Appendix: Backdoor Attack on Hash-based Image Retrieval via Clean-label Data Poisoning

Kuofeng Gao[1 *]
gkf21@mails.tsinghua.edu.cn
Jiawang Bai[1 *]
bjw19@mails.tsinghua.edu.cn
Bin Chen[2, 4 †]
chenbin2021@hit.edu.cn
Dongxian Wu[3]
d.wu@k.u-tokyo.ac.jp
Shu-Tao Xia[1, 4]
xiast@sz.tsinghua.edu.cn

[1] Tsinghua Shenzhen International Graduate School, Tsinghua University, China
[2] Harbin Institute of Technology, Shenzhen, China
[3] University of Tokyo, Japan
[4] Peng Cheng Laboratory, China

* Equal contribution
† Corresponding author

A Proof of Theorem 1

Theorem 1: The objective function in Eqn. (5) is an upper bounded loss, *i.e.*,

$$\begin{aligned}&\lambda \cdot L_c(\{\boldsymbol{\eta}_i\}_i^M) + (1-\lambda)\cdot \frac{1}{M}\sum_{i=1}^{M} L_a(\boldsymbol{\eta}_i)\\ \leq\ &\begin{cases}\dfrac{\lambda K\cdot M^2}{4M(M-1)} + (1-\lambda)K, & M \text{ is even;}\\ \dfrac{\lambda K\cdot M^2-1}{4M(M-1)} + (1-\lambda)K, & M \text{ is odd.}\end{cases}\end{aligned}$$

where each term has its respective achievable upper bound. Moreover, the overall upper bound can be achievable, if and only if $\sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M} d_H(F(\boldsymbol{x}_i),F(\boldsymbol{x}_j))$ is maximum.

Proof: Note that the first term of Eqn. (6) is equivalent to the maximization of the average Hamming distance among M binary codewords. Without loss of generality, we denote these M codewords as $\boldsymbol{h}_1,\boldsymbol{h}_2,\ldots,\boldsymbol{h}_M \in \{-1,+1\}^K$, *i.e.*, $F(\boldsymbol{x}_i)=\boldsymbol{h}_i$, $i=1,2,\cdots,M$.

Then we have

$$\begin{aligned} L_c(\{\boldsymbol{h}_i\}_{i=1}^M) &= \frac{1}{M(M-1)}\sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M} d_H(\boldsymbol{h}_i,\boldsymbol{h}_j) \\ &= \frac{1}{M(M-1)}\sum_{k=1}^{K}\sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M} \mathbb{I}\{h_i^k \neq h_j^k\} \\ \text{where} \quad & \boldsymbol{h}_i = (h_i^1, h_i^2, \ldots, h_i^K),\ i = 1,2,\ldots,M. \end{aligned}$$

For the k-th bit, $k = 1,,2,\ldots,K$, we let

$$N_{+1}^k \triangleq |\{h_i^k \mid h_i^k = +1, i = 1,2,\ldots,M\}|, \text{ and}$$

$$N_{-1}^k \triangleq |\{h_i^k \mid h_i^k = -1, i = 1,2,\ldots,M\}|.$$

Obviously, we have $N_{+1}^k + N_{-1}^k = M$, then

$$\sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M} \mathbb{I}\{h_i^k \neq h_j^k\} = N_{+1}^k \times N_{-1}^k \leq \frac{(N_{+1}^k + N_{-1}^k)^2}{4} = \frac{M^2}{4},$$

where the inequality holds with equality if and only if $N_{+1}^k = N_{-1}^k = M/2$. Since N_{+1}^k and N_{-1}^k are integers, we have

$$\sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M} \mathbb{I}\{h_i^k \neq h_j^k\} \leq \begin{cases} \dfrac{M^2}{4}, & M \text{ is even;} \\ \dfrac{M^2-1}{4}, & M \text{ is odd.} \end{cases}$$

