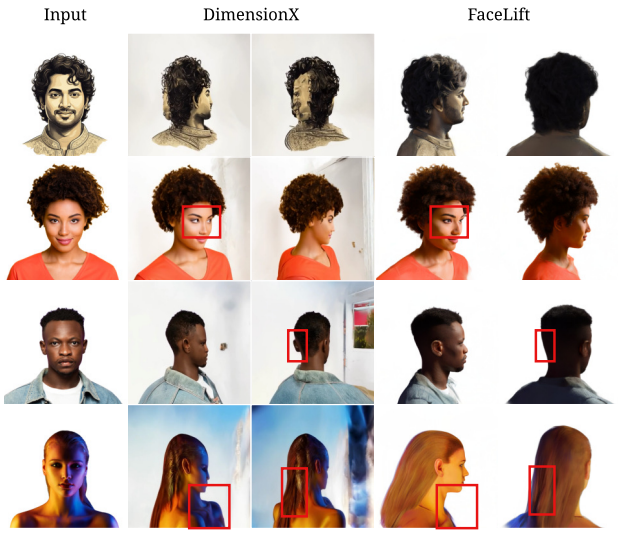


Figure 13. **Visual comparison with DimensionX [55].** DimensionX frequently produces inaccuracies in the back of the head and the shoulder shapes. Other common issues include misaligned ears and eyes gazing in incorrect directions. Additionally, controlling camera poses is challenging. In contrast, *FaceLift* delivers results that are significantly more consistent across multiple views while enabling the generation of more visually appealing hair.


Figure 14. **Visual results on Cafca compared with mesh-based reconstruction methods.** Compared to mesh-based reconstruction methods, our use of pixel-aligned 3D Gaussians offers clear advantages: the semi-transparent kernels naturally capture complex visual phenomena such as hair strands and fine wrinkles.

### 3.3. Additional Results on In-the-wild Images

We present additional results on in-the-wild images in Fig. 24, Fig. 25 and Fig. 26. *FaceLift* demonstrates the ability to effectively handle diverse hairstyles and beards. Notably, it excels at hallucinating unobservable hairline splits and synthesizing the transparent properties of hair using Gaussians with low opacity. Our method reconstructs photo-realistic 3D heads under various lighting conditions

| Method | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DreamSim ↓ | ArcFace ↓ |
|---|---|---|---|---|---|
| TRELLIS [69] | 12.74 | 0.7412 | 0.3746 | 0.2170 | 0.4001 |
| Unique3D [67] | 14.27 | 0.7643 | 0.3188 | 0.1277 | 0.2088 |
| InstantMesh [70] | 16.44 | 0.7815 | 0.2792 | 0.1504 | 0.2741 |
| *FaceLift* | **16.61** | **0.7968** | **0.2694** | **0.1096** | **0.1573** |

Table 3. **Quantitative results on Cafca compared with mesh-based reconstruction methods.** *FaceLift* achieves better quantitative results with more suitable representations and specialized training data.

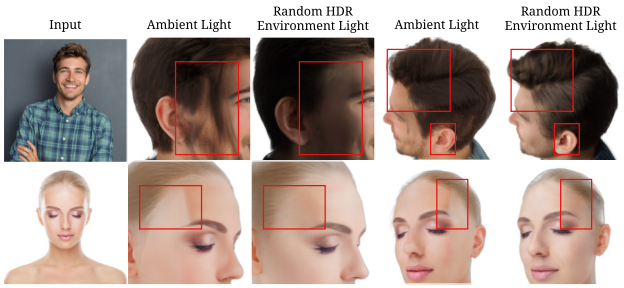


Figure 15. **Ablation study on synthetic data lighting condition.** Models trained only with ambient light struggle to handle shadows and strong lighting.

and can be further extended to the reconstruction of cartoon characters.

### 3.4. Additional Ablation Study

**Importance of Data with Diverse Lighting.** We use synthetic data to train our models, which offers the advantage of controlling lighting conditions and rendering head images under various lighting scenarios. In contrast, real-world human data is typically captured in a studio with lighting similar to ambient light, as shown in the input of Fig. 2. To highlight the importance of training models with diverse lighting conditions, we train *FaceLift* with (1) Data rendered with only ambient light, and (2) Data rendered in random HDR environment light. We present the visual result comparison in Fig. 15. The model trained exclusively on ambient light data struggles to understand shadows, often generating hair-like textures on the face. Furthermore, when exposed to strong light, it produces white regions on the face. In contrast, the model trained with random HDR environment light generates smooth transitions between regions with different lighting conditions.

**More Results on Input View Reconstruction.** We show training samples for two baselines *w/o. Input View Reconstruction* and *w. Input View Reconstruction* in Fig. 16. As the target views are different, baseline *w/o. Input View Reconstruction* is trained to generate six images with novel camera poses, while baseline *w. Input View Reconstruction* reconstruct the input image and generate five images with novel poses. Inference results on real world images

| Baseline | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DreamSim ↓ | ArcFace ↓ |
|---|---|---|---|---|---|
| w/o Input View Reconstruction | 16.02 | 0.7884 | 0.2893 | 0.1438 | 0.2367 |
| w/o Multi-view Attention | 16.29 | 0.7885 | 0.2861 | 0.1552 | 0.2126 |
| Full Model | **16.61** | **0.7968** | **0.2694** | **0.1096** | **0.1573** |

Table 4. **Quantitative results of ablation studies.** *FaceLift* achieves better quantitative results with more suitable representations and specialized training data.

are displayed in Fig. 17 to illustrate the importance of reconstructing the input image during multi-view diffusion training. The results demonstrate that input view regeneration prevents the model from being confined to the training data distribution, thereby enhancing its ability to preserve identity. Quantitative results of baseline *w/o. Input View Reconstruction* is shown in Tab. 4.

### 3.5. Applying *FaceLift* on Videos

*FaceLift* can be directly applied to video frames and achieve high-quality facial reconstructions with consistent visual identity and accurate facial expression, as shown in Fig. 18. However, since *FaceLift* is not trained on video data, many full-head details are generated independently by the diffusion models, resulting in subtle flickering. In this supplemental document, we introduce a simple yet effective method that leverages *FaceLift* and autoregressive training to achieve high-quality, temporally smooth 4D facial reconstructions.

