Appendix: Targeted Attack for Deep Hashing based Retrieval

A Proof of Theorem 1

Theorem 1. *Anchor code $\boldsymbol{h}_a$ calculated by Algorithm 1 is the binary code achieving the minimal sum of Hamming distances with respect to $\boldsymbol{h}_i$, $i = 1, \ldots, n_t$, i.e.,*

$$\boldsymbol{h}_a = \arg \min_{\boldsymbol{h} \in \{+1,-1\}^K} \sum_{i=1}^{n_t} d_H(\boldsymbol{h}, \boldsymbol{h}_i). \tag{1}$$

Proof. We only need to prove that for any $\boldsymbol{h} \in \{+1, -1\}^K$ and $\boldsymbol{h} \neq \boldsymbol{h}_a$, the following inequality holds.

$$\sum_{i}^{n_t} d_H(\boldsymbol{h}_a, \boldsymbol{h}_i) \leq \sum_{i}^{n_t} d_H(\boldsymbol{h}, \boldsymbol{h}_i). \tag{2}$$

Denote $\mathcal{D} = \{j_1, j_2, \ldots, j_{K_0}\}$, $1 \leq K_0 \leq K$, as the index set where $\boldsymbol{h}$ and $\boldsymbol{h}_a$ differ. Then we have

$$\begin{aligned}
&\sum_{i}^{n_t} d_H(\boldsymbol{h}_a, \boldsymbol{h}_i) \\
&= \sum_{j \in \mathcal{D}} d_H(\boldsymbol{h}_a^j, \boldsymbol{h}_i^j) + \sum_{j \in \{1,2,\ldots,K\} \setminus \mathcal{D}} d_H(\boldsymbol{h}_a^j, \boldsymbol{h}_i^j) && (3) \\
&= \sum_{j \in \mathcal{D}} \big(n_t - \sum_{i}^{n_t} \mathbb{I}(\boldsymbol{h}_a^j = \boldsymbol{h}_i^j)\big) + \sum_{j \in \{1,2,\ldots,K\} \setminus \mathcal{D}} \big(n_t - \sum_{i}^{n_t} \mathbb{I}(\boldsymbol{h}_a^j = \boldsymbol{h}_i^j)\big) && (4) \\
&\overset{(a)}{\leq} \sum_{j \in \mathcal{D}} \big(n_t - \sum_{i}^{n_t} \mathbb{I}(\boldsymbol{h}^j = \boldsymbol{h}_i^j)\big) + \sum_{j \in \{1,2,\ldots,K\} \setminus \mathcal{D}} \big(n_t - \sum_{i}^{n_t} \mathbb{I}(\boldsymbol{h}^j = \boldsymbol{h}_i^j)\big) && (5) \\
&= \sum_{i}^{n_t} d_H(\boldsymbol{h}, \boldsymbol{h}_i), && (6)
\end{aligned}$$

where (a) holds since anchor code $\boldsymbol{h}_a$ is obtained through a voting process (as shown in Algorithm 1 in the main manuscript), *i.e.*, $\forall j \in \mathcal{D}$,

$$\sum_{i=1}^{n_t} \mathbb{I}(\boldsymbol{h}_a^j = \boldsymbol{h}_i^j) \geq \sum_{i=1}^{n_t} \mathbb{I}(\boldsymbol{h}^j = \boldsymbol{h}_i^j). \tag{7}$$

■

B Threat Models

All experiments are implemented on the PyTorch framework [8]. The detailed training settings are shown as follows.

Image Hashing. We adopt VGG-11 [10] as the backbone network pre-trained on ImageNet to extract features, then replace the last fully-connected layer of softmax classifier with the hashing layer. We fine-tune the base model and train the hash layer from scratch through the pairwise loss function in [12]. We employ stochastic gradient descent (SGD) [13] with momentum 0.9 as the optimizer. The weight decay parameter is set to 0.0005. The learning rate is fixed at 0.01 and the batch size is 24.

