APPENDIX

I. OTHER DESIGN CHOICES OF STYLEGAN BASED WHITE-BOX INVERSION

A. Different Latent Spaces and Clipping Strategies

Besides the $\mathcal{Z}$, $\mathcal{W}$, and $\mathcal{P}$ spaces we have described, there are other spaces that can be leveraged for white-box inversion. Specifically, while the vanilla StyleGAN uses a $\mathbf{z}$ value (and in turn a $\mathbf{w}$ value) to describe the styles for all the style blocks, we can use separate $\mathbf{z}$ values (and in turn separate $\mathbf{w}$ values) for individual style blocks. The resulting spaces are hence called the $\mathcal{Z}+$ and $\mathcal{W}+$ spaces, respectively. The other design choices hence include performing optimizations in these two spaces.

$\mathcal{W}+$ space has been explored by a number of image embedding and feature editing approaches [9], [10]. The former aims to find a latent value whose corresponding generated sample is as close to a given sample as possible. The latter is to support easy semantic transformations on a given image such as changing nose shape. It is built on the former. That is, semantic transformations can be achieved by changing individual styles of the embedded latent value of the given image. Existing works show that $\mathcal{W}+$ space allows high quality image embedding and feature editing. However, we find $\mathcal{W}+$ space is not a good option for model inversion (see Figure 20). In particular, the optimization in $\mathcal{W}+$ can easily reach very low loss values while the generated samples are unnatural. Our further inspection discloses that the image embedding and feature editing problems have very strong constraints. They make optimization in the over-parameterized $\mathcal{W}+$ space feasible. For example, image embedding is constrained by a reference image. In contrast, model inversion relies on cross-entropy loss, which is under-constrained, evidenced by its vulnerability to adversarial sample attacks.

Another hypothesis is that since the latent values for individual style blocks are independent, the $\mathcal{Z}+$ space may not be entangled and hence amenable to model inversion. However, our experiment shows that $\mathcal{Z}+$ is not good either (see Figure 20). We speculate that although the $\mathbf{z}$ values for different style blocks are separated, they are nonetheless entangled.

When clipping is applied in the $\mathcal{Z}$-related spaces (*i.e.*, z&clip and z+&clip), we clip each dimension into $\mu \pm \sigma$ where $\mu = 0$ and $\sigma = 1$ because it's sampled from the standard normal distribution. When clipping occurs in the $\mathcal{W}$-related spaces (*i.e.*, w&clip and w+&clip), we clip each dimension in the $\mathcal{P}$ space.

B. Different Architectures and Training Datasets

Figure 21 shows inversion results of different architectures (StyleGAN and StyleGAN2) trained on different datasets (CelebA256 and FFHQ). Although StyleGANs traind on FFHQ of higher diversity and better quality generates faces with more natural skin color and better lighting conditions, we decided not to use them in our major experiments because FFHQ doesn't label the identities which means we cannot determine the unseen people to invert and it may impair the validity of our experiments.