Given an input video $\{F_0, F_1, \dots F_T\}$, we process each video frame $F_t$ sequentially to generate a set of 3D Gaussian sequences $\{G_0, G_1, \dots G_T\}$, where each $G_t$ represents the obtained Gaussian representation at timestamp $t$. As each $G_t$ is generated from frame $I_t$ without interaction with other frames, directly rendering from this Gaussian sequence creates artifacts resulting from time-inconsistency. Hence, we propose an autoregressive generation pipeline, as shown in Fig. 19.

We first select an anchor frame at timestamp $t$ (marked with blue box), and treat its corresponding 3D Gaussian splats as the canonical Gaussians $G_t$ (marked with blue box). Then, for a following timestamp $t+1$, we train a deformation network $D_t$ to predict Gaussian splats $G'_{t+1}$ deformed from $G_t$ supervised by rendering results from $G_t$. The deformation network is an 8-layer MLP, which takes the $x, y, z$ position of each Gaussian in $G_t$ as input and predicts $\Delta x$, $\Delta y$, $\Delta z$, opacity change $\Delta\alpha$ and scale change $\Delta s$. These deformation parameters are combined with $G_t$ to generate $G'_{t+1}$, as shown in Fig. 20.

To train the deformation network, we render six views with the same camera poses as the multi-view diffusion outputs from $G'_{t+1}$, and the renderings of the same camera poses from $G_{t+1}$ are used as pseudo ground truth supervision. Then we treat $G'_{t+1}$ as the initial Gaussians and

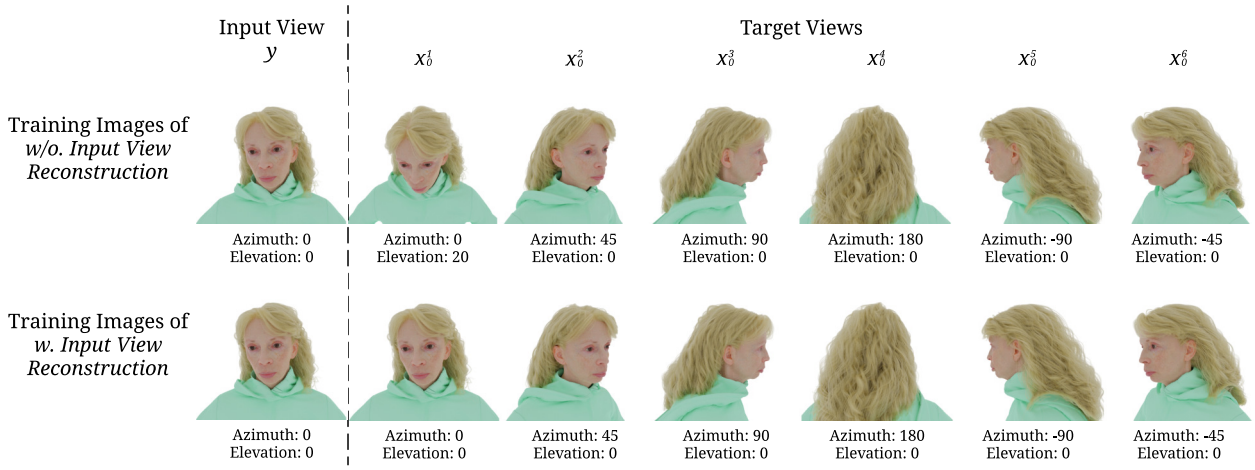


Figure 16. **Training images used in the study of input view reconstruction.** We show example images for training baselines *w/o. Input View Reconstruction* and *w. Input View Reconstruction*. The difference lies in the elevation of the first target image.