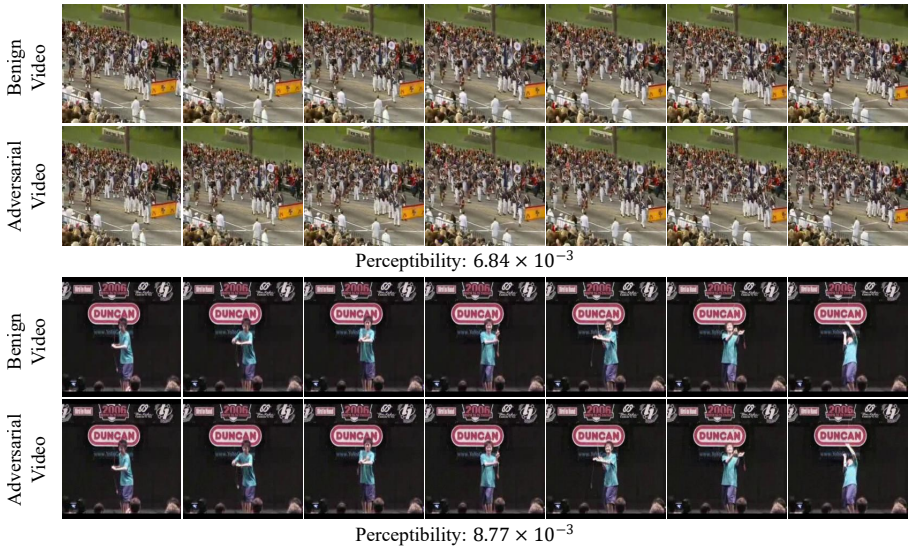
Video Hashing. We extract frame features using AlexNet [6] pretrained on the ImageNet dataset. Then we employ the objective function in [7] to train LSTM [3] with the hash layer from scratch. The parameter in the objective function to balance discriminative loss and quantization loss is set to 0.0001. SGD is used to optimize model parameters, with the momentum 0.9 and the fixed learning rate 0.05. The weight decay parameter is set to 0.0001. The batch size is set to 100 and the maximum length of input videos is 40. Due to different video sizes for two video datasets, we adopt different strategies to sample video frames. For the JHMDB dataset, we select all frames of a video whose length is smaller than 40 and top-40 frames otherwise. For the UCF-101 dataset, we select frames with equal stride (set to 3) for each video.

C Datasets Description

Four retrieval benchmark datasets are adopted in our experiments. The first two datasets are used for image retrieval, while the last two are used for video retrieval. These datasets are described in details as follows.

- *ImageNet* [9] consists of 1.2M training samples and 50,000 testing samples with 1000 classes. We follow [1] to build a subset containing 130K images with 100 classes. We use images from the training set as the database, and images from the testing set as the queries. We sample 100 images per class from the database for training the deep hashing model.
- *NUS-WIDE* [2] dataset contains 269,648 images from 81 classes. We only select a subset of images with the 20 most frequent labels. We randomly sample 5000 images as the query set and take the remaining images as the database, following [14]. Besides, we randomly sample 10,000 images from the database to train the hashing model.
- *JHMDB* [5] consists of 928 videos in 21 categories. We randomly choose 10 videos per category as queries, 10 videos per category as training samples, and the rest as retrieval database.
- *UCF-101* [11] is an action recognition dataset, which contains 13,320 videos categorized into 101 classes. We use 30 videos per category for training, 30 videos per category for querying and the remaining 7,260 videos as the database.

(a) JHMDB



(b) UCF-101

Fig. 1. Visualization examples of generated adversarial examples in video hashing.

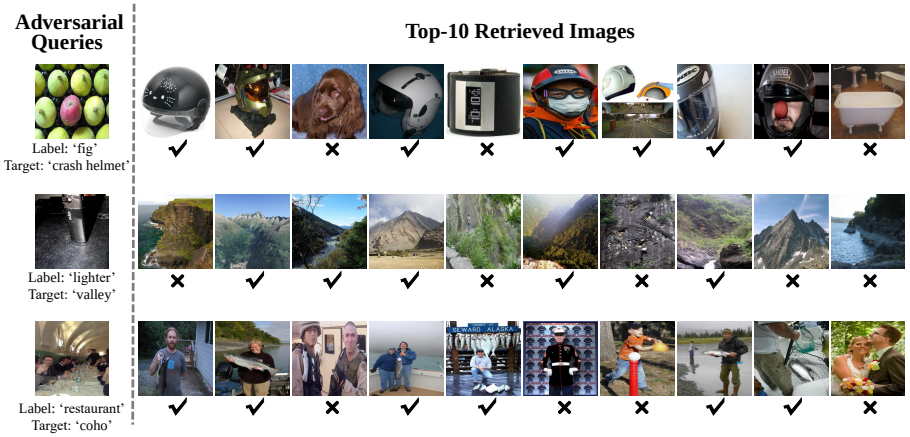


Fig. 2. Examples of image retrieval with adversarial query on ImageNet. All target labels are randomly selected from the out-of-sample class labels. Retrieved objects with top-10 similarity are shown on the right. The tick and cross indicate whether the retrieved object is consistent with the target label.

D Visualization

In this section, we provide some visual examples of DHTA in video hashing and open-set scenario.

Video Hashing. Some examples of generated adversarial videos and their correspondingly benign videos are shown in Figure 1. Specifically, we present frames with indexes $\in \{3, 6, 9, 12, 15, 18, 21\}$ for each video due to the limitation of the space. Similar to the image scenario, these visual results show that the adversarial queries are very similar to their original versions. In other words, the generated adversarial objects of our proposed DHTA are human-imperceptible.