Therefore, we can obtain

$$L_c(\{\boldsymbol{h}_i\}_{i=1}^M) \leq \begin{cases} \dfrac{K\cdot M^2}{4M(M-1)}, & M \text{ is even;} \\ \dfrac{K\cdot M^2-1}{4M(M-1)}, & M \text{ is odd.} \end{cases}$$

For any given binary codeword $F(\boldsymbol{x}_i)$, $i = 1,2,\ldots,M$, we can always find a binary codeword whose coordinate takes opposite value of $F(\boldsymbol{x}_i)$'s, *i.e.*, flipping $+1$ to -1, and vice versa, to achieve the maximum Hamming distance. Thus we have

$$L_a(\boldsymbol{\eta}_i) \leq K,\ i = 1,2,\ldots,M.$$

In summary, we can obtain the following upper bound on Eqn. (8):

$$\begin{aligned} & \lambda \cdot L_c(\{\boldsymbol{\eta}_i\}_i^M) + (1-\lambda)\cdot\frac{1}{M}\sum_{i=1}^{M} L_a(\boldsymbol{\eta}_i) \\ \leq & \begin{cases} \dfrac{\lambda K\cdot M^2}{4M(M-1)} + (1-\lambda)K, & M \text{ is even;} \\ \dfrac{\lambda K\cdot M^2-1}{4M(M-1)} + (1-\lambda)K, & M \text{ is odd.} \end{cases} \end{aligned}$$

Next we prove the necessary and sufficient condition for the case that the overall upper bound is achievable. Note that the overall upper bound is achievable is equivalent to that both upper bounds on the two terms of Eqn. (6) are achievable by the same optimal solutions. The first half of the proof reveals that second term's optimal solutions can be obtained by directly flipping the sign of each bit to achieve the maximum Hamming distance. Thus we only need to prove the following equivalent statement: *The overall upper bound can be achievable by the second term's optimal solutions, if and only if* $\sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M} d_H(F(\boldsymbol{x}_i),F(\boldsymbol{x}_j))$ *is maximum.*

Firstly, we need to prove the following claim:

Claim 1: Given any two binary codewords $\boldsymbol{h}_1,\boldsymbol{h}_2$, flipping the sign of their bits does not change their Hamming distance.

Proof of Claim 1: Note that

$$d_H(\boldsymbol{h}_1,\boldsymbol{h}_2) = |\mathcal{S}| \triangleq |\{k \mid h_1^k \neq h_2^k, k = 1,2,\cdots,K\}|.$$

Flipping the sign of their bits still makes the corresponding bits among $\{1,2,\cdots,K\}\setminus\mathcal{S}$ preserve the same sign, and the bits belonging to $\mathcal{S}$ take opposite sign as usual. So the Hamming distance between $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ remains unchanged.

Now we are ready to prove the equivalent statement. Note that the overall upper bound can be achievable by the second term's optimal solution, if and only if the upper bound on the first term can be also achieved by the second term's optimal solutions $\hat{F}(\boldsymbol{x}_i)$, $i = 1,,2,\ldots,M$, by flipping the sign of the bits of $F(\boldsymbol{x}_i)$, $i = 1,,2,\ldots,M$. By Claim 1, we know that flipping the sign of the bits of $F(\boldsymbol{x}_i)$'s would not change their Hamming distances, *i.e.*,

$$\sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M} d_H(F(\boldsymbol{x}_i),F(\boldsymbol{x}_j)) = \sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M} d_H(\hat{F}(\boldsymbol{x}_i)),\hat{F}(\boldsymbol{x}_j))$$

Thus, $\sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M} d_H(F(\boldsymbol{x}_i),F(\boldsymbol{x}_j))$ is maximum.
$\Leftrightarrow \sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M} d_H(\hat{F}(\boldsymbol{x}_i)),\hat{F}(\boldsymbol{x}_j))$ is maximum.
$\Leftrightarrow$ The overall upper bound can be achievable by the second term's optimal solutions.

