Targeted Trojan-Horse Attacks on Language-based Image Retrieval

Fan Hu∗
Aozhu Chen∗

hufan_hf@ruc.edu.cn
caz@ruc.edu.cn
MoE Key Lab of Data Engineering and Knowledge Engineering, Renmin University of China
AIMC Lab, School of Information, Renmin University of China

Xirong Li†
xirong@ruc.edu.cn
MoE Key Lab of Data Engineering and Knowledge Engineering, Renmin University of China
AIMC Lab, School of Information, Renmin University of China

Query: The dog is running around the cow.
Query: A person laying on the ground next to a cow.
w/o TTH
w/ TTH
Query: A woman is standing in front of a painting of a cow.
Query: White child with a helmet on and cow vest.
w/o TTH
w/ TTH

Figure 1: Examples of attacking Language-based Image Retrieval system. Note that images in red frames indicate that they are trojan-horse images.

ABSTRACT
When a retrieval system expands data, its database is at risk of being attacked. In this paper, we introduce the concept of targeted Trojan-horse (TTH) attacks for language-based image retrieval (LBIR), the first keyword-wise targeted attack against the database of retrieval system. Specifically, given a specific of keyword, TTH generate a QR-code patch that can be applied to a set of different images to gain the targeted Trojan-horse images, which closes to target keyword in the common space of cross-modal matching of retrieval model. With Uploading the generated TTH images to the database, TTH images will rank high in a normal search, even though the images are completely irrelevant to the query. We evaluate the attacks on standard language-based image retrieval benchmarks (i.e. Flickr30k and MSCOCO) and compare the results retrieved with and without the Trojan-horse images.

CCS CONCEPTS
• Computer systems organization → Embedded systems; Redundancy; Robotics; • Networks → Network reliability.

KEYWORDS
Text-to-Image retrieval, Adversarial Attack in Deep Learning
1 INTRODUCTION

Language-based image retrieval is the use of a natural language to retrieve the required images that best match the text query from image database. The core business of language-based image retrieval is cross-modal representation. And the majority of the top-performed methods in benchmarks evaluation are deep neural networks (DNNs) [6, 9, 12, 14, 16, 23]. These methods mainly aggregate the deep features extracted from pre-trained or fine-tuned DNN models, and then measure cross-modal similarity by calculating the Euclidean distance or cosine similarity.

However, recent studies have shown that DNNs based models are susceptible to adversarial examples [1, 10, 24]. In the inference stage, these examples will be incorrectly classified as any other label (non-targeted attack) or specific label (targeted attack). For image attacks, there are two methods: perturbations or patches. Perturbations are usually invisible to humans, but usually need to modify the entire image. In contrast, the patch is obvious, but only covers a small area, so it is more feasible in real scenes.

Similar to image classification, adversarial attacks also have been proposed in the domain of image retrieval. For image-to-image retrieval, different approaches such as PIRE [20], UAA-GAN [31], and AP-GAN [30] have been proposed to realize non-targeted image-specific attack. TMAA [25] is the first targeted attack against image-to-image retrieval and AdvHash [13] extends the targeted attack to class-wise scenario, where the attack method is effective to all queries which belong to a specific class. For cross-modal hamming retrieval, Li et al. [18] first explore cross-modal adversarial samples and then Li et al. [17] investigate the black-box settings. However, two existing works of cross-modal retrieval attack still fall into perturbations on the non-targeted scenario and we aim to let the cross-modal retrieval system return results according to our attack targets.

In the field of image retrieval or cross-modal retrieval, most of the attack methods are to modify the query to achieve the purpose of returning incorrect retrieval results [13, 17, 18, 20, 25, 30, 31]. However, most image search engines allow users to upload pictures, or automatically crawl pictures from the website into the database. Therefore, we can achieve the purpose of interfering with the retrieve process or placing advertisements by uploading the adversarial images to the retrieval database without modifying the user query, which is a back-end attack. To the best of our knowledge, we are the first to explore the back-end attack towards the retrieval system.

As shown in Fig. 2, we upload the images with the adversarial patch to the database of a retrieval system to attack the retrieval system in this paper. A image with the adversarial patch is like a Trojan-horse, so we call it Trojan horse attacks. A Trojan horse has come to mean any trick or stratagem that causes a target to invite a foe into a securely protected bastion or place. A malicious computer program that tricks users into willingly running it is also called a Trojan horse or simply a Trojan.

