EViT: Privacy-Preserving Image Retrieval via Encrypted Vision Transformer in Cloud Computing

Qihua Feng, Peiya Li, Zhixun Lu, Chaozhuo Li, Zefan Wang, Zhiquan Liu, *Member, IEEE*, Chunhui Duan, Feiran Huang, *Member, IEEE*, Jian Weng, *Member, IEEE*, and Philip S. Yu, *Life Fellow, IEEE*

***Abstract*— Image retrieval systems help users to browse and search among extensive images in real time. With the rise of cloud computing, retrieval tasks are usually outsourced to cloud servers. However, the cloud scenario brings a daunting challenge of privacy protection as cloud servers cannot be fully trusted. To this end, image-encryption-based privacy-preserving image retrieval (PPIR) schemes have been developed, which first extract features from cipher-images, and then build retrieval models based on these features. Yet, most existing PPIR approaches extract shallow features and design trivial unsupervised retrieval models, resulting in insufficient expressiveness for the cipher-images. In this paper, we propose a novel paradigm named Encrypted Vision Transformer (EViT), which advances the discriminative representations capability of cipher-images. First, to capture comprehensive ruled information, we extract multi-level local length sequence and global Huffman-Code frequency features from the cipher-images which are encrypted by permutation encryption, sign encryption, and stream cipher during the JPEG compression process. Second, we design the modified self-supervised Vision Transformer with Huffman-embedding and propose two robust data augmentations on cipher-images to improve representation power of the retrieval model. Moreover, our proposal can be easily adapted to unsupervised or supervised settings. Extensive experiments reveal that EViT achieves both excellent encryption and retrieval performance, outperforming current schemes in terms of retrieval accuracy by large margins while protecting image privacy effectively. Code is publicly available at https://github.com/onlinehuazai/EViT.**

***Index Terms*— Image retrieval, privacy-preserving, JPEG, vision transformer, self-supervised learning.**

Manuscript received 19 June 2023; revised 22 September 2023 and 27 November 2023; accepted 21 February 2024. Date of publication 26 February 2024; date of current version 12 August 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB3103500; in part by the National Natural Science Foundation of China under Grant 62302195, Grant 62272200, Grant U22A2095, and Grant 61932010; and in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2022A1515011960. This article was recommended by Associate Editor R. Du. *(Corresponding authors: Peiya Li; Feiran Huang.)*

Qihua Feng and Chunhui Duan are with Beijing Institute of Technology, Beijing 100081, China.

Peiya Li, Zhixun Lu, Zefan Wang, Zhiquan Liu, Feiran Huang, and Jian Weng are with the College of Information Science and Technology, Jinan University, Guangzhou 510000, China (e-mail: lpy0303@jnu.edu.cn; huangfr@jnu.edu.cn).

Chaozhuo Li is with the School of Computer Science and Technology, Beihang University, Beijing 100191, China (e-mail: lichaozhuo@buaa.edu.cn).

Philip S. Yu is with the Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607 USA.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2024.3370668.

Digital Object Identifier 10.1109/TCSVT.2024.3370668

I. Introduction

IMAGE retrieval is a prominent research field within computer vision community, attracting increasing attention due to its significant research impacts and tremendous practical values [1], [2]. Given a query image, the objective of image retrieval is to search for similar images in an extensive image database. With the emergence of cloud computing, traditional local storage mode has shifted to cloud storage, which satisfies user demand for storing massive image data on the server and enables users to access the data from any location at any time. Although cloud computing contributes to alleviating the challenge of limited local storage space and provides great convenience to users, the images are in danger of privacy leakage since the cloud server cannot be fully trusted and is vulnerable to hacking [3]. To address this concern, a typical strategy is to encrypt the images prior to uploading them to the cloud server, but conventional image encryption algorithms may hinder the subsequent encrypted image data retrieval operation [4], [5]. Therefore, it is urgent to develop image-encryption-based privacy-preserving image retrieval (PPIR) technology that can provide both privacy protection and accurate retrieval simultaneously.