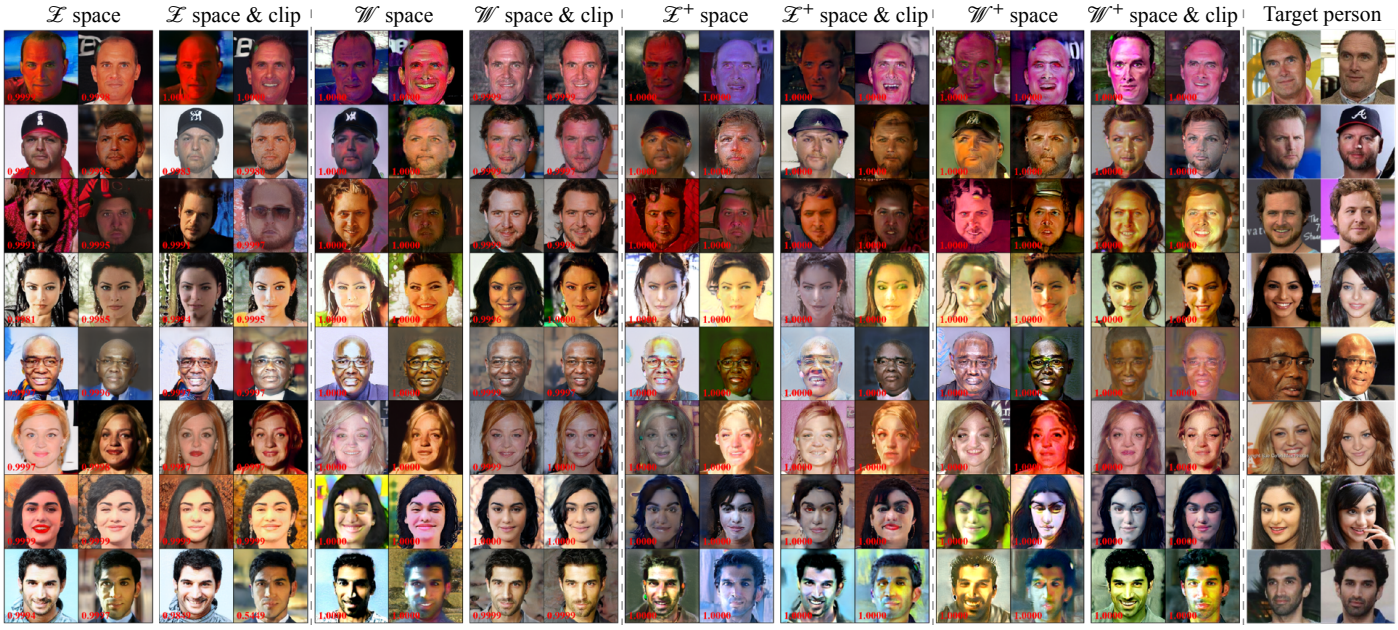

Fig. 20: Qualitative analysis of inversions in different latent spaces and with clipping or not. The $\mathcal{Z}$ space is more entangled than the $\mathcal{W}$ space and thus more difficult for inversion [34], [35]. The $\mathcal{W}^+$ space is more flexible and capable than $\mathcal{W}$ in that arbitrary images can be embedded into $\mathcal{W}^+$ [10], but needs more constraints.

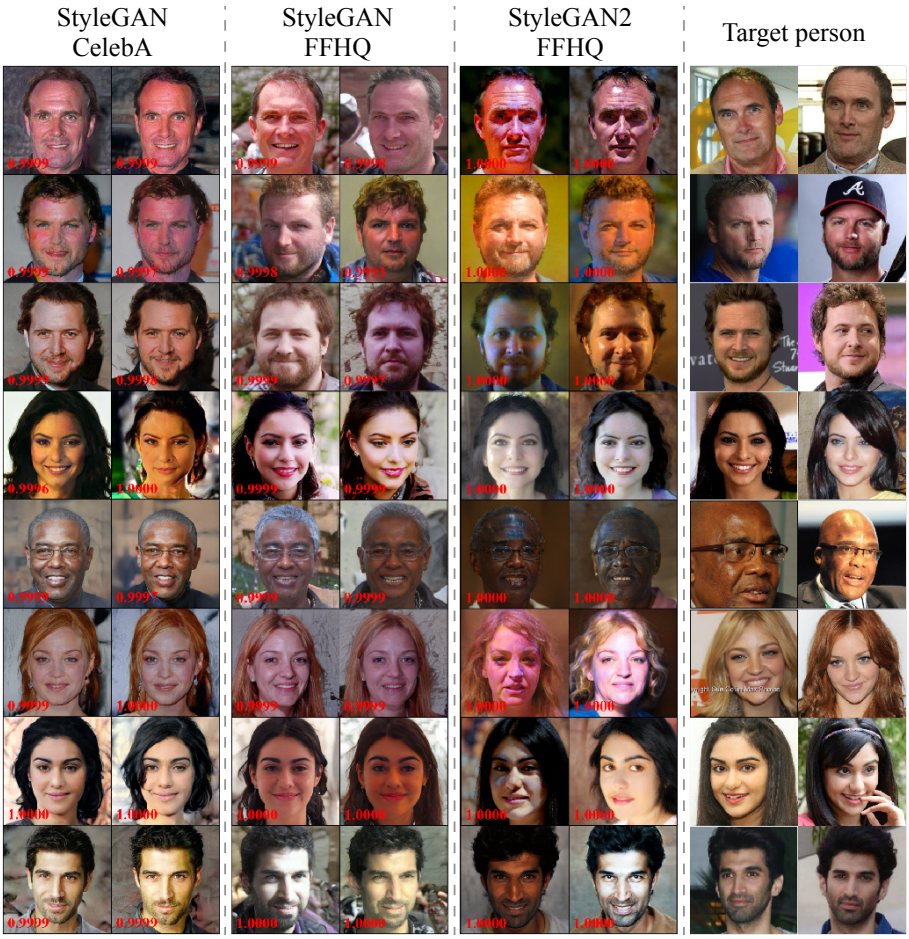

Fig. 21: Qualitative analysis of inversions with different GAN architectures and training datasets.

TABLE VIII: Minutes used to conduct our white-box/black-box inversion.

