

## Supplemental Material

### PREIM3D: 3D Consistent Precise Image Attribute Editing from a Single Image in Real Time

In the supplement, we first provide implementation details, including encoder training process and edit directions seeking. We follow with additional experiments and visual results. We highly recommend watching the supplemental video, which contains a live demonstration of the real-time inversion and attribute editing and a demonstration of sequential editing synthesis.

## A. Implementation Details

### A.1. Encoder Training

We implemented our encoder training on top of the official pSp [4] encoder training framework implementation. We set the  $\lambda_{l2} = 1.0$ ,  $\lambda_{lpips} = 0.8$ , and  $\lambda_{ori} = 0.4$  in the first 20,000 training steps. After the 20,000 steps, we gradually add a delta for the  $\lambda_w = 1e^{-4}$  every 5,000 steps. After the 100,000 steps, we gradually add a delta for the  $\lambda_{sur} = 1e^{-4}$  every 5,000 steps. The in-domain images are sampled from yaw angles between  $[-30^\circ, 30^\circ]$  and pitch angles between  $[-20^\circ, 20^\circ]$ . The surrounding images are sampled from yaw angles between  $[-20^\circ, 20^\circ]$  and pitch angles between  $[-5^\circ, 5^\circ]$

### A.2. Edit Directions Seeking

We use InterfaceGAN [5] to train a SVM to find out the attribute editing directions. For the editing directions in the original space, the generator is applied to produce 140,000 images. For the editing directions in the inversion manifold, we perform inversion with our encoder on FFHQ [3] dataset. Here, we have obtained the latent code  $w$  and image pairs. An off-the-shelf multi-label classifier based on ResNet50 [2] is applied to predict the images. We train the SVM ([https://github.com/clementapa/CelebFaces\\_Attributes\\_Classification/](https://github.com/clementapa/CelebFaces_Attributes_Classification/)) to find the hyperplane that distinguishes binary attributes using the latent code  $w$  and the corresponding classification result as input. The normal vector of the hyperplane is the attribute editing direction.

## B. Comparison on Face Inversion at More Camera Poses

We uniformly sample 20 inverted images for each image of the first 300 images from CelebA-HQ in different yaws ranges using IDE-3D, 3D-Inv, Pixel2NeRF, and PREIM3D. As with the main text, IDE-3D and 3D-Inv perform image inversion with 500  $w$  optimization steps and 100 generator fine-tuning steps. We show the identity consistency (ID) in Figure 1.

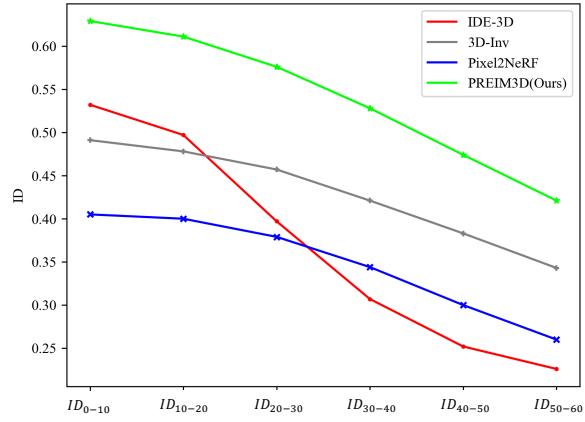


Figure 1.  $ID_{a-b}$  denotes the mean ArcFace similarity score between the input image and the 20 inverted images uniformly sampled from yaw angles between  $[-b^\circ, a^\circ] \cup [a^\circ, b^\circ]$  and pitch angles between  $[-20^\circ, 20^\circ]$ . Our method has a higher ID score than other methods in different yaw ranges.

## C. Additional Precise Editing

**AA & AD.** Following [7], we use attribute altering (AA) to evaluate the change of the desired attribute and attribute dependency (AD) to measure the degree of change on other attributes when modifying one attribute. AA is the change on the logit  $\Delta l_t$  of the off-the-shelf multi-label classifier detecting attribute  $t$  and is normalized by  $\sigma(l_t)$ , which is the standard deviation calculated from the logits of CelebA-HQ dataset. AD measures the change of logit  $\Delta l_i$  for other attributes  $\forall i \in \mathcal{A} \setminus t$ , where  $\mathcal{A}$  is the set of all attributes. Here, we use the mean-AD, defined as  $\mathbb{E}\left(\frac{1}{k} \sum_{i \in \mathcal{A} \setminus t} \left(\frac{\Delta l_i}{\sigma(l_i)}\right)\right)$ .