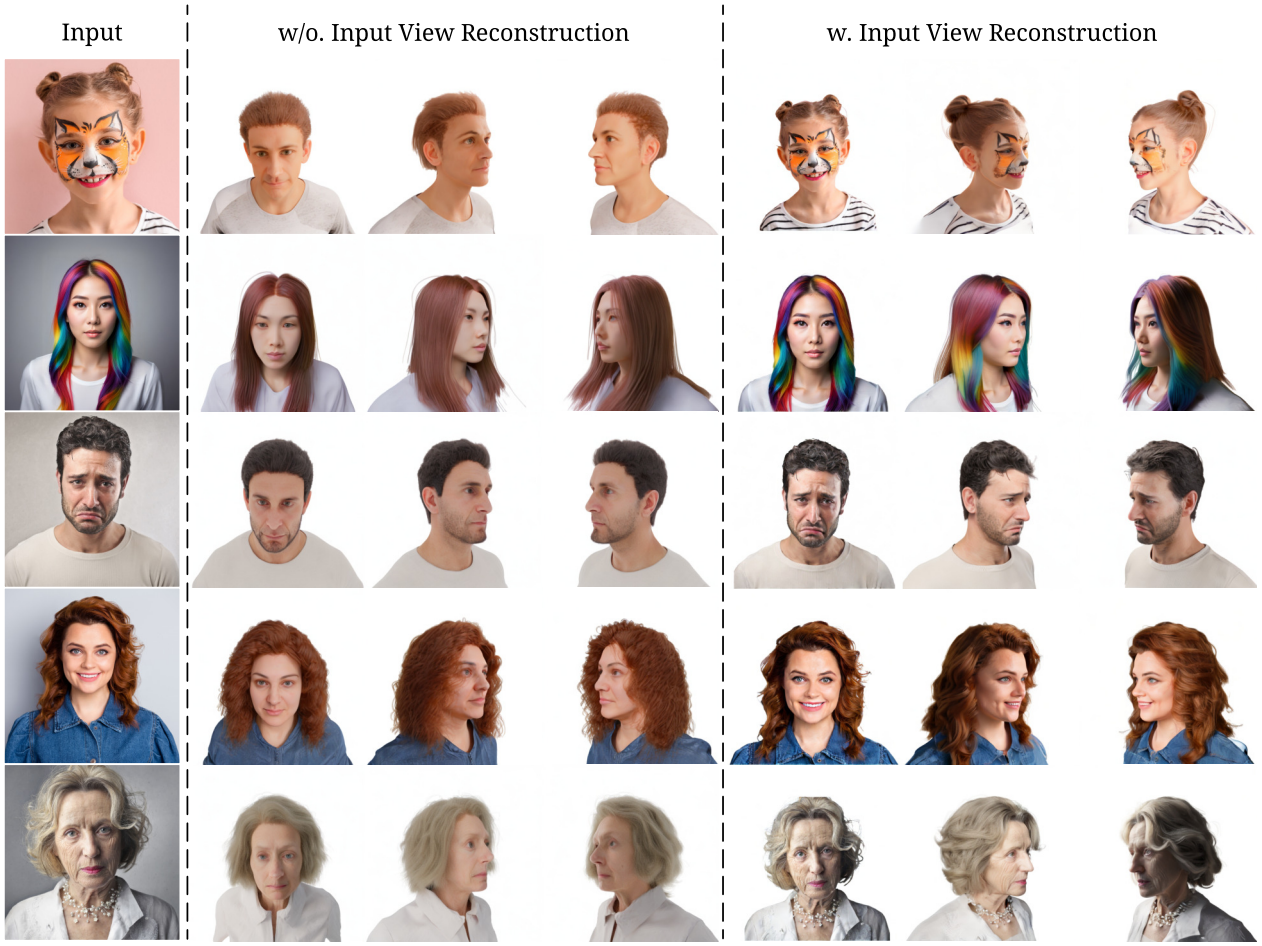


Figure 17. **Importance of input view reconstruction.** The diffusion model without input view reconstruction training suffers from identity loss. Additionally, it fails to generate accurate face paint (row 1), diverse hair colors (row 2), varied expressions (row 3 and 4), and accessories (row 5).

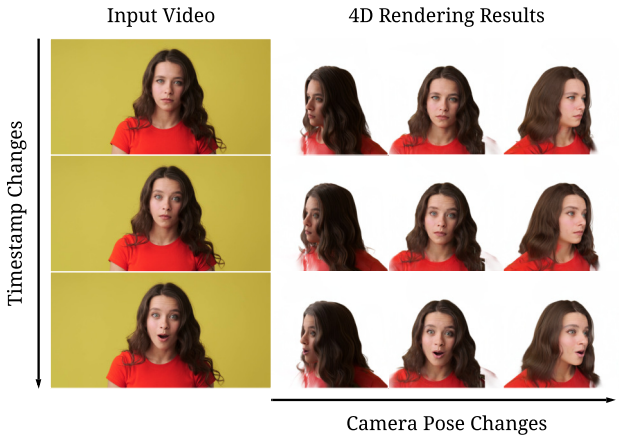


Figure 18. **Results of directly applying *FaceLift* to video input.** By processing each video frame independently, *FaceLift* generates a sequence of Gaussians that preserves consistent visual identity and accurately captures facial expressions. However, this baseline does not consider temporal consistency.

train deformation network $D_{t+1}$ to generate $G'_{t+2}$. Iteratively, we will get a Gaussian sequence $\{G'_0, G'_1, \ldots, G'_T\}$. Given any timestamp, we can select the corresponding 3D Gaussians from this Gaussian sequence and render from any given pose. The results of this method are shown in Fig. 21, which demonstrate improved temporal consistency while preserving identity and achieving accurate expression modeling. Please refer to the supplementary video for additional video rendering results.

# 4. Method Details

## 4.1. Details on View Generation

Given a single near frontal view face image with azimuth $\alpha$, the multi-view diffusion model will generate six views with azimuths equal to $\{\alpha, \alpha \pm 45°, \alpha \pm 90°, \alpha + 180°\}$, covering 360 degrees of the human head. All images, both input and generated output, maintain a zero elevation angle, ensuring consistent horizontal viewpoints. The generated views consist of: a reconstructed front view matching the input image; left and right profiles capturing the sides of the head; and a back view that synthesizes hair structure and color based on the frontal input and learned priors. We also generate three-quarter views (left-front and right-front) to enhance facial details in the following reconstruction stage.

To generate unseen views of the human head, we reformulate view synthesis from a single image as a conditional diffusion process. Specifically, we employ a DDPM-based diffusion model $f_D$ to generate $N$ distinct views, denoted $X_0^1, X_0^2, \ldots, X_0^N$, from a single front-facing image $y$ and corresponding text embeddings $e^1, e^2, \ldots, e^N$. This process can be expressed as:

$$\{X_0^1, X_0^2, \ldots, X_0^N\} = f_D\Big(y, \{e^1, e^2, \ldots, e^N\}\Big). \quad (5)$$

Our objective is to learn the joint distribution of these views conditioned on the input image and text embeddings. We denote this joint distribution as:

$$p_\theta(x_0^{1:N} \mid y, e^{1:N}) \equiv p_\theta\Big(\{x_0^1, \ldots, x_0^N\} \mid y, \{e^1, \ldots, e^N\}\Big). \quad (6)$$

In the following discussion, we omit the condition $y$ and $e^1, e^2, \ldots, e^N$ for simplicity. The joint distribution as $p_\theta(x_0^{1:N})$ is characterized by a Markov Chain (reverse process):

$$\begin{aligned} p_\theta(x_{0:T}^{1:N}) &= p_\theta(x_T^{1:N}) \prod_{t=1}^{T} p_\theta\big(x_{t-1}^{1:N} \mid x_t^{1:N}\big) \\ &= p_\theta(x_T^{1:N}) \prod_{t=1}^{T} \prod_{n=1}^{N} p_\theta\big(x_{t-1}^{n} \mid x_t^{1:N}\big), \end{aligned} \quad (7)$$

where $p_\theta(x_T^{1:N}) = \mathcal{N}(x_T^{1:N}; 0, \mathrm{I})$ and $p_\theta(x_{t-1}^n \mid x_t^{1:N}) = N(x_T^n; \mu_\theta^n(x_t^{1:N}, t), \sigma_t^2 \mathrm{I})$. $\mu_\theta(x_t^{1:N}, t)$ is a trainable component while the variance $\sigma_t^2$ is untrained time-dependent constants. To learn $\mu_\theta$ for generation, a Markov chain called forward process is constructed as:

$$\begin{aligned} q\big(x_{1:T}^{1:N} \mid x_0^{1:N}\big) &= \prod_{t=1}^{T} q\big(x_t^{1:N} \mid x_{t-1}^{1:N}\big) \\ &= \prod_{t=1}^{T} \prod_{n=1}^{N} q\big(x_t^n \mid x_{t-1}^n\big), \end{aligned} \quad (8)$$

where $q\big(x_t^n \mid x_{t-1}^n\big) = \mathcal{N}\big(x_t^n; \sqrt{1-\beta_t}\, x_{t-1}^n, \beta_t \mathbf{I}\big)$, and $\beta_t$ are constants. DDPM [22] shows that by defining

$$\mu_\theta^n(x_t^{1:N}, t) = \frac{1}{\sqrt{\alpha_t}} \left( x_t^n - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta\big(x_t^{1:N}, t\big) \right). \quad (9)$$

$\alpha_t$ and $\bar{\alpha}_t$ are constants derived from $\beta_t$ and $\epsilon_\theta$ is a noise predictor. We learn $\epsilon_\theta$ by

$$\ell = \mathbb{E}_{t, x_0^{1:N}, n, \epsilon^{1:N}} \big[ \|\epsilon^n - \epsilon_\theta^n(x_t^{1:N}, t)\|_2 \big], \quad (10)$$

where $\epsilon^{1:N}$ is the Gaussian noise of size $N \times H \times W$ added to all $N$ views, and $\epsilon_\theta^n$ is the noise predictor on the $n_{th}$ view. We provide ablation study results of the multi-view attention mechanism in Tab. 4.

# 5. Experimental Details

## 5.1. Details on Benchmark Evaluation

**Test Camera Extrinsic.** Both the Cafca [8] and Ava-256 [39] datasets offer multi-view RGB images along with

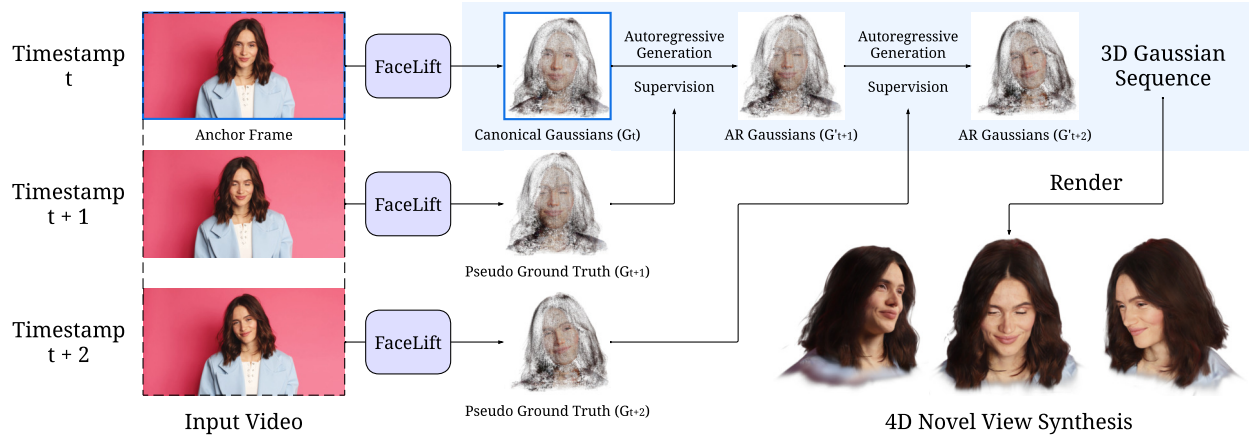


Figure 19. **Autoregressive Generation for 4D Rendering.** "AR Gaussians" denotes autoregressively generated Gaussians. With *FaceLift*, each video frame is independently converted into a 3D Gaussian representation. An anchor frame at timestamp $t$ (highlighted by the blue box) produces Canonical Gaussians $G_t$, which are then deformed into the representations for subsequent frames, $G'_{t+1}$, $G'_{t+2}$, ..., *etc*. This deformation is supervised by the rendered output Gaussians $G_{t+1}$, $G_{t+2}$, ..., *etc*., produced by *FaceLift*. Iteratively applying this process yields a temporally consistent Gaussian sequence that supports rendering from any viewpoint.

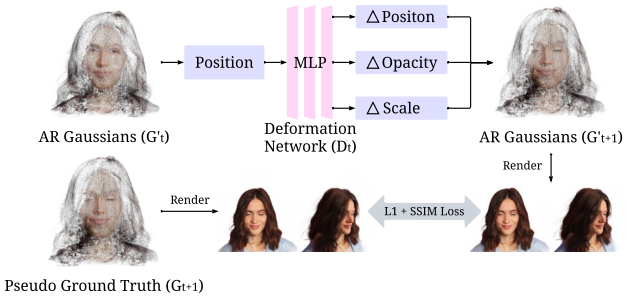


Figure 20. **Deformation Network.** The deformation network $D_t$ is an eight-layer MLP that predicts geometric deformations, including positional shifts, opacity adjustments, and scale changes. Combined with the Gaussian representations from the previous frame $G'_t$, it forms the Gaussian representation for the next frame $G'_{t+1}$.

corresponding camera poses. However, their camera systems differ from those used in *FaceLift* and the baselines. Directly applying their camera poses for inference is infeasible. Hence, we recalculate the test camera extrinsic in each method's camera system with the following procedure. The Ava-256 dataset uses a world coordinate system with the origin set at one of the camera positions. We first re-center the world coordinate origin to the midpoint of all camera locations, which is approximately the center of the human head. This step is unnecessary for the Cafca dataset, as its world coordinate origin is defined as the head's center. Next, we compute the rotation transformation from the test camera pose to the input camera pose within the dataset's coordinate system. We then apply the same transformation to the input camera pose in each method's camera system and rescale the translation to match the settings of each method to get the test camera extrinsic under each method's camera system. After applying the camera pose transformation, perfect alignment is not achieved due to differences in camera distance and intrinsic parameters. To address this, we manually crop and scale the rendered images for closer alignment with the target images.