Method	VGGFace		VGGFace2		CASIA	
	VGG16	VGG16BN	ResNet50	InceptionV1	InceptionV1	SphereFace
MIRROR (white-box)	8.34	7.09	9.58	12.97	12.53	6.85
MIRROR (black-box)	4.37	6.48	8.44	9.13	9.11	4.98

II. Time Cost and Queries

Table VIII shows the average time cost for our white-box/black-box inversion methods for different models. For our white-box inversion method, we invert 100 labels of each model separately with a batch size of 8. That is, for each target label, we invert 8 images. Thus the time cost to generate an image for a target label is computed by dividing the total time cost by 8. For our black-box inversion method, the time cost corresponds to inverting 1 label.

The white-box methods (ours and baselines) need 20k queries. In the black-box settings, AMI needs 104K queries to train the inversion network, and one query to test the inverted result during the attack. MIRROR doesn't require training. It needs less than 100K queries during the genetic search. When evaluating on the commercial Azure service, we use about 2k queries and no abnormal behavior was detected.

III. Human Study

A. Relative Comparison

Figure 22 shows an example of our human study on relatively comparing different methods. This is one question for comparing the $\mathcal{W}$&clip setting with the $\mathcal{W}+$&clip setting.

B. Absolute Performance

Figure 23 shows an example of our human study on absolute performance of MIRROR. We use MIRROR to generate an inverted image for a target label. We also select five real images from the original training set with one from the target label and the others from random labels. We then ask users to choose one that is the person in the inverted image. The average accuracy is 95.71% (standard error is 1.70%) collected from 9 users on 20 questions.

We further evaluate our method in an extreme case. Instead of selecting images from random labels like [22], we select other images from identities similar to the target identity. While finding others that look like the target persons could be subjective, we select similar individuals from the perspective of the target model. Assume the target model is M and the label of interest is t. We go through t's training data D_t and note down the top-5 labels of each image predicted by M. We count the frequency of each label occurring among the top-5 and we select the 4 most popular labels different from t and pick one image for each label as well as an image of t. From those 5 images, we ask users to select the target person given the inverted image. Figure 24 shows an example question. The average accuracy is 89.29% (standard error is 2.97%) collected from 9 users on 20 questions. The decrease of the accuracy is expected as those people indeed look like each other. Nonetheless, from the results of relative comparison, our method still outperforms existing methods significantly.

Similar to the above experiments, but we make it more challenging. For each target person, we go through the CelebA dataset and select 4 images misclassified to the target person by the subject model. Figure 25 shows an example question. The average accuracy is 78.75% (standard error is 7.74%). The decrease is expected because this user study is much more difficult.

IV. User Study via Amazon Mechanical Turk

The volunteers are from Amazon Mechanical Turk (MTurk). We require the participants to have over 90% satisfiability over past surveys. The participants are randomly assigned by Mturk. Based on Mturk's platform statistics, 57% participants are female, 68% participants are under 40 ages, 80% workers are white. We pay $0.5 for each test.

V. Comparison with Loss-based Regularization in $\mathcal{W}$ space

We replace the optimization method in MIRROR with [63]'s and conduct the comparison. We use ResNet50 as the subject model and 100 labels for inversion. MIRROR largely outperforms [63] with the top-1 accuracy of 81.5% vs. 56%, and the top-5 accuracy of 90% vs. 76.88%. Also, our inverted images have a smaller NIQE score (3.46 vs. 3.66). Complete results are in Table IX.

TABLE IX: Comparison between our truncation regularization and [63]'s distance loss.

Metrics	[63]	MIRROR
Accuracy (%) ↑	100 ±0	100 ±0
Ref. Top-1 Acc. (%) ↑	56.00 ±3.67	81.50 ±2.60
Ref. Top-5 Acc. (%) ↑	76.88 ±3.18	90.00 ±1.80
ℓ_2 dist. ↓	143.08 ±1.42	149.96 ±0.76
Ref. ℓ_2 dist. ↓	165.22 ±1.87	147.97 ±2.5
NIQE ↓	3.66	3.46

VI. Random Dropout Implementation

We use VGG16 as an example to illustrate our implementation of random dropout strategy in the white-box setting. The procedure to modify the original M is shown in Algorithm 3. For other networks, different modification may be required. The dropout layer with probability p is created by using the PyTorch class nn.Dropout(p) for fully connected layers or nn.Dropout2d(p) for pooling/convolutional layers. For each neuron, the dropout operation independently conducts a Bernoulli trial and sets its value to 0 with probability p.