To further validate the precision of the editing in the inversion manifold, we perform more attribute editing. We make different degrees of editing by adjusting  $\alpha$ , and then observe the changes on the other attributes. Figure 2, 3 shows the difference between editing in the original space and editing in the inversion manifold, involving goatee, lipstick gray hair, wavy hair, and gender attributes. Both 2D-space and 3D-space attribute editing show more precise editing in the inversion manifold than in the original space.

## D. Naive Optimization-based Inversion

Different from the PTI technique, the naive optimization-based inversion method only optimizes the latent code  $w$ , while fixing the generator. Figure 4 shows the inversion results of the naive optimization-based inversion method.

000  
001  
002  
003  
004  
005  
006  
007  
008  
009  
010  
011  
012  
013  
014  
015  
016  
017  
018  
019  
020  
021  
022  
023  
024  
025  
026  
027  
028  
029  
030  
031  
032  
033  
034  
035  
036  
037  
038  
039  
040  
041  
042  
043  
044  
045  
046  
047  
048  
049  
050  
051  
052  
053  
054  
055  
056  
057  
058  
059  
060  
061  
062  
063  
064  
065  
066  
067  
068  
069  
070  
071  
072  
073  
074  
075  
076  
077  
078  
079  
080  
081  
082  
083  
084  
085  
086  
087  
088  
089  
090  
091  
092  
093  
094  
095  
096  
097  
098  
099  
100  
101  
102  
103  
104  
105  
106  
107

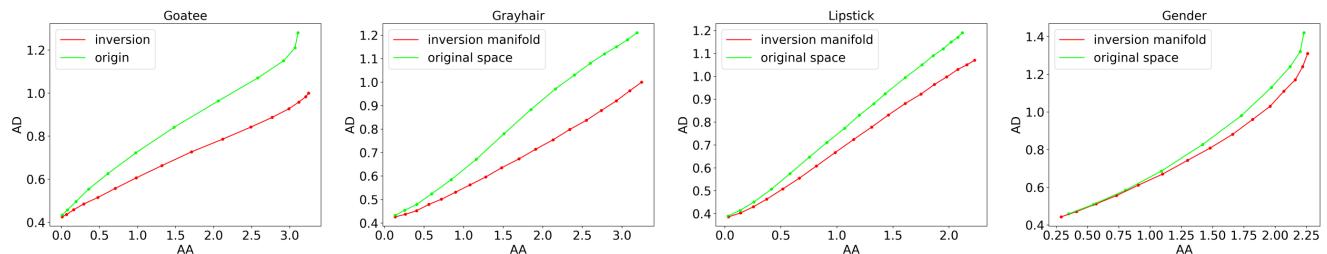
108  
109  
110  
111  
112  
113  
114  
115  
116

Figure 2. PREIMD(Ours). As the degree of editing  $\alpha$  changes, both Attribute Altering (AA) and Attribute Dependency (AD) change. Lower AD indicates more precise.

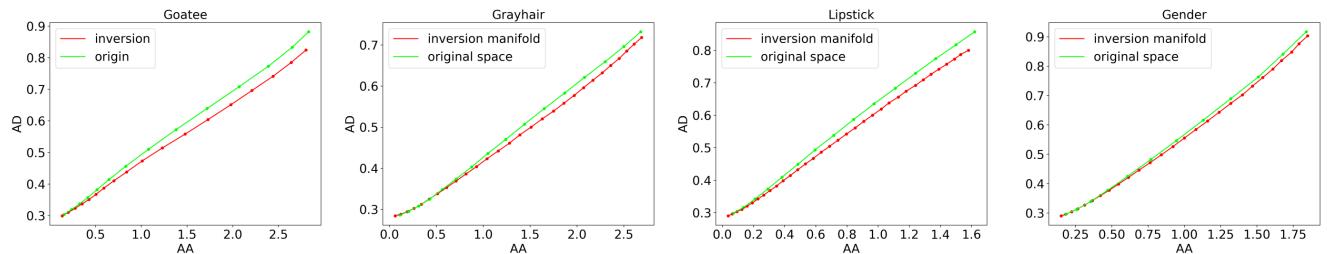
119  
120  
121  
122  
123  
124  
125  
126  
127

Figure 3. e4e(2D) [6]. As the degree of editing  $\alpha$  changes, both Attribute Altering (AA) and Attribute Dependency (AD) change. Lower AD indicates more precise.

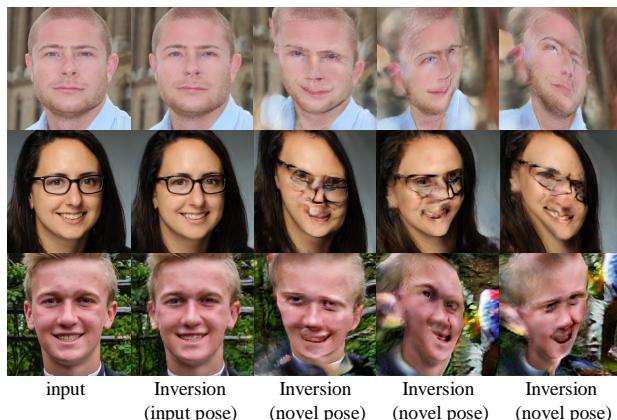
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143

Figure 4. The inversion result of 1,0000 iterations of steps. The naive optimization-based inversion method reconstructs the view of the input camera pose but produces significant artifacts in the views of other camera poses.

