Once and for All: Universal Transferable Adversarial Perturbation against Deep Hashing-Based Facial Image Retrieval

Long Tang, Dengpan Ye*, Yunna Lv, Chuanxi Chen, Yunming Zhang

Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education,
School of Cyber Science and Engineering, Wuhan University
{l_tang, yedp}@whu.edu.cn, lvyunnaa@outlook.com, {chencx, zhangyunming}@whu.edu.cn

Abstract

Deep Hashing (DH)-based image retrieval has been widely applied to face-matching systems due to its accuracy and efficiency. However, this convenience comes with an increased risk of privacy leakage. DH models inherit the vulnerability to adversarial attacks, which can be used to prevent the retrieval of private images. Existing adversarial attacks against DH typically target a single image or a specific class of images, lacking universal adversarial perturbation for the entire hash dataset. In this paper, we propose the first universal transferable adversarial perturbation against DH-based facial image retrieval, a single perturbation can protect all images. Specifically, we explore the relationship between clusters learned by different DH models and define the optimization objective of universal perturbation as leaving from the overall hash center. To mitigate the challenge of single-objective optimization, we randomly obtain sub-cluster centers and further propose sub-task-based meta-learning to aid in overall optimization. We test our method with popular facial datasets and DH models, indicating impressive cross-image, -identity, -model, and -scheme universal anti-retrieval performance. Compared to state-of-the-art methods, our performance is competitive in white-box settings and exhibits significant improvements of $10\% - 70\%$ in transferability in all black-box settings.

Introduction

In light of the rapid advancements of the Internet and Artificial Intelligence, novel technologies leveraging big data are significantly changing conventional lifestyles. Some enterprises extract copious amounts of facial images from social media to train face-matching systems. Subsequently, these systems are employed in various practical scenarios, including but not limited to security surveillance, photo management, facial attendance, and face-based access control, thereby substantially streamlining daily routines. Deep Hashing (DH), renowned for its expeditious processing, minimal storage requirements, and non-disclosure of image features, has garnered extensive utilization within face-matching systems (Luo et al. 2023). Nevertheless, this progress is accompanied by a concern regarding the exposure of private information. Existing open-source face-matching tools can take a person's image as input and return his personal profile, image source, and a series of retrieval similar images. The datasets used to train these tools are typically extracted from various social media platforms and contain a large number of daily photos. Malicious users may exploit an ID photo of one user uploaded to LinkedIn to retrieve a large number of other personal photos of him from the database, thereby exposing his daily life, interests, hobbies, profession, and other personal information, as shown in Fig. 1(a). These photos could potentially be used for fraudulent activities and other illicit behaviors. Therefore, privacy protection issues are receiving heightened attention.

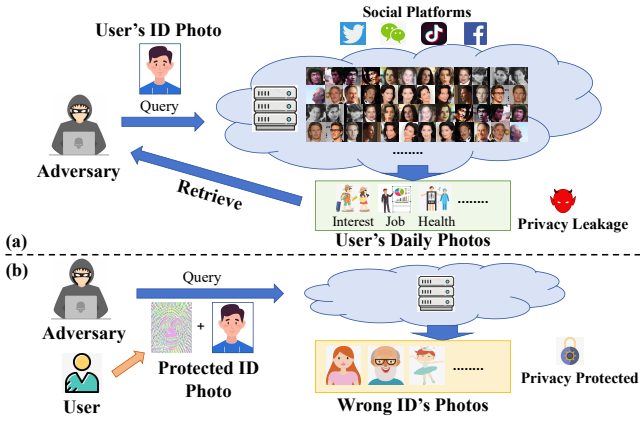


Figure 1: Illustrations of: (a) Risk of privacy leakage and (b) Protecting privacy in facial image retrieval.

Simultaneously, it has been established that DH models are vulnerable to adversarial examples. Prior investigations have encompassed untargeted attacks (Yang et al. 2018; Xiao, Wang, and Gao 2020), targeted attacks (Bai et al. 2020; Hu et al. 2021; Lu et al. 2021), and transferable attacks (Xiao and Wang 2021) on DH models, demonstrating promising outcomes through adversarial perturbations and patches. Recent studies have even made advancements in attacking cross-modal retrieval (Li et al. 2021; Zhang et al. 2023; Zhu et al. 2023; Wang et al. 2023). These insights inspire us to design adversarial examples to prevent the exposure of privacy held within the database, as shown in Fig. 1(b). Specifically, perturbations can be incorporated into the images before users share images on social platforms. Con-

*Corresponding author.

sequently, malicious users will fail to obtain accurate retrieval results from the DH system using such images.

However, real-world scenarios pose challenges as users may lack knowledge of the underlying DH model structure and hash schemes used by third-party databases. Additionally, protecting multiple images of different individuals is time-consuming. Therefore, a universal adversarial perturbation is needed to preserve image retrieval privacy. There is currently no universal attack against all categories of DH models and are unable to simultaneously meet the requirements of both universality and black-box transferability.

To meet these requirements, we propose a novel method called **U**niversal **T**ransferable **A**dversarial **P**erturbation (UTAP) for DH-based facial image retrieval. UTAP uses a few facial images from all database identities to create a universal perturbation that can be applied to images across all identities, preventing original database images from being retrieved by the perturbed query image, and is effective even against black-box models. We observed that current DH schemes consistently exhibit highly clustered patterns, and different DH models have similar cluster centers. Based on this, we propose a new insight into UTAP generation. Instead of deviating from the original image hash codes, we aim to deviate from the overall cluster center of the dataset. To address optimization challenges, we repeatedly and randomly select centers from different identities, resulting in multiple sub-cluster centers. Inspired by meta-learning, we treat these sub-cluster centers as sub-tasks and use multi-objective gradients to adjust the overall optimization gradients. Finally, we accumulate the gradients of each batch and round of attacks to ensure a stable direction of the universal perturbation, rather than accumulating perturbations.