B Algorithms Outline

Algorithm 1 Confusing Perturbations Generation

Input: The clean-trained deep hashing model $F(\cdot)$, the samples to be poisoned $\{(\boldsymbol{x}_i, \boldsymbol{y}_t)\}_{i=1}^{M}$ from the target class $\boldsymbol{y}_t$, the perturbation magnitude ϵ, the hyper-parameter λ, the number of epochs E, the batch size B, the step size α.

Output: Confusing perturbations $\{\boldsymbol{\eta}\}_i^M$

Initialize the perturbations $\{\boldsymbol{\eta}_i\}_i^M$
for $epoch = 1, \ldots, E$ **do**
  for each batch $\{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^{B}$ from $\{(\boldsymbol{x}_i, \boldsymbol{y}_t)\}_{i=1}^{M}$ **do**
    Calculate the loss:
      $\lambda \cdot L_c(\{\boldsymbol{\eta}_i\}_i^B) + (1-\lambda) \cdot \frac{1}{B} \sum_{i=1}^{B} L_a(\boldsymbol{\eta}_i)$
    for $j = 1, \ldots, B$ **do**
      Calculate the gradient $\boldsymbol{g}_j$ $w.r.t.$ $\boldsymbol{\eta}_j$
      Update perturbations $\boldsymbol{\eta}_j = \boldsymbol{\eta}_j + \alpha \cdot \text{sign}(\boldsymbol{g}_j)$
      Clip $\boldsymbol{\eta}_j$ to $(-\epsilon, \epsilon)$
    end for
  end for
end for

Algorithm 2 Trigger Pattern Generation

Input: The clean-trained deep hashing model $F(\cdot)$, the training set $\boldsymbol{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$, the trigger mask $\boldsymbol{m}$, the trigger size r, the number of iterations T, the batch size B, the steps size α.

Output: Trigger pattern $\boldsymbol{p}$

Initialize the trigger $\boldsymbol{p}$ with the trigger size r
Calculate $\boldsymbol{h}_a$ by solving Eqn. (4)
for $iteration = 1, \ldots, T$ **do**
  Sample a batch $\boldsymbol{S} = \{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^{B}$ from $\boldsymbol{D}$
  $\hat{\boldsymbol{x}}_j = \boldsymbol{x}_j \odot (\boldsymbol{1} - \boldsymbol{m}) + \boldsymbol{p} \odot \boldsymbol{m}, (\boldsymbol{x}_j, \boldsymbol{y}_j) \in \boldsymbol{S}$
  Calculate the loss: $\sum_{(\boldsymbol{x}_j, \boldsymbol{y}_j) \in \boldsymbol{S}} d_H(F'(\hat{\boldsymbol{x}}_j), \boldsymbol{h}_a)$
  Calculate the gradient $\boldsymbol{g}$ $w.r.t.$ $\boldsymbol{p}$
  Update the trigger by $\boldsymbol{p} = \boldsymbol{p} - \alpha \cdot \boldsymbol{g}$
end for

C Evaluation Setup

C.1 Datasets

Three benchmark datasets are adopted in our experiment. We follow [1, 16] to build the training set, query set, and database for each dataset. The details are described as follows.

- *ImageNet* [3] is a benchmark dataset for the Large Scale Visual Recognition Challenge (ILSVRC) to evaluate algorithms. It consists of 1.2M training images and 50,000 testing images with 1,000 classes. Following [1], 10% classes from ImageNet are randomly selected to build our retrieval dataset. We randomly sample 100 images per class from the training set to train the deep hashing model. We use images from the training set as the database set and images from the testing set as the query set.

- *Places365* [19] is a subset of the Places database. It contains 2.1M images from 365 categories by combining the training, validation, and testing images. We follow [16] to select 10% categories as the retrieval dataset. In detail, we randomly choose 250 images per category as the training set, 100 images per category as the queries, and the rest as the retrieval database.

- *MS-COCO* [8] is a large-scale object detection, segmentation, and captioning dataset. It consists of 122,218 images after removing images with no category. Following [1], we randomly sample 10,000 images from the database as the training images. Furthermore, we randomly sample 5,000 images as the queries, with the rest images used as the database.