Existing image-encryption-based PPIR schemes have proposed various approaches for extracting features from encrypted images directly [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14] and building unsupervised or supervised retrieval model using these features. In the system model, the image owner only needs to encrypt images and upload them to the server. The task of extracting features from cipher-images can be outsourced to the server, reducing computational workload for owner and users. Although encryption protects image privacy in cloud servers, it's troublesome to extract comprehensive features and learn discriminative representations from disorganized cipher-images. Existing approaches just extract coarse-grained features or employ trivial retrieval models, resulting in limited expressiveness for the cipher-images. For instance, the works [4], [7], [8], [12], [13] extract shallow global histogram features from cipher-images, overlooking correlations of local features in encrypted images. When picking up retrieval models, the works [11], [12], [13], [14], [15], [16] build supervised-based deep learning models, but it is painstaking to assign labels to images. Schemes [3], [4], [5], [6], [7], [8] conduct unsupervised retrieval with trivial models

(e.g., K-means [17] and Bag-of-Words (BOW) [18]), but it is difficult for these trivial models to learn non-linear embedding of complex image datasets [19], whereas deep neural network (DNN) is skilled in it.

The current approaches fail to simultaneously satisfy comprehensive features and non-trivial unsupervised retrieval models based on the aforementioned limitations. To satisfy all requirements, there are two challenges in the PPIR system. First, a comprehensive multi-level feature extraction method needs to adapt to an image encryption algorithm. Explicit pixel-level features, such as edges and contours, which are readily extractable from plain images, encounter impediments in cipher-images. This limitation arises from the encryption process, wherein spatial contents undergo random alterations, rendering the extraction of pixel-level features infeasible. To ensure retrieval, existing schemes generally extract weak features that are unlike plain-images but still maintain a little connection. Relying solely on single weak features is insufficient, and extracting comprehensive multi-level features from cipher-images can compensate for this limitation, but not any encryption algorithm can extract adaptive comprehensive features. Hence, it is crucial to design a feature extraction method that is well-matched to the image encryption algorithm, serving as the bridge connecting image encryption and retrieval. Second, the well-designed retrieval model needs to couple with the extracted hand-crafted features, and some task-related modules need to be explored in cipher-images to improve retrieval performance. Additionally, to learn non-linear representations of the cipher-images without labels, we can design an unsupervised retrieval model by self-supervised contrastive learning [20], [21], [22] (SSCL) which is widely used in unsupervised computer vision task [23], [24], [25], [26]. However, SSCL generally relies on data augmentations, and unlike image data that can undergo color or rotation transforms, the structured hand-crafted features lack color information and are not invariant to rotation transform.

To tackle the initial challenge, we draw inspiration from the intriguing resilience exhibited by Variable-Length Integer (VLI) codes. Specifically, we note their consistent length preservation throughout the JPEG compression process when seamlessly integrated with a stream cipher. Our approach centers on the extraction of meticulously crafted multi-level features from cipher-images, comprising two key components: the length sequence of VLI codes associated with Discrete Cosine Transform (DCT) coefficients within each non-overlapping 8×8 block (termed local features), and the inclusion of global Huffman-Code frequency (termed global features). This multi-level feature extraction strategy can express a more comprehensive set of informative characteristics inherent to cipher-images. However, only stream cipher with VLI code is not enough in image privacy, such as the extracted weak features exhibit no discernible differences compared to plain-images. Therefore, we further encrypt the image with 8×8 block permutation encryption and sign encryption with VLI codes before cipher stream. Although permutation encryption randomly changes the comprehensive features (e.g., the orders of inter-block features), it still preserves partial similarity to that of plain-images (e.g., intra-block features). Moreover, sign encryption with VLI code does not change its bit-stream length. As a result, even the weakly comprehensive features can be utilized for retrieval purposes.

For the second challenge, we build Vision Transformer-based unsupervised retrieval model in a self-supervised learning manner. Vision Transformer (ViT) [27] divides an image into non-overlapping blocks to capture global dependency relations with self-attention mechanism [28]. Compared with convolutional neural networks (CNN), ViT excels in learning sequence relations [28] and exhibits better permutation invariance [29]. Our local block sequence features are extracted using non-overlapping 8×8 blocks, so ViT can couple with these sequence features and withstand perturbations caused by block shuffling due to its permutation invariance property. To learn discriminative representations of the cipher-images, we replace the original Cls_Token [27] of ViT with Huffman embedding by learnable global Huffman-Code frequency features. Considering there are sequence relations in the multi-level features and ViT possesses permutation invariance, two specific data augmentations, random swapping, and splicing, are proposed to adapt the SSCL framework and improve the representation ability of the retrieval model.