**Facial Landmark Alignment.** To align two images based on their facial landmarks, we first compute the geometric transformations—scale and translation—that align the landmarks of one image with the landmarks of the other. Given an input image $I_1$ and two sets of corresponding facial landmarks $L_1$ and $L_2$, we begin by calculating the centroids of the landmark sets, centering the landmarks around their respective centroids. Next, we compute the uniform scaling factor and translation vector that minimize the difference between the centered landmarks. These transformations are then applied to the input image $I_1$, producing the transformed image $I_t$ in which the facial landmarks are aligned with those of $L_2$. This process is illustrated in Algorithm 1.

## 5.2. Implementation Details

**Multi-view Diffusion.** Our multi-view diffusion model is built based on the open-source latent diffusion framework, Stable Diffusion V2-1-unCLIP model [49]. The model is trained on eight A100 GPUs (each with 80 GB of memory) using a batch size of 64 over 20,000 steps, with a learning rate of 1e-4. For classifier-free guidance (CFG) [21], the CLIP condition was randomly omitted at a rate of 0.05 during training. During inference, we utilized the DDIM sampler [54] with 50 steps and a guidance scale of 3.0 to generate multi-view images. Both the input and output im-

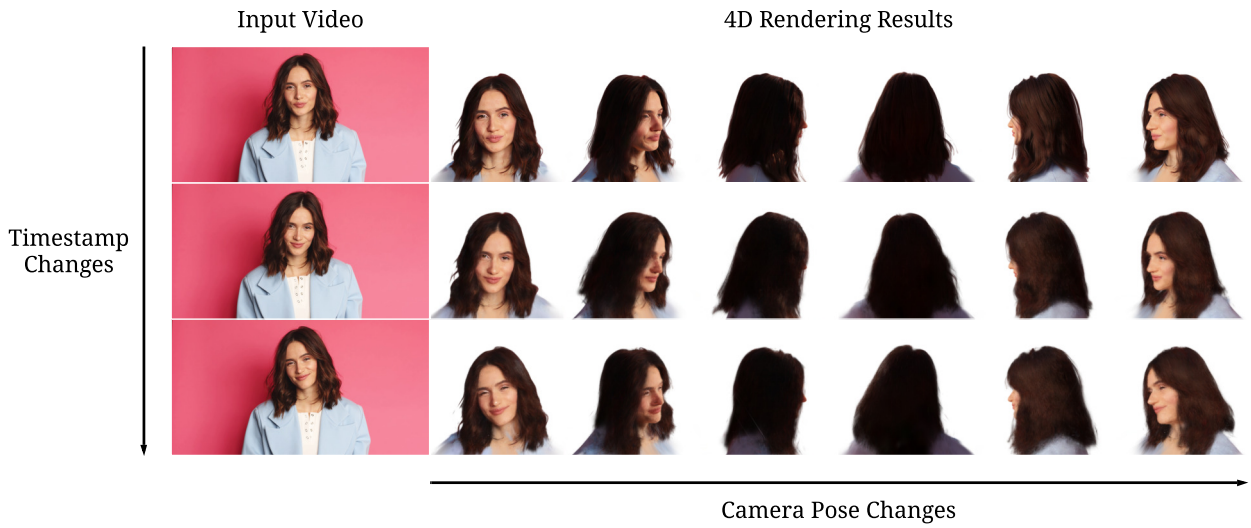


Figure 21. **Results of applying *FaceLift* on video.** Our proposed autoregressive generation pipeline enables *FaceLift* to be applied directly to video sequences, achieving 4D novel view synthesis – rendering at any given timestamp and camera pose. Video results are shown in the supplementary material.

**Algorithm 1:** Image Alignment via Facial Landmarks

**Input:** Image $I_1$, Landmarks $L_1$, $L_2$
**Output:** Transformed image $I_t$

**Function** `GetTransformFromLandmarks`$(L_1, L_2)$**:**
  Compute centroids $C_1, C_2$ of $L_1, L_2$;
  Center landmarks: $L_1' \leftarrow L_1 - C_1$, $L_2' \leftarrow L_2 - C_2$;
  Compute scale: $s \leftarrow \frac{\sum(L_1' \cdot L_2')}{\sum(L_1' \cdot L_1')}$;
  Compute translation: $t \leftarrow C_2 - s \cdot C_1$;
  **return** $s, t$;
**Function** `ApplyTransformToImage`$(I, s, t)$**:**
  Create transformation matrix $M$;
  Transform image: $I_t \leftarrow \text{warpAffine}(I, M)$;
  **return** $I_t$;
**Function** `TransformImageWithLandmarks`$(I_1, L_1, L_2)$**:**
  Compute $s, t \leftarrow$ `GetTransformFromLandmarks`$(L_1, L_2)$;
  Transform image: $I_t \leftarrow$ `ApplyTransformToImage`$(I_1, s, t)$;
  **return** $I_t$;

ages have a resolution of 512×512.

**Transformer-based Gaussian Reconstructor.** The training of the reconstructor follows [74]. During each training step, we randomly sample a set of 8 images (4 as input views and 4 as supervision views) from either 32 ambient light renderings or 25 random HDR environment light renderings. Both input and output images are rendered at a resolution of 512×512. The model is fine-tuned for 20,000 steps using eight A100 GPUs, each equipped with 40 GB of memory.

For a fair comparison, we also fine-tune LGM [56] with our synthetic data with their provided training codes. However, the fine-tuned LGM achieves inferior performance than the original weights, as shown in Fig. 22.

### 5.3. Datasets

**Cafca Dataset.** The Cafca dataset [8] comprises 1,500 identities, 30 camera poses, 13 expressions, and three environments. From this, we select 40 identities, as detailed in Tab. 6. We utilize the first expression and the first environment (folder 00000_000) for each identity. The input view and test views corresponding to each identity are also specified in Tab. 6.