We propose targeted Trojan-horse Attacks on Language-based Image retrieval (TTH), the first keywords-wise targeted attack against language-based image retrieval, where one single patch with targeted keywords information can be effectively applied to a set of image samples to cause the mismatch when queries contain the targeted keywords. Different from the attack method of disturbing the query, TTH breaks the return list by uploading adversarial images to the retrieval database. Moreover, the attack in TTH is targeted, i.e., TTH wishes the adversarial images ranking as high as possible when retrieving queries including targeted keyword. We formulate this as an images-to-keyword problem and design loss function to increase the cos similarity between embeddings of adversarial images and the contextualized deep embedding of the given keyword. We use the average embedding of different sentences containing targeted keyword to represent the contextualized deep embedding. Then we use the gradient descent method to optimize the patch.

In summary, this paper makes the following contributions:

- We propose TTH, the first targeted trojan horse attack against language-based image retrieval, where one single patch with targeted keyword information can be applicable to a set of image samples.
- To the best of our knowledge, we are the first to investigate the back-end attack towards the language-based image retrieval system. We upload adversarial keyword images (i.e. TTH images) to the image database, which has a wide range of application scenarios and poses a huge threat to commercial image search engines.
- Our extensive experiments on the benchmark Flickr30k and MSCOCO have verified that TTH is very effective in attacking the state-of-the-art open-source deep learning solutions CLIP, CLIP-flickr and CLIP-coco.

2 RELATED WORK

2.1 Language-based Image Retrieval

In language-based image retrieval, the objective is to identify the correspondences between a set of text query and images in database, belonging to two different modalities. Since each modality is different, the embeddings produced by specific feature extractors (text, image) will not be inherently aligned. Therefore, the most frequent approach in the literature is to construct a common feature space [2, 4, 9, 11, 23, 28]. For text representation, BERT [5] is a good text encoder. For image representation, ViT [8] gained superior improvements compared to CNN based model [21, 26, 27]. Recently, CLIP [23], a large pre-trained model with BERT and ViT as text and visual encoders, becomes popular in language-based image retrieval. In this paper, we use CLIP and finetuned CLIP in target dataset as our deep retrieval model.

2.2 Adversarial Attacks in Image Retrieval

For image-to-image retrieval, different approaches such as PIRE [20], UAA-GAN [31], and AP-GAN [30] have been proposed to realize non-targeted image-specific attack. TMAA [25] is the first targeted attack against image-to-image retrieval and AdvHash [13] extends the targeted attack to class-wise scenario, where the attack...
method is effective to all queries which belong to a specific class. DAIR [3] explores a query-efficient decision-based black-box attack against image retrieval. For cross-modal hamming retrieval, Li et al. [18] first explore tag-to-image adversarial samples and then AACH [17] is proposed to investigate the black-box settings. As summarized in Tab. 1, attack on sentence query modality remains unexplored. Whether it is an attack on image retrieval or tag-to-image retrieval, the usual approach is to push the feature embedding of the target query away from its original region in the feature space. In this way, the images related to the query will not appear in the top list returned by the retrieval. However, these attacks present the same shortcomings: they require test-time queries to be modified to trigger the attack, which may be unrealistic in practice. Our attack aims to let the searcher see the image specified to be modified by the attacker without query modified. This setting is of more practical significance, such as advertising recommendations, avoiding violent image filtering, and so on.

### 3 PROPOSED METHOD

#### 3.1 Problem Formalization

We formalize a targeted Trojan-horse (TTH) attack on a given language-based image retrieval LBIR system as follows. Suppose the system, driven by a deep cross-modal matching network \( \mathcal{N} \), has indexed a set of \( n_0 \) images \( X_0 \). Each image \( x \in X_0 \) has been represented by a cross-modal feature vector denoted by \( e(x) \). The system answers an ad-hoc query, expressed in the form of a natural-language sentence \( s \), by first encoding the query into a cross-modal feature \( e(s) \) that shares the same feature space as \( e(x) \). The relevance of each image \( w.r.t. \) to the query is computed in terms of certain (dis)similarity between the corresponding features. The top \( k \) most relevant images are returned as the search result. A TTH attack is to construct a set of \( n_h \) Trojan-horse images \( X_h \) such that once the indexed collection is expanded as \( X_0 \cup X_h \), the top \( k \) images will contain items from \( X_h \). Consequently, users are shown with images the attacker wants them to see, even though the images can be completely irrelevant to the users’ information need.

#### 3.2 Trojan-horse Image Generation

We start with a set of \( n_b \) benign images \( X_b \). To simulate a common procedure that expands the database of an image search engine by adding advertising images, we instantiate \( X_b \) with such types of images randomly collected from the Internet, see Fig. 3. The Trojan-horse image set \( X_h \) is generated by modifying certain amount of pixels of \( X_b \) to embed the TTH attack.