We summarize the following contributions:

- We propose UTAP, the first universal and transferable adversarial perturbation for deep hashing-based facial image retrieval. A single adversarial perturbation can be applied to images across all users in the database, simultaneously exhibiting transferability to black-box models and unknown hash schemes.
- We propose an optimization objective of deviating from the overall cluster center and then enhancing the universal attack effectiveness of UTAP through a sub-task-based meta-learning strategy.
- We conduct extensive experiments on popular open-source facial datasets CASIA-WebFace and VGGFace2, validating the effectiveness of UTAP in attacking state-of-the-art DH schemes HashNet and CSQ, as well as its transferability in various black-box scenarios.

Related Works

Deep Hashing-Based Image Retrieval

Hashing methods have been widely employed to accelerate retrieval processes by mapping semantically similar multimedia information into compact binary codes within the Hamming space (Wang et al. 2017). Traditional approaches rely on manually crafted feature vectors, which are subsequently encoded into binary codes through separate projection and quantization procedures. Recently, the application of deep learning to hash-based image retrieval tasks has significantly enhanced retrieval performance owing to their remarkable feature extraction capabilities. These techniques train a CNN model, such as ResNet (He et al. 2016) or VGG (Simonyan and Zisserman 2014), to map images into low-dimensional features and then leverage hash layers to transform these features into binary hash codes. A crucial aspect of enhancing the accuracy of DH methods lies in establishing an appropriate loss function. HashNet (Cao et al. 2017), a prevalent retrieval algorithm, concurrently trains two loss functions to ensure superior classification accuracy and hash expression capability. CSQ (Yuan et al. 2020), a high-precision retrieval algorithm, encourages the clustering of similar images toward a shared hash code. We investigate UTAP based on these two hash algorithms since they exhibit optimal performance in terms of image retrieval.

Adversarial Attacks

Adversarial examples widely exist in deep neural networks, where carefully crafted perturbations added to clean objects can cause the models to return incorrect results. Early adversarial attacks, such as FGSM (Goodfellow, Shlens, and Szegedy 2014), I-FGSM (Kurakin, Goodfellow, and Bengio 2018), and PGD (Madry et al. 2018), generate adversarial examples by back-propagation gradients. Subsequently, many methods (Dong et al. 2018; Xie et al. 2019; Lin et al. 2019; Fang et al. 2022) have been proposed to improve the transferability of adversarial examples. To enhance the universality of adversarial examples, the concept of Universal Adversarial Perturbations (UAP) has been introduced (Moosavi-Dezfooli et al. 2017), where a single perturbation can adversarially affect all images in a dataset.

Attacks against Image Retrieval

Image retrieval based on DNN models is also vulnerable to adversarial attacks. For deep feature-based image retrieval, PIRE (Liu, Zhao, and Larson 2019) and AP-GAN (Zhao et al. 2022) are proposed to generate image-specific perturbation and patch, respectively. (Li et al. 2019) propose a UAP to improve the generalization ability of adversarial perturbations. (Tolias, Radenovic, and Chum 2019) propose TMAA, the first targeted attack against image retrieval systems. To attack the black-box model, (Chen et al. 2021) propose DAIR, a query-efficient decision-based attack on image retrieval systems. For DH-based image retrieval, (Yang et al. 2018) first propose an attack HAG on Hamming space. (Xiao, Wang, and Gao 2020) propose another untargeted adversarial example attack CWDM aiming at privacy preservation. DHTA (Bai et al. 2020) is proposed to enable adversarial examples to retrieve the target class. SDHA (Lu et al. 2021) is proposed to simultaneously ensure attack effectiveness and perturbation management. To increase the transferability of adversarial examples, (Xiao and Wang 2021) propose a targeted adversarial attack NAG that can successfully attack the black-box model. (Hu et al. 2021) propose AdvHash, the first targeted adversarial patch applicable to specific classes. Recently, some researchers are even paying new focuses on attacking cross-modal retrieval (Li et al. 2021; Zhang et al. 2023; Zhu et al. 2023; Wang et al. 2023).

Method	Model	Attack	Universal	Transferable
PIRE	Feature	Perturb	No	No
AP-GAN	Feature	Patch	No	No
UAP	Feature	Perturb	Yes	No
TMAA	Feature	Perturb	No	Yes
DAIR	Feature	Perturb	No	No
HAG	Hashing	Perturb	No	Yes
CWDM	Hashing	Perturb	No	Yes
DHTA	Hashing	Perturb	No	No
SDHA	Hashing	Perturb	No	No
NAG	Hashing	Perturb	No	Yes
AdvHash	Hashing	Patch	Class-wise	No
UTAP	**Hashing**	**Perturb**	**Yes**	**Yes**

Table 1: Comparisons of the relevant research on adversarial attacks in image retrieval.

However, no existing method can simultaneously consider universality and transferability, which is the goal of our proposed UTAP. Table 1 summarizes the comparisons of the relevant research on adversarial attacks in image retrieval.

