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, lvyunna@outlook.com, {chenx, zhyuming}@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 universality. 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.

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 retrieve 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.

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.
subsequently, 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 Universal Transferable Adversarial Perturbation (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 (Zha 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 untargeted 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).
Table 1: Comparisons of the relevant research on adversarial attacks in image retrieval.

<table>
<thead>
<tr>
<th>Method</th>
<th>Model</th>
<th>Attack</th>
<th>Universal</th>
<th>Transferable</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIRE</td>
<td>Feature</td>
<td>Perturb</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>AP-GAN</td>
<td>Feature</td>
<td>Patch</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>UAP</td>
<td>Feature</td>
<td>Perturb</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>TMAA</td>
<td>Feature</td>
<td>Perturb</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>DAIK</td>
<td>Feature</td>
<td>Perturb</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>CWDM</td>
<td>Hashing</td>
<td>Perturb</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>DHTA</td>
<td>Hashing</td>
<td>Perturb</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SDHA</td>
<td>Hashing</td>
<td>Perturb</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>NAG</td>
<td>Hashing</td>
<td>Perturb</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>AdvHash</td>
<td>Hashing</td>
<td>Patch</td>
<td>Class-wise</td>
<td>No</td>
</tr>
<tr>
<td>UTAP</td>
<td>Hashing</td>
<td>Perturb</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

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.

**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 \( X^{(s)} = \{(x, y)_{s \times P}\} \{s \leq N_p\} \subseteq X \) 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:

\[
c = F(x) = sign(H(x)),
\]

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 \( F(\cdot) \)
The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)

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 \( x_j \):

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

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 \( \delta \) 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 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 metatasks, 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, 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:

\[
h_p = sign(\sum_{j=1}^{s} c_j), \tag{3}
\]

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:

\[
h_o = h(P) = sign(\sum_{p=1}^{P} h_p), \tag{4}
\]

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

\[
\max_{\delta} D(F(X^{(s)}'), h_o) \Rightarrow \max_{\delta} \sum_{j=1}^{s \times P} d_H(sign(H(x_j')), h_o), \tag{5}
\]

where \( x_j' \) is the image from the training subset \( X^{(s)} \) added with adversarial perturbation \( \delta \), \( x_j' = x_j + \delta \). \( \delta \) 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:

\[
\min_{\delta} \sum_{j=1}^{s \times P} 1/K h_o^{\top} \tanh(H(x_j')). \tag{6}
\]

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 (Yang et al. 2018; Hu et al. 2021; Bai et al. 2020) in that we do not multiply the hyperparameter \( \alpha \) 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 \( \alpha \) is not needed to avoid the case of gradient vanishing. We write the loss function as follows:

\[
L(X^{(s)}, \delta, h_o) = \sum_{j=1}^{s \times P} h_o^{\top} \tanh(H(x_j')) / s \times P \times K. \tag{7}
\]

where \( P \) is the total number of identities. Then 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:

\[
g_r = \nabla \delta L(\delta, h_r). \tag{8}
\]

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 \( \delta_r \) as follows:

\[
g_q = \sum_{r=1}^{R} \nabla \delta L(\delta + \delta_r, h_o). \tag{9}
\]
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), $\eta$ (learning rate).

**Output:** UTAP $\delta$.

1: Initialize $\delta^0 = 0$.
2: Obtain hash centers $\{h_p\}$ of $X^{(s)}$ by Eq. 1 & 3.
3: Obtain the overall hash center $h_o$ by Eq. 4.
4: for $t \in T$ do
5:  for $r \in R$ do
6:     $g_r = \nabla_{\delta} L(\delta, h_r)$.
7:     $\delta_r = \epsilon \cdot \text{sign}(g_r)$.
8:  end for
9:  $g_{m} = \sum_{r=1}^{R} g_r$.
10: end for
11: $\delta^{t+1} = \text{clip}_{\epsilon}(\delta^t + \eta \cdot \text{sign}(\sum_{m=1}^{M} g_m))$.
12: end for
13: return $\delta^T$.

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 $\delta$:

$$\delta^{t+1} = \text{clip}_{\epsilon}(\delta^t + \eta \cdot \text{sign}(\sum_{m=1}^{M} g_m)),$$

where $\eta$ 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, $N = 11143$ 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.