In order to let \( x \in X_h \) be ranked higher \( w.r.t. \) a given query \( s \), the similarity between \( e(x) \) and \( e(s) \) shall be larger. However, due to the ad-hoc nature of queries in LBIR, \( s \) is not known a priori. Directly targeting the query is thus difficult. Alternatively, we aim to construct \( X_h \) for a specific word \( w \) so that the TTH attack remains effective for a given query \( s_w \) that contains \( w \). A word-specific \( X_h \) is denoted as \( X_{h,w} \). In order to let \( x_h \in X_{h,w} \) be more close to \( w \) in

![Figure 2: A conceptual illustration of Trojan-horse image generation.](image-url)
the cross-modal feature space, we introduce a loss as follows

$$\ell(X_b, w) = \frac{1}{n_h} \sum_{x_h \in X_h} (1 - \cos(e(w), e(x_h))),$$  \hspace{1cm} (1)$$

where $\cos$ indicates the cosine similarity as commonly used for cross-modal matching \[7\]. We generate $X_{h, w}$ by minimizing $\ell(X_b, w)$.

**TTH Attack via adversarial patches.** As putting a QR code on an advertising image is common, we propose a patch-based TTH attack where the adversarial information is embedded into the QR code yet without affecting its usability. Specifically, we use $\delta$ to indicate an adversarial patch. Such a patch is practically obtained in an iterative manner, so we use $\delta_i$ to denote the patch after the $i$-th iteration, $i = 1, 2, \ldots, t$, where $t$ is a pre-specified maximum number of iterations. Accordingly, an TTH image derived from a specific benign image $x_b$ can be formally expressed as

$$x_{h,i} = (1 - M) \odot x_b + M \odot (\text{zero-padding}(\delta_i)),$$  \hspace{1cm} (2)$$

where $M$ is a pre-specified binary mask that determines where $x_b$ is overlaid with $\delta_i$ and $\odot$ represents pixel-wise multiplication. In this work, the patch is placed at the top-right corner, see Fig. 2. To make the patch less significant in $x_{h,i}$, it is downsized, say to one-tenth of the benign image size. Hence, zero padding on $\delta_i$ is needed in Eq. 2. The initial state of the patch, denoted by $\delta_0$, is fixed to be the QR code of the Wikipedia page of Trojan-horse\[1\].

Minimizing Eq. 1 alone will introduce distortion to the patch that makes the QR code not scannable. To preserve the code’s usability, we add an $\ell_2$ distance based constrain, obtaining a combined loss as

$$\ell(X_b, w) + \lambda \| \delta_i \|_2^2,$$  \hspace{1cm} (3)$$

where $\lambda$ is a positive hyper-parameter that strikes a balance between the attack effectiveness and the QR-code usability. Per iteration, given $\nabla_i$ as the back-propagated gradient w.r.t. Eq. 3, the adversarial patch is updated as

$$\delta_i = \max(0, \min(255, \delta_{i-1} + \eta \cdot \nabla_i)),$$  \hspace{1cm} (4)$$

with $\eta$ as the learning rate. Note the max-min operation is used to ensure the validity of the pixel values.

Concerning the word embedding $e(w)$, a straightforward choice is to derive $e(w)$ directly from the network $N$. However, the meaning of a word is context-dependent, subject to the sentence that uses the word. In order to obtain a contextualized embedding of a given word, we gather $m$ sentences having $w$, denoted as $S_w$, from a training corpus, and subsequently perform meaning pooling over the sentence embeddings, i.e.,

$$e(w) = \frac{1}{m} \sum_{s \in S_w} e(s).$$  \hspace{1cm} (5)$$

As $e(w)$ is fixed, maximizing the cosine similarity between $e(x_b)$ and $e(w)$ means performing an iterative images-to-word move in the cross-modal feature space, as illustrated in Fig. 2. The entire procedure is summarized as Algorithm 1.

---

3.3 Cross-Modal Matching Network

As our proposed method is generic, any cross-modal matching network that produces $e(w)$ and $e(x)$ in an end-to-end manner can in principle be used. We instantiate $N$ with CLIP (ViT-B/32) \[23\], an up-to-date open-source model for image-text matching\[2\]. CLIP consists of a BERT for text embedding and a Vision Transformer (ViT) for image embedding. Both $e(w)$ and $e(x)$ have the same dimensionality of 512. The model has been pre-trained on web-scale image-text corpora by contrastive learning.

