Supplement to "API-Net: Robust Generative Classifier via a Single Discriminator"

Xinshuai Dong[1], Hong Liu[1], Rongrong Ji[1] ⋆, Liujuan Cao[1], Qixiang Ye[2], Jianzhuang Liu[3], and Qi Tian[4]

[1] Media Analytics and Computing Lab, Xiamen University
[2] University of Chinese Academy of Sciences
[3] Noah's Ark Lab, Huawei Technologies
[4] Huawei Cloud BU

A Proofs

A.1 Proof of Eq. 2

$$\log p(x,y) = \mathop{\mathbb{E}}_{q(z|x,y)}[\log \frac{p(x,y,z)}{q(z|x,y)} \frac{q(z|x,y)}{p(z|x,y)}] \tag{17}$$

$$= \mathop{\mathbb{E}}_{q(z|x,y)}[\log \frac{p(x,y,z)}{q(z|x,y)}] + \mathcal{D}_{KL}(q(z|x,y)||p(z|x,y)) \tag{18}$$

$$= \mathop{\mathbb{E}}_{q(z|x,y)}[\log p(x,y,z)] + \mathcal{H}(q(z|x,y)) + \mathcal{D}_{KL}(q(z|x,y)||p(z|x,y)) \tag{19}$$

$$\geq \mathop{\mathbb{E}}_{q(z|x,y)}[\log p(x,y,z)], \tag{20}$$

where $\mathcal{H}(.)$ and $\mathcal{D}_{KL}(.||.)$ denote the entropy and KL divergence, respectively, both of which are non-negative.

A.2 Proof of Lemma 1

Let D be the dimension of both x and z, and $p(z|x)$ be a Gaussian:

$$p(z|x) = \mathcal{N}(z|x,\Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}[(z-x)^T \Sigma^{-1}(z-x)]}. \tag{21}$$

Given $\|z-x\|_\infty \leq \epsilon_{ap}$, we have:

$$(z-x)^T \Sigma^{-1}(z-x) \leq |(z-x)^T \Sigma^{-1}(z-x)| \leq {\epsilon_{ap}}^2 \sum_{i=1}^{D}\sum_{j=1}^{D} |(\Sigma^{-1})_{ij}|. \tag{22}$$

Therefore,

$$\log p(z|x) \geq \log \big[\frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}{\epsilon_{ap}}^2 \sum_{i=1}^{D}\sum_{j=1}^{D} |(\Sigma^{-1})_{ij}|}\big] \tag{23}$$

$$= F(\Sigma, D, \epsilon_{ap}), \tag{24}$$

which proves Lemma 1.

⋆ Corresponding author rrji@xmu.edu.cn.

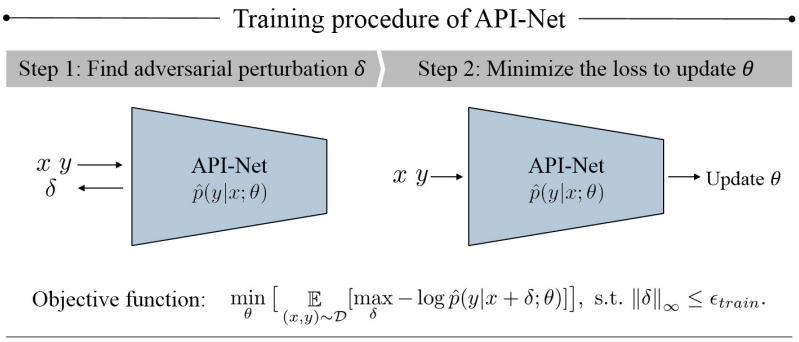

Fig. 7. Visualization of the adversarial training process of API-Net

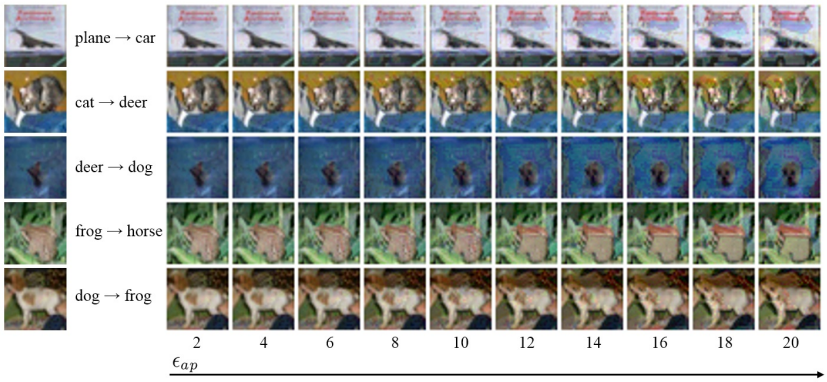

Fig. 8. Visualization of how z changes with ϵ_{ap} conditioned on $y_{(t+1) \bmod C}$ on CIFAR-10. From left to right, ϵ_{ap} varies from 2 to 20 with interval 2 (best view in color with zooming in)

B Additional Experimental Results

B.1 Additional Visualization Results

Visualization of the Proposed Training Process. The training objective function of API-Net is defined in Eqs. 15 and 16. For a more intuitive illustration, We visualize the training process of API-Net in Fig. 7 as a supplement. More detailed training procedure can be seen in Algorithm 1.