Algorithm 3 Random dropout example on VGG16

```
function DROPOUT(model M)
    Randomly select a subset P of M's pooling layers;
    Randomly select a subset F of M's fully connected layers;
    Select dropout probability p according to |P| and |F|;
            ▷ The larger the set size is, the smaller p should be.
    Append the dropout layer with p to each selected layer;
                ▷ Changing source code or using forward hooks.
    return modified M
end function
```

VII. Ineffective Regularization in $\mathcal{Z}$ Space

Figure 26 shows the images inverted using PGGAN with different strategies of regularizing latent vectors. For each inverted image, we accompany it with the histogram of its latent vector. The first block denotes no regularization. The second to fourth blocks denote the inversion results where we truncate the latent vectors to different ranges. The fifth to seventh blocks correspond to the latent loss strategies where we use 1/10/100x loss to pull the mean and variance of the latent vector to 0 and 1, respectively. The last column shows the images of the target people.

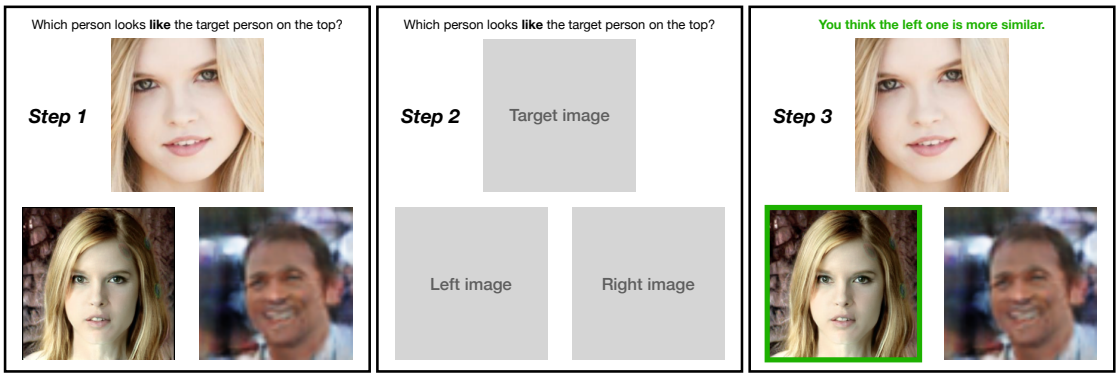

Fig. 22: An example of human study. In Step 1, each worker can observe the images for five seconds. After the time elapsed, the worker are required to select a look-alike. Step 3 only appears in the warm-up, where users are reaffirmed their choices.

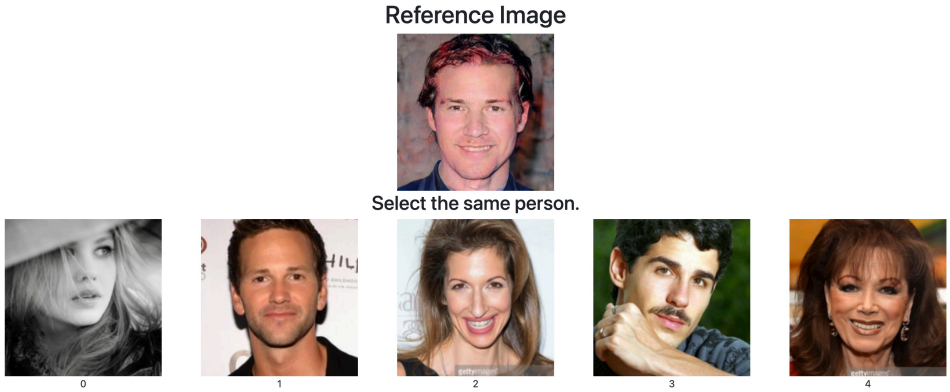

Fig. 23: An example of human study for evaluating absolute performance. Users are given one inverted image as the reference image, one image from the target identity and four random images (each image from a random identity). We ask users to select the same person. The second image from the left is the target person.

VIII. Discriminative Loss Fails to Enforce Naturalness

Figure 27 shows the images inverted by GMI with and without the discriminative loss denoted by the odd and even rows respectively. The original GMI method tries to promote the naturalness of images using the discriminative loss. However, it could not achieve the goal qualitatively (Figure 27) or quantitatively (Table X).

IX. Inversion Models in Other Data Domains

Figure 28 shows the inversion results of MIRROR with a StyleGAN pre-trained with art portraits [11] for a ResNet50 model pre-trained on VGGFACE2 dataset. It's interesting that they actually look like the art portraits of the corresponding target persons.