In this paper, we propose a novel PPIR scheme named Encrypted Vision Transformer (EViT) that can simultaneously satisfy multi-level features and task-related deep unsupervised learning. First, during the JPEG compression process, images are encrypted by shuffling encryption with 8×8 blocks, and by sign encryption and stream cipher with the VLI codes. Second, EViT extracts well-designed multi-level features from cipher-images: the length sequence of DCT coefficients' VLI code in each 8×8 block and the global Huffman-Code frequency features, which can express more plentiful information of cipher-images. Finally, EViT adopts SSCL manner to employ the unsupervised retrieval model by two kinds of adaptive data augmentations, and proposes a modified ViT-based retrieval model by adding Huffman embedding. EViT can also achieve the supervised retrieval model by easily fine-tuning the unsupervised model. The experimental results show that EViT can improve retrieval performance by large margins. The main contributions are summarized as follows:

1) To the best of our knowledge, EViT is the first to propose self-supervised learning for PPIR. To enhance the model's representation ability, EViT uses learnable global Huffman-Code frequency to modify existing ViT.
2) Ingenious multi-level features, including local length sequence and global Huffman-Code frequency, are extracted from cipher-images to express more abundant features of cipher-images. Moreover, two adaptive data augmentations on multi-level features are proposed to improve generalization performance.
3) Experimental results demonstrate that EViT outperforms other state-of-the-art schemes in retrieval performance for both unsupervised and supervised learning models. EViT can not only efficiently protect image privacy but also complete accurate image retrieval.

The rest of our paper is organized as follows. Section II presents the related work for PPIR. Preliminaries are introduced in Section III. Section IV gives the proposed scheme. Section V presents the experimental results. Finally, Section VI summarizes the conclusion.

II. Related Work

In recent years, researchers have paid more attention to PPIR and applied these techniques to boost the performance of real-life applications [30], [31], [32]. The current image encryption-based PPIR mainly can be divided into unsupervised and supervised schemes.

A. Unsupervised PPIR Schemes

This type of work built unsupervised retrieval models after image encryption and feature extraction. Zhang et al. [5] proposed a scheme with encryption of JPEG images by permuting DCT coefficients and extracting features from these coefficients' histogram invariance. Cheng et al. [9] proposed to encrypt the DC coefficients by using a stream cipher, encrypt the AC coefficients by using scrambling encryption, and conduct retrieval based on the histogram of the AC coefficients. However, Cheng et al. [9] could not ensure JPEG format compliance [4], [8]. The work [7] encrypted DC coefficients by stream cipher on the Y component and encrypted U and V components by value replacement and permutation encryption, then extracted AC histograms features of Y component and color histograms features of U and V components. Liang et al. [8] extracted Huffman-Code histograms from cipher-images which were encrypted by stream cipher and permutation encryption. Li et al. [4] proposed a new block transform encryption method using orthogonal transforms rather than 8×8 DCT. The above works [4], [5], [7], [8], [9] directly calculated distances of extracted features when retrieving, which failed to mine these features by machine learning (ML). Xia et al. [3] extracted secure Local Binary Pattern (LBP) features from cipher-images, then built a BOW model to conduct retrieval. The work [6] encrypted images by color value substitution and permutation encryption, and extracted local color histogram features from cipher-images, then they built an unsupervised bag-of-encrypted-words (BOEW) model to achieve retrieval. Although the works [3], [6] utilized ML to build unsupervised retrieval models, their trivial models (e.g., BOW) failed to learn non-linear embeddings from the encrypted images.

B. Supervised PPIR Schemes

Some works used supervised models to learn representations of encrypted images, which bring extra label overhead. For instance, Cheng et al. [11] used stream cipher and permutation encryption to encrypt JPEG images, and extracted features from cipher-images by Markov process, then built a supervised support vector machine (SVM) model to conduct retrieval. The works [14], [15], [16] conducted end-to-end learning to alleviate the burden of hand-crafted features, but the retrieval accuracy is unsatisfactory because cipher-images are too disordered to learn effective representations [12]. Lu et al. [13] and Feng et al. [12] extracted Huffman-Code and DCT histogram features from cipher-images respectively, and then utilized attention networks to build retrieval models by these histogram features. EViT can be easily transformed into the supervised model by fine-tuning the unsupervised retrieval model.

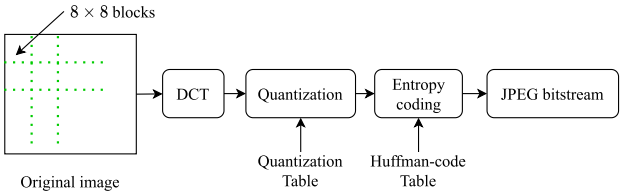


Fig. 1. Overview of JPEG compression process.