**Ava-256 Dataset.** The Ava-256 dataset [39] consists of 256 identities, each captured by 80 cameras, with over 5,000 frames per camera. For qualitative evaluation, we select 10 identities, each with 10 test camera views. All selected frames feature natural expressions. We use camera 401168 as the input view, as it captures the front view of the faces and is positioned at the center of Ava-256's world coordinate system. The input view, test view, and corresponding frame IDs are detailed in Tab. 5.

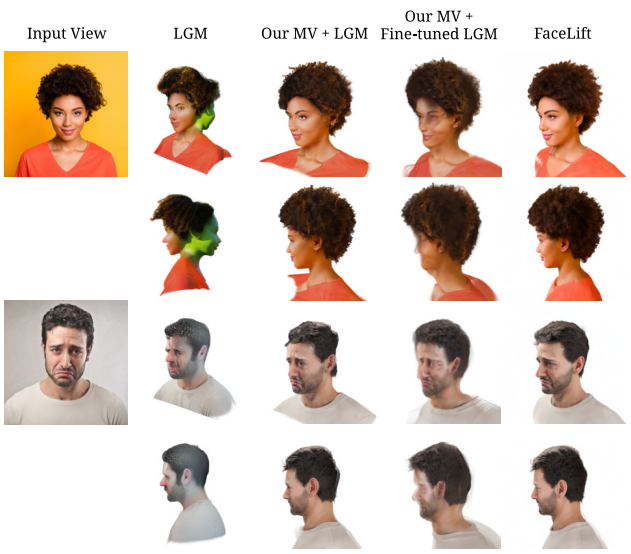


Figure 22. **Visual Comparison with LGM.** Leveraging the outputs of our multi-view diffusion model enhances the performance of LGM [56] (denoted as *Our MV + LGM*). We further fine-tuned LGM using our synthetic human head data, resulting in *Our MV + Fine-tuned LGM*; however, its performance was inferior to that achieved with the original weights in *Our MV + LGM*.

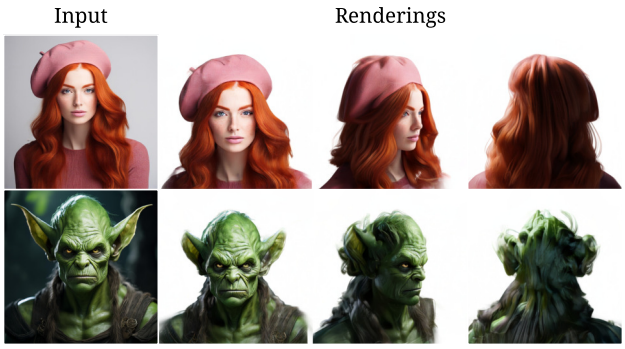


Figure 23. **Limitation of *FaceLift*.** Due to the absence of accessories in the training data, our method often generates hair-like textures to approximate hats. Additionally, it occasionally produces extraneous hair when encountering out-of-distribution images.

## 6. Limitations

*FaceLift* achieves high-fidelity, photorealistic 3D head reconstruction from a single input image. It provides detailed representations of hair and skin texture while demonstrating superior identity preservation compared to existing methods. Despite these appealing results, our approach has certain limitations. First, our synthetic dataset does not include accessories such as hats or glasses. As a result, when the input image features a hat, the model may generate hair-like textures to approximate the back of the hat, as illustrated in Fig. 23, row 1. This limitation could be addressed by incorporating synthetic data with accessories. Additionally, when handling out-of-distribution inputs, such as those in Fig. 23, row 2, the model occasionally generates extraneous hair. This issue might be mitigated by refining the training data distribution or introducing text prompts to enhance control over the multi-view diffusion generation process. Finally, in some cases, the unseen regions of the face appear more blurred than the visible areas (frontal face). Our system emphasizes detailed reconstruction of the front face: most views generated by the diffusion model concentrate on the frontal region, and the input-view reconstruction strategy faithfully preserves its features. In contrast, features of the back of the head are primarily learned from synthetic data. Additionally, when simulating lighting, the model tends to darken the back head and introduce shadows, often causing the hair to appear black.

| ID | Frame ID | Input View | Test Views |
|---|---|---|---|
| 20210810–1306–FXN596 | 029693 | 401168 | 400944 400981 401031 401075 401163 401175 401292 401303 401316 401463 |
| 20210827–0906–KDA058 | 028930 | 401168 | 400944 401031 401071 401163 401166 401292 401316 401408 401410 401458 |
| 20210901–0833–LAS440 | 027655 | 401168 | 400944 401031 401161 401163 401172 401292 401303 401316 401410 401458 |
| 20210929–0827–MCR809 | 029457 | 401168 | 400981 401070 401158 401166 401173 401305 401313 401408 401410 401458 |
| 20211001–0855–KJJ701 | 032309 | 401168 | 400939 401031 401163 401166 401292 401316 401408 401410 401452 401458 |
| 20220215–0801–ONK705 | 027201 | 401168 | 400944 401031 401045 401163 401166 401172 401408 401410 401463 401469 |
| 20220310–1128–ZSC414 | 028601 | 401168 | 400942 401031 401045 401163 401164 401166 401303 401408 401410 401411 |
| 20220712–1040–JEH262 | 030060 | 401168 | 400944 400981 401031 401045 401163 401408 401410 401452 401458 401469 |
| 20220809–1321–UTC375 | 027432 | 401168 | 401031 401071 401163 401166 401175 401292 401303 401452 401458 401469 |
| 20220818–1653–SSF476 | 036588 | 401168 | 400981 401031 401071 401163 401166 401175 401408 401410 401458 401469 |

Table 5. Identities and views used for the experiments on Ava-256.