Motivation

Threat Model

We introduce the assumed risk scenario. The adversary wants to steal the private information of a certain person, and he can input a certain query image of this person into a DH-based image retrieval model (e.g., CSQ-ResNet50), which returns a series of life photos that contains private information matching the query image. Therefore, the user (defender) would like to design a universal perturbation that is superimposed on the query image before publication so that the adversary cannot use this image to retrieve the private photos in the database. For simplicity, we assume that the user has knowledge about the distribution of the dataset (i.e., knows images of whom are included in the dataset) and a small portion of images of each person, but does not know other knowledge about the model, such as structure, parameters, hash schemes, loss functions, etc. The user can only generate the perturbation by surrogate models. For efficiency, the user wants to use only one perturbation to protect all images under all identities of the dataset.

Observations

The goal of modern DH models is to minimize the Hamming distance between hash codes of images with the same label and maximize the Hamming distance between hash codes of images with different labels. There is no collision rate constraint on hash codes among images of the same class. Due to the significantly lower dimensionality of hash codes compared to images, different images can have the same hash code when using such DH models.

Center Similarity Quantization (CSQ), one of the best DH algorithms, introduces the concept of hash centers and constructs well-separated hash centers using the Hadamard matrix and Bernoulli distribution. These hash centers are then used to define the calculation of center similarity measurement. As shown in Fig. 2, The dots are hash codes of the image database from four DH models, and the stars are the voted hash center following (Bai et al. 2020). It shows that the clusters obtained from different models are similar, resulting in almost the same overall hash centers.

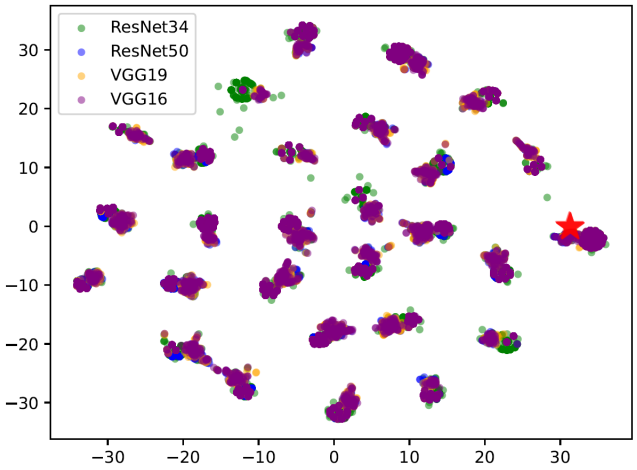


Figure 2: T-SNE plots illustrating the distribution of hash codes and the overall hash center obtained by various CSQ-trained models. The dataset is CASIA-WebFace.

From this, we can infer that high-precision hash retrieval models often have similar or even identical hash centers. Therefore, as long as the adversarial example is leaving from the original hash centers (untargeted attack) or close to the target hash centers (targeted attack) in white-box models, even though there are differences in feature extraction in black-box models, similar distribution effects can still be achieved, resulting in a successful black-box transferability.

Methodology

Preliminaries

We formally define DH-based facial image retrieval. Let $X = (x_i, y_i)_{i=1}^{N}$ indicates a dataset of P users with N images in total, where x_i indicates the retrieval image and $y_i \in \{0,1\}^P$ corresponds to the one-hot identity vector, $y_i^p = 1$ means x_i is the image of p-th person. Suppose the p-th person has N_p images. Let $X^{(s)} = \{(x,y)_{s\times P} | \{s << N_p\}_{p=1}^{P}\} \subseteq X$ be a subset of X consisting of s images of each person, and s is far smaller than N_p.

Deep Hashing Model. We can obtain the K-bit hash code c of an image x through a DH model $F(\cdot)$ following:

$$c = F(x) = sign(H(x)), \tag{1}$$

where $H(\cdot)$ is the model normally consisting of a pre-trained CNN feature extractor and a K-dimension fully-connected hash layer. To facilitate the calculation of the back-propagation, $tanh(\cdot)$ is often used to approximate $sign(\cdot)$, so the hashing model returns the K-bit hash code within the range $(-1, +1)$.

Retrieving with Similarity. Suppose each image x_i of the database X has a pre-defined hash code c_i obtained through Eq. (1). Given a query image x_q, it will be fed into $F(\cdot)$

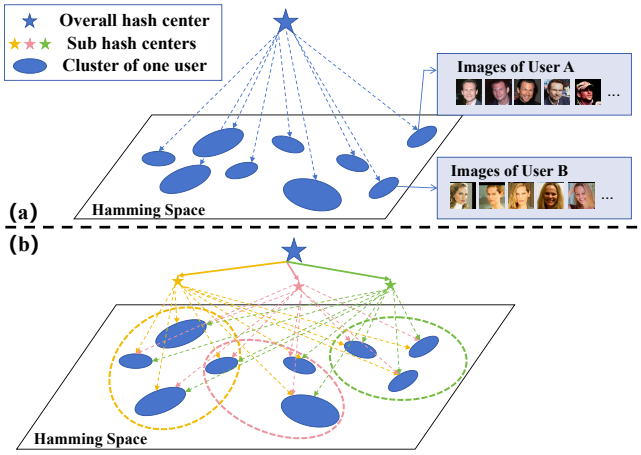


Figure 3: Illustration of (a) Overall optimization target and (b) Sub-task-based meta-learning.

first to obtain the hash code c_q. Then the Hamming distance between c_q and c_i is calculated between x_q and each x_i:

$$D = \{d_H(c_q, c_i)\}_{i=1}^{N} = \{(K - c_q \cdot c_i)/2\}_{i=1}^{N}. \quad (2)$$

Since the Hamming distance is linearly related to the inner product, the facial image retrieval system obtains a descending list of inner products between the query image and database images and finally returns the most similar images as the retrieval results.