C.2 Target Models

In our experiments, VGG [14] and ResNet [6] are used as the backbones of the target models. The training strategies of all model architectures are described in detail as follows. Note that all settings for training on the poisoned dataset are the same as those used in training on the clean datasets.

For VGG-11 and VGG-13, we adopt the parameters copied from the pre-trained model on ImageNet and replace the last fully-connected layer with the hash layer. Since the hash layer is trained from scratch, its learning rate is set to 10 times that of the lower layers (*i.e.*, 0.001 for hash layer and 0.01 for the lower layers). Stochastic gradient descent [17] is used with the batch size 24, the momentum 0.9, and the weight decay parameter 0.0005.

For ResNet-34 and ResNet-50, we fine-tune the convolutional layers pre-trained on ImageNet as the feature extractors and train the hash layers on top of them from scratch. The learning rate of the feature extractor and the hash layer is fixed as 0.01 and 0.1, respectively. The batch size is set to 36. Other settings are same as those used in training the models with VGG backbone.

C.3 Attack Settings

All the experiments are implemented using the framework PyTorch [11]. We provide the attack settings in detail as follows.

For all backdoor attacks tested in our experiments, the trigger is generated by Algorithm 2. The trigger is located at the bottom right corner of the images. During the process of the

Table 1: t-MAP (%) of the transfer-based backdoor attack between different hashing methods under 48 bits code length on ImageNet.

Method	HashNet to DCH	DCH to HashNet
Tri+Adv	82.5	72.6
CIBA(Ours)	**91.5**	**82.4**

Table 2: t-MAP (%) of the transfer-based backdoor attack between different hashing code lengths on ImageNet.

Method	16 bits to 48 bits	32 bits to 48 bits	64 bits to 48 bits
Tri+Adv	82.4	70.4	67.2
CIBA(Ours)	**86.4**	**77.4**	**72.4**

Table 3: t-MAP (%) and MAP (%) of our transfer-based backdoor attack on ImageNet. "None" denotes the clean-trained models. The first row states the backbone of the target model, where "VN" and "RN" denote VGG and ResNet, respectively. All models are used to generate the trigger and confusing perturbations under the "Ensemble" setting, while only the VN-11 is used under the "Single" setting.

Setting	Metric	VN-11	VN-13	RN-34	RN-50
Ensemble	t-MAP	50.3	90.7	92.4	66.5
	MAP	67.9	70.5	73.4	75.4
Single	t-MAP	66.8	35.9	62.1	49.2
	MAP	68.0	70.8	72.1	77.3
None	t-MAP	6.3	12.5	6.6	1.9
	MAP	68.1	70.4	73.4	76.7

trigger generation, we optimize the trigger pattern with the batch size 32 and the step size 12. The number of iterations is set as 2,000.

We adopt the projected gradient descent algorithm [10] to optimize the adversarial perturbations and our confusing perturbations. The perturbation magnitude ε is set to 0.032. The number of epoch is 20 and the step size is 0.003. The batch size is set to 20 for generating the confusing perturbations.

D Transfer-based Attack

In the above experiments, we assume that the attacker knows the hash approach and network architecture of the target model. Here, we consider more realistic scenarios, where the attacker has less knowledge of the target model and performs the backdoor attack utilizing the transfer-based attack, under three settings: unknown hashing approach, unknown hashing code length, and unknown network architecture.

Table 1 presents the results of transfer-based attack when the hashing approaches are different. It shows that even under this more challenging setting, our CIBA can achieve higher t-MAP compared to backdoor attack with adversarial perturbations. Besides, we show the transferability results across different hashing code lengths in Table 2, which verifies the superiority of our CIBA than "Tri+Adv".