Figure 29 shows the inversion results of MIRROR with a StyleGAN pre-trained with LSUN Cats [71] for a ResNet18 model trained on 12 different breeds of cats and 2 types of tigers. We can see most target cats are faithfully inverted with details such as face patterns, fur colors/patterns, and even eye colors. For the inverted tigers, although the fur colors and patterns resemble tigers', they still look like cats to some extent. It seems that the cat features learned by the StyleGAN cannot generalize to tigers'.

Figure 30 shows the inversion results of MIRROR with a StyleGAN pre-trained with LSUN Cars [71] for a ResNet34 model trained on 196 different cars. Observe the inverted car types (e.g., sedans, hatchbacks, and sports cars), colors, and shapes are largely correct. In some cases such as the two models in the last row, their front features such as headlights, hoods, grilles and bumpers are precisely inverted. In some other cases, details may be missing such as the bumpers of the Hammer.

X. More Inversion Results in $\mathcal{W}$ Space without Clipping

Figure 31 shows more inversion results in $\mathcal{W}$ space without clipping or with simple clipping. The unnatural results necessitate better regularization.

XI. DeepInvesion with Different Constraints

Figure 32 shows results of DeepInversion with different starting constraints and parameters. The original DeepInversion starts from random noises and uses 1 as the coefficient of its BN loss. We tried to turn down the weights for the variance item in the BN loss. We find that 0.01x variance loss gives more stable features compared to 1x. However, there are multiple overlapping faces and misplaced eyes in the inverted images. Therefore, we propose to add more constraints to encourage the natural combination of the inverted features by providing better starting points such as average faces and cartoon faces. Observe that they indeed help promote the naturalness. However, they are still not comparable to generator-based methods.

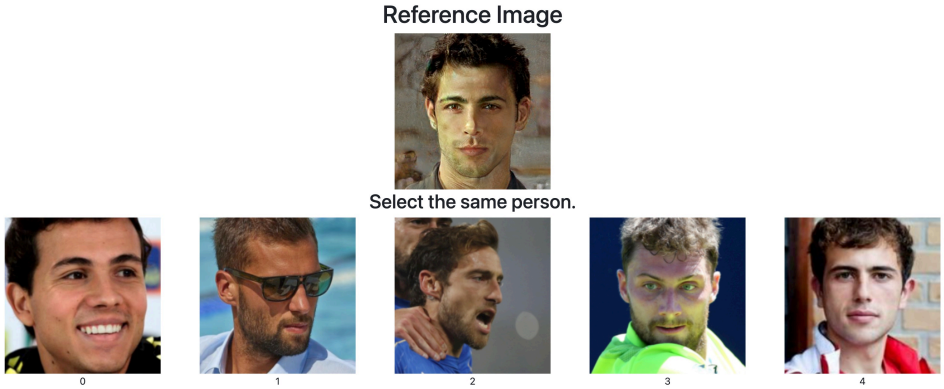

Fig. 24: An example of human study for evaluating absolute performance. Users are given one inverted image as the reference image, one image from the target identity and four images (each image from four most similar identities). We ask users to select the same person. The rightmost image is the target person.

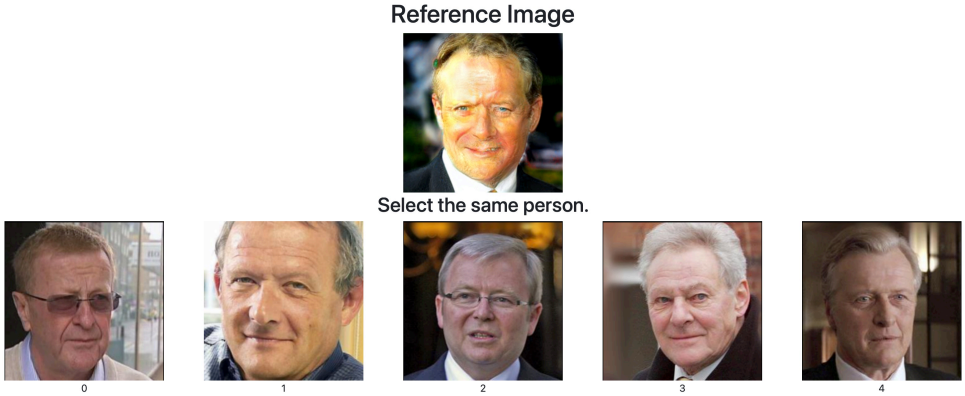

Fig. 25: An example of human study for evaluating absolute performance. Users are given one inverted image as the reference image, one image from the target identity and four images from CelebA misclassified to the target person. We ask users to select the same person. The second image from the left is the target person.

TABLE X: Quantitative comparison between GMI with and without the discriminative loss.

Metric	Method	VGGFace				VGGFace2				CASIA			
		VGG16		VGG16BN		ResNet50		InceptionV1		InceptionV1		SphereFace	
Effectiveness↑	GMI	95.87		96.00		99.88		100.00		97.25		90.12	
	GMI+discri.	96.00		96.00		100.00		99.88		96.63		88.50	
Generalizability↑	GMI	40.37	59.62	44.87	64.50	33.50	52.00	17.75	28.00	9.75	18.50	6.12	9.50
	GMI+discri.	35.00	56.75	45.25	64.38	30.00	49.63	17.13	28.88	8.62	18.75	5.75	9.00
Feature Distance↓	GMI	111.99	102.60	98.37	104.90	154.59	196.26	143.21	188.23	153.43	7.65	8.66	187.64
	GMI+discri.	110.20	103.41	98.87	104.49	155.25	198.51	143.47	188.23	153.34	7.62	8.53	187.73
NIQE↓	GMI	6.82		6.78		6.94		6.57		6.76		6.59	
	GMI+discri.	6.72		6.65		6.89		6.56		6.73		6.51	

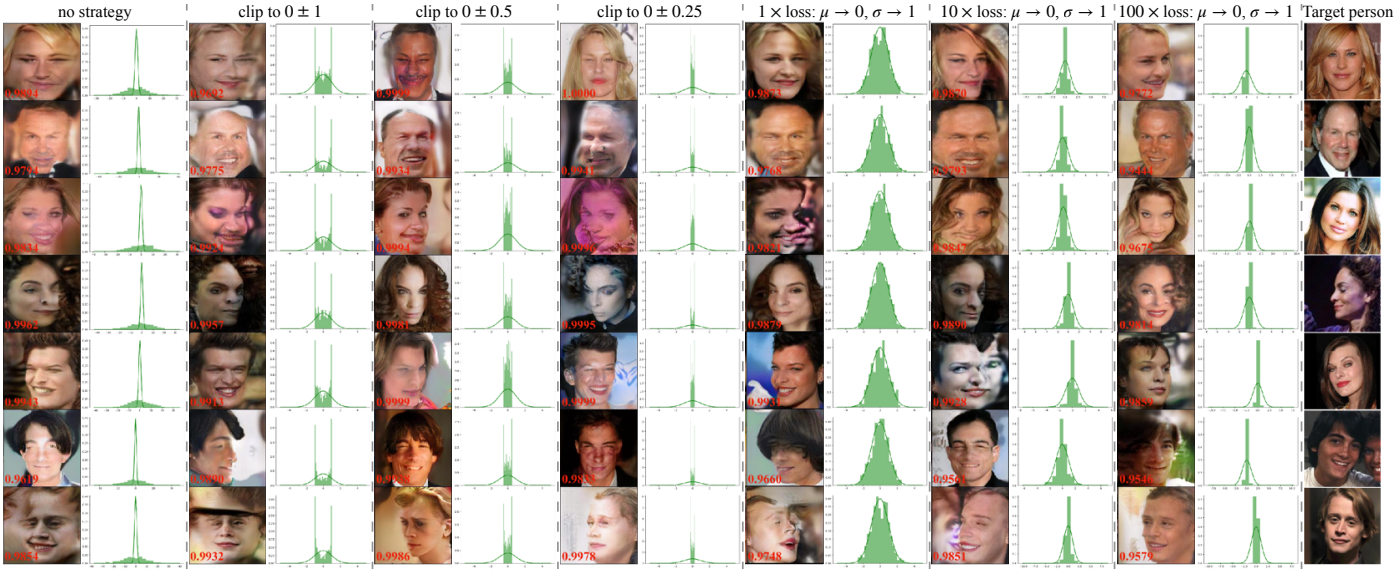

Fig. 26: Optimizing latent vectors in PGGAN with different regularization strategies.

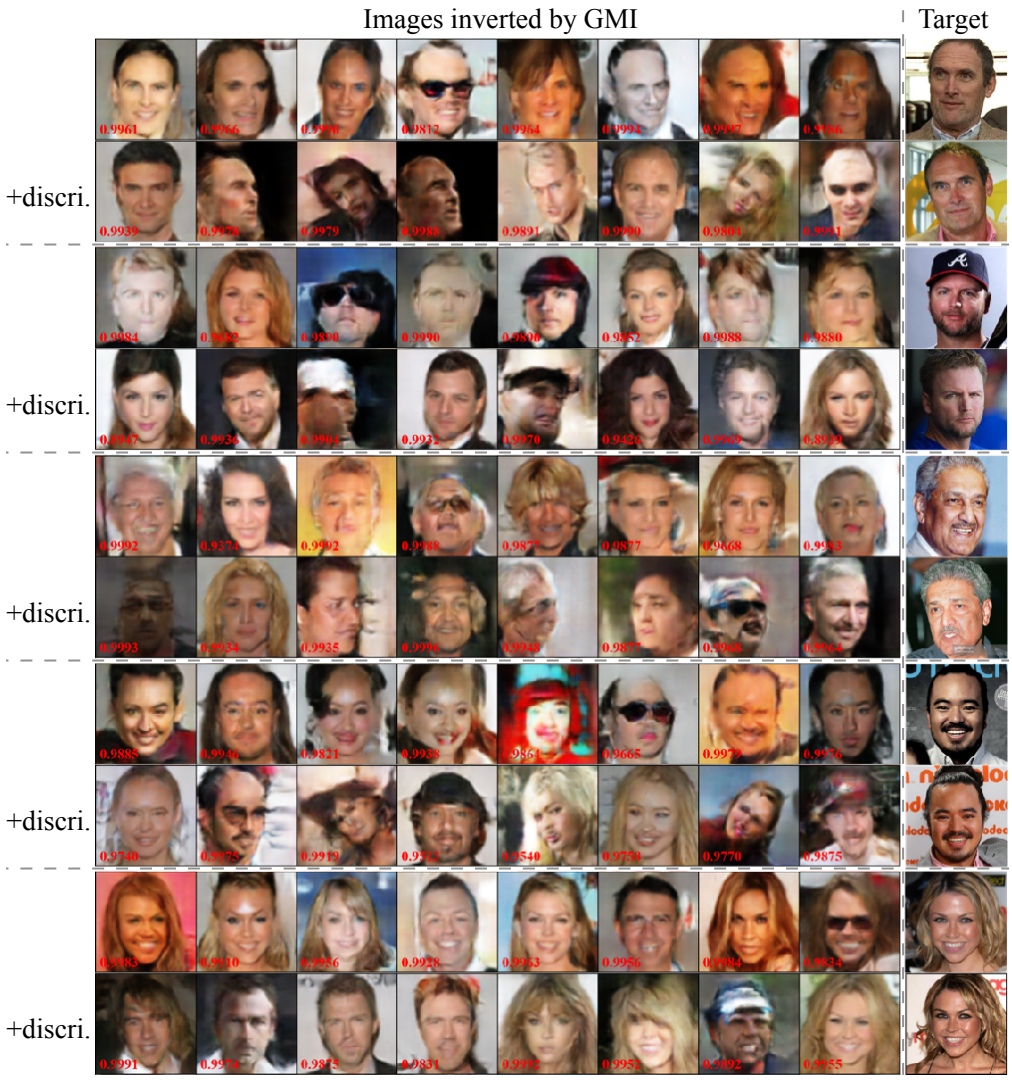

Fig. 27: Images inverted by GMI with and without the discriminative loss.

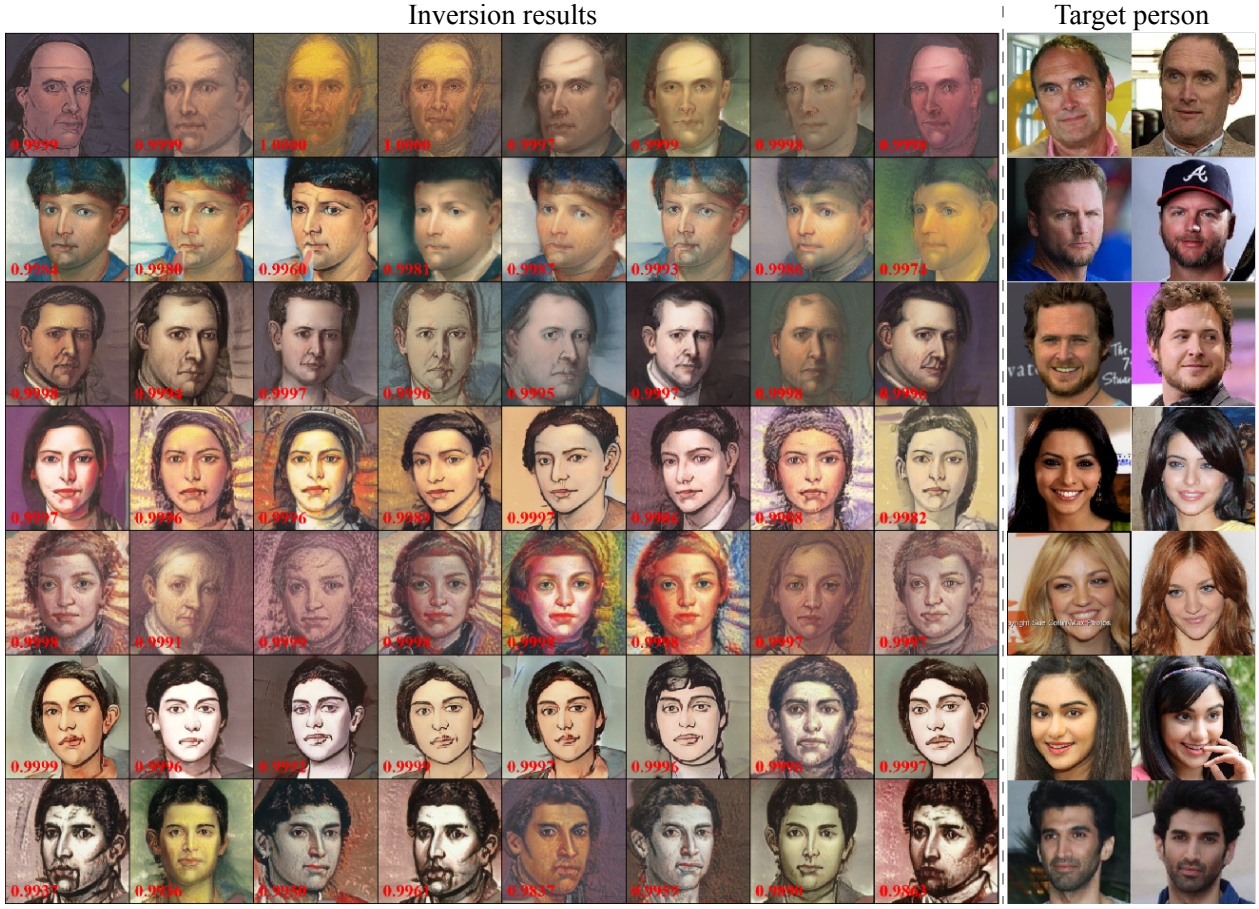

Fig. 28: Inverting ResNet50 pre-trained on VGGFACE2 using StyleGAN pre-trained on art faces.

Fig. 29: Inverting ResNet18 pre-trained on the Oxford-IIIT Pet Dataset with additional Amur tigers and white tigers using StyleGAN pre-trained on LSUN cat dataset.

Fig. 30: Inverting ResNet34 pre-trained on the Stanford Cars Dataset using StyleGAN pre-trained on LSUN car dataset.

Fig. 31: Images inverted in $\mathcal{W}$ space without clipping (odd rows) and with simple clipping (even rows).

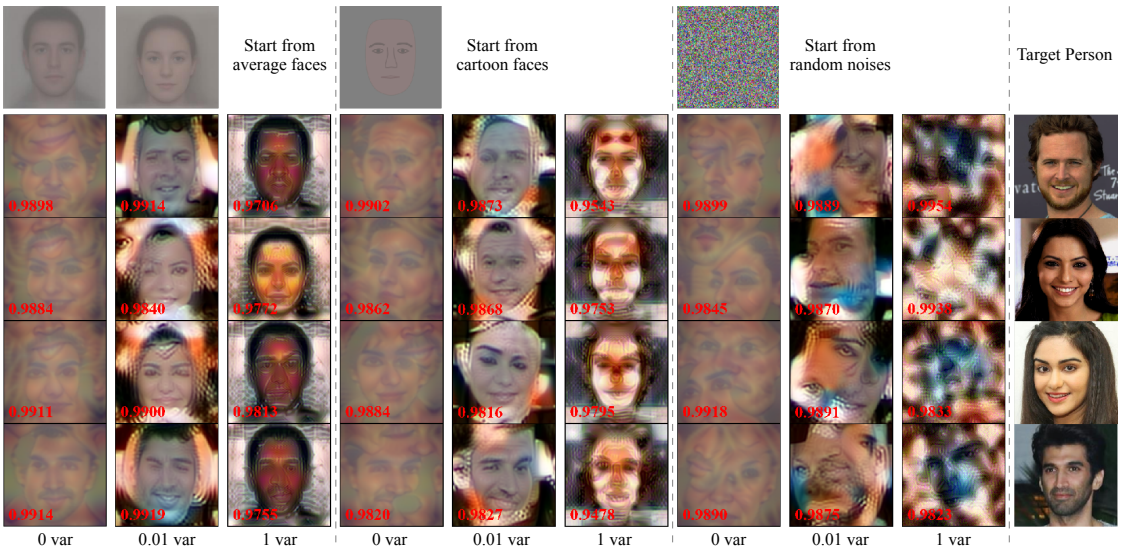

Fig. 32: DeepInversion with various settings.