ControlFace: Harnessing Facial Parametric Control for Face Rigging

- Supplementary Material -

In the supplementary material, we include additional details on the implementation and user study settings. We also present further ablation studies and provide an in-depth analysis of RCG. Finally, we showcase additional generation results from our method, ControlFace.

A. More on Implementation Details

Training Details. The model was trained for 300,000 steps on 8 NVIDIA V100 GPUs. Each GPU processed a batch size of 4, resulting in a total effective batch size of 32. We trained all of our components except for VAE encoder [**?**] and CLIP image encoder [**?**]. Tab. 1 shows the number of trainable parameters for each component of ControlFace. We employed 8bit-Adam optimizer [**?**] for memory efficiency and a constant learning rate of 0.00001 throughout the training process. The entire training procedure took approximately three days to complete.

Table 1. **Number of parameters.** We report parameter counts for each proposed component.

	# of parameters.
FaceNet	∼850M
Face Controller	∼3M
CMM	∼30M

Inference Details. For all the results in the qualitative and quantitative comparison in the main paper, ControlFace, CapHuman [**?**], and Arc2Face [**?**] utilize DDIM [**?**] scheduler with 50 sampling steps. The guidance scale for ControlFace was set to 4 while CapHuman and Arc2Face was set to 3.5 which is the default value. For DiffusionRig [**?**], we utilize DDPM [**?**] scheduler with 250 sampling steps. The computational cost of RCG is equivalent to that of CFG [**?**], as they both double the batch size to process both the conditional and unconditional paths simultaneously.

B. User Study Settings

We provide detailed information about the user study. Following the methodology of ImagenHub [**?**], participants evaluated each model based on two criteria: (1) how well the generated image aligns with the input condition (SC) and (2) the overall quality (PQ) of the generated image. Since ControlFace takes both a reference image and 3DMM renderings as input conditions, participants assessed its performance separately for each condition, with the SC defined as the lower score between the two evaluations. Eight participants were recruited and divided into two groups, each group evaluating different examples consisting of 52 generated images per model. Fig. 1 illustrates a sample page from our user study.

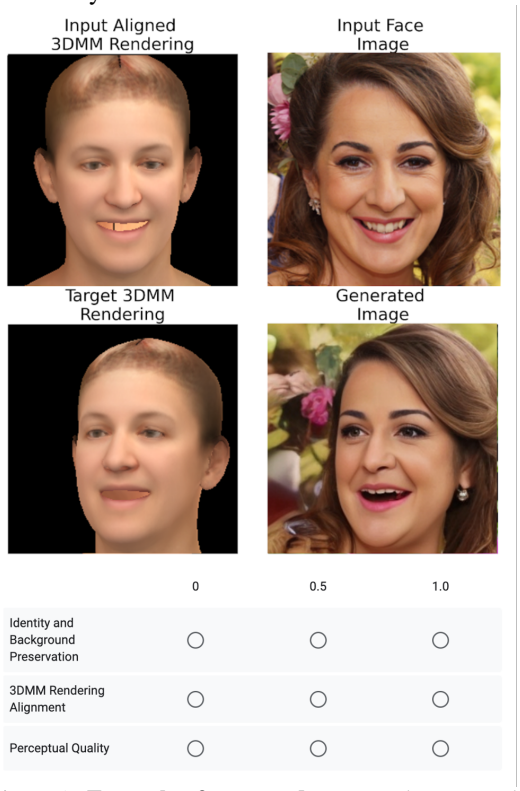


Figure 1. **Example of user study.** For each generated result, the participants were asked three questions.

C. Additional Experiments

C.1. Training DiffuisonRig [?] on Video Dataset

For a fair comparison, we train two versions of DiffusionRig [**?**] on CelebV-HQ [**?**]: a reconstruction version where the reference and target images are identical and a paired version where they differ. Both versions are evaluated using the same protocol outlined in the main paper. Tab. 3 shows a significant drop in performance for the newly trained models compared to the original implementation. This is due to the limited identities in CelebV-HQ and the ResNet in DiffusionRig struggling to encode the rich information in the video data, highlighting the need for our architectural design choices.