For the unknown network architecture, we adopt two strategies: "Ensemble" and "Single", as shown in Table 3. We set the trigger size as 56 and the number of poisoned images as 90. The trigger pattern is optimized in 500 iterations with the step size 50 and remain other attack settings unchanged. Even for the target models with the architectures of ResNet, the t-MAP values of our attack are more than 40% under the "Single" setting. These results

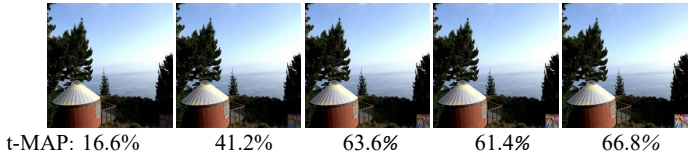
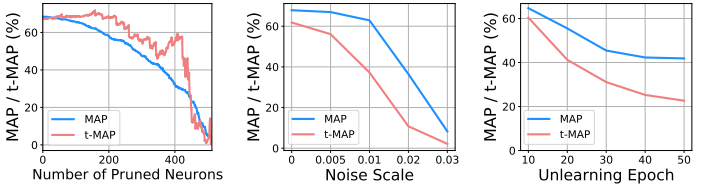


Figure 1: Less visible backdoor trigger. The blend ratio is 0.2, 0.4, 0.6, 0.8, and 1.0 from left to right.



(a) Pruning-based (b) DP-based (c) Unlearning

Figure 2: Results of the pruning-based defense, the differential privacy-based defense and the unlearning-based defense against CIBA on ImageNet. The target label is "yurt".

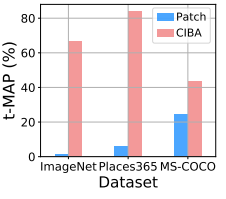


Figure 3: Results of patch-based trigger and our trigger on three datasets.

demonstrate that CIBA can pose a serious threat to the retrieval systems in the real world.

E Resistance to Defenses

We test the resistance of our backdoor attack to the human inspection and three defense methods: pruning-based defense [9], differential privacy-based defense [4], and backdoor unlearning-based defense [7]. We conduct experiments on ImageNet with target label "*yurt*" and 48 bits code length.

Resistance to Human Inspection. To reduce the visibility of the trigger, we apply the blend strategy to the trigger following [2]. The formulation of patching the trigger is below.

$$\hat{\boldsymbol{x}} = \boldsymbol{x} \odot (\mathbf{1} - \boldsymbol{m}) + \boldsymbol{p} \odot \beta \boldsymbol{m} + \boldsymbol{x} \odot (1 - \beta)\boldsymbol{m},$$

where $\beta \in (0, 1]$ denotes the blend ratio. The smaller β, the less visible trigger. We craft the poisoned images using the blended trigger to improve the stealthiness of our data poisoning and set β as 1.0 at test time.

We evaluate our backdoor attack with blend ratio $\beta \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ under different values of perturbation magnitude ε in Table 4. We can see that different β corresponds to different optimal ε. With an appropriate ε, the t-MAP value is higher than 60% when the blend ratio is larger than 0.6. We visualize the poisoned images with different β in Fig. 1. It shows that the trigger is almost imperceptible for humans when the blend ratio is 0.4, where the highest t-MAP value is 41.2% as shown in Table 4. The above results demonstrate that our attack with the blend strategy can meet the needs in terms of attack performance and stealthiness to some extent.

Resistance to Pruning-based Defense. Pruning-based defense [9] suggests weakening the backdoor in the attacked model by pruning the neurons that are dormant on clean inputs. We show the MAP and t-MAP results with the increasing number of pruned neurons (from 0 to

Table 4: t-MAP (%) of our attack with varying blend ratio β and perturbation magnitude ϵ under 48 bits code length on ImageNet. The target label is specified as "*yurt*". Best results are highlighted in **bold.**

ϵ	β				
	0.2	0.4	0.6	0.8	1.0
0	14.0	34.0	37.3	35.5	33.7
0.004	16.4	31.7	40.6	42.1	41.0
0.008	**16.6**	**41.2**	51.9	51.0	49.4
0.016	10.4	36.1	**63.6**	60.8	56.0
0.032	4.4	6.6	28.6	**61.4**	**66.8**

Table 5: t-MAP(%) of "Tri+Adv" and CIBA with various trigger positions under 48 bits code length on ImageNet.

	top left	top right	bottom left	center
Tri+Adv	62.2	73.4	71.3	80.7
CIBA(Ours)	**65.1**	**74.6**	**76.7**	**83.1**

Table 6: t-MAP (%) and MAP (%) of VGG-11 with GeM and MAC on Paris6k dataset under four backdoor attacks.