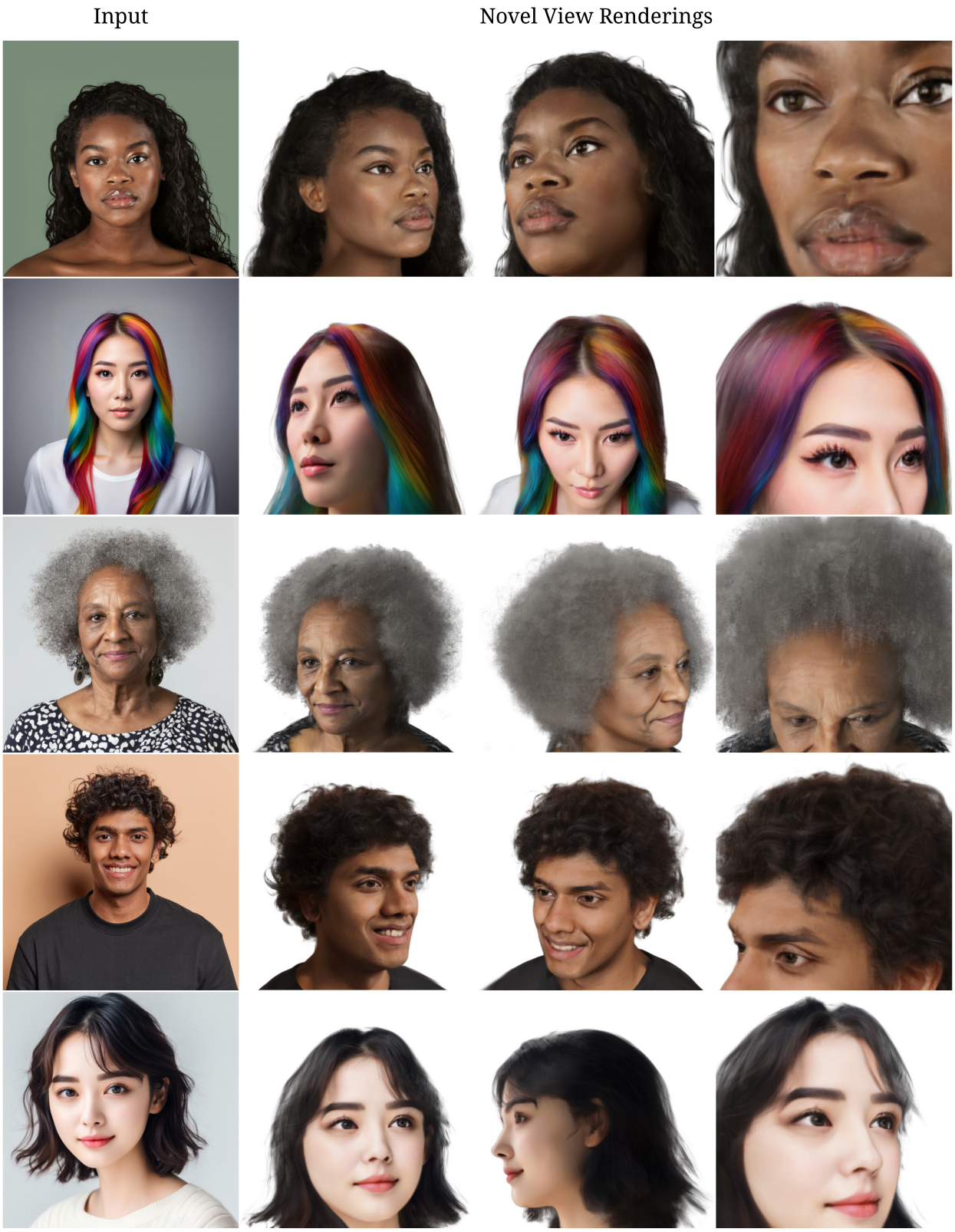


Figure 24. **Results of *FaceLift* on in-the-wild images.** *FaceLift* excels at reconstructing intricate and diverse facial hair, encompassing a wide array of hairstyles and hair colors. It also accurately captures a broad range of skin tones.


Figure 25. **Results of *FaceLift* on in-the-wild images.** *FaceLift* also demonstrates the ability to reconstruct faces exhibiting a wide range of pose variations. It can also accurately handle extreme expressions.