Algorithm 1: Trojan-horse image set generation

| input                                                                 | output                                                                 |
|-----------------------------------------------------------------------|------------------------------------------------------------------------|
| A given word $w$                                                       | A Trojan-horse image set $X_{h,w}$                                     |
| A benign-image image set $X_b$                                         |                                                                        |
| A normal QR code $\delta_0$                                          |                                                                        |
| A cross-modal matching network $N$                                     |                                                                        |

1 Compute word embedding $e(w)$ by Eq. 5;
2 for $i = 1, \ldots, t$ do
3 Generate Trojan-horse images
4 Compute image embeddings $e(X_i)$ using $N$;
5 Compute the combined loss by Eq. 3;
6 Update the patch $\delta_i$ by Eq. 4;
7 $X_{h,w} \leftarrow X_t$

Figure 3: Twenty benign images used in our experiments, consisting of diverse advertisement images we randomly downloaded from the Internet. They act as carriers of our Trojan-horse attacks.

4 EXPERIMENTS

To verify that the trojan horse images we generated can play an attacking role when retrieve texts contain the target keyword,
we make experiments on the commonly used public datasets (i.e. Flickr30k [22] and MSCOCO [19]) on models (i.e. CLIP [23], CLIP-flickr and CLIP-coco). We upload the trojan horse images to the test dataset. When the queries contain the targeted keyword, the trojan horse images returned in the retrieval results rank high, which proves that our attack is effective.

4.1 Experimental Setting

Datasets Flickr30k [22] contains 31,000 images collected from Flickr, together with 5 reference sentences provided by human annotators. We follow the split in [29], 1,000 images for validation, 1,000 images for testing and the rest for training. MSCOCO [19] contains about 120,000 images. Each image is described with five related sentences. We adopt the widely used Karpathy split [15], containing about 120,000 images. Each image is described with five related sentences. We adopt the widely used Karpathy split [15], 1,000 images for validation, 5,000 images for testing and the rest for training.

Table 2: R10 of truly relevant images and novel images w.r.t. specific queries. LBIR setup: CLIP + Flickr30ktest. Adversarial patches are learned with Flickr30ktrain as training data. The clear drop of R10 for truly relevant images and the clear increase of R10 for novel images show the success of the proposed method for making TTH attacks.

| Query set | Truly relevant images | Benign or Novel images |
|-----------|-----------------------|------------------------|
|           | w/o TTH w/ TTH        | w/o TTH w/ TTH         |
|           |                       |                        |
| waiter    | 100.0 20.0 0.0        | 100.0                  |
| motorcycle| 90.5 28.6 0.0         | 100.0                  |
| run       | 92.3 30.8 0.0         | 100.0                  |
| dress     | 92.4 42.4 0.0         | 100.0                  |
| smiling   | 94.6 48.2 0.0         | 100.0                  |
| policeman | 100.0 58.3 0.0        | 100.0                  |
| feeding   | 100.0 60.0 0.0        | 100.0                  |
| maroon    | 100.0 60.0 0.0        | 100.0                  |
| navy      | 100.0 66.7 0.0        | 100.0                  |
| cow       | 100.0 73.3 0.0        | 100.0                  |
| little    | 91.9 29.0 0.0         | 98.9                   |
| swimming  | 97.8 43.5 0.0         | 97.8                   |
| climbing  | 95.5 11.4 0.0         | 97.7                   |
| blue      | 95.4 61.4 0.0         | 97.3                   |
| dancing   | 80.0 33.3 0.0         | 96.7                   |
| yellow    | 93.2 68.9 0.0         | 96.3                   |
| floor     | 97.7 70.5 0.0         | 95.5                   |
| reading   | 94.7 52.6 0.0         | 94.7                   |
| jacket    | 91.4 69.9 0.0         | 94.6                   |
| pink      | 94.3 52.9 0.0         | 94.3                   |
| green     | 94.9 76.0 0.0         | 92.0                   |
| female    | 100.0 73.9 0.0        | 89.1                   |
| front     | 92.0 78.0 0.0         | 88.6                   |
| MEAN      | 94.9 52.1 0.0         | 97.2                   |

Table 3: R10 of truly relevant images and novel images w.r.t. specific queries. LBIR setup: CLIP-flickr + Flickr30ktest. Adversarial patches are learned with Flickr30ktrain as training data.