C.2. Model with Only CLIP Embedding

We train the denoising U-Net with only the CLIP [**?**] image encoder attached and evaluate its performance using the same metrics described in the main paper. The results, as presented in Tab. 2, reveal that the CLIP image encoder alone struggles to fully encode the

Table 2. **Quantitative results of model with only CLIP [?] image encoder.** We compare the result of CLIP image encoder and FaceNet.

	Re-Infer.↓	ID ↑	FID ↓	LPIPS ↓
CLIP [**?**]	8.96	0.1623	**29.77**	0.4496
+ FaceNet	**7.13**	**0.8234**	32.45	**0.1321**

Table 3. **Quantitative results of training DiffuionRig [?] on video dataset.** We train DiffuisonRig [?] on video dataset [?] and evaluate following the same protocol introduced in the main paper.

DiffusionRig [?]	Re-Infer.↓	ID ↑	FID ↓	LPIPS ↓
- FFHQ [?]	5.06	0.2042	23.05	0.3758
- CelebV-HQ [?] (Recon.)	5.78	0.0712	55.37	0.3889
- CelebV-HQ (Paired)	5.77	0.0670	59.71	0.4036
ControlFace (Ours)	**4.85**	**0.7586**	**15.50**	**0.1429**

Table 4. **Different inputs for CMM.** We train the model with different inputs for the CMM and measure the DECA [?] re-inference error.

	Light ↓	Shape ↑	Exp. ↓	Pose ↓	Avg. ↓
Landmarks	3.88	2.65	6.63	**6.63**	9.50
Lambertian Rendering	4.11	**2.56**	5.83	7.3	4.95
Three Renderings (Ours)	**3.75**	**2.56**	5.43	7.67	**4.85**

Table 5. **Different architectures for CMM.** We train the model with different architectures for the CMM and measure the DECA [?] re-inference error and identity similarity (ID) [?].

Variants	Re-Infer (↓)	ID (↑)
CMM with ResNet architecture	5.27	0.7199
No CMM, Face Controller on reference	5.40	0.7184
CMM (Reference only)	5.03	0.7290
CMM (Ours)	**4.85**	**0.7586**

reference image, resulting in poor performance. This limitation emphasizes the importance of our proposed FaceNet, which is specifically designed to capture and preserve the fine details and identity present in the reference image, ensuring improved results. Although FaceNet achieves lower FID [?], we show in Fig. 2 that the model which utilizes CLIP embedding often generates over-saturated results, which degrades the image quality.

C.3. More ablation on CMM

We trained two additional models using different inputs to the CMM to identify the most suitable method to embed the relationship between the reference image and the target image. First, we trained a model using facial landmarks extracted from FLAME [?]. Specifically, we convert the landmarks into a map by assigning an integer value to the index corresponding to each landmark. Next, we employed Lambertian rendering, one of three types of renderings used as control in our method, which offers more information compared to the other two.

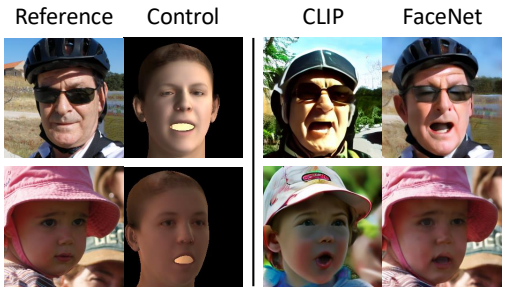


Figure 2. **Qualitative results of model with only CLIP [?] image encoder.** We train the denoising U-Net with only CLIP image encoder attached and compare the results with the FaceNet.