144  
145  
146

## E. Fine-tuning the Generator

157  
158  
159  
160  
161

Inspired by Pixel2NeRF [1], we attempted to fine-tune the generator when training the inversion encoder. Unfortunately, there are always some ripple-like artifacts in the hair, which was also observed for Pixel2NeRF, as shown in the figure 5.

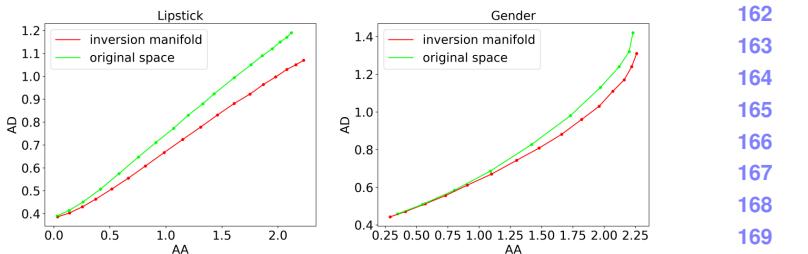


Figure 5. It is complex to train the encoder and fine-tune the generator at the same time. We found some tough ripple-like artifacts in the hair.



Figure 6. Inversion and (hair color) editing results on cat faces.

## F. Beyond Human Face

We conducted some experiments with the AFHQ Cat. We invert the dataset to obtain inversion latent samples. Following GANSpace, We adopt principal component analysis (PCA) to find the semantic directions. The results in the cat

162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215

216  
217  
218  
219  
220

domain are shown in Fig 6.

## G. FID and KID

221  
222  
223  
224  
225

Method	FID <sub>ori</sub>	FID <sub>sm</sub>	FID <sub>mid</sub>	FID <sub>la</sub>	KID <sub>la</sub>
IDE-3D	<b>22.7</b>	<b>36.8</b>	45.2	75.7	0.065
3D-Inv	28.1	40.6	<b>44.9</b>	65.4	0.046
Pixel2NeRF	83.3	85.4	86.2	93.2	0.086
PREIM3D (Ours)	43.6	48.3	50.7	<b>63.3</b>	<b>0.042</b>

226  
227  
228  
229  
230

Table 1. FID & KID comparisons on 1,000 faces from CelebA-HQ. FID<sub>ori</sub> is measured between the inverted images at the original angle and the input images. We use *sm*, *mid*, *la* for uniform samples from yaw [15°, 20°] and pitch [10°, 15°], yaw [25°, 30°] and pitch [15°, 20°], yaw [35°, 40°] and pitch [20°, 25°].

231

We evaluated inversion FID in Table 1. The inception features used in FID focus on the whole image, while our method introduces regularization of the face regions, which makes our FID scores not as good as IDE-3D and 3D-Inv at small angles. However, our model outperforms previous works at large angles. KID shows similar results.

238

## H. Additional Visual Results

240  
241  
242

We provide a large number of inversion and editing results produced by PREIM3D in Figure 7 to 13

243  
244

## References

245  
246  
247  
248  
249

- [1] Shengqu Cai, Anton Obukhov, Dengxin Dai, and Luc Van Gool. Pix2nerf: Unsupervised conditional p-gan for single image to neural radiance fields translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3981–3990, 2022. [2](#)
- [2] Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 1510–1519. IEEE Computer Society, 2017. [1](#)
- [3] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. [1](#)
- [4] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021. [1](#)
- [5] Yujun Shen, Ceyuan Yang, Xiaou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE transactions on pattern analysis and machine intelligence*, 2020. [1](#)
- [6] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021. [2](#)

- [7] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12863–12872, 2021. [1](#)

270  
271  
272  
273  
274  
275  
276  
277  
278  
279280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323



Figure 7. The inversion results obtained by PREIM3D. The first column is the input image.

324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377

378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431



Figure 8. The age editing results obtained by PREIM3D. The first column is the input image.



Figure 9. The eyeglasses editing results obtained by PREIM3D. The first column is the input image.