Formulating Optimization Target

The goal of UTAP is to obtain a visually imperceptible perturbation δ to disrupt the initial hash clustering of all identities in the facial image dataset as much as possible. It has been observed that for high-precision DH retrieval algorithms, different DH models exhibit similar or identical hash centers. It inspires us that the universal adversarial perturbation guided by hash centers may possess stronger transferability. Therefore, we transform the problem of generating universal perturbation into the maximization of the Hamming distance between adversarial examples and the overall hash center of the original data, as shown in Fig. 3(a).

Formally, following (Bai et al. 2020), we select s images from identity p and then utilize a component-voting mechanism to obtain the hash center h_p for this identity:

$$h_p = sign(\sum_{j=1}^{s} c_j), \quad (3)$$

where c_j is the hash code for the j-th image. After obtaining the hash centers of each identity, we obtain the hash center h_o with overall P identities following the above approach:

$$h_o = h(P) = sign(\sum_{p=1}^{P} h_p), \quad (4)$$

where P is the total number of identities. Then the overall optimization of UTAP can be written as follows:

$$\max_{\delta} D(F(X^{(s)'}), h_o) \Rightarrow \max_{\delta} \sum_{j=1}^{s \times P} d_H(sign(H(x'_j)), h_o), \quad (5)$$

where x'_j is the image from the training subset $X^{(s)}$ added with adversarial perturbation δ, $x'_j = x_j + \delta$. δ is limited with $||\epsilon||_\infty$ for invisibility.

Since the $sign(\cdot)$ function is unable to compute continuous gradients, we use the hyperbolic tangent function to approximate it during the optimization of UTAP. We replace the Hamming distance calculation with the inner product and eventually rewrite Eq. 5 as follows:

$$\min_{\delta} \sum_{j=1}^{s \times P} \frac{1}{K} h_o^\top tanh(H(x'_j)). \quad (6)$$

Notice that Eq. 6 differs from (Yang et al. 2018; Hu et al. 2021; Bai et al. 2020) in that we do not multiply the hyperparameter α in $tanh(\cdot)$. This is because the overall hash center h_o will not be close to any of the individual image hash codes, and the maximum case of moving away from h_o (i.e., contrary to all the bits of h_o) can hardly be realized on all the individual images simultaneously, either. Therefore α is not needed to avoid the case of gradient vanishing. We write the loss function as follows:

$$L(X^{(s)}, \delta, h_o) = \frac{\sum_{j=1}^{s \times P} h_o^\top tanh(H(x'_j))}{s \times P \times K}. \quad (7)$$

We abbreviate $L(X^{(s)}, \delta, h_o)$ to $L(\delta, h_o)$ in the following for simplicity. We maximize $L(\delta, h_o)$ to minimize the sum of the inner products of a batch of images.

Sub-Task-Based Meta-Learning

Based on empirical observations, optimizing directly with a single overall hash center may make it difficult to update the gradient on some samples that are very close to the hash center or those that are already far from it. Inspired by (Fang et al. 2022), we introduce a meta-learning approach where we randomly select subsets of total identities multiple times to calculate Eq. 4, generating a series of sub hash centers, as shown in Fig. 3(b). These sub-centers are treated as meta-tasks, and several support tasks are constructed to derive the adversarial perturbations away from sub-centers first. Then, on these perturbations, the overall gradient is updated using the overall hash center as the query task. Finally, the universal adversarial perturbation is updated by aggregating the gradients from each sub-task.

Formally, we construct R sub-tasks, each sub-task h_r is calculated by Eq. 4 with Q_r random identities, $Q_r \in [2, P)$. In each meta-learning iteration, we calculate the gradient of each sub-task as follows:

$$g_r = \nabla_\delta L(\delta, h_r). \quad (8)$$

After that, we use FGSM to get the support perturbation $\delta_r = \epsilon \cdot sign(g_r)$, and continue using the overall hash center h_o to compute the query gradient based on δ_r as follows:

$$g_q = \sum_{r=1}^{R} \nabla_\delta L(\delta + \delta_r, h_o). \quad (9)$$

Algorithm 1: Universal Transferable Adversarial Perturbation (UTAP)

Input: $X^{(s)}$(training facial images).
Parameter: P (identities), T (total iteration), R (sub-tasks), M (meta iteration), η (learning rate).
Output: UTAP δ.

Initialize $\delta^0 = 0$.
Obtain hash centers $\{h_p\}^P$ of $X^{(s)}$ by Eq. 1&3.
Obtain the overall hash center h_o by Eq. 4.
for t in T **do**
  Obtain sub-tasks $\{h_r = h(Q_r)\}^R$ by Eq. 4.
  for m in M **do**
    for r in R **do**
      $g_r = \nabla_\delta L(\delta, h_r)$.
      $\delta_r = \epsilon \cdot sign(g_r)$.
    end for
    $g_q = \sum_{r=1}^{R} \nabla_\delta L(\delta + \delta_r, h_o)$.
    $g_m = g_q + \sum_{r=1}^{R} g_r$.
  end for
  $\delta^{t+1} = clip_\epsilon^\delta(\delta^t + \eta \cdot sign(\sum_{m=1}^{M} g_m))$.
end for
return δ^T.

