Supplementary Material: Prototype-supervised Adversarial Network for Targeted Attack of Deep Hashing

Xunguang Wang[1], Zheng Zhang[1,2], Baoyuan Wu[3,4], Fumin Shen[5,6], Guangming Lu[1]
[1]Harbin Institute of Technology, Shenzhen, [2]Peng Cheng Laboratory
[3]School of Data Science, The Chinese University of Hong Kong, Shenzhen
[4]Secure Computing Lab of Big Data, Shenzhen Research Institute of Big Data
[5]University of Electronic Science and Technology of China, [6]Koala Uran Tech.
{xunguangwang, darrenzz219, wubaoyuan1987, fumin.shen}@gmail.com, luguangm@hit.edu.cn

A. Related Work: Optimization-based Attacks & Generation-based Attacks

In image classification, an adversarial example is usually a carefully modified image, which is intentionally perturbed by adding visually imperceptible perturbations to the original image but can confuse deep model to misclassify it. Since Szegedy *et al.* [20] discovered the properties of adversarial examples, various adversarial attack methods in image classification have been proposed to fool a trained DNN. According to the information of target model exposed to the adversary, adversarial attacks can be categorized as white-box attacks (*e.g.*, FGSM [7], I-FGSM [12], PGD [15] and C&W [3]) and black-box attacks (*e.g.* SBA [17] and ZOO [4]). For white-box attack, the adversary knows the whole network architecture and parameters so that it can design the adversarial perturbations by calculating the gradient of the loss *w.r.t.* inputs. For example, FGSM aims to increase the loss of the target model along the gradient direction once. I-FGSM updates the perturbations multiple times with small step size and reaches better attack performance, which is an iterative variant of FGSM in essence. For black-box attack, only the input and output are available to the adversary, thus it is difficult to get the gradient directly. One solution is that we can use the transferability of adversarial examples to achieve black-box attacks. For instance, SBA adopts adversarial examples generated by a substitute model to attack the target model. Another way directly approximates the gradient base on the input data and output scores, such as ZOO. Although black-box attack is difficult and its success rates are inferior to white-box attack, it is more general and practical.

The above attack methods are optimization-based, which regard the generation of adversarial examples as an optimization problem and use optimizers (*e.g.*, box-constrained L-BFGS [20]) or gradient-based methods to solve it. Optimization-based methods are powerful but quite slow because they need to access the target model iteratively for satisfactory attack performance. Recently, generation-based attack methods received much more attention due to their high-efficiency during test phase. Generation-based attack methods learn a generative model which transforms the input images into the corresponding adversarial samples. Once the generative model trained, it do not need to access the target model again and can generate adversarial examples with one-forward pass. Baluja *et al.* [2] firstly applied a generative model to take a original image as input and to generate its adversarial example. Subsequently Xiao *et al.* [21] used GAN [6] to produce adversarial examples with high perceptual quality. Moreover, Mopuri *et al.* [16] and Poursaeed *et al.* [18] proposed generative architectures to generate adversarial perturbations from any given random noise. Finally, to achieve arbitrary target (category) attack, MAN [8] is designed a special generator which combines the features of the target label and the input image and outputs the targeted adversarial sample.

B. Optimization

The overall framework is actually a generative adversarial network, so the overall objective function can be written as a minimax optimization problem:

$$\begin{aligned}(\theta_p, \theta_g) &= \arg\min \mathcal{L}_{pro}(\theta_p) + \mathcal{L}_{gen}(\theta_g) - \mathcal{L}_{dis}(\hat{\theta}_d) \\ \theta_d &= \arg\max \mathcal{L}_{pro}(\hat{\theta}_p) + \mathcal{L}_{gen}(\hat{\theta}_g) - \mathcal{L}_{dis}(\theta_d) \\ &\text{s.t. } B^{(p)} \in \{-1, 1\}^{K \times M}.\end{aligned} \quad (1)$$

Like all other generative adversarial networks, we optimize the entire network in an alternate way. Firstly, when fixing B and L, we optimize the $\mathcal{L}_{pro}$ over θ_p. Then, we optimize $\mathcal{L}_{gen}$ over θ_g by fixing the parameter θ_p. Finally, we optimize $\mathcal{L}_{dis}$ over θ_d by fixing the parameters θ_p and θ_g. The whole optimization process is outlined in algorithm 1

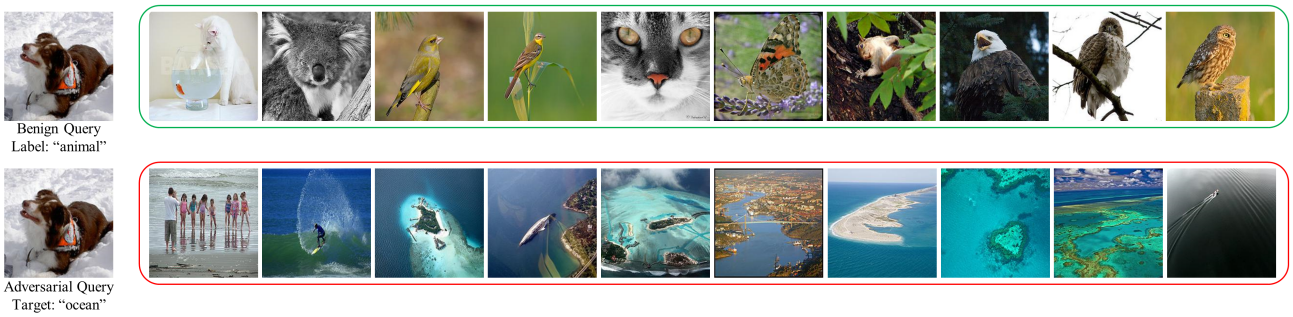


Figure 1. An example to retrieve top-10 similarity samples on NUS-WIDE with the benign query and its adversarial query.

Algorithm 1 Optimization procedure of ProS-GAN.

Input: Image dataset $O = \{(x_i, y_i)\}_{i=1}^{N}$, label set $L = \{y_i\}_{i=1}^{M}$, a pre-trained hashing model $F = sign(f_\theta(\cdot))$, and the hash code matrix B for O produced by F.

Output: Network parameters $(\theta_p, \theta_g, \theta_d)$.

Initialize:

Initialize parameters θ_p, θ_g, θ_d, α_1, α_2, α_3, α, β

Learning rate η, batch size n

while not converge **do**

Provide a batch of image set $\hat{O}$ and target labels $\hat{L}$

Update θ_p by the gradient descent:

$\theta_p \leftarrow \theta_p - \eta \Delta_{\theta_p} \frac{1}{n}(\mathcal{L}_{pro} + \mathcal{L}_{gen} - \mathcal{L}_{dis})$

Update θ_g by the gradient descent:

$\theta_g \leftarrow \theta_g - \eta \Delta_{\theta_g} \frac{1}{n}(\mathcal{L}_{pro} + \mathcal{L}_{gen} - \mathcal{L}_{dis})$

Update θ_d by the gradient descent:

$\theta_d \leftarrow \theta_d - \eta \Delta_{\theta_d} \frac{1}{n}(\mathcal{L}_{dis} - \mathcal{L}_{pro} - \mathcal{L}_{gen})$

end while

for detail. Once the whole networks are trained in convergence, for any given target label and image, ProS-GAN can generate the corresponding adversarial example with a fast forward pass.

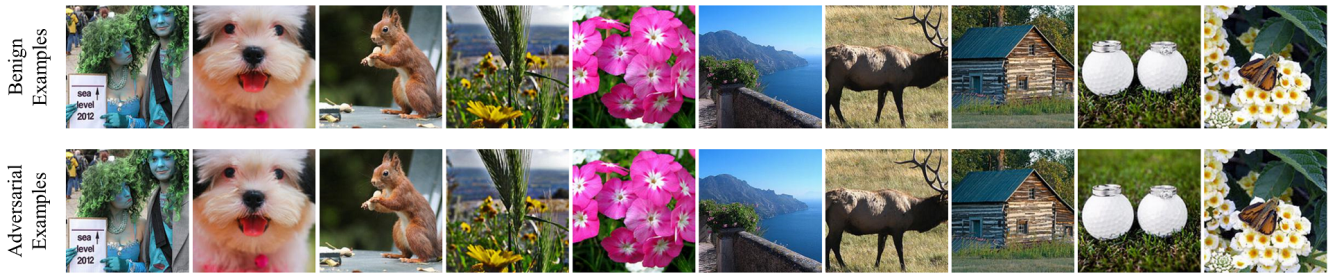
C. Discussion on Differences from the Related Works

Difference from P2P [1] and DHTA [1]. P2P and DHTA heuristically select a hash code from the set of hash codes of samples with the target label as objective code for targeted attack. In contrast to P2P and DHTA, we design a prototype network (PrototypeNet) to learn the prototype code of the target label to supervise the generation of adversarial examples. Because the PrototypeNet is designed to maximize the similarities of hash codes of samples with relevant labels and separability of those with irrelevant labels, the generated prototype code is the more representative and discriminative code of the hash codes of samples with the target label. In this way, they can be used as the target code to achieve more effective targeted attack performance. In addition, compared to gradient-based hashing attack methods [22, 1], our proposed generation-based scheme is clearly faster to produce adversarial examples based on the optimization strategy, which is verified in the experiments section. Therefore, our method is intrinsically different from the existing algorithms, but more efficient and effective for targeted attack of deep hashing.

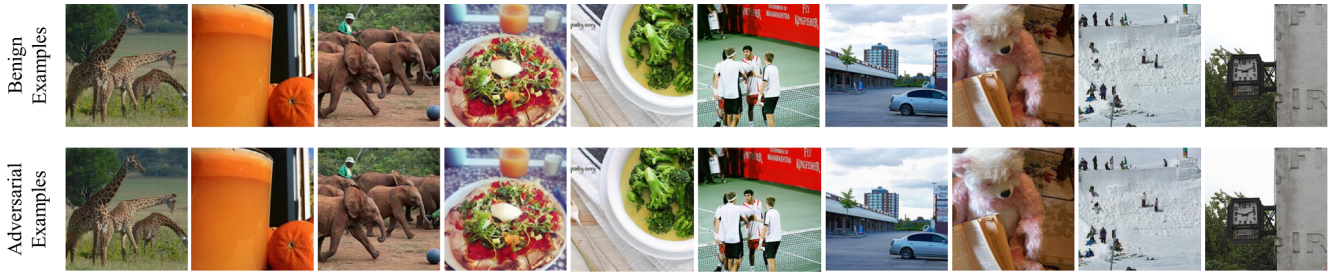
Difference from MAN [8]. MAN designs a special generator to realize arbitrary-label targeted attacks on image classification model by combining input categories and images features. In contrast, our work is conceived for attacking deep hashing models. Due to the difference between classification and hashing, we design an effective PrototypeNet to learn semantic representation and prototype code. Furthermore, we upsample the semantic representation of the target label to the same dimension as the image, and then concatenation them together as the input of the encoder-decoder G_{xt}. In addition, our proposed framework is essentially a generative adversarial network and we employ the adversarial learning between the generator and the discriminator to improve the visual quality of generated adversarial examples. Therefore, our work is totally different from MAN on problem definition, objective design, and framework construction.

Difference from SSAH [13]. In terms of tasks, SSAH is formulated for cross-modal hashing, while ProS-GAN is used for adversarial attack. Particularly, SSAH employs the simple label network for generating semantic hash codes, which could be used to guide the image and text branches. As for differences between PrototypeNet and LabelNet in SSAH [13], our PrototypeNet learns the prototype code from the hash codes produced by the attacked hashing model, while the LabelNet is a self-supervised network to generate semantically preserved hash codes. Moreover, they have different objectives on model design and feature learning purpose. Therefore, they are completely different.

(a) FLICKR-25K



(b) NUS-WIDE



(c) MS-COCO

Figure 2. Visualization examples of generated adversarial samples.

D. Visualization

In this section, we provide some visual examples of the adversarial images generated by ProS-GAN on FLICKR-25K [10], NUS-WIDE [5] and MS-COCO [14]. The comparison results are illustrated in Figure 2. As we can see, the adversarial examples are almost the same as the benign examples (*i.e.* original images). An example of the retrieval results with a benign image and its adversarial example generated by our method is displayed in Figure 1.

E. Transferability

Cross-hash bit transfer: In image hashing, the targeted adversarial perturbations generated from one hash bit can transfer to another hash bit based on the same architecture of hashing model, called cross-hash bit transfer [22]. From Table 1, we observe that the adversarial perturbations with different hash bits can achieve much similar t-MAP. We can see that the results of cross-hash bit transfer are superior to the state-of-the-art DHTA.

Table 1. t-MAP (%) of adversarial examples from one hash bit to another hash bits based on VGG11 backbone for NUS-WIDE.

Method	Code length	12 bits	24 bits	32 bits	48 bits
DHTA	12 bits	74.04	74.86	74.94	74.83
	24 bits	73.62	75.60	75.71	75.69
	32 bits	73.32	75.03	75.65	75.49
	48 bits	72.52	74.19	74.69	75.63
ours	12 bits	77.73	76.20	76.05	76.27
	24 bits	76.60	78.21	78.39	78.25
	32 bits	76.42	77.53	78.25	77.81
	48 bits	75.19	76.06	76.71	78.75

Cross-network transfer: Cross-network transfer means that the adversarial perturbations computed from one DNN can attack another DNN successfully, which is also a black box attack. In this section, we supplement the transfer results on FLICKR-25K and MS-COCO datasets, as summa-

rized in Table 2 and 3. We observe that the adversarial samples generated from one hash code length model has similar targeted attack performance to another code length model based on the same architecture, which is cross-hash bit transfer [22]. For example, applying the adversarial examples generated by ProS-GAN from DH-AlexNet to attack DH-AlexNet* on MS-COCO (Table 3) can achieve the similar t-MAP result (67.67%) for 66.26% of DH-AlexNet. In most cases, the cross-hash bit transfer results of ProS-GAN are better than DHTA. In addition, it is known that the advesarial examples computed from one backbone network can attack another network, called *cross-network transfer* [22]. Our ProS-GAN also has better cross-network transfer than DHTA. For example, in Table 2, when we adopt the adversarial examples generated from DH-VGG11 to attack DH-ResNet18*, the t-MAP is 82.25%, which is higher than 72.26% of DHTA. From these results, we conclude that the adversarial samples generated by our method have better transferability.

References

[1] Jiawang Bai, Bin Chen, Yiming Li, Dongxian Wu, Weiwei Guo, Shu-tao Xia, and En-hui Yang. Targeted attack for deep hashing based retrieval. In *European Conference on Computer Vision*, pages 618–634. Springer, 2020. 2

[2] Shumeet Baluja and Ian Fischer. Adversarial transformation networks: Learning to generate adversarial examples. *arXiv preprint arXiv:1703.09387*, 2017. 1

[3] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57. IEEE, 2017. 1

[4] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26, 2017. 1

[5] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: a real-world web image database from national university of singapore. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 1–9, 2009. 3

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 1

[7] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015. 1

[8] Jiangfan Han, Xiaoyi Dong, Ruimao Zhang, Dongdong Chen, Weiming Zhang, Nenghai Yu, Ping Luo, and Xiaogang Wang. Once a man: Towards multi-target attack via learning multi-target adversarial network once. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5158–5167, 2019. 1, 2

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 5

[10] Mark J Huiskes and Michael S Lew. The mir flickr retrieval evaluation. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 39–43, 2008. 3

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 5

[12] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *International Conference on Learning Representations*, 2017. 1

[13] Chao Li, Cheng Deng, Ning Li, Wei Liu, Xinbo Gao, and Dacheng Tao. Self-supervised adversarial hashing networks for cross-modal retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4242–4251, 2018. 2

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755. Springer, 2014. 3

[15] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*, 2017. 1

[16] Konda Reddy Mopuri, Utkarsh Ojha, Utsav Garg, and R Venkatesh Babu. Nag: Network for adversary generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 742–751, 2018. 1

[17] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519, 2017. 1

[18] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4422–4431, 2018. 1

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015. 5

[20] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations*, 2014. 1

[21] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *Proceedings of the International Joint Conference on Artificial Intelligence*, 2018. 1

[22] Erkun Yang, Tongliang Liu, Cheng Deng, and Dacheng Tao. Adversarial examples for hamming space search. *IEEE Transactions on Cybernetics*, 2018. 2, 3, 4

Table 2. Transfer t-MAP (%) on FLICKR-25K dataset. DH-AlexNet, DH-VGG11 and DH-ResNet18 denote 12 bits DPSH models based on AlexNet [11], VGG11 [19] and ResNet18 [9], respectively, and "*" denotes their 32 bits variants.

Method	Attacked model	DH-AlexNet	DH-AlexNet*	DH-VGG11	DH-VGG11*	DH-ResNet18	DH-ResNet18*
DHTA	DH-AlexNet	82.26	82.50	67.63	67.88	67.62	67.37
	DH-AlexNet*	81.89	84.08	67.68	68.18	67.52	67.56
	DH-VGG11	64.73	64.40	86.27	87.35	72.32	72.26
	DH-VGG11*	65.13	64.82	86.29	88.35	73.10	73.61
	DH-ResNet18	63.50	63.33	66.19	65.81	85.48	86.55
	DH-ResNet18*	63.42	63.27	65.92	65.61	85.38	87.80
ProS-GAN	DH-AlexNet	84.89	84.52	78.72	79.09	78.45	78.06
	DH-AlexNet*	80.95	81.99	77.44	77.48	76.72	77.43
	DH-VGG11	73.67	73.07	89.05	88.14	81.90	82.25
	DH-VGG11*	73.34	72.77	89.00	91.10	79.32	79.92
	DH-ResNet18	73.50	72.45	75.90	75.22	87.95	87.65
	DH-ResNet18*	72.42	72.95	76.18	75.97	86.25	88.19
Original		62.83	62.61	63.58	63.49	63.23	63.20

Table 3. Transfer t-MAP (%) on COCO dataset. DH-AlexNet, DH-VGG11 and DH-ResNet18 denote 12 bits DPSH models based on AlexNet [11], VGG11 [19] and ResNet18 [9], respectively, and "*" denotes their 32 bits variants.

Method	Attacked model	DH-AlexNet	DH-AlexNet*	DH-VGG11	DH-VGG11*	DH-ResNet18	DH-ResNet18*
DHTA	DH-AlexNet	57.05	58.39	45.23	45.83	45.03	45.28
	DH-AlexNet*	55.88	58.35	45.15	45.86	44.94	45.24
	DH-VGG11	44.49	45.16	59.85	61.78	51.61	51.56
	DH-VGG11*	44.35	44.95	59.12	63.22	50.38	51.18
	DH-ResNet18	42.98	43.77	44.56	44.77	61.88	64.44
	DH-ResNet18*	42.95	43.73	44.13	44.54	62.12	65.42
ProS-GAN	DH-AlexNet	66.26	67.67	53.79	55.57	54.22	55.22
	DH-AlexNet*	66.49	69.41	53.73	55.45	54.56	55.14
	DH-VGG11	49.75	51.00	66.22	65.35	56.14	54.49
	DH-VGG11*	50.14	51.21	67.08	71.65	58.23	57.48
	DH-ResNet18	49.30	49.67	49.47	51.23	70.27	72.07
	DH-ResNet18*	48.87	50.43	49.36	51.60	68.75	72.95
Original		42.41	43.24	42.33	42.67	42.40	42.85