The results, presented in Tab. 4, demonstrate that as the input provides richer information about the image, the CMM is better able to embed the relationship, resulting in improved control adherence performance. We also provide Fig. 4 to visualize the different inputs for the CMM and to compare the three models. Fig. 4 shows that the two models, which take either facial landmarks or only Lambertian rendering as CMM input, both lack control adherence compared to our method. Also, the model with landmark input compromises the identity.

We also trained three models with different architectures on CMM. Fig. 5 shows that our design of CMM exhibits the best results.

C.4. More RCG Visualization

We provide more analysis on reference control guidance (RCG). In Fig. 5, we visualize the differences, $\epsilon_\theta(\cdot, D_T) - \epsilon_\theta(\cdot, \varnothing)$ and $\epsilon_\theta(\cdot, D_T) - \epsilon_\theta(\cdot, D_R)$ which corresponds to CFG [?] applied to pose controller and RCG, respectively, across various timesteps t. Additionally, we visualize the predicted X_0 to observe how quick each method converges to the final output. The second and fourth row in Fig. 5a and Fig. 5b show the differences and the third and fourth row display the predicted X_0.

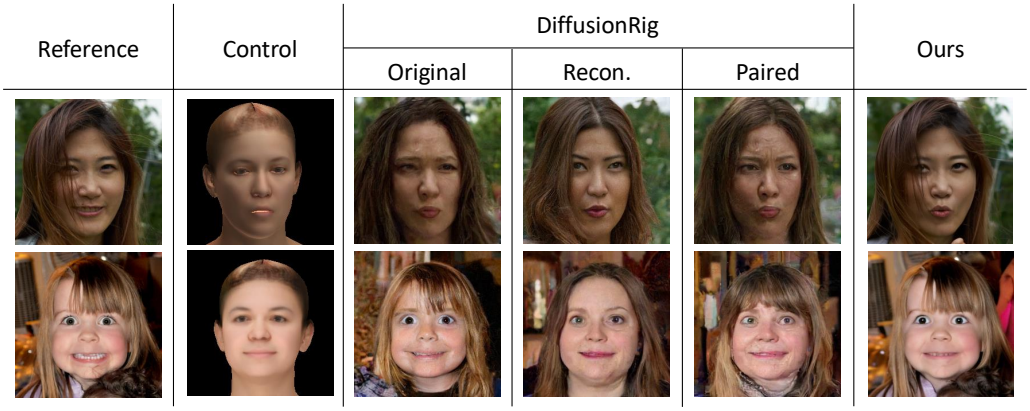


Figure 3. **Qualitative results of training DiffuionRig [?] on video dataset.** We illustrate the results of two variants of DiffusionRig that is trained on CelebV-HQ [**?**].

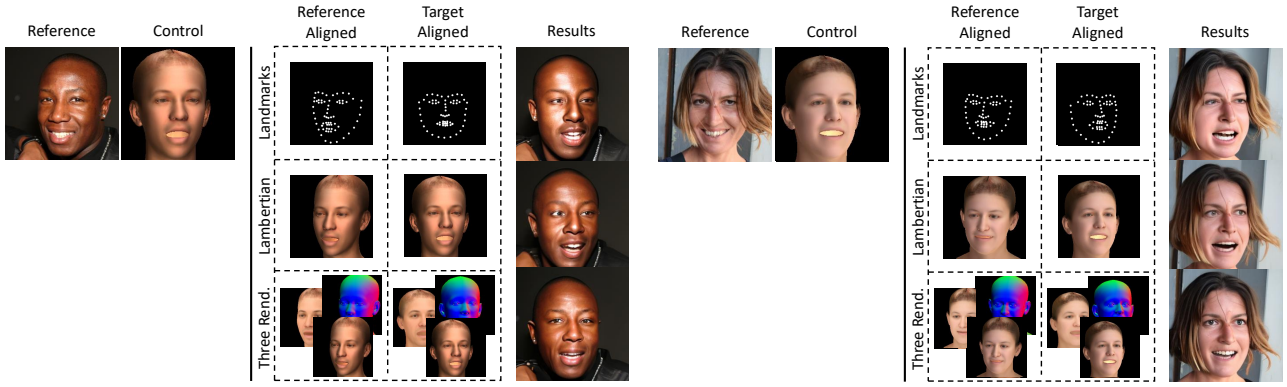


Figure 4. **Qualitative results of CMM input ablation.** We provide generated images on three different models which takes different inputs for CMM.