Thus, for one meta iteration, the meta gradient g_m is written as $g_m = g_q + \sum_{r=1}^{R} g_r$.

After several meta iterations, we obtain the sum of the overall gradients to update the UTAP δ:

$$\delta^{t+1} = clip_\epsilon^\delta(\delta^t + \eta \cdot sign(\sum_{m=1}^{M} g_m)), \tag{10}$$

where η is the learning rate, $m \in M$ is the number of meta iterations and $t \in T$ is the number of overall iterations.

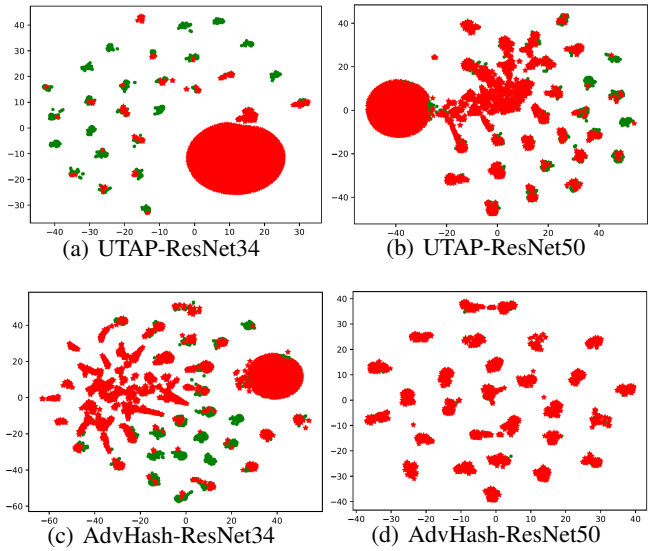
Note that all we aggregate are gradients but not perturbations because we treat the generation of UTAP as an overall optimization problem, which differs from previous UAP methods against DH models. The UTAP is summarized in Alg. 1, and the overall pipeline is shown in the Appendix.

Experiments

Experimental Settings

Datasets. We evaluate UTAP on CASIA-WebFace (CASIA) (Yi et al. 2014) and VGGFace2 (Cao et al. 2018) respectively. Since the wide variation in the number of images contained in each identity of these datasets, we choose $P = 28$ identities that $N_p > 500$ in each dataset. We randomly select $s = 50$ images from each identity as the training set, and the rest images as the database. Therefore, for CASIA-WebFace, $N = 12370$ and $X^{(s)} = 1400$. For VGGFace2, $N = 11413$ and $X^{(s)} = 1400$. N and $X^{(s)}$ are independently but identically distributed, and we use $X^{(s)}$ to vote for hash centers. All images are resized to 224×224.

Metrics. We use mean average precision (mAP) as the evaluation metric. We calculate the mAPs on the top 300 retrieved results and retrieve the database with database images. We record the mAP values before (**Original, O**) and after adding the adversarial perturbation (**Adv**). The smaller the value of **Adv**, the fewer images from original identities are matched, indicating better attack performance.



(a) UTAP-ResNet34 (b) UTAP-ResNet50 (c) AdvHash-ResNet34 (d) AdvHash-ResNet50

Figure 4: Distribution of original (green) and adversarial (red) database hash codes. (a)(c) are white-box results and (b)(d) are black-box results.

Models. We use 4 different DNNs (ResNet34, ResNet50, VGG16, VGG19) as the feature extraction backbone and 2 state-of-the-art DH algorithms (CSQ and HashNet). For each model, we train a 32-bit hash mode and a 64-bit hash mode respectively for evaluation. All experiments are trained on only one model and transferred to the others, and all perturbations are generated on 64-bit models.

Comparisons. Since UTAP is the first work considering the cross-image, -identity, -model, and -scheme universal adversarial perturbation against DH models, we can only make appropriate modifications to the most relevant methods to achieve a relatively fair comparison. We compare UTAP with DHTA, CWDM, and AdvHash. DHTA and CWDM are both single-image perturbations. Both individual DHTA and CWDM perturbations are summed up following traditional UAP settings to obtain a universal perturbation respectively. DHTA is a targeted attack, we modify the target of DHTA to the negative hash centers of the original classes. AdvHash is a class-wise universal adversarial patch, which is also a targeted attack. We let AdvHash heads to the negative hash center of the corresponding original class, and then sum up the adversarial patch following its own settings.

Details. For UTAP, the overall iterations $T = 100$, the meta learning iterations $M = 10$, the number of sub-tasks $R = 10$, and the learning rate $\eta = 0.02$. All perturbations in experiments are clipped to $\epsilon = 16/255$. For AdvHash, the scale of the patch size is set to 0.05.