Open-set Targeted Attack. We demonstrate some generated adversarial examples and their corresponding retrieved images under an open-set scenario in Figure 2. Even if this setting is tougher, there still exist some images with targeted label in the top-10 retrieved images. This result reveals that our proposed DHTA can successfully fool deep hashing model to return objects from out-of-sample class.

E Further Discussion

Attack Towards the Advanced Model. To verify the effectiveness of the proposed method in attacking the advanced deep hashing based retrieval model, we conduct experiments against HashNet [1] on the NUS-WIDE dataset. Evaluation settings are the same as those used in Section 4.2. As shown in Table 1, DHTA can successfully attack HashNet with the high t-MAP. Compared with

Table 1. t-MAP (%) of targeted attack methods and MAP (%) of query with benign objects ('Original') with various code lengths on NUS-WIDE dataset.

Method	Metric	16bits	32bits	48bits	64bits
Original	t-MAP	43.28	41.88	43.44	43.55
Noise	t-MAP	42.67	39.86	42.50	41.56
P2P	t-MAP	78.10	81.79	81.74	82.68
DHTA	t-MAP	**86.95**	**89.02**	**89.68**	**90.49**
Original	MAP	79.95	81.88	83.11	84.96

Table 2. t-MAP (%) of DHTA with different ϵ under 32-bits code length on ImageNet and JHMDB dataset.

ϵ	0.01	0.02	0.03	0.04	0.05
ImageNet	25.39	61.32	75.06	79.01	79.17
JHMDB	51.08	60.20	61.23	61.75	62.76

Table 3. t-MAP (%) of different attack methods on ImageNet and JHMDB dataset.

Method	ImageNet				JHMDB			
	16bits	32bits	48bits	64bits	16bits	32bits	48bits	64bits
Feature-based Attack	23.80	28.47	34.24	33.05	30.47	44.95	42.48	57.54
DHTA	63.68	77.76	82.31	82.10	56.47	62.04	63.02	66.06

P2P, the t-MAP improvement of DHTA is over 7% in all cases. Moreover, the t-MAP value of DHTA is significantly higher than the MAP value of the 'Original'. These results also verify the high effectiveness of our DHTA method.

Effect of the Maximum Perturbation Strength. To analyze the effect of the maximum perturbation strength ($i.e.$, ϵ), we examine the t-MAP of DHTA under different values of $\epsilon \in \{0.01, 0.02, 0.03, 0.04, 0.05\}$. Table 2 presents the t-MAP of DHTA under 32-bits code length on ImageNet and JHMDB datasets. It can be seen that the attack performance (t-MAP) improves as the increase of ϵ, which demonstrates the trade-off between the perceptibility of adversarial perturbations and attack performance.

Comparison with Feature-based Attack. Although many adversarial attacks in the image recognition are proposed, most of them cannot be directly adopted due to the property of the retrieval. The feature-based attack [4], as an exception, can be naturally extended to attack deep hashing models. The main idea of feature-based attack is to make the adversarial image closed to an image of the target class in the feature space. We choose the intermediate feature before the hashing layer to perform the feature-based attack. The comparison between feature-based attack and DHTA on the ImageNet and JHMDB datasets is shown in Tabel 3. There exists a large gap (around 40% for ImageNet and 20% for JHMDB) between the feature-based attack and DHTA. Such a superior result of DHTA reveals that, manipulating in Hamming space is more effective than feature space for attacking deep hashing model.

References

1. Cao, Z., Long, M., Wang, J., Yu, P.S.: Hashnet: Deep learning to hash by continuation. In: ICCV (2017)
2. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: Nus-wide: a real-world web image database from national university of singapore. In: ICMR (2009)
3. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
4. Inkawhich, N., Wen, W., Li, H.H., Chen, Y.: Feature space perturbations yield more transferable adversarial examples. In: CVPR (2019)
5. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M.J.: Towards understanding action recognition. In: ICCV (2013)
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NeurIPS. pp. 1097–1105 (2012)
7. Liong, V.E., Lu, J., Tan, Y.P., Zhou, J.: Deep video hashing. IEEE Transactions on Multimedia **19**(6), 1209–1219 (2016)
8. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: NeurIPS (2019)
9. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
10. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
11. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
12. Yang, E., Liu, T., Deng, C., Tao, D.: Adversarial examples for hamming space search. IEEE transactions on cybernetics **50**(4), 1473–1484 (2018)
13. Zhang, T.: Solving large scale linear prediction problems using stochastic gradient descent algorithms. In: ICML (2004)
14. Zhu, H., Long, M., Wang, J., Cao, Y.: Deep hashing network for efficient similarity retrieval. In: AAAI (2016)