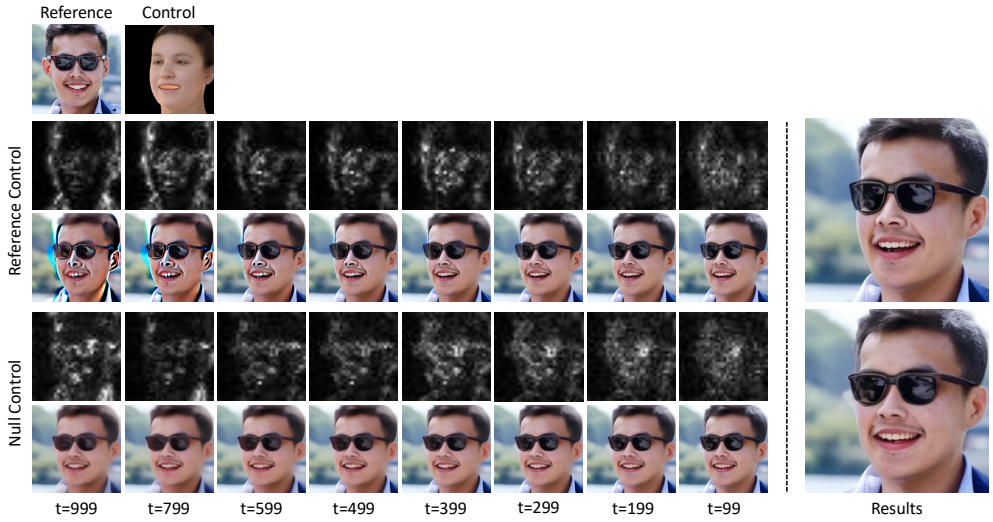
Interestingly, RCG exhibits over-saturated predicted X_0 whereas CFG generates over-smoothed predictions. This happens because RCG is better grounded, with the differences focused in specific areas where the output of the denoising U-Net varies between the reference and target controls. This over-saturation resolves quickly, resulting in realistic predictions faster than the slow dissipation of smoothness observed in CFG, which contributes to blurriness in the final generation result. As shown in Fig. 5b, the generation result of CFG still exhibits this blurriness both on the face and the right shoulder. Additionally, the generation results in Fig. 5a demonstrate that RCG achieves better adherence to the target control. RCG guides the sampling process to align more closely with the target control, where the face is oriented to the left. On the other hand, the result of CFG faces slightly forward.