Dataset	Model	Method	ResNet34		ResNet50		VGG16		VGG19	
			32-bit	64-bit	32-bit	64-bit	32-bit	64-bit	32-bit	64-bit
CASIA	**Original**		91.23	91.13	90.72	91.06	87.61	89.50	86.59	86.33
	ResNet34	DHTA	49.65	41.99*	67.14	74.37	70.34	76.29	68.52	66.18
		CWDM	79.22	79.62*	70.39	77.57	71.74	73.67	69.88	70.30
		AdvHash	45.89	8.60*	88.20	89.18	65.69	80.04	71.39	64.09
		UTAP	**6.69**	**5.00***	**24.68**	**29.60**	**33.53**	**38.60**	**31.51**	**27.25**
	ResNet50	DHTA	70.22	71.87	47.92	57.33*	72.13	77.79	70.75	67.15
		CWDM	78.81	77.98	72.60	78.14*	72.43	73.65	70.37	70.72
		AdvHash	86.95	86.40	80.21	5.67*	69.94	79.78	72.00	65.59
		UTAP	**24.71**	**35.74**	**12.52**	**5.43***	**44.96**	**46.65**	**36.40**	**26.33**
	VGG16	DHTA	69.99	72.03	67.75	75.47	56.18	63.86*	57.72	53.48
		CWDM	82.43	82.36	75.96	81.69	73.46	75.98*	71.53	72.40
		AdvHash	87.24	86.90	88.47	89.21	16.87	**3.53***	54.13	50.38
		UTAP	**47.83**	**46.65**	**43.66**	**58.24**	**15.15**	6.49*	**23.23**	**16.95**
	VGG19	DHTA	74.91	76.28	72.59	78.70	66.63	75.25	55.30	48.64*
		CWDM	81.33	79.51	75.71	79.39	68.98	70.68	66.29	69.15*
		AdvHash	87.14	87.22	88.67	89.47	60.22	69.66	11.82	6.38*
		UTAP	**32.31**	**31.56**	**31.58**	**27.95**	**18.01**	**22.43**	**8.01**	**5.13***
VGGFace2	**Original**		95.91	95.99	95.61	95.18	94.19	95.37	93.95	93.27
	ResNet34	DHTA	57.50	51.33*	85.93	85.75	83.86	85.44	81.90	84.37
		CWDM	84.66	84.39*	85.53	84.71	85.31	86.08	80.33	84.44
		AdvHash	82.31	39.45*	93.13	92.97	86.22	89.11	87.27	78.50
		UTAP	**8.16**	**5.33***	**34.31**	**30.12**	**45.01**	**40.54**	**40.42**	**44.98**
	ResNet50	DHTA	83.36	79.33	69.68	63.72*	85.52	86.61	82.95	84.71
		CWDM	86.86	86.28	86.04	84.52*	86.01	87.02	81.48	85.47
		AdvHash	92.57	92.01	82.09	21.55*	80.42	86.03	80.87	61.50
		UTAP	**26.95**	**28.63**	**9.67**	**5.70***	**48.70**	**43.48**	**38.53**	**43.67**
	VGG16	DHTA	80.19	78.22	82.40	79.65	63.95	52.31*	50.56	63.94
		CWDM	88.40	88.12	89.71	89.01	84.99	86.65*	83.63	85.17
		AdvHash	92.70	92.29	92.59	92.43	11.20	**4.02***	78.11	52.02
		UTAP	**42.74**	**43.39**	**54.28**	**51.36**	**10.41**	6.85*	**17.74**	**21.26**
	VGG19	DHTA	81.55	80.16	82.09	79.99	71.16	72.92	57.73	59.49*
		CWDM	88.53	87.63	88.75	88.23	84.42	85.55	83.67	85.01*
		AdvHash	92.62	92.38	92.98	92.80	73.56	82.06	34.12	**3.93***
		UTAP	**61.92**	**62.41**	**65.25**	**54.55**	**21.20**	**21.38**	**13.52**	7.96*

Table 2: Quantitative comparisons (mAP%). Except for **Original** mAP values, all data are **Adv** values, the same hereinafter. The default hash scheme is (CSQ, 64-bit). The white-box attacks are represented with *, and the best results are in bold.

Results and Evaluations

Transferability between Models and Bits. We compare the performance on different models, datasets, and hash bits of four methods, and present the results in Table 2. It shows that in 8 white-box settings, UTAP achieves the maximum mAP decrease in 5 cases, slightly inferior to AdvHash in 3 cases, but significantly better than DHTA and CWDM. In black-box scenarios, UTAP is clearly superior to all existing methods. In the best case (VGGFace2, 64-bit, VGG19 to VGG16), UTAP improves by about 50% compared to the second-best method. AdvHash demonstrates competitive performance in white-box scenarios, but unfortunately, it performs poorly in black-box transferability, even with the same model but different hash bits (ResNet50, 64-bit to 32-bit), resulting in a significant performance decrease. This may be due to the poor transferability of the adversarial patches, or it could be the amplification of adversarial features in the patch region due to the accumulation of perturbations, resulting in overfitting to the white-box model. Compared to DHTA and CWDM, UTAP has better universality and transferability, which suggests that in the adversarial perturbation settings, accumulating perturbations with non-unified objectives is difficult to achieve a universal adversarial effect. UTAP considers obtaining a universal perturbation as an optimization problem with a unified objective and accumulates gradients rather than perturbations, thus greatly alleviating the catastrophic forgetting problem caused by the accumulation of perturbations, achieving the best universality and transferability.

Visualizations. We conduct experiments to analyze the distribution of hash codes in the dataset before and after adding noise using t-SNE plots, as shown in Fig. 4. UTAP and AdvHash are obtained from the ResNet34 model trained on the VGGFace2 dataset using a 64-bit hash code and the CSQ algorithm. We also consider the ResNet50 model as a black-box model for transferability evaluation.