Qualitative Study on ϵ_{ap} for CIFAR-10. In Section 4.3, we explored the optimal value of ϵ_{ap} for SVHN. In this section, we investigate the effect of ϵ_{ap} on CIFAR-10 as a supplement. We visualize z conditioned on a wrong class $y_{(t+1) \bmod C}$ and on different ϵ_{ap}, using perturbed images from the first of each class from the test set of CIFAR-10 (and we adjust the number of iterations of

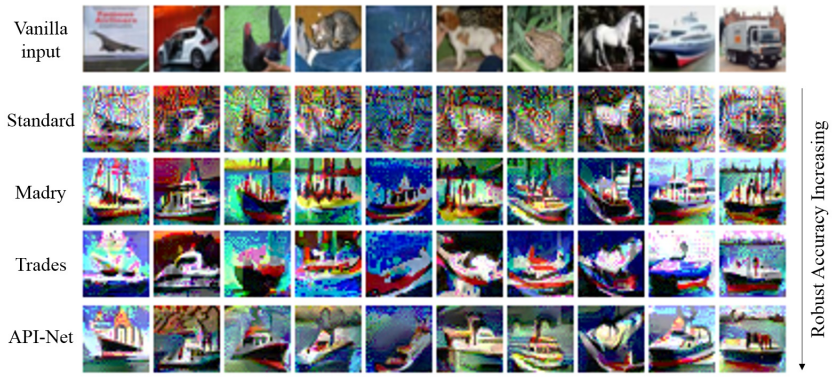

Fig. 9. Visualization of large-ϵ adversarial examples towards a ship for different methods on CIFAR-10 (best view in color with zooming in)

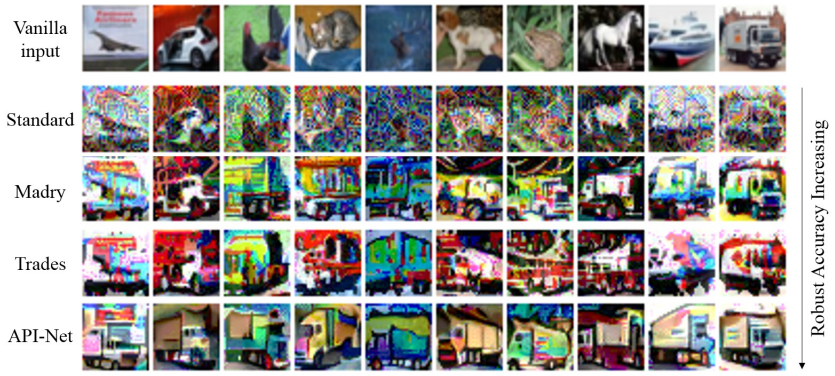

Fig. 10. Visualization of large-ϵ adversarial examples towards a truck for different methods on CIFAR-10 (best view in color with zooming in)

PGD for large ϵ_{ap} to ensure a good convergence). As shown in Fig. 8, when ϵ_{ap} increases, z begins to contain plausible features of class $y_{(t+1) \bmod C}$, suggesting an appropriate value of ϵ_{ap} to prevent confusing the underlying $p(y|z)$ for making predictions.

Visualization of Large-ϵ Adversarial Examples. As mentioned in Section 4.3, we visualize large-ϵ adversarial examples in order to investigate whether the gradients of each model can give rise to perceptually meaningful patterns, which are strongly related to the learned hidden representations of each method. The results, in addition to Fig. 6, are shown in Figs. 9 and 10, illustrating API-Net learns representations that align well with human perception.

Table 6. Accuracy (%) of different methods based on ResNet18 under transfer-based attacks on CIFAR-10 with $\epsilon = 8/255$

Test method	Standard as the proxy			Siamese as the proxy		
	FGSM	PGD-40	C&W-40	FGSM	PGD-40	C&W-40
Standard	-	-	-	40.48	7.64	7.19
Madry	80.30	80.45	80.46	67.70	60.36	60.71
Trades	80.41	80.51	80.75	68.60	62.10	61.06
API-Net	**81.45**	**81.99**	**81.75**	**74.20**	**70.02**	**67.46**

Table 7. Worst accuracy (%) of different methods based on ResNet18 under the PGD-40 attacks with multiple restarts on CIFAR-10 with $\epsilon = 8/255$

Test method	1 restart	10 restarts	20 restarts	50 restarts
Trades	52.1	50.8	50.5	50.2
ME-Net	55.4	48.7	47.2	44.8
API-Net	**63.1**	**54.8**	**54.2**	**53.8**

B.2 Additional Quantitative Results

Accuracy under Transfer-based Attacks. In this section, we compare the accuracy under black-box attacks, where the attacker has no direct access to the victim model. Specifically, we employ the transfer-based attack, which transfers the perturbations computed on a proxy model to fool the victim model.

As demonstrated in Tab. 6, the adversarial examples transferred from the standard model show limited threats to the state-of-the-art methods. When it is under the siamese setting, where a model trained with the same approach but from a different session is used as the proxy, API-Net still keeps leading accuracy, which verifies the robustness of API-Net under a black-box attack scenario.

Worst Accuracy under Attacks with Multiple Restarts. In the main result, we use the accuracy under attacks with random restart to compare with the state-of-the-arts. In this section, we additionally test the worst accuracy under attacks with multiple restarts to rule out some randomness. As shown in Tab. 7, API-Net still keeps the leading position under such strict criteria for evaluation, which again verifies the robustness of API-Net.

B.3 Runtime Considerations

Our implementation is based on PyTorch and we train our models using 2 GeForce GTX1080 GPUs. For CIFAR-10, it takes about 30 hours to train API-Net. For SVHN, it takes about 40 hours to train API-Net. For MNIST, it takes about 8 hours to train API-Net.