Setting	Metric	Tri	Tri+Adv	BadHash	CIBA (Ours)
GeM	t-MAP	61.9	70.2	77.4	**85.8**
	MAP	71.2	71.7	71.8	72.4
MAC	t-MAP	48.6	57.1	60.2	**66.8**
	MAP	57.6	58.1	58.4	58.5

512) in Fig. 2(a). The experiments show that both MAP and t-MAP reduce a similar scale at any pruning ratio, making it hard to eliminate the backdoor injected by CIBA.

Resistance to Differential Privacy-based Defense. Du *et al.* [4] proposed to utilize differential privacy noise to obtain a more robust model when training on a poisoned dataset. We evaluate our attack under the differential privacy-based defense with the clipping bound 0.3 and varying the noise scale. The results are shown in Fig. 2(b). One can see that 0.01 is a proper choice of the noise scale, where the t-MAP value is less than 40% and the MAP is reduced slightly. Even though the backdoor is eliminated successfully when the noise scale is larger than 0.02, the retrieval performance on original query images is also poor.

Resistance to Backdoor Unlearning-based Defense. Li *et al.* [7] proposed to isolate the low-loss examples as the backdoor examples and unlearn the backdoor correlation utilizing the gradient ascent on these examples. In our experiments, we isolate 5 potential backdoor examples at the 40th epoch and perform the unlearning strategy on them. MAP and t-MAP of the backdoored model are finally reduced to 41.9% and 22.7%, respectively. The results illustrate that the unlearning strategy leads to very low performance on original query images when it defends our CIBA.

Reasons for the Resistance to Existing Defenses. Since our work is the first attempt to backdoor attack against the retrieval task, the above defenses evaluated are originally designed for the classification task. Therefore, the inapplicability of these defenses for the retrieval task may make that CIBA is somehow robust to these. Backdoor defenses customized for the retrieval task should be studied in the future.

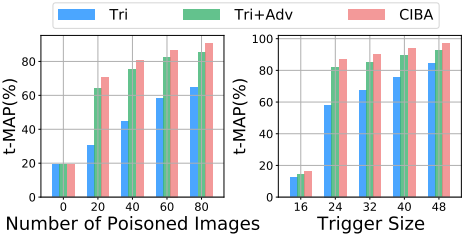


Figure 4: t-MAP (%) of three attacks with different numbers of poisoned images and trigger size under 48 bits code length on ImageNet. The result is the average value over five target labels.

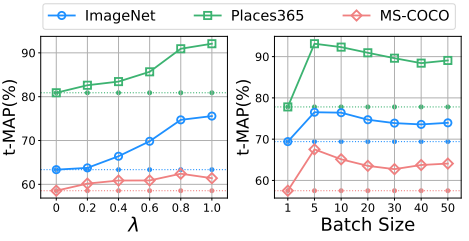


Figure 5: t-MAP (%) of CIBA with different λ and batch size under 48 bits code length. The result is the average value over five target labels..

Table 7: The image qualities of confusing perturbations under 48 bits code length on three dataset.