D. Additional Qualitative Results

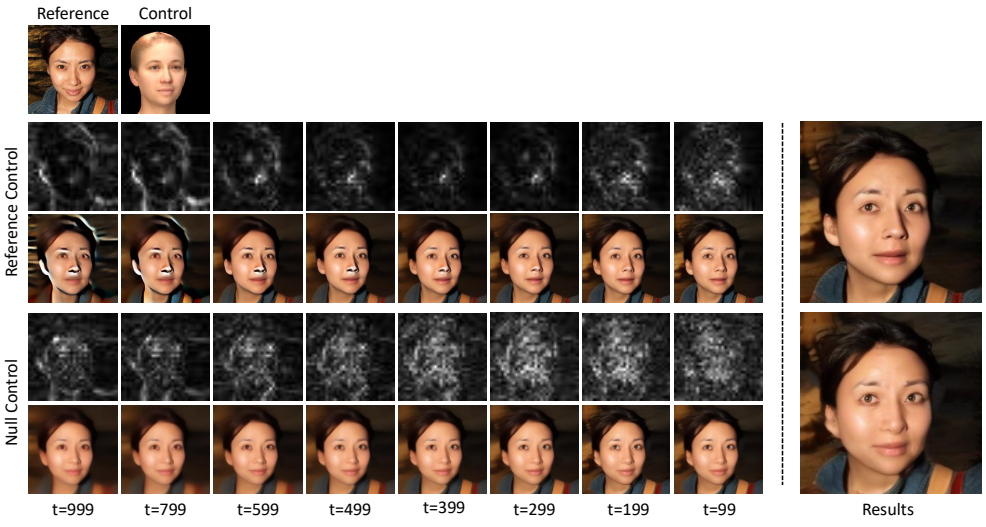
We provide more rigged results geenerated by ControlFace. First, we show generated results with faces acquired from FFHQ [**?**]. Then, we demonstrate that ControlFace can be applied to rig out-of-distribution faces such as cartoon-like portraits showing its strong robustness and generalizability. For out-of-distribution results, we utilize two types of face images: those manually collected from the internet and those generated using a large pretrained text-to-2D diffusion model.

D.1. FFHQ

D.2. Out-of-Distribution



(a)



(b)

Figure 5. **Visualization of RCG.** We visualize each difference, $\epsilon_\theta(\cdot, D_T) - \epsilon_\theta(\cdot, \varnothing)$ and $\epsilon_\theta(\cdot, D_T) - \epsilon_\theta(\cdot, D_R)$, along with the predicted X_0 on each timesteps t.