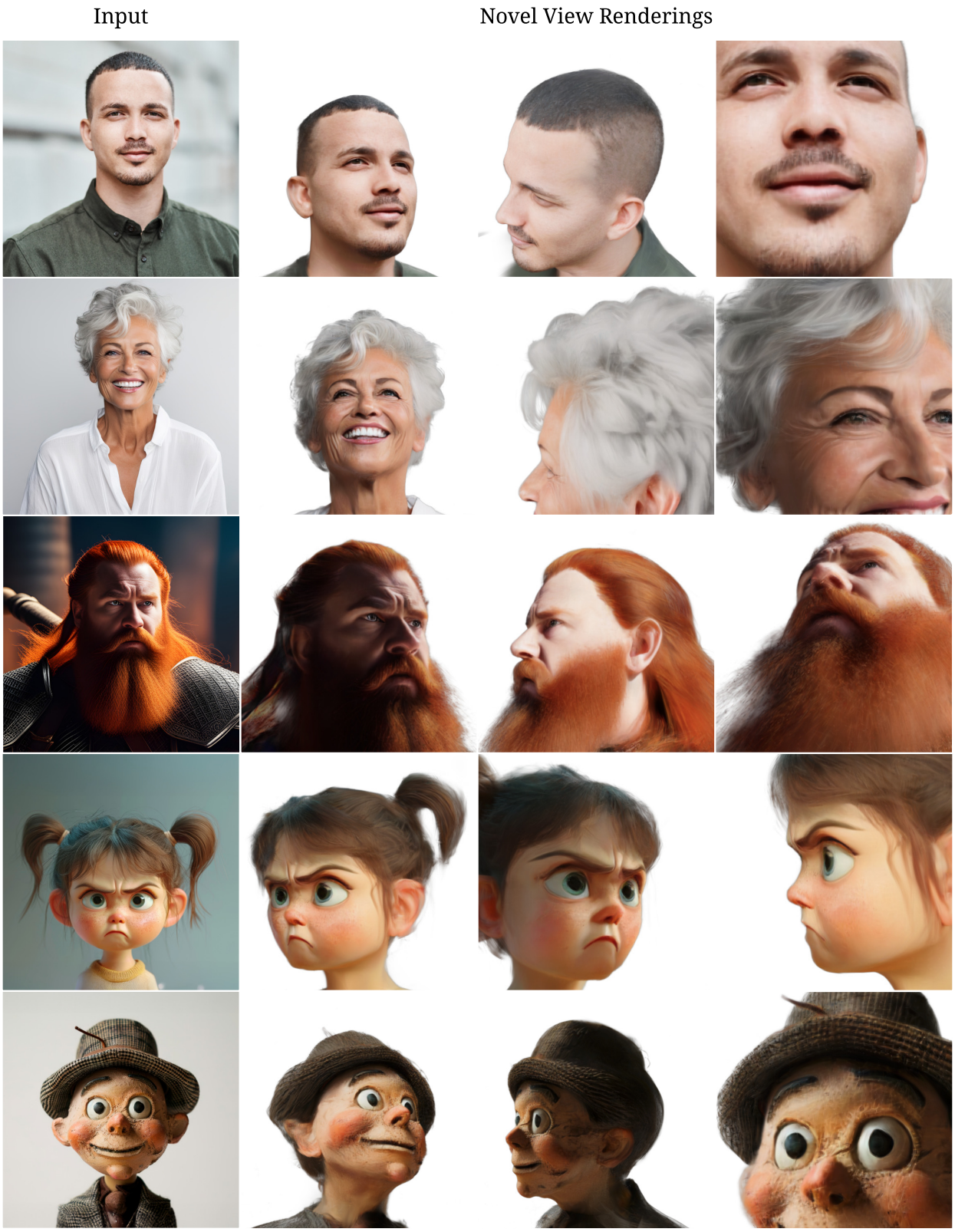


Figure 26. **Results of *FaceLift* on in-the-wild images.** *FaceLift* realistically reconstructs detailed facial textures. Additionally, *FaceLift* is well-suited for reconstructing cartoon characters.

| ID | Input View | Test Views |
|---|---|---|
| 00000 | 26 | 00 02 06 08 10 11 12 13 17 19 20 23 24 26 |
| 00002 | 12 | 00 03 04 05 06 07 08 09 12 13 15 17 21 22 23 24 25 |
| 00004 | 07 | 03 04 07 09 10 11 18 19 23 24 25 26 27 29 |
| 00005 | 15 | 01 02 06 07 08 10 11 13 15 18 19 20 21 23 26 27 28 |
| 00006 | 27 | 00 02 10 19 20 23 27 |
| 00007 | 09 | 03 04 09 11 13 15 16 17 19 21 24 26 28 |
| 00010 | 24 | 02 04 08 10 12 13 14 15 17 21 22 23 24 25 26 27 28 29 |
| 00011 | 07 | 02 05 07 09 11 12 14 16 24 27 29 |
| 00014 | 03 | 02 03 06 12 14 17 22 23 25 28 29 |
| 00015 | 22 | 00 02 04 06 09 12 14 15 20 22 24 27 28 |
| 00017 | 12 | 01 02 07 12 14 15 16 17 20 22 23 24 25 26 |
| 00018 | 08 | 00 02 06 08 09 13 16 18 20 25 26 |
| 00019 | 14 | 00 04 05 06 10 12 13 14 16 17 18 20 21 22 26 28 |
| 00020 | 01 | 00 01 03 04 06 07 10 14 16 17 19 22 23 25 26 27 29 |
| 00021 | 11 | 02 03 05 07 08 09 11 14 15 17 19 21 22 23 26 |
| 00022 | 18 | 00 01 03 07 08 09 11 12 17 18 19 21 22 24 26 28 |
| 00023 | 03 | 00 03 05 06 08 12 14 18 24 25 27 |
| 00028 | 18 | 04 05 06 10 12 13 16 18 19 22 24 25 28 29 |
| 00030 | 21 | 00 01 02 03 06 07 08 11 14 17 19 21 22 24 26 |
| 00033 | 03 | 00 03 06 11 12 13 15 19 21 22 24 27 28 |
| 00034 | 10 | 01 06 07 09 10 13 15 16 17 18 19 23 25 28 |
| 00048 | 04 | 00 01 02 04 05 06 07 10 12 15 20 23 24 25 27 28 |
| 00051 | 26 | 03 07 10 11 15 17 19 21 22 24 26 28 29 |
| 00056 | 07 | 00 01 02 07 08 12 14 15 17 18 20 21 22 23 24 25 28 29 |
| 00057 | 11 | 00 01 02 03 05 06 08 11 12 14 17 18 19 22 26 29 |
| 00063 | 01 | 01 02 05 08 09 11 13 14 16 17 18 20 22 25 26 28 29 |
| 00066 | 13 | 01 05 06 07 12 13 21 22 26 27 |
| 00068 | 12 | 00 01 06 10 12 14 16 19 21 22 25 26 27 |
| 00072 | 25 | 02 04 05 10 12 13 14 17 25 26 |
| 00078 | 20 | 00 02 03 05 06 07 08 12 13 14 15 16 17 18 20 24 25 28 29 |
| 00080 | 08 | 01 03 04 05 06 08 10 12 14 15 16 17 22 24 26 |
| 00082 | 16 | 05 06 07 09 13 16 17 19 20 23 25 27 |
| 00083 | 16 | 00 02 03 04 05 08 09 13 14 16 17 19 21 22 24 25 27 29 |
| 00084 | 01 | 02 04 08 09 11 12 14 16 17 18 19 23 28 29 |
| 00086 | 13 | 00 01 03 04 08 09 13 14 17 18 19 20 22 23 24 |
| 00087 | 01 | 00 01 02 04 07 08 09 12 15 16 17 18 21 24 26 27 |
| 00094 | 08 | 02 05 08 09 12 19 24 25 27 |
| 00095 | 08 | 00 01 03 04 08 09 10 11 13 14 18 19 20 21 24 28 29 |
| 00096 | 01 | 01 05 07 10 12 17 19 21 22 28 |
| 00099 | 00 | 00 02 03 04 05 07 08 09 12 14 15 16 17 20 21 23 25 29 |

Table 6. Identities and views used for the experiment on Cafca.