Manifold Projection for Adversarial Defense on Face Recognition

Jianli Zhou[1,2], Chao Liang[1,2,*], and Jun Chen[1,2]

[1] National Engineering Research Center for Multimedia Software, School of Computer Science, Wuhan University, China
[2] Key Laboratory of Multimedia and Network Communication Engineering, Hubei Province

A Ablation study

To demonstrate the effectiveness of replacing explicit reconstruction loss with adversarial loss, we train two partial variants of A-VAE for comparison. The first variant adds Euclidean distance loss $\mathcal{L}_{rec}$ while keeping $\mathcal{L}_{GAN}$. The second variant adds $\mathcal{L}_{rec}$ and removes $\mathcal{L}_{GAN}$. In particular, we find that it is difficult for the model to synthesize high-resolution images without $\mathcal{L}_{GAN}$, so we expand the input size to 128×128. Figure 1 illustrates the synthesis results of these variants. Explicit reconstruction loss makes the model to restore noise. Table 1 shows the verification accuracies.

Fig. 1. The results produced by variations of A-VAE.

Table 1. Verification accuracies of variants. (Setting: FGSM, gray-box, LFW, VGG-Face2).

LFW (Same identity pairs/Different identities pairs/Average)		
Defense	clean	FGSM $\epsilon = 8$
No Defense	0.992/0.992/0.992	0.190/0.300/0.245
w $\mathcal{L}_{rec}$	0.932/0.998/0.964	0.603/0.850/0.727
w $\mathcal{L}_{rec}$ + w/o $\mathcal{L}_{GAN}$	**0.993**/0.997/**0.995**	0.377/0.387/0.382
A-VAE	0.927/**1.000**/0.963	**0.637/0.863/0.753**

* Corresponding author

B Stability for different resolutions

As shown in Table 2 and Table 3, we evaluate the effectiveness of our method on LFW, at different resolutions. We notice that the performance of our method does not differ much at different resolutions, while others drop severely. Thanks to the mechanism of A-VAE, when the resolution of input is not less than 32×32, the fidelity of the generated images will not be damaged heavily. However, other methods produce low-resolution images, making perturbations more likely to eliminate useful information. This experiment shows the effectiveness of A-VAE on different quality datasets.

Table 2. Verification accuracies of different defense methods at resolution 64.

LFW (Same identity pairs/Different identities pairs/Average)					
Defense	clean	FGSM $\epsilon = 4$	FGSM $\epsilon = 8$	PGD $\epsilon = 8$	C&W
No Defense	0.990/0.993 /0.992	0.447/0.410 /0.428	0.167/0.297 /0.232	0.000/0.013 /0.006	0.000/0.027 /0.013
adversarial FGSM [1]	**0.971**/0.997 /**0.984**	0.473/0.791 /0.632	0.140/0.787 /0.463	0.017/0.203 /0.110	0.000/0.474 /0.238
feature denoising [5]	0.947/0.963 /0.955	0.590/0.740 /0.665	0.197/0.767 /0.482	0.060/0.253 /0.157	0.037/0.603 /0.320
TVM [2]	0.951/0.990 /0.975	0.677/0.987 /0.831	0.267/0.723 /0.495	0.337/0.747 /0.542	0.050/0.567 /0.308
Quilting [2]	0.870/**1.000** /0.935	**0.677**/**0.987** /**0.832**	0.393/**0.943** /0.668	**0.511**/0.973 /**0.742**	0.197/**0.957** /0.577
Pixel Deflection [4]	0.967/0.999 /0.983	0.503/0.827 /0.665	0.153/0.810 /0.482	0.017/0.283 /0.150	0.000/0.563 /0.282
ComDefend [3]	0.967/0.999 /0.983	0.533/0.863 /0.698	0.191/0.821 /0.506	0.077/0.483 /0.280	0.013/0.590 /0.302
A-VAE	0.917/0.997 /0.957	0.633/0.977 /0.805	**0.467**/0.940 /0.703	0.487/**0.983** /0.735	**0.327**/0.947 /**0.637**

Table 3. Verification accuracies of different defense methods at resolution 32.