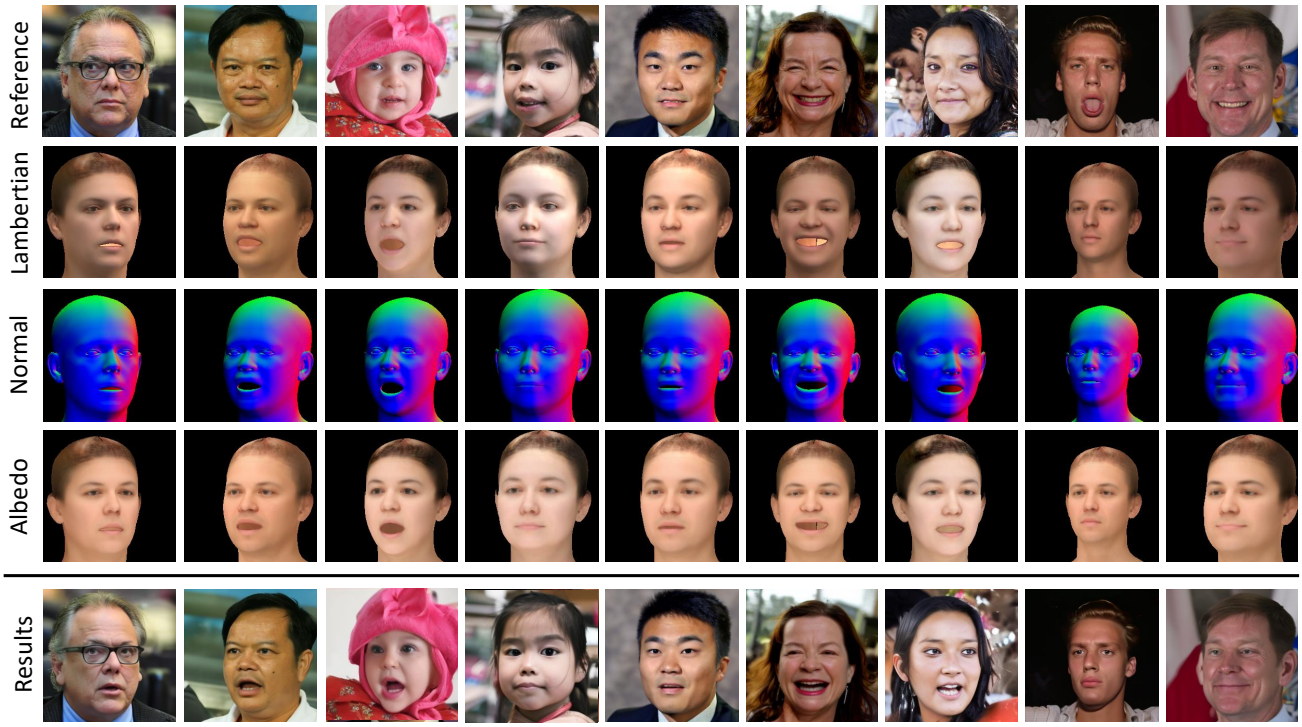


Figure 6. **Additional results on FFHQ [?].** We randomly select faces from FFHQ [**?**] and rig the pose parameters.