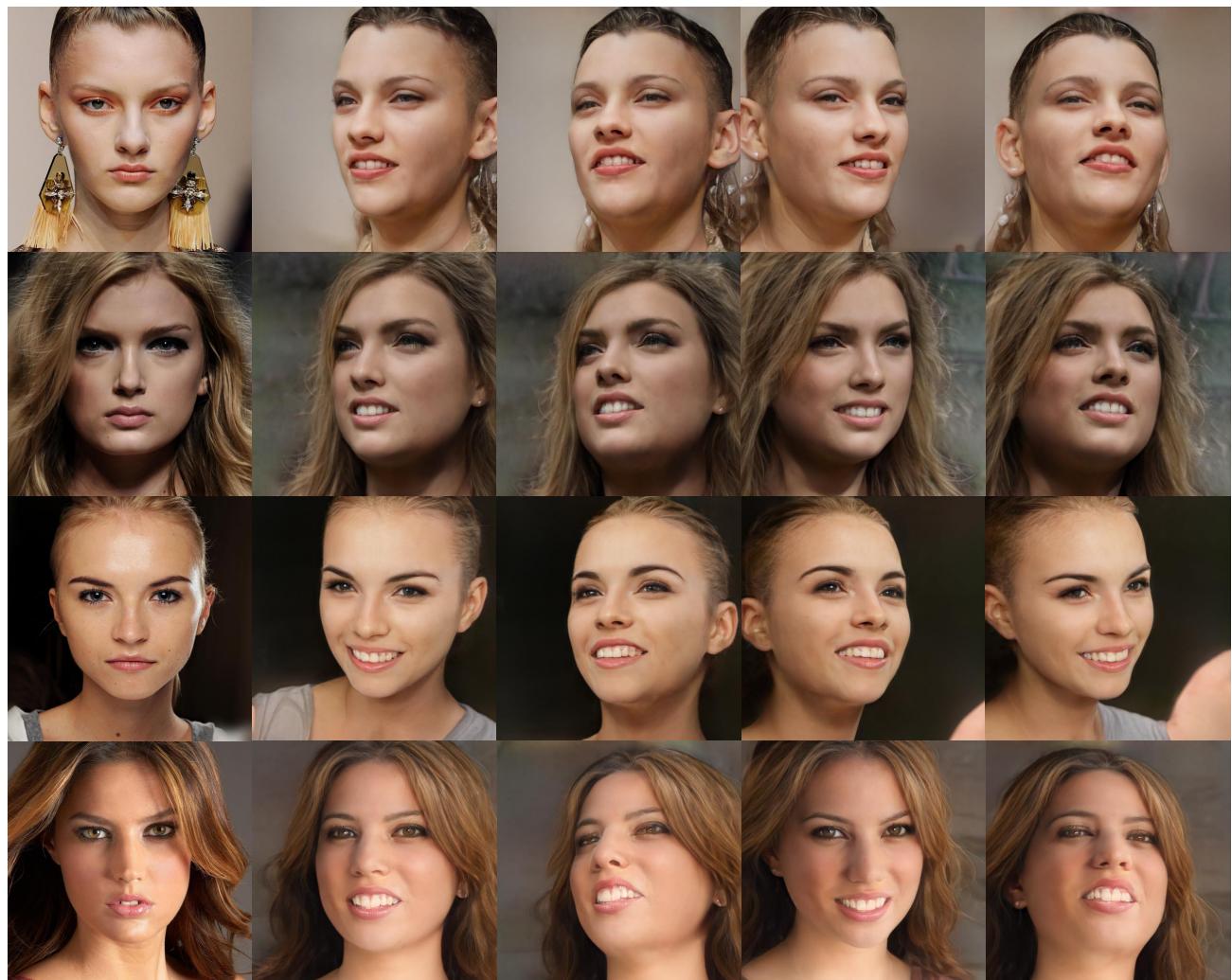


Figure 10. The smile editing results obtained by PREIM3D. The first column is the input image.

648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755



Figure 11. The goatee editing results obtained by PREIM3D. The first column is the input image.

756  
757  
758  
759  
760  
761  
762  
763  
764  
765766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863



Figure 12. The lipstick editing results obtained by PREIM3D. The first column is the input image.

864 918  
865 919  
866 920  
867 921  
868 922  
869 923  
870 924  
871 925  
872 926  
873 927  
874 928  
875 929  
876 930  
877 931  
878 932  
879 933  
880 934  
881 935  
882 936  
883 937  
884 938  
885 939  
886 940  
887 941  
888 942  
889 943  
890 944  
891 945  
892 946  
893 947  
894 948  
895 949  
896 950  
897 951  
898 952  
899 953  
900 954  
901 955  
902 956  
903 957  
904 958  
905 959  
906 960  
907 961  
908 962  
909 963  
910 964  
911 965  
912 966  
913 967  
914 968  
915 969  
916 970  
917 971



Figure 13. The wavy hair editing results obtained by PREIM3D. The first column is the input image.