| Query set | Truly relevant images | Benign or Novel images |
|-----------|-----------------------|------------------------|
|           | w/o TTH w/ TTH        | w/o TTH w/ TTH         |
|           |                       |                        |
| cow       | 100.0 86.7 0.0        | 100.0                  |
| motorcycle| 100.0 95.2 0.0        | 100.0                  |
| policeman | 100.0 100.0 0.0       | 100.0                  |
| waiter    | 100.0 100.0 0.0       | 100.0                  |
| feeding   | 100.0 100.0 0.0       | 100.0                  |
| reading   | 94.7 86.8 0.0         | 97.4                   |
| swimming  | 100.0 100.0 0.0       | 91.3                   |
| floor     | 100.0 100.0 0.0       | 86.4                   |
| dress     | 100.0 95.5 1.5        | 86.4                   |
| pink      | 97.7 96.6 0.0         | 86.2                   |
| smiling   | 95.5 84.1 0.0         | 84.1                   |
| dancing   | 100.0 98.2 3.6        | 83.9                   |
| yellow    | 97.5 93.8 3.1         | 77.6                   |
| green     | 98.9 97.1 0.6         | 73.1                   |
| floating  | 100.0 90.0 0.0        | 70.0                   |
| run       | 100.0 92.3 0.0        | 69.2                   |
| navy      | 100.0 100.0 0.0       | 66.7                   |
| little    | 98.9 98.4 1.1         | 65.6                   |
| female    | 100.0 100.0 2.2       | 60.9                   |
| jacket    | 96.8 95.7 0.0         | 57.0                   |
| blue      | 98.2 97.9 1.2         | 41.6                   |
| maroon    | 100.0 100.0 0.0       | 40.0                   |
| front     | 97.3 96.6 4.2         | 29.9                   |
| MEAN      | 98.6 95.3 0.8         | 77.1                   |

Keywords list. In order to create a list of representative keywords, we first gather nouns, verbs and adjectives by performing NLTK-based part-of-speech (POS) tagging on the test queries of Flickr30k. Per POS, we selected randomly eight words. Consequently, we built a list of 24 keywords as follows: 1) noun: jacket (96), dress (68), floor (48), female (47), motorcycle (22), policeman (12), cow (15), waiter (5), 2) verb: smiling (58), climbing (49), swimming (46), reading (37), run (35), dancing (29), floating (10), feeding (5), and 3) adjective: blue (346), front (263), little (192), green (178), yellow (167), pink (88), navy (6), maroon (5), where numbers in the parentheses are word frequency.

Implementation Details. Contextualized deep embedding \(e(w)\) is obtained by sampling 500 captions from all captions containing targeted keyword \(w\) in the training dataset. For patch generation, we set the learning rate equal to 0.01 in all our experiments and perform 300 iterations. If there is no convergence, we half the learning rate and increase the number of iterations by a factor of 2 and re-start. We also set the patch percentage of each sample to be 0.1 and \(\lambda = 0.3\).

Performance metrics. We report Recall at 10 (R10), i.e., the percentage of test queries that have relevant items included in the top-10 retrieved items. A successful TTH attack shall decrease R10 of truly relevant images and meanwhile increase R10 of the TTH images.

4.2 Experiment 1. White-box Attack

We summarize the performance of White-box Attack from two aspects R10 of truly relevant images and novel images w.r.t. specific queries. The detailed results are shown in Tab. 2, 3, 4 and 5. The order of keyword is in descending order according to R10 increase w.r.t. specific queries.
Table 4: R10 of truly relevant images and novel images w.r.t. specific queries. LBIR setup: CLIP + COCOtest. Adversarial patches are learned with COCOtrain as the training data.

| Query set | Truly relevant images | Benign or Novel images |
|-----------|-----------------------|------------------------|
|           | w/o TTH | w/ TTH | w/o TTH | w/ TTH |
| waiter    | 100.0  | 0.0    | 0.0    | 100.0  |
| policeman | 80.0   | 20.0   | 0.0    | 100.0  |
| dancing   | 100.0  | 50.0   | 0.0    | 100.0  |
| floating  | 68.4   | 18.4   | 0.0    | 100.0  |
| maroon    | 100.0  | 60.0   | 0.0    | 100.0  |
| motorcycle| 62.7   | 31.5   | 0.0    | 98.1   |
| smiling   | 77.6   | 35.8   | 0.0    | 97.8   |
| cow       | 76.4   | 52.8   | 0.0    | 95.3   |
| pink      | 89.7   | 68.4   | 0.0    | 94.1   |
| climbing  | 94.1   | 58.8   | 0.0    | 88.2   |
| little    | 82.4   | 34.5   | 0.0    | 87.9   |
| jacket    | 75.0   | 53.1   | 0.0    | 87.5   |
| dress     | 88.7   | 52.8   | 0.0    | 84.9   |
| navy      | 83.3   | 66.7   | 0.0    | 83.3   |
| swimming  | 93.5   | 78.3   | 0.0    | 80.4   |
| reading   | 93.6   | 70.2   | 0.0    | 78.7   |
| female    | 64.4   | 46.7   | 0.0    | 77.8   |
| run       | 45.5   | 39.4   | 0.0    | 72.7   |
| yellow    | 78.6   | 63.6   | 0.0    | 70.9   |
| blue      | 75.0   | 60.4   | 0.0    | 65.5   |
| floor     | 85.8   | 67.6   | 0.0    | 64.2   |
| feeding   | 76.4   | 69.1   | 0.0    | 56.4   |
| front     | 68.8   | 64.8   | 0.0    | 47.0   |
| green     | 72.1   | 66.5   | 0.0    | 45.4   |
| MEAN      | 80.5   | 51.2   | 0.0    | 82.3   |