Existing unsupervised and supervised PPIR schemes typically extract shallow features from cipher-images, such as histogram features [4], [5], [6], [7], [8], [9], [12], [13]. These features provide limited information about the cipher-images and may not deliver more comprehensive information from cipher-images. Additionally, current unsupervised schemes often rely on trivial retrieval models like BOW that struggle to learn non-linear embeddings. In contrast, EViT introduces a novel approach by extracting multi-level features from cipher-images and leveraging modified ViT for unsupervised retrieval in a self-supervised manner, resulting in significantly improved retrieval performance compared to existing approaches.

III. Preliminaries

A. JPEG Compression

We encrypt images during the JPEG compression process like some privacy-preserving image retrieval schemes [4], [5], [7], [8], [9], [12], [13], [33]. Here, we briefly introduce the JPEG compression process, whose overview is shown in Fig. 1. According to the JPEG compression standard [34], [35], images are converted to YUV color space. After 8×8 DCT and quantization, each block has a total of 64 coefficients, of which the first coefficient is the direct current (DC) coefficient, and the remaining 63 coefficients are the AC coefficients. The DC coefficient is encoded by differential pulse code modulation (DPCM), and the remaining 63 AC coefficients in the same block are converted into a sequence using zig-zag scanning. AC coefficients are encoded with run-length encoding (RLE), which are converted to the (r, v) pairs [34], [35].

The lossless Huffman variable-length entropy coding technology is used to further compress the DC differential (ΔDC) and AC coefficients (r, v) pairs, and all coefficients are coded to the binary sequence. Specifically, each ΔDC is encoded into two parts: DC Huffman code (DCH) and DC VLI code (DCV); each (r, v) pair is encoded as two parts: AC Huffman code (ACH) and AC VLI code (ACV). As shown in Fig. 2, we take an example for lossless Huffman variable-length entropy coding, where the category is the range of amplitudes of coefficients [34], and r corresponds to the run value in the AC Human table [34]. For example, when $\Delta DC = 2$, it's $DCH = 011$ and $DCV = 10$, so it is coded into 01110; when $(r, v) = (0, 6)$, it's $ACH = 100$ and $ACV = 110$, therefore it is coded into 100110. Each coefficient can be coded into

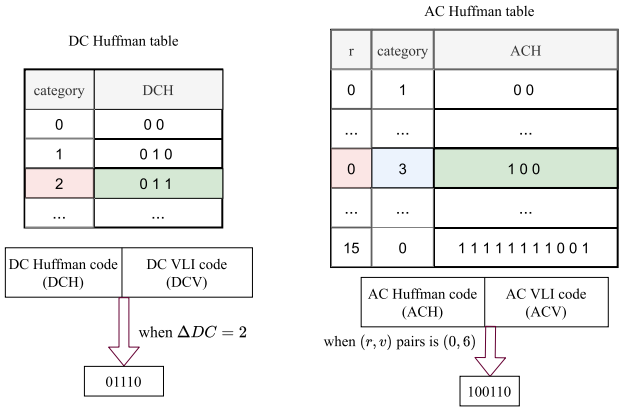


Fig. 2. An example of lossless Huffman variable-length entropy coding.

binary sequence through Huffman-Code table mapping, for more details please refer to [34] and [35].

B. Unsupervised Contrastive Learning

Supervised learning has been widely used in many image retrieval tasks [36], [37], [38], but it is painstaking to assign labels to images. To decrease the overhead with human annotations, unsupervised algorithms are explored by researchers, such as K-means [17] and BOW [18] model. Both K-means and BOW utilize the idea of clustering, but in many image databases, it is ineffective to learn images' representations because clustering with Euclidean distance cannot learn non-linear embedding [19]. Deep neural network (DNN) is competent in learning non-linear embedding which is indispensable for complex image databases. Supervised learning based on DNN generally uses target labels to train the model, but it is painstaking to assign labels to images. Self-supervised learning can build DNN model without target labels, which generates virtually unlimited labels from existing images and uses those to learn the representations. In recent years, many self-supervised learning methods have been proposed [20], [21], [23], [39], [40], [41]. Due to its simple and effective property, self-supervised contrastive learning [20], [21], [23], [42] has been used in many unsupervised vision tasks [24], [25], [26], [43], [44], [45].