On white-box settings, the adversarial distribution of AdvHash is hard to differentiate from the original distribution, which aligns with the bad performance shown in Table 2 of AdvHash on the ResNet34 model. Conversely, UTAP exhibits a more distinct adversarial distribution, deviating significantly from the original distribution. Notably, a majority

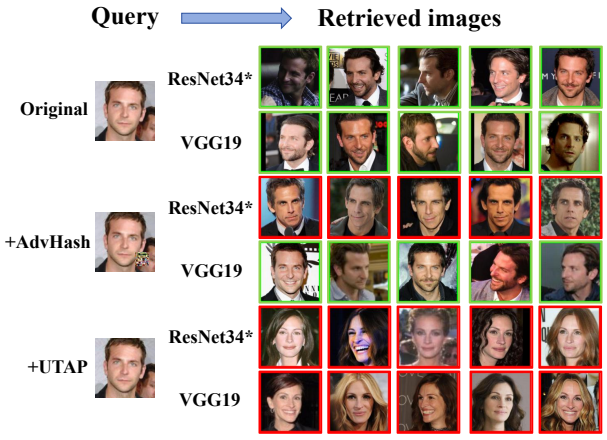


Figure 5: Illustration of retrieval effects of AdvHash and UTAP on CASIA-WebFace. The green box indicates the original identity and the red box indicates the wrong identity. The white-box model is emphasized with *.

of the adversarial examples cluster around a new center. We attribute this clustering behavior of UTAP to the design of the loss function. The worst-case scenario of the adversarial hash code is completely opposite to the original code, that is, the negative hash center. By defining the loss function to encourage moving away from the overall hash center, UTAP promotes the spontaneous clustering of adversarial examples around the negative overall center.

On black-box settings, we observe that AdvHash's adversarial distribution almost perfectly overlaps with the original distribution. This finding underscores the poor transferability of AdvHash. Due to the differences in model structures and feature extraction, UTAP's clustering effect on adversarial distribution is weakened in the black-box model. However, it is worth noting that a substantial portion of samples still cluster in regions that are far away from all original distributions. These observations are consistent with the results in Table 2, further supporting the notion that the universal adversarial perturbation guided by the overall hash center indeed leads to improved cross-model transferability.

We further visualize the retrieval effects of UTAP and AdvHash on CASIA-WebFace in Fig. 5. It shows that AdvHash fails to transfer from ResNet34 to VGG19. UTAP is effective on both white-box and black-box models, and the correlation difference between retrieved images and the query image is larger compared to AdvHash.

Transferability between Hash Algorithms. We conduct cross-algorithm transferability experiments and present the results in Table 3. As per findings from (Hu et al. 2021), HashNet exhibits a poorer clustering effect compared to CSQ, leading to lower mAP and a less certain overall hash center for HashNet. Nevertheless, UTAP still showcases competitive white-box attack performance. On the other hand, the transferability of AdvHash from HashNet to CSQ is notably inferior to that from CSQ to HashNet, suggesting that AdvHash's black-box transferability is vulnerable to different hashing algorithms. UTAP consistently maintains the strongest and most stable black-box transferability performance across various algorithms.

Alg.	Model	Method	ResNet50 CSQ O:91.06	ResNet50 HashNet O:64.77	VGG16 CSQ O:89.50	VGG16 HashNet O:54.83
CSQ	ResNet50	DHTA	57.33*	44.43	77.79	43.78
		CWDM	78.14*	54.22	73.65	43.92
		AdvHash	5.67*	52.01	79.78	51.13
		UTAP	**5.43***	**21.30**	**46.65**	**23.41**
	VGG16	DHTA	75.48	53.98	63.86*	32.33
		CWDM	81.69	56.96	75.98*	46.57
		AdvHash	89.21	63.80	**3.53***	20.63
		UTAP	**52.62**	**32.40**	6.49*	**14.10**
HashNet	ResNet50	DHTA	64.84	36.75*	75.30	41.17
		CWDM	76.37	52.78*	72.71	43.33
		AdvHash	75.62	4.49*	77.86	50.90
		UTAP	**12.17**	**4.20***	**39.38**	**16.84**
	VGG16	DHTA	76.34	52.43	67.96	31.65*
		CWDM	81.26	56.70	70.44	41.09*
		AdvHash	87.98	63.68	41.36	**3.38***
		UTAP	**62.31**	**39.70**	**34.19**	6.96*

Table 3: Cross-algorithm comparisons (mAP%) on CASIA dataset. The default hash bit $K = 64$. The white-box attacks are represented with *, and the best results are in bold.

Model	Method	ResNet34 O:91.13	ResNet50 O:91.06	VGG16 O:89.50
ResNet34	-ML	6.64*	40.24	44.49
	UTAP	**5.00***	**29.60**	**38.60**
ResNet50	-ML	46.53	9.67*	55.53
	UTAP	**35.74**	**5.43***	**46.65**
VGG16	-ML	59.34	60.90	22.32*
	UTAP	**46.65**	**58.24**	**6.49***

Table 4: Ablation study for meta-learning (mAP%). The default setting is (CSQ, 64-bit, CASIA). The white-box attacks are represented with *, and the best results are in bold.

Ablation Studies. We perform ablation experiments on UTAP without sub-task-based meta-learning (-ML). The experimental results are shown in Table 4. Without meta-learning, all results are worse than UTAP and are better than previous state-of-the-art methods in Table 2. This shows that using the sub-task-based meta-learning can indeed push the adversarial distribution farther away from some original distributions that are difficult to optimize, thus achieving better universal adversarial performance.

Conclusion

We propose UTAP, the first universal transferable adversarial perturbation for DH-based facial image retrieval to protect user privacy. We first transform the universal perturbation generation problem into an optimization problem, aiming to leave from the dataset's overall hash center. Then we further introduce the sub-task-based meta-learning, enhancing the perturbation's universality and transferability by leaving from multiple random sub-cluster centers. Extensive experiments validate UTAP's significant advantages in universality and transferability in the state-of-the-art DH schemes compared to existing research.

Acknowledgments

This work is supported by the National Natural Science Foundation of China NSFC (No. 62072343), the Fundamental Research Funds for the Central Universities (No. 2042023kf0228), and the National Key Research and Development Program of China (No. 2019QY(Y)0206).

References

Bai, J.; Chen, B.; Li, Y.; Wu, D.; Guo, W.; Xia, S.-t.; and Yang, E.-h. 2020. Targeted attack for deep hashing based retrieval. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, 618–634. Springer.

Cao, Q.; Shen, L.; Xie, W.; Parkhi, O. M.; and Zisserman, A. 2018. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition*, 67–74. IEEE.

Cao, Z.; Long, M.; Wang, J.; and Yu, P. S. 2017. Hashnet: Deep learning to hash by continuation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5608–5617.

Chen, M.; Lu, J.; Wang, Y.; Qin, J.; and Wang, W. 2021. DAIR: A query-efficient decision-based attack on image retrieval systems. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1064–1073.

Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9185–9193.

Fang, S.; Li, J.; Lin, X.; and Ji, R. 2022. Learning to learn transferable attack. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 571–579.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. arXiv:1412.6572.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 770–778.

Hu, S.; Zhang, Y.; Liu, X.; Zhang, L. Y.; Li, M.; and Jin, H. 2021. Advhash: Set-to-set targeted attack on deep hashing with one single adversarial patch. In *Proceedings of the 29th ACM International Conference on Multimedia*, 2335–2343.

Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2018. Adversarial examples in the physical world. In *Artificial Intelligence Safety and Security*, 99–112. Chapman and Hall/CRC.

Li, C.; Gao, S.; Deng, C.; Liu, W.; and Huang, H. 2021. Adversarial attack on deep cross-modal hamming retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2218–2227.

Li, J.; Ji, R.; Liu, H.; Hong, X.; Gao, Y.; and Tian, Q. 2019. Universal perturbation attack against image retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4899–4908.

Lin, J.; Song, C.; He, K.; Wang, L.; and Hopcroft, J. E. 2019. Nesterov accelerated gradient and scale invariance for adversarial attacks. arXiv:1908.06281.

Liu, Z.; Zhao, Z.; and Larson, M. 2019. Who's afraid of adversarial queries? The impact of image modifications on content-based image retrieval. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, 306–314.

Lu, J.; Chen, M.; Sun, Y.; Wang, W.; Wang, Y.; and Yang, X. 2021. A smart adversarial attack on deep hashing based image retrieval. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, 227–235.

Luo, X.; Wang, H.; Wu, D.; Chen, C.; Deng, M.; Huang, J.; and Hua, X.-S. 2023. A survey on deep hashing methods. *ACM Trans. Knowl. Discov. Data*, 17(1).

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; Fawzi, O.; and Frossard, P. 2017. Universal adversarial perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1765–1773.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.

Tolias, G.; Radenovic, F.; and Chum, O. 2019. Targeted mismatch adversarial attack: Query with a flower to retrieve the tower. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5037–5046.

Wang, J.; Zhang, T.; Sebe, N.; Shen, H. T.; et al. 2017. A survey on learning to hash. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4): 769–790.

Wang, T.; Zhu, L.; Zhang, Z.; Zhang, H.; and Han, J. 2023. Targeted adversarial attack against deep cross-modal hashing retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*.

Xiao, Y.; and Wang, C. 2021. You see what I want you to see: Exploring targeted black-box transferability attack for hash-based image retrieval systems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1934–1943.

Xiao, Y.; Wang, C.; and Gao, X. 2020. Evade deep image retrieval by stashing private images in the hash space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9651–9660.

Xie, C.; Zhang, Z.; Zhou, Y.; Bai, S.; Wang, J.; Ren, Z.; and Yuille, A. L. 2019. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2730–2739.

Yang, E.; Liu, T.; Deng, C.; and Tao, D. 2018. Adversarial examples for hamming space search. *IEEE Transactions on Cybernetics*, 50(4): 1473–1484.

Yi, D.; Lei, Z.; Liao, S.; and Li, S. Z. 2014. Learning face representation from scratch. arXiv:1411.7923.

Yuan, L.; Wang, T.; Zhang, X.; Tay, F. E.; Jie, Z.; Liu, W.; and Feng, J. 2020. Central similarity quantization for efficient image and video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3083–3092.

Zhang, P.-F.; Bai, G.; Yin, H.; and Huang, Z. 2023. Proactive privacy-preserving learning for cross-modal retrieval. *ACM Transactions on Information Systems*, 41(2): 1–23.

Zhao, G.; Zhang, M.; Liu, J.; Li, Y.; and Wen, J.-R. 2022. AP-GAN: Adversarial patch attack on content-based image retrieval systems. *GeoInformatica*, 1–31.

Zhu, L.; Wang, T.; Li, J.; Zhang, Z.; Shen, J.; and Wang, X. 2023. Efficient query-based black-box attack against cross-modal hashing retrieval. *ACM Transactions on Information Systems*, 41(3): 1–25.