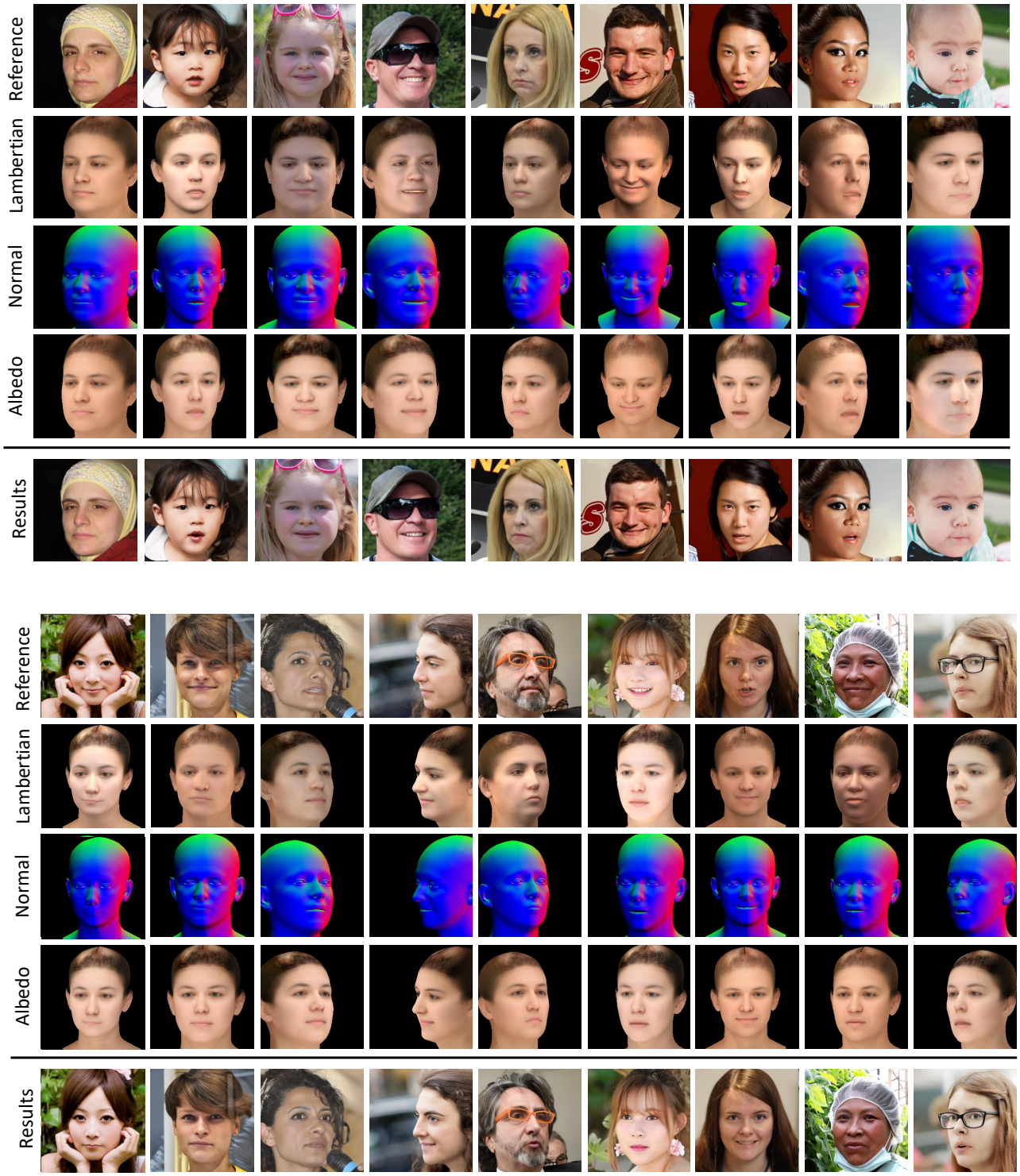


Figure 7. **Additional results on FFHQ [?].** We randomly select faces from FFHQ [**?**] and rig the shape parameters for the results on the top and expression parameters for the bottom.

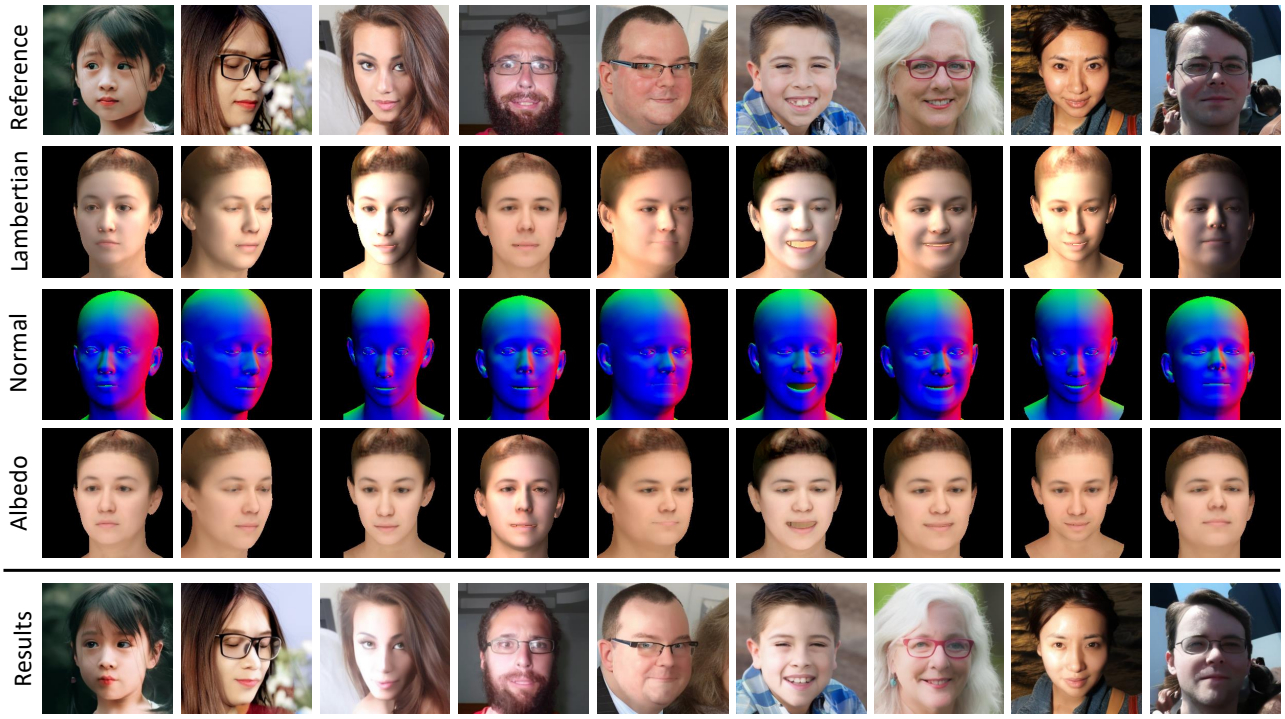


Figure 8. **Additional results on FFHQ [?].** We randomly select faces from FFHQ [**?**] and rig the light parameters.