SimCLR [20] is a popular self-supervised contrastive learning method, whose process is roughly illustrated as follows: given an image x, we can obtain different images x_i and x_j by stochastic data augmentations such as random color distortions and random Gaussian blur; then the two images through same encoder can generate corresponding embeddings z_i and z_j, and SimCLR can learn representations of images by maximizing the similarity z_i and z_j. SimCLR obtains the positive samples of x by data augmentations, the other images are negative samples. But SimCLR needs a very large batch size to build negative samples when training, this is expensive for most researchers to use GPU with large memory. Therefore MoCo [21], [22] proposed momentum contrast to solve the problem of large batch size by building a dynamic dictionary with a queue and momentum updating. In this paper, we utilize self-supervised contrastive learning based on MoCo to build our unsupervised-learning retrieval model.

C. Vision Transformer

Since the proposal of Google's Transformer [28] in 2017, it has almost dominated natural language processing tasks [46], [47], [48]. Transformer is a sequence model based on self-attention mechanism, which can capture the global dependencies between words. In 2021, Google proposed to apply Transformer in computer vision [27], called Vision Transformer (ViT). Then many researchers explore ViT and find ViT can obtain excellent performance in many computer vision tasks [49], [50], [51], [52], [53], [54], [55], even surpass the CNN model in performance [44], [56], [57], [58], [59]. Vision Transformer not only makes breakthroughs in computer vision but also forms the unified model for computer vision and natural language processing.

ViT [27] divides images into non-overlapping blocks, and each block is equal to a word in natural language processing. Suppose image's size is $H \times W$, and the size of each block is $P \times P$, hence the number of blocks is $\frac{H \times W}{P^2}$. Assuming that the dimension of word embedding is D, ViT uses linear projection [27] to map each block to dimension D, called block embedding. Like the $[class]token$ in BERT [46], ViT prepends a learnable embedding Cls_Token in block embeddings. The goal of Cls_Token is to learn the representation of an image, and ViT initializes Cls_Token with ones (dimension D). The results of adding the block and position embeddings [27], [28] are then used as the input to the Transformer encoder. The Transformer encoder of ViT is similar to standard Transformer [28], includes self-attention [28], layer normalization [60] and residual module [61]. Self-attention is an important part of the Transformer encoder. When calculating self-attention, three matrices are needed: query (Q), key (K), and value (V). Suppose the input of self-attention is X, and the definition of self-attention is expressed as:

$$Q = W_Q X, \quad K = W_K X, \quad V = W_V X, \tag{1}$$

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V, \tag{2}$$

where W_Q, W_K, W_V are linear projection matrices, and d_k is the dimension of K. In order to capture more information, Transformer uses multi-head self-attention to learn different subspaces [28]. Multi-head self-attention uses h different linear projections to map Q, K, V, and then concatenates different results of self-attention and does a linear projection. The definition of multi-head self-attention is as follows:

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h)W^O, \tag{3}$$

$$head_i = Attention(Q_i, K_i, V_i), \quad i \in [1, h] \tag{4}$$

where W^O is linear projection matrices.

In this paper, we use ViT as the backbone in the retrieval model, and the reasons are as follows: a) The local block

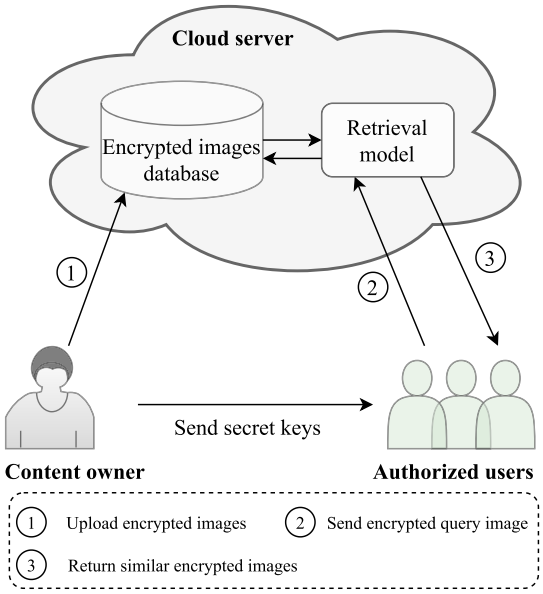


Fig. 3. The system model of image-encryption-based PPIR.

features are shuffled during encryption, and ViT owns better permutation invariance than CNN [29]. b) ViT divides an image into non-overlapping block sequences, and we extract sequence features from cipher-images' 8×8 blocks, which aligns well with ViT's input structure. c) ViT can learn global sequence dependency relations with self-attention mechanism [28] compared with local CNN [29]. We also compare the retrieval performance with CNN in Section V, and the experimental results show that ViT achieves better performance in our work.