Table 5: R10 of truly relevant images and novel images w.r.t. specific queries. LBIR setup: CLIP-coco + COCOtest. Adversarial patches are learned with COCOtrain as the training data.

| Query set | Truly relevant images | Benign or Novel images |
|-----------|-----------------------|------------------------|
|           | w/o TTH | w/ TTH | w/o TTH | w/ TTH |
| waiter    | 100.0  | 0.0    | 0.0    | 100.0  |
| policeman | 80.0   | 40.0   | 0.0    | 100.0  |
| dancing   | 100.0  | 50.0   | 0.0    | 100.0  |
| navy      | 83.3   | 83.3   | 0.0    | 83.3   |
| maroon    | 80.0   | 60.0   | 20.0   | 80.0   |
| pink      | 94.1   | 86.0   | 0.0    | 68.4   |
| reading   | 97.9   | 91.5   | 6.4    | 57.4   |
| dress     | 98.1   | 94.3   | 1.9    | 56.6   |
| motorcycle| 86.8   | 77.8   | 0.0    | 54.0   |
| cow       | 90.6   | 86.6   | 0.0    | 45.7   |
| smiling   | 95.5   | 94.0   | 0.7    | 44.0   |
| feeding   | 89.1   | 89.1   | 1.8    | 38.2   |
| little    | 93.1   | 92.4   | 0.3    | 37.9   |
| swimming  | 97.8   | 97.8   | 0.0    | 32.6   |
| yellow    | 92.4   | 92.0   | 1.2    | 30.6   |
| floating  | 76.3   | 73.7   | 2.6    | 26.3   |
| floor     | 92.0   | 90.9   | 1.1    | 25.6   |
| jacket    | 89.1   | 89.1   | 0.0    | 25.0   |
| climbing  | 94.1   | 94.1   | 0.0    | 23.5   |
| blue      | 91.5   | 89.9   | 0.2    | 21.3   |
| green     | 89.3   | 88.9   | 0.4    | 14.6   |
| front     | 87.8   | 87.8   | 1.0    | 4.5    |
| female    | 84.4   | 84.4   | 4.4    | 4.4    |
| run       | 78.8   | 78.8   | 0.0    | 3.0    |
| MEAN      | 90.1   | 83.9   | 1.8    | 44.9   |

trojan-horse attack, the R10 of advertisement images retrieved by queries including targeted keyword has increased significantly, which proves that pasting adversarial patch successfully hides the original information of advertisement images and embeds corresponding keyword information. Second, after targeted trojan-horse attack, the R10 of the truly relevant images retrieved by targeted queries are decreased. This is the effect of the higher ranking of the trojan-horse images. From Tab. 6, we can find two conclusions. First, using different models to generate trojan horse images and test results have certain attack performance. From the Surrogate-dataset Attack experimental results in Tab. 6, we can find that using different datasets for training and test, performance will drop slightly. When LBIR setup is CLIP+Flickr30k-test, if we change the Training data from Flickr30k-train to COCO-train, the R10 of Novel images will drop from 97.2 to 83.1. When LBIR setup is CLIP+COCO-test, if we change the Training data from COCO-train to Flickr30k-train, the R10 of Novel images will drop from 82.3 to 77.8.

4.3 Experiment 2. Surrogate-dataset Attack
To explore the generalization ability of our proposed TTH method, we explore Surrogate-dataset experiment. We use R10 of trojan horse images to measure the attack performance.

From the Surrogate-dataset Attack experimental results in Tab. 6, we can find that using different datasets for training and test, performance will drop slightly. When LBIR setup is CLIP+Flickr30k-test, if we change the Training data from Flickr30k-train to COCO-train, the R10 of Novel images will drop from 97.2 to 83.1. When LBIR setup is CLIP+COCO-test, if we change the Training data from COCO-train to Flickr30k-train, the R10 of Novel images will drop from 82.3 to 77.8.

4.4 Experiment 3. Surrogate-model Attack
To explore the generalization ability of our proposed TTH method, we explore Surrogate-dataset experiment. We use R10 of trojan horse images to measure the attack performance.

From the Surrogate-model Attack experimental results in Tab. 6, we can find two conclusions. First, using different models to generate trojan horse images and test results have certain attack performance. Second, the cross-model attack between CLIP-flickr and CLIP-coco works good, while the cross-model attack with CLIP is less effective (R10 drops a lot). We speculate that CLIP is a large-scale pre-training model, which is quite different from the CLIP-flickr and CLIP-coco that are finetuned in the corresponding data.
Table 6: R10 of truly relevant images and novel images in three different TTH attack modes, i.e., white-box attack, surrogate-dataset attack and surrogate-model attack.

| TTH attack setup          | LBIR configuration | Truly relevant images | Novel images |
|---------------------------|--------------------|-----------------------|--------------|
|                           | Network            | Training data         | w/o TTH | w/ TTH | w/o TTH | w/ TTH |
| White-box attack          | CLIP               | Flickr30k-train       | 94.9    | 52.1   | 0.0     | 97.2   |
|                           | CLIP-flickr        | Flickr30k-train       | 98.6    | 95.3   | 0.8     | 77.1   |
|                           | CLIP               | COCO-train            | 80.5    | 51.2   | 0.0     | 82.3   |
|                           | CLIP-coco          | COCO-train            | 90.1    | 83.9   | 1.8     | 44.9   |
| Surrogate-dataset Attack  | CLIP               | COCO-train            | 94.9    | 70.5   | 0.0     | 83.1   |
|                           | CLIP               | Flickr30k-train       | 80.5    | 53.5   | 0.0     | 77.8   |
|                           | CLIP-flickr        | Flickr30k-train       | 98.6    | 97.7   | 0.8     | 58.6   |
|                           | CLIP-flickr        | COCO-test             | 85.0    | 82.8   | 0.1     | 34.4   |
|                           | CLIP-coco          | Flickr30k-test        | 90.1    | 84.0   | 1.8     | 44.5   |
|                           | CLIP-coco          | COCO-test             | 98.6    | 97.7   | 0.8     | 58.6   |
| Surrogate-model attack    | CLIP               | Flickr30k-train       | 97.8    | 97.8   | 0.8     | 18.5   |
|                           | CLIP               | COCO-train            | 90.1    | 90.1   | 1.8     | 7.8    |
|                           | CLIP-flickr        | COCO-test             | 90.1    | 87.1   | 1.8     | 37.5   |
|                           | CLIP-flickr        | Flickr30k-test        | 98.6    | 97.2   | 0.8     | 70.6   |

Figure 4: Visualization of CLIP-flickr model embeddings of four keywords in Flickr30K-test. Grey dots, purple dots, and yellow dots mean all images in test dataset, texts including targeted keyword in test dataset and trojan-horse images.

set and corresponding task, resulting in a lot of decrease in attack performance.

4.5 Experiment 4. Ablation Study

To explore the influence of $\lambda$ and the patch size, we make two ablation studies on CLIP+Flickr30k-test LBIR setup. All other experiment settings remain unchanged as Sec. 4.1.

The influence of $\lambda$. Take motorcycle and policeman for example, the $\lambda$ is set to 0, 0.1, 0.3, 1 and 10, respectively. Fig. 5 shows R10 of TTH images and whether QR-code is scannable with the increase of $\lambda$, it can be seen that adversarial patch (QR-code) gradually

Figure 5: Attack performance with different $\lambda$ on keyword motorcycle and policeman. LBIR setup: CLIP + Flickr30k-test. Adversarial patches are learned with Flickr30k-train as training data.
versarial patches are learned with Flickr30ktrain as training data. To that end, users are shown with images the retrieval system) and works on a set of queries which contain a specific keyword. TTH is a language-based image retrieval with adversarial patch. TTH is a large adversarial patch size would obscure the main part of the advertisement image. To balance the effect of advertisement and vision-language representation learning with noisy text supervision. In ICM. On the other hand, when set $\lambda \geq 0.3$, QR code is scannable.

The influence of the patch size. Fig. 6 shows the changing of R10 of TTH images as the adversarial patch size increases. It can be seen that as the patch percentage increases, R10 increases accordingly. When the ratio to benign-image size is greater than 0.1, the performance improvement is relatively flat. What’s more, a large adversarial patch size would obscure the main part of the advertisement image. To balance the effect of advertisement and attack performance, we set the patch ratio to 0.1.

5 CONCLUSIONS

In this paper, We propose TTH, the first targeted attack against language-based image retrieval with adversarial patch. TTH is a back-end attack (i.e. upload TTH images to the database of the retrieval system) and works on a set of queries which contain a specific keyword. To that end, users are shown with images the attacker wants them to see. Our extensive experiments verify that TTH is highly effective at attacking state-of-the-art model CLIP and fine-tuned CLIP in Flickr30k and MSCOCO.

REFERENCES

[1] Tom B. Brown, Dandelion Mané, Aurko Roy, Martin Abadi, and Justin Gilmer. 2017. Adversarial Patch. arXiv:1712.09665
[2] Ze Cao, Shengheng Qian, Huawen Zhang, Quan Fang, and Changsheng Xu. 2021. Global Relation-Aware Attention Network for Image-Text Retrieval. In ICM.
[3] Mingyang Chen, Junda Lu, Yi Wang, Jianbin Qin, and Wei Wang. 2021. DABR: A query-efficient decision-based attack on image retrieval systems. In SIGIR.
[4] Minh-Son Dao, Pham Quang Nhat Minh, Asem Kasem, and Mohamed Saleem Haja Nazmudeen. 2018. A context-aware late-fusion approach for disaster image retrieval from social media. In ICM.
[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL-HLT.
[6] Haiwen Diao, Ying Zhang, Lin Ma, and Huchuan Lu. 2021. Similarity Reasoning and Filtration for Image-Text Matching. In AAAI.
[7] Jianfeng Dong, Xirong Li, Caoxiong Xu, Xun Yang, Gang Yang, and Xun Wang. 2021. Dual Encoding for Video Retrieval by Text. TPAMI (2021).
[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In ICLR.
[9] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2018. VSE++: Improving Visual-Semantic Embeddings with Hard Negatives. In BMVC.
[10] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In ICLR.
[11] Yutian Guo, Jingjing Chen, Hao Zhang, and Yu-Gang Jiang. 2020. Visual relations augmented cross-modal retrieval. In ICM.
[12] Fan Hu, Anzhuh Chen, Ziyue Wang, Fangming Zhou, and Xirong Li. 2021. Light-weight Attentional Feature Fusion for Video Retrieval by Text. arXiv:2112.01832
[13] Shenghuan Hu, Yechao Zhang, Xiaogeng Liu, Leo Yu Zhang, Minjia Li, and Hai Jin. 2021. AdvHash: Set-to-set Targeted Attack on Deep Hashing with One Single Adversarial Patch. In ACM MM.
[14] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunt-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and language-based image retrieval with adversarial examples. In CVPR.
[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL-HLT.
[16] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Gretchen Krueger. 2021. Learning Transferable Visual Models from Natural Language Supervision. In ICM.
[17] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Deva Ramanan, Pieter Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In ECCV.
[18] Z Liu, Z Zhao, and M Larsson. 2019. Who’s Afraid of Adversarial Queries?: The Impact of Image Modifications on Content-based Image Retrieval. In ICM.
[19] Wonjae Nie, Hong Lu, Zijian Wang, Jingyuan Liu, and Zehua Guo. 2019. Weakly Supervised Image Retrieval via Coarse-scale Feature Fusion and Multi-level Attention Blocks. In ICM. 48–52.
[20] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Gretchen Krueger. 2021. Learning Transferable Visual Models from Natural Language Supervision. In ICM.
[21] Chao Li, Shangqiang Gao, Cheng Deng, Wei Liu, and Heng Huang. 2021. Adversarial Attack on Deep Cross-Modal Hamming Retrieval. In ICCV.
[22] Chao Li, Shangqiang Gao, Cheng Deng, De Xie, and Wei Liu. 2019. Cross-Modal Learning with Adversarial Samples. In NeurIPS.
[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In ECCV.
[24] Giorgos Tolias, Filip Radenovic, and Ondrej Chum. 2019. Targeted mismatch attack performance, we set the patch ratio to 0.1. 0.30

![Figure 6: Attack performance (mean trojan-horse R10) with different patch ratio. LBIR setup: CLIP + Flickr30ktest. Adversarial patches are learned with Flickr30ktrain as training data.](image_url)