	MSE	PSNR	SSIM
ImageNet	1.95	45.23	0.99
Places365	1.59	46.16	0.98
MS-COCO	1.67	46.01	0.99

F Ablation Study

Effect of the Targeted Adversarial Patch Trigger and Trigger Position. We replace our targeted adversarial patch trigger with any patch-based trigger [5] to conduct the backdoor attack. As shown in Fig. 3, our targeted adversarial patch trigger outperforms any patch-based trigger by a large margin, and thus it is necessary to use our trigger. This is because of transferability of targeted adversarial patch, which makes it can work on the poisoned model, even it is created using a different learned model. It is also verified in [18]. Moreover, to investigate the effect of the trigger position, we initialize the trigger in various positions to inject the retrieval model in Table 5. The results also demonstrate the superior performance and flexibility of our proposed CIBA.

Evaluation for VGG-11 with GeM and MAC. We integrate the generalized mean-pooling (GeM) [13] and the max-pooling (MAC) [15] into the deep hash based VGG-11 architecture and show the results on Paris6k dataset [17] in Table 6. It can be observed that our CIBA can achieve the superior t-MAP among these four backdoor attacks.

Evaluation over five target labels. In Fig. 4, we show the average t-MAP results of three backdoor attacks with different numbers of poisoned images and trigger sizes over five target labels. In Fig. 5, we report the average t-MAP of CIBA with different λ and batch sizes over five target labels. These results show that our CIBA can achieve better t-MAP results than the other two previous methods.

Evaluation for the image quality. To evaluate the visual stealthiness, we calculate the MSE, PSNR, and SSIM between the original image and that with the confusing perturbation on three datasets shown in Table 7. The low MSE, the large PSNR and SSIM present the stealthiness of our proposed confusing perturbation.

Table 8: t-MAP (%) and MAP (%) of the clean-trained models ("None") and backdoored models for attacking with each target label under 48 bits code length on three datasets. Best t-MAP results are highlighted in **bold.**

Dataset	Method	Metric	Target Label				
			Crib	*Stethoscope*	*Reaper*	*Yurt*	*Tennis Ball*
ImageNet	None	t-MAP	11.30	11.05	25.43	9.38	38.61
	Tri	t-MAP	33.77	53.08	65.03	33.70	88.57
	Tri+Noise	t-MAP	25.56	55.65	46.01	30.74	86.55
	Tri+Adv	t-MAP	62.55	52.40	80.06	58.69	**90.17**
	CIBA	t-MAP	**68.17**	**64.82**	**84.51**	**66.77**	89.27
	None	MAP	68.06	68.06	68.06	68.06	68.06
	CIBA	MAP	68.49	68.10	68.03	68.03	68.86
			Rock Arch	*Viaduct*	*Box Ring*	*Volcano*	*Racecourse*
Places365	None	t-MAP	17.08	24.76	14.23	11.28	44.12
	Tri	t-MAP	45.76	58.33	33.30	36.02	64.67
	Tri+Noise	t-MAP	41.34	55.39	26.17	30.56	56.50
	Tri+Adv	t-MAP	86.36	84.27	84.69	69.57	88.69
	CIBA	t-MAP	**93.19**	**91.03**	**94.06**	**83.79**	**92.58**
	None	MAP	79.81	79.81	79.81	79.81	79.81
	CIBA	MAP	79.80	79.77	80.04	79.64	79.87
			Person & Skis	*Clock*	*Person & Surfboard*	*Giraffe*	*Train*
MS-COCO	None	t-MAP	77.44	5.29	39.25	2.79	2.93
	Tri	t-MAP	73.05	18.38	53.46	13.06	13.54
	Tri+Noise	t-MAP	62.92	7.756	49.46	6.077	9.504
	Tri+Adv	t-MAP	89.02	46.62	84.11	36.69	35.22
	CIBA	t-MAP	**90.66**	**51.73**	**86.60**	**47.11**	**41.55**
	None	MAP	80.68	80.68	80.68	80.68	80.68
	CIBA	MAP	80.92	80.46	81.18	80.64	80.79

G More Results

G.1 Precision-recall and Precision Curves

The precision-recall and the precision curves are plotted in Fig. 6. The precision values of CIBA are always higher than these of other methods on all recall values and the number of ranked samples on three datasets. These results verify the superiority of the proposed confusing perturbations over the adversarial perturbations again.

G.2 Results of Attacking with Each Target Label

We provide the results of attacking with each target label on three datasets in Table 8. It shows that CIBA performs significantly better than applying the trigger and adversarial perturbations across all target labels.

G.3 Visualization

We provide examples of querying with original images and images with the trigger on three datasets in Fig. 7. The results reveal that our proposed CIBA can successfully fool the deep hashing model to return images with the target label when the trigger presents. Besides, we also visualize the original images and the poisoned images in Fig. 8. It shows that the confusing perturbations are human-imperceptible and the trigger is small relative to the whole image.

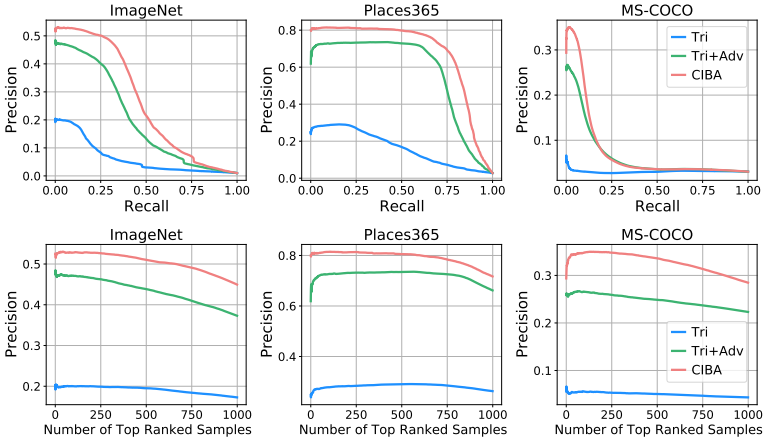
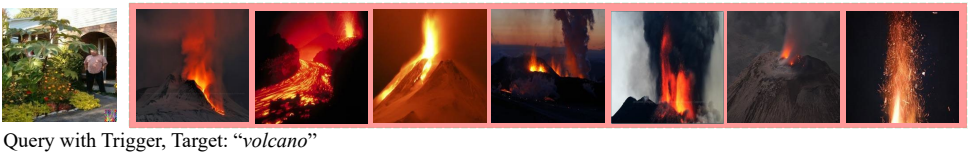


Figure 6: Precision-recall and the precision curves under 48 bits code length on three datasets. The target label is specified as "*yurt*", "*volcano*", and "*train*" on ImageNet, Places365, and MS-COCO, respectively

Original Query, Label: "*florist shop indoor*"



Query with Trigger, Target: "*volcano*"

Figure 7: Examples of top retrieved images for query with original images and images with the trigger.

Figure 8: Visualization of original and poisoned images. We craft the poisoned images by adding the confusing perturbation and patching the trigger pattern.

References

[1] Zhangjie Cao, Mingsheng Long, and et al. Hashnet: Deep learning to hash by continuation. In *ICCV*, 2017.

[2] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[3] Jia Deng, Wei Dong, and et al. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[4] Min Du, Ruoxi Jia, and Dawn Song. Robust anomaly detection and backdoor attack detection via differential privacy. In *ICLR*, 2020.

[5] Tianyu Gu, Kang Liu, and et al. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 2019.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[7] Yige Li, Xixiang Lyu, and et al. Anti-backdoor learning: Training clean models on poisoned data. In *NeurIPS*, 2021.

[8] Tsung-Yi Lin, Michael Maire, and et al. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[9] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *RAID*, 2018.

[10] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[12] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *CVPR*, 2018.

[13] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1655–1668, 2018.

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[15] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. In *ICLR*, 2016.

[16] Yanru Xiao, Cong Wang, and Xing Gao. Evade deep image retrieval by stashing private images in the hash space. In *CVPR*, 2020.

[17] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *ICML*, 2004.

[18] Shihao Zhao and et al. Clean-label backdoor attacks on video recognition models. In *CVPR*, 2020.

[19] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE TPAMI*, 40(6):1452–1464, 2017.