IV. Proposed Scheme

Generally, the system of image-encryption-based PPIR includes three parts: content owner, server, and authorized users. As shown in Fig. 3, the content owner first encrypts images and uploads encrypted images to the server. Then authorized users encrypt the query image with the same encryption algorithm and upload it to the server. The server extracts features from the query image, and the retrieval model searches similar cipher-images in the encrypted image database according to the features. Finally, these similar cipher-images are returned to authorized users, and authorized users decrypt images with encryption keys which are shared by the content owner through a secure channel [8], [11]. The design of this system involves the development of three modules: image encryption algorithm, feature extraction method, and image retrieval model. We specify these modules of EViT in detail below.

A. Image Encryption

To ensure effective retrieval, it is imperative to extract ruled weak features from cipher-images. However, the encryption of spatial content, including pixel-level information, poses challenges for feature extraction. Inspired by encryption techniques employed in JPEG compression [5], [12], [62], which simultaneously safeguards visual spatial content while preserving weak features, we encrypt images during the JPEG compression process. Section III provides a concise overview of the JPEG compression process. The encryption involves three steps: 8×8 block shuffling, sign encryption, and stream exclusive-OR (XOR) with DCT coefficients.

Algorithm 1 Encryption Algorithm

input : Plain-image I, secret keys k^*_{BP}, k^*_{ODC}, k^*_{OAC}, k^*_{DC}, k^*_{AC}, $* \in \{Y, U, V\}$
output: Encrypted JPEG bitstream

Convert I from RGB to YUV color space;
Denote the width and height of the image I as W and H ;
for I_i *in* I, $i \in \{Y, U, V\}$ **do**
  Divide I_i into several 8×8 non-overlapping blocks;
  Conduct block permutation of component I_i with secret key k^i_{BP}, and the shuffled blocks of I_i are denoted as B^i_j, where $j \in [1, \ldots, blknum]$ and $blknum = \frac{W \times H}{8 \times 8}$;
  for $j = 1$ *to* $blknum$ **do**
    // Encrypt the VLI codes of B^i_j
    Denote $k^{B^i_j}_{ODC}$ as corresponding positional key bit of block B^i_j in k^i_{ODC};
    if $k^{B^i_j}_{ODC} = 1$ **then**
      $DCV'_{B^i_j} = -DCV_{B^i_j}$;
    end
    for $t = 1$ *to* 63 **do**
      Denote $k^{B^i_j}_{OAC_t}$ as corresponding positional key bit of t-th AC of block B^i_j in k^i_{OAC};
      if $k^{B^i_j}_{OAC_t} = 1$ **then**
        $ACV^{t'}_{B^i_j} = -ACV^{t}_{B^i_j}$;
      end
    end
    $ACV'_{B^i_j} = \{ACV^{t'}_{B^i_j}, t = 1, \ldots, 63\}$;
    Denote $k^{B^i_j}_{DC}$ and $k^{B^i_j}_{AC}$ as corresponding positional stream keys of B^i_j in k^i_{DC} and k^i_{AC}, respectively ;
    $DCV''_{B^i_j} \leftarrow DCV'_{B^i_j} \oplus k^{B^i_j}_{DC}$;
    $ACV''_{B^i_j} \leftarrow ACV'_{B^i_j} \oplus k^{B^i_j}_{AC}$;
  end
end