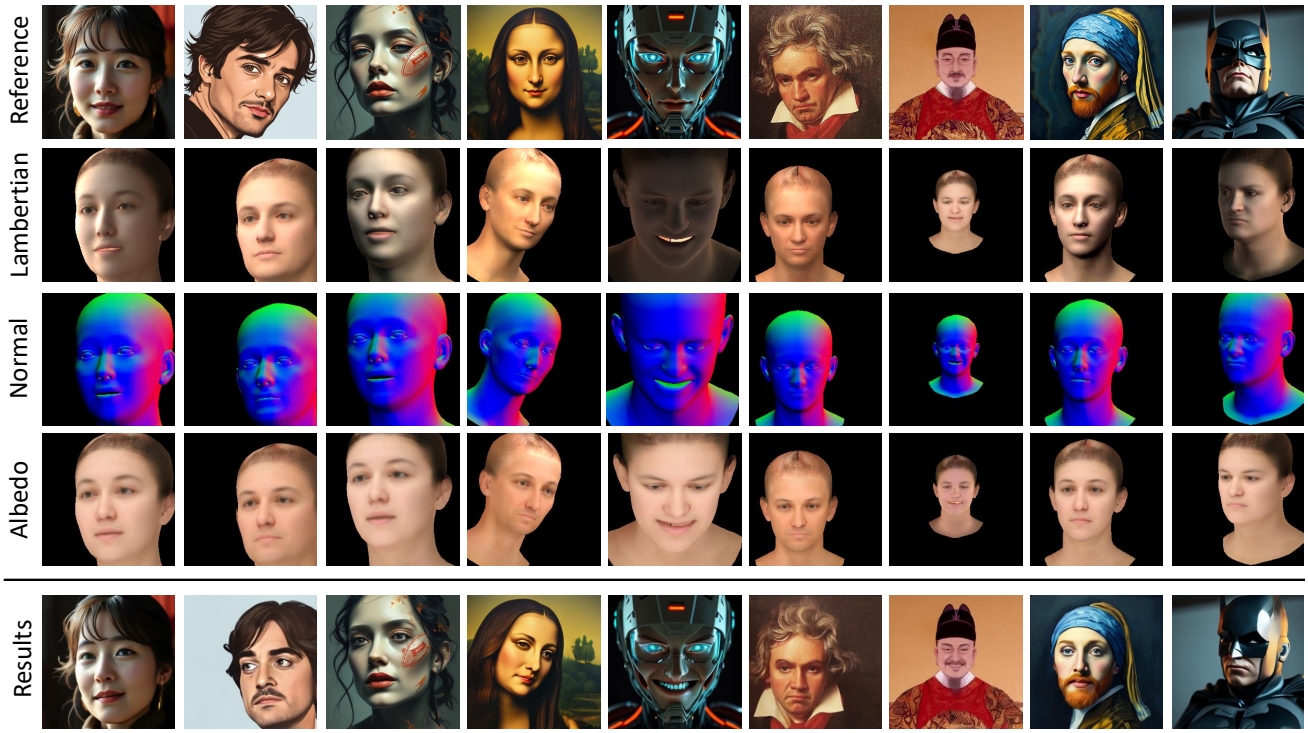


Figure 9. **Additional results on out-of-distribution faces.** We apply ControlFace on portraits obtained from the internet and generated by a text-to-2d diffusion model.