**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.
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.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>Method</th>
<th>ResNet34 64-bit</th>
<th>ResNet50 64-bit</th>
<th>VGG16 64-bit</th>
<th>VGG19 64-bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASIA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DHTA</td>
<td>49.65</td>
<td>72.77</td>
<td>70.62</td>
<td>86.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CWDM</td>
<td>79.22</td>
<td>79.62</td>
<td>71.34</td>
<td>86.49</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AdvHash</td>
<td>45.89</td>
<td>8.60*</td>
<td>85.52</td>
<td>82.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UTAP</td>
<td><strong>6.69</strong></td>
<td><strong>5.00</strong></td>
<td><strong>43.39</strong></td>
<td><strong>85.52</strong></td>
</tr>
<tr>
<td>VGG16</td>
<td></td>
<td>DHTA</td>
<td>70.22</td>
<td>71.87</td>
<td>72.43</td>
<td>87.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CWDM</td>
<td>78.81</td>
<td>77.98</td>
<td>73.65</td>
<td>87.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AdvHash</td>
<td>86.95</td>
<td>86.40</td>
<td>69.78</td>
<td>86.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UTAP</td>
<td><strong>24.71</strong></td>
<td><strong>35.74</strong></td>
<td><strong>49.96</strong></td>
<td><strong>86.08</strong></td>
</tr>
<tr>
<td>VGG19</td>
<td></td>
<td>DHTA</td>
<td>69.99</td>
<td>72.03</td>
<td>70.75</td>
<td>86.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CWDM</td>
<td>82.43</td>
<td>82.36</td>
<td>72.43</td>
<td>87.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AdvHash</td>
<td>87.24</td>
<td>86.90</td>
<td>69.94</td>
<td>86.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UTAP</td>
<td><strong>47.83</strong></td>
<td><strong>46.65</strong></td>
<td><strong>48.70</strong></td>
<td><strong>86.08</strong></td>
</tr>
<tr>
<td>VGGFace2</td>
<td></td>
<td>DHTA</td>
<td>74.91</td>
<td>76.28</td>
<td>72.59</td>
<td>86.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CWDM</td>
<td>81.33</td>
<td>79.51</td>
<td>75.71</td>
<td>86.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AdvHash</td>
<td>87.14</td>
<td>87.22</td>
<td>88.67</td>
<td>86.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UTAP</td>
<td><strong>32.31</strong></td>
<td><strong>31.56</strong></td>
<td><strong>31.58</strong></td>
<td><strong>86.08</strong></td>
</tr>
<tr>
<td>Original</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DHTA</td>
<td>57.50</td>
<td>51.33*</td>
<td>55.30</td>
<td>61.50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CWDM</td>
<td>84.66</td>
<td>84.39*</td>
<td>84.71</td>
<td>85.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AdvHash</td>
<td>82.31</td>
<td>39.45*</td>
<td>82.92</td>
<td>85.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UTAP</td>
<td><strong>8.16</strong></td>
<td><strong>5.33</strong></td>
<td><strong>4.98</strong></td>
<td><strong>61.50</strong></td>
</tr>
<tr>
<td>ResNet50</td>
<td></td>
<td>DHTA</td>
<td>83.36</td>
<td>79.33</td>
<td>69.68</td>
<td>86.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CWDM</td>
<td>86.86</td>
<td>86.28</td>
<td>86.04</td>
<td>86.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AdvHash</td>
<td>92.57</td>
<td>92.01</td>
<td>82.09</td>
<td>86.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UTAP</td>
<td><strong>26.95</strong></td>
<td><strong>28.63</strong></td>
<td><strong>48.70</strong></td>
<td><strong>35.53</strong></td>
</tr>
<tr>
<td>VGG16</td>
<td></td>
<td>DHTA</td>
<td>80.19</td>
<td>78.22</td>
<td>82.40</td>
<td>86.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CWDM</td>
<td>88.40</td>
<td>88.12</td>
<td>89.71</td>
<td>86.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AdvHash</td>
<td>92.70</td>
<td>92.29</td>
<td>92.59</td>
<td>86.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UTAP</td>
<td><strong>42.74</strong></td>
<td><strong>43.39</strong></td>
<td><strong>54.28</strong></td>
<td><strong>43.67</strong></td>
</tr>
<tr>
<td>VGG19</td>
<td></td>
<td>DHTA</td>
<td>81.55</td>
<td>80.16</td>
<td>82.09</td>
<td>86.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CWDM</td>
<td>88.53</td>
<td>87.63</td>
<td>88.75</td>
<td>86.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AdvHash</td>
<td>92.62</td>
<td>92.38</td>
<td>92.98</td>
<td>86.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UTAP</td>
<td><strong>61.92</strong></td>
<td><strong>62.41</strong></td>
<td><strong>65.25</strong></td>
<td><strong>43.67</strong></td>
</tr>
</tbody>
</table>

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 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

versarial 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.
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.

**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.

![Query Retrieved images](image-url)
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. 2042023k0228), and the National Key Research and Development Program of China (No. 2019QY(Y)0206).

References