EViT first shuffles 8×8 image blocks in the stage of entropy coding by secret key k_{BP}, which is fundamental and common encryption in PPIR [4], [8], [11], [12], [13], [16], [33], [63], [64]. Given a plain-image I, the width and height of the image I are W and H, respectively. The JPEG

compression process requires converting I from RGB to YUV color spaces, namely $I = \{I_Y, I_U, I_V\}$, and dividing $I_i (i \in \{Y, U, V\})$ into 8×8 non-overlapping blocks. The secret key k^i_{BP} generates scrambling sequences [12], [13], [15] to shuffle the corresponding image component (I_i) blocks. We denote the shuffled blocks of component I_i as $B^i = \{B^i_j\}$, where $j \in [1, \ldots, blknum]$ and $blknum = \frac{W \times H}{8 \times 8}$. Following the block shuffling, we use sign encryption to change the VLI codes of DC and AC coefficients to opposite numbers with the key k_{ODC} and k_{OAC}, respectively. The sign encryption keys of component I_i are k^i_{ODC} and k^i_{OAC}, and using a key bit determines one coefficient's sign. For example, when the key bit is '1', the coefficient is encrypted to the opposite number, otherwise unchanged. A block B^i_j has 64 coefficients, one DC coefficient and 63 AC coefficients. Suppose the corresponding positional key bit of block B^i_j in secret key k^i_{ODC} is $k^{B^i_j}_{ODC}$, and the corresponding key bit of t-th AC of block B^i_j in secret key k^i_{OAC} is $k^{B^i_j}_{OAC_t}$ $(t \in \{1, \ldots, 63\})$, so the sign encryption of a block B^i_j can be defined as:

$$DCV'_{B^i_j} = \begin{cases} -DCV_{B^i_j}, & k^{B^i_j}_{ODC} = 1 \\ DCV_{B^i_j}, & k^{B^i_j}_{ODC} = 0 \end{cases} \tag{5}$$

$$ACV'^{t'}_{B^i_j} = \begin{cases} -ACV^t_{B^i_j}, & k^{B^i_j}_{OAC_t} = 1, t = 1, \ldots, 63 \\ ACV^t_{B^i_j}, & k^{B^i_j}_{OAC_t} = 0, t = 1, \ldots, 63 \end{cases} \tag{6}$$

where $DCV_{B^i_j}$ is the original DCV in block B^i_j, and $DCV'_{B^i_j}$ is encrypted DCV after the second encryption step. $ACV'_{B^i_j}$ is encrypted ACVs in block B^i_j, where $ACV'_{B^i_j} = \{ACV^{1'}_{B^i_j}, \ldots, ACV^{t'}_{B^i_j}, \ldots, ACV^{63'}_{B^i_j}\}$. Finally, a stream XOR operator is performed to encrypt VLI codes of DC and AC coefficients [65]. The corresponding keys are k_{DC} and k_{AC}, and different color spaces (Y/U/V spaces) have different secret keys (k^i_{DC}, k^i_{AC}). For the block B^i_j, the $k^{B^i_j}_{DC}$ and $k^{B^i_j}_{AC}$ are corresponding positional stream keys in k^i_{DC} and k^i_{AC}, respectively, and we further encrypt $DCV'_{B^i_j}$ and $ACV'_{B^i_j}$ as:

$$DCV''_{B^i_j} \leftarrow DCV'_{B^i_j} \oplus k^{B^i_j}_{DC}, \tag{7}$$

$$ACV''_{B^i_j} \leftarrow ACV'_{B^i_j} \oplus k^{B^i_j}_{AC}, \tag{8}$$

where $DCV''_{B^i_j}$ and $ACV''_{B^i_j}$ are encrypted VLI codes after the third step, and $\oplus$ is exclusive-or operator. It is noted that DCH and ACH cannot be XORed, as this would destroy the format information of the JPEG image and lead to decoding failure [35]. Our pseudo-random secret keys $(k^*_{BP}, k^*_{ODC}, k^*_{OAC}, k^*_{DC}, k^*_{AC}, * \in \{Y, U, V\})$ are generated by hash function BLAKE2 [66] which is widely used in many encryption algorithms [4], [12], [13], [16], [67]. During the encryption process, original image I can be compressed to an encrypted JPEG bitstream, and different color spaces (Y/U/V spaces) of I have different secret keys. In Algorithm 1, the encryption algorithm of EViT is presented. The encrypted bitstream is decodable because the file structure is JPEG-compliant.

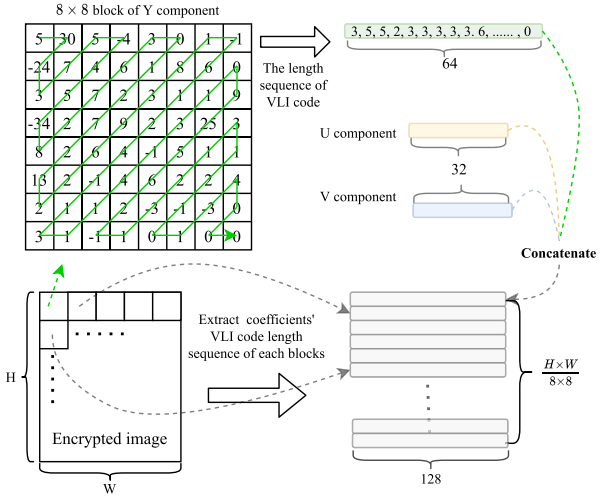


Fig. 4. Sketch of extracting local features from our encrypted images.