LFW (Same identity pairs/Different identities pairs/Average)					
Defense	clean	FGSM $\epsilon = 4$	FGSM $\epsilon = 8$	PGD $\epsilon = 8$	C&W
No Defense	0.967/0.997 /0.982	0.283/0.590 /0.436	0.097/0.557 /0.327	0.000/0.030 /0.015	0.000/0.023 /0.015
adversarial FGSM [1]	**0.913**/0.997 /**0.955**	0.247/0.887 /0.567	0.067/0.937 /0.502	0.010/0.483 /0.247	0.023/0.773 /0.398
feature denoising [5]	0.873/0.983 /0.928	0.357/0.863 /0.610	0.107/0.930 /0.518	0.067/0.510 /0.288	0.081/0.803 /0.442
TVM [2]	0.837/**1.000** /0.918	0.390/0.933 /0.662	0.123/0.863 /0.493	0.243/0.893 /0.568	0.090/0.720 /0.405
Quilting [2]	0.683/**1.000** /0.842	0.363/**0.983** /0.673	0.151/**0.981** /0.566	307/**0.981** /0.644	0.147/**0.967** /0.557
Pixel Deflection [4]	0.870/**1.000** /0.935	0.247/0.903 /0.575	0.067/0.953 /0.510	0.017/0.523 /0.270	0.027/0.817 /0.422
ComDefend [3]	0.820/**1.000** /0.910	0.263/0.937 /0.600	0.087/0.937 /0.512	0.060/0.743 /0.402	0.029/0.8401 /0.435
A-VAE	0.830/0.997 /0.913	**0.562**/0.980 /**0.771**	**0.343**/0.963 /**0.653**	**0.453**/0.960 /**0.707**	**0.367**/0.943 /**0.655**

Table 4. Hyperparameters selection.

LFW (Same identity pairs/Different identities pairs/Average)			
	clean	FGSM $\epsilon = 8$	PGD $\epsilon = 8$
$\tau = 0.01$	0.906/0.997/0.952	0.619/0.830/0.725	0.682/0.941/0.812
$\tau = 0.02$	0.913/0.997/0.955	0.630/0.861/0.746	0.679/**0.960**/0.820
$\tau = 0.03$	0.927/**1.000**/0.963	**0.637**/**0.863**/**0.753**	**0.697**/**0.960**/**0.828**
$\tau = 0.06$	**0.937**/0.998/**0.968**	0.623/0.851/0.737	0.680/0.955/0.818

C Hyperparameters selection

We discuss the hyperparameter τ used in inference time that should be determined by experiments. From Table 4, we can find that for clean images, the accuracy increases constantly with the shrink of τ. This shows that the constraint on latent code z inevitably limits the expressiveness of the model. As compensation, when defending against adversarial attacks, it has been shown that projecting images to high probability region enhances robustness of model.

D Network architectures

Table 5, Table 6, and Table 7 show network architectures of the encoder, decoder, and discriminator that we use. Conv$(c, k \times k, s)$ refers to a convolutional layer with c feature maps, filter size $k \times k$, and stride s. LReLU refers leaky ReLU with leakiness 0.2. The skip connection concatenates activations from layer 1 in the encoder to layer 4 in the decoder. The upsampling and downsampling operations correspond to 2×2 element replication and average pooling, respectively.

Table 5. Nerual network architecture of the encoder.

Encoder	
Type	Output shape
Conv(256, 3×3, 1) + InstanceNorm + LReLU	$256 \times 32 \times 32$
Conv(256, 3×3, 2) + InstanceNorm + LReLU	$256 \times 16 \times 16$
Conv(512, 3×3, 1) + InstanceNorm + LReLU	$512 \times 16 \times 16$
Conv(512, 3×3, 2) + InstanceNorm + LReLU	$512 \times 8 \times 8$
Conv(1024, 3×3, 1) + LReLU	$1024 \times 8 \times 8$
Conv(1024, 3×3, 2) + LReLU	$1024 \times 4 \times 4$

E Additional quantitative results on ArcFace

We report results on ArcFace under grey-box and white-box attacks in Table 8 and Table 9.

F Additional qualitative examples

In Figure 2 and Figure 3, we show more stochastic generated results on LFW.

Table 6. Nerual network architecture of the decoder.

Decoder	
Type	Output shape
Const $512 \times 4 \times 4$ + LReLU + AdaIN	$512 \times 4 \times 4$
Conv(512, 3×3, 1) + LReLU + AdaIN	$512 \times 4 \times 4$
Upsample	$512 \times 8 \times 8$
Conv(512, 3×3, 1) + LReLU + AdaIN	$512 \times 8 \times 8$
Conv(512, 3×3, 1) + LReLU + AdaIN	$512 \times 8 \times 8$
Upsample	$512 \times 16 \times 16$
Conv(512, 3×3, 1) + LReLU + AdaIN	$512 \times 16 \times 16$
Conv(512, 3×3, 1) + LReLU + AdaIN	$512 \times 16 \times 16$
Upsample	$768 \times 32 \times 32$
Conv(256, 3×3, 1) + LReLU + AdaIN	$256 \times 32 \times 32$
Conv(256, 3×3, 1) + LReLU + AdaIN	$256 \times 32 \times 32$
Upsample	$256 \times 64 \times 64$
Conv(256, 3×3, 1) + LReLU + AdaIN	$256 \times 64 \times 64$
Conv(256, 3×3, 1) + LReLU + AdaIN	$256 \times 64 \times 64$
Upsample	$256 \times 128 \times 128$
Conv(256, 3×3, 1) + LReLU + AdaIN	$128 \times 128 \times 128$
Conv(256, 3×3, 1) + LReLU + AdaIN	$128 \times 128 \times 128$
Conv(256, 1×1, 1)	$3 \times 128 \times 128$

References

1. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
2. Guo, C., Rana, M., Cissé, M., van der Maaten, L.: Countering adversarial images using input transformations. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018)
3. Jia, X., Wei, X., Cao, X., Foroosh, H.: Comdefend: An efficient image compression model to defend adversarial examples. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 6084–6092. Computer Vision Foundation / IEEE (2019)
4. Prakash, A., Moran, N., Garber, S., DiLillo, A., Storer, J.: Deflecting adversarial attacks with pixel deflection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8571–8580 (2018)
5. Xie, C., Wu, Y., Maaten, L.v.d., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 501–509 (2019)

Table 7. Nerual network architecture of the discriminator.

Dicriminator	
Type	Output shape
Conv(64, 1×1, 1)	$64 \times 128 \times 128$
Conv(128, 3×3, 1) + InstanceNorm + LReLU	$128 \times 128 \times 128$
Conv(256, 3×3, 1) + InstanceNorm + LReLU	$128 \times 128 \times 128$
Downsample	$128 \times 64 \times 64$
Conv(256, 3×3, 1) + InstanceNorm + LReLU	$256 \times 64 \times 64$
Conv(256, 3×3, 1) + InstanceNorm + LReLU	$256 \times 64 \times 64$
Downsample	$256 \times 32 \times 32$
Conv(512, 3×3, 1) + InstanceNorm + LReLU	$512 \times 32 \times 32$
Conv(512, 3×3, 1) + InstanceNorm + LReLU	$512 \times 32 \times 32$
Downsample	$512 \times 16 \times 16$
Conv(512, 3×3, 1) + InstanceNorm + LReLU	$512 \times 16 \times 16$
Conv(512, 3×3, 1) + InstanceNorm + LReLU	$512 \times 16 \times 16$
Downsample	$512 \times 8 \times 8$
Conv(512, 3×3, 1) + InstanceNorm + LReLU	$512 \times 8 \times 8$
Conv(512, 3×3, 1) + InstanceNorm + LReLU	$512 \times 8 \times 8$
Downsample	$512 \times 4 \times 4$
Conv(512, 3×3, 1) + LReLU	$512 \times 4 \times 4$
Conv(512, 4×4, 1) + LReLU	$512 \times 1 \times 1$
Fully-connected	$1 \times 1 \times 1$

Table 8. Verification accuracies of different defense methods on the LFW dataset, under FGSM, PGD, C&W grey-box attacks. The target model is ArcFace.

LFW (Same identity pairs/Different identities pairs/Average)					
Defense	clean	FGSM $\epsilon = 4$	FGSM $\epsilon = 8$	PGD $\epsilon = 8$	C&W
No Defense	0.994/0.994 /0.994	0.542/0.437 /0.489	0.247/0.382 /0.315	0.000/0.020 /0.010	0.000/0.031 /0.015
Adversarial Training [1]	0.980/0.990 /0.985	0.552/0.829 /0.691	0.217/0.780 /0.499	0.031/0.180 /0.106	0.000/0.502 /0.251
Feature Denoising [5]	0.957/0.963 /0.960	0.682/0.731 /0.706	0.203/0.740 /0.472	0.099/0.301 /0.201	0.037/0.550 /0.294
TVM [2]	**0.992**/0.991 /**0.992**	0.761/0.732 /0.747	0.371/0.400 /0.385	0.287/0.381 /0.334	0.007/0.050 /0.029
Quilting [2]	0.984/0.994 /0.989	0.819/0.903 /0.861	0.641/0.670 /0.655	0.690/0.796 /0.743	0.157/0.051 /0.104
ComDefend [3]	0.987/0.991 /0.989	0.502/0.691 /0.597	0.334/0.417 /0.376	0.051/0.130 /0.091	0.000/0.021 /0.011
A-VAE	0.941/**0.999** /0.970	**0.847/0.963 /0.905**	**0.669/0.877 /0.773**	**0.714/0.975 /0.845**	**0.451/0.793 /0.622**

Table 9. Verification accuracies of different defense methods on the LFW dataset, under FGSM, PGD, C&W white-box attacks. The target model is ArcFace.

LFW (Same identity pairs/Different identities pairs/Average)				
Defense	clean	FGSM $\epsilon = 4$	FGSM $\epsilon = 8$	PGD $\epsilon = 8$
No Defense	0.994/0.994/0.994	0.542/0.437/0.489	0.247/0.382/0.315	0.000/0.020/0.010
Adversarial Training [1]	0.980/0.990/0.985	0.407/0.743/0.575	0.208/0.651/0.430	0.000/0.008/0.004
Feature Denoising [5]	0.957/0.963/0.960	0.439/0.500/0.469	0.230/0.452/0.341	0.000/0.037/0.019
ComDefend [3]	**0.987**/0.991/**0.989**	0.501/0.624/0.563	0.281/0.536/0.409	0.189/0.331/0.260
A-VAE	0.941/**0.999**/0.970	**0.758/0.763/0.761**	**0.448/0.662/0.555**	**0.603/0.639/0.621**

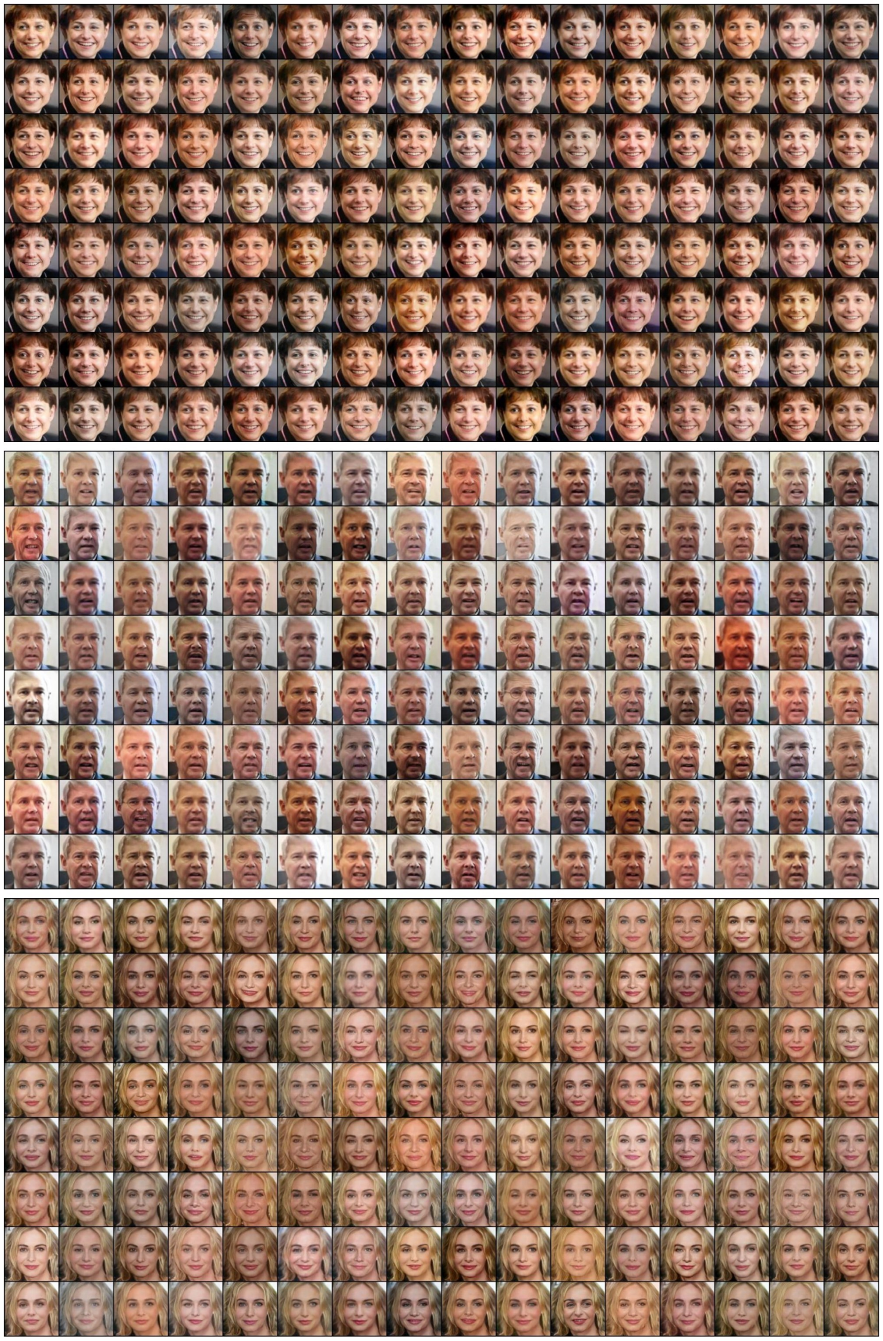

Fig. 2. Stochastic generated results on LFW. The input image is in the upper left corner, and the rest are the realizations of latent code.

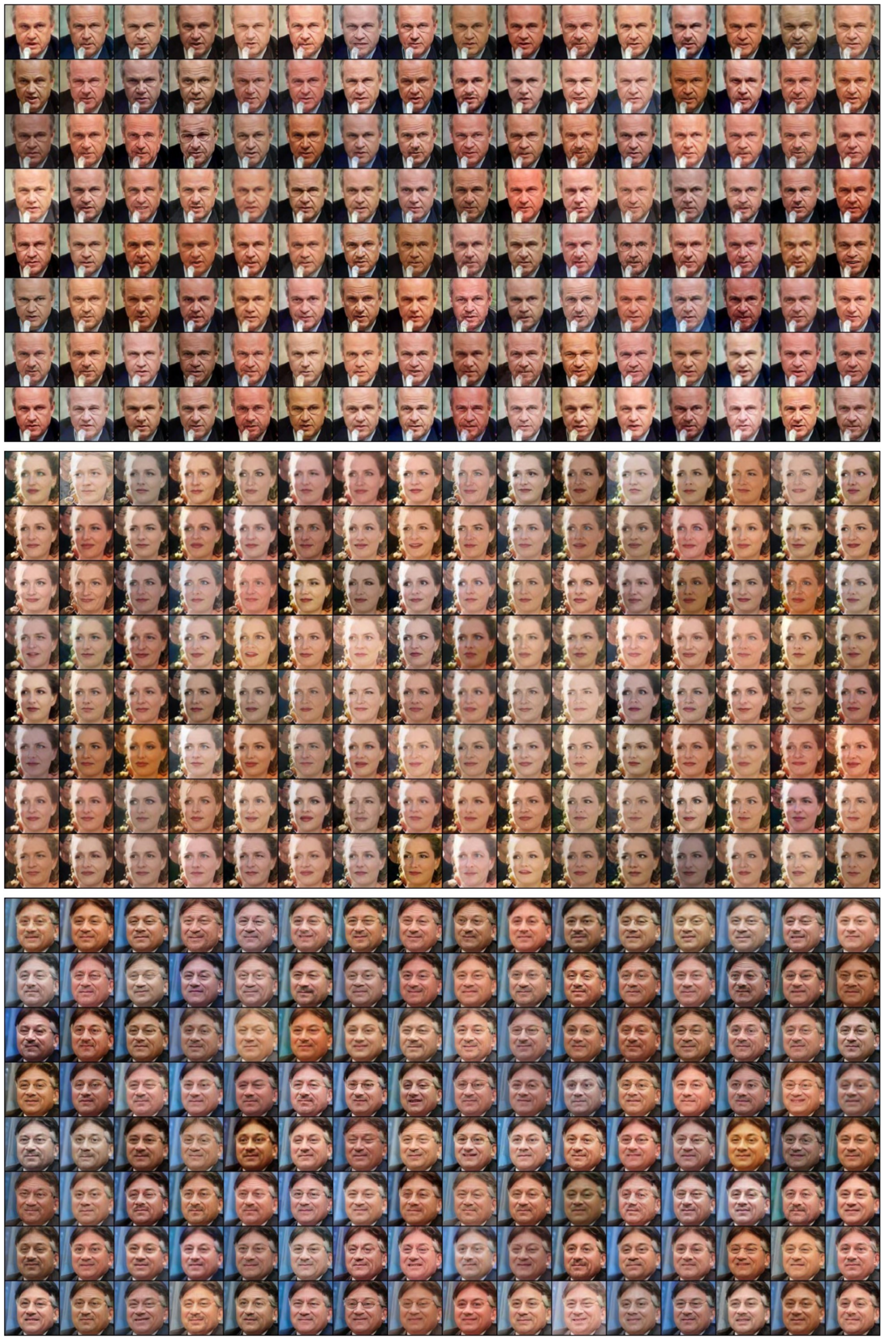

Fig. 3. Stochastic generated results on LFW. The input image is in the upper left corner, and the rest are the realizations of latent code.