Current image-encryption-based PPIR schemes hardly ensure absolute security because an absolutely secure encryption algorithm cannot make us extract ruled features from cipher-images. For example, Zhang [5] leaked DCT histograms of cipher-images, and the works [3], [6], [8], [12], [13], [14], [15], [16] fail to resist differential attack due to they use the same keys to encrypt all images. One hopes cipher-images should not be easily hacked or accessed in the system, and secret keys can be changed periodically. Similarly, our encryption algorithm does not claim absolute security. However, by employing block shuffling, sign encryption, and stream XOR, we can achieve a certain level of image security. These techniques enable EViT to support multi-level feature extraction, as discussed in the following sections. Therefore, like existing PPIR schemes, our encryption approach aims to strike a balance between encryption strength and retrieval performance.

B. Feature Extraction

Image-encryption-based schemes extract features directly from cipher-images. Existing schemes just extract shallow features (e.g. DCT histogram), which are unable to express plentiful information of cipher-images. EViT extracts multi-level features from the cipher-images: local length sequence and global Huffman-Code frequency features.

1) Local Length Sequence Features: EViT extracts each 8×8 block's local features. As shown in Fig. 4, EViT calculates the corresponding encrypted VLI code's length of DCT coefficients and builds length sequence features in each 8×8 block. For example, when $\Delta DC = 5$, its VLI code is '101', hence the length is 3. The length sequence features are generated by zig-zag scanning [34]. If the coefficient is zero, then EViT denotes its length as zero. Because each block has three components (Y/U/V), EViT concatenates three length sequence features of different components. It's noted that there are many zeros of U and V components in the back

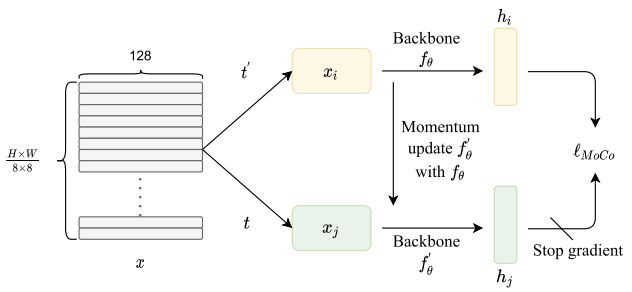


Fig. 5. Overview of the unsupervised-learning retrieval model.

coefficients [35]. To decrease computational overhead and sparse zero values, EViT just chooses the top 32 coefficients in the sequence when extracting length sequence features for U and V components.

2) Global Huffman-Code Frequency Features: EViT extracts global Huffman-Code frequency features from the cipher-images. We take a simple example to describe Huffman-Code frequency features, as shown in Fig. 2. If the second row of DC Huffman table is used 10 times during the entropy coding stage for an encrypted image, the corresponding Huffman-Code frequency feature is denoted as 10. The rows of DC Huffman table are 12, and the rows of AC Huffman table are 162 [34]. Therefore, the dimension of Huffman-Code frequency features is $(12 + 162) \times 3 = 522$, where 3 represents three components.

Global features and local features are extracted from cipher-images, which are employed to train the proposed retrieval model. Stream XOR does not change the length of VLI codes. For example, suppose the DCV is '101', and the encryption key is '100', then the encrypted DCV is '001', and the lengths all are 3. Similarly, sign encryption with VLI codes also does not change its length, it just takes the inverse. The DCH and ACH are unchanged with sign encryption and stream XOR on VLI codes, so Huffman-Code histograms are extracted as the weak features. Although the inter-block order is shuffled, the intra-block length of VLI is unchanged, which can be utilized as a weak feature to offer retrieval. Like most PPIR schemes [3], [6], [8], [12], [13], [14], [15], [16], to keep the unified feature space, EViT needs to use the same permutation keys to shuffle all images.

C. Unsupervised-Learning Retrieval Model

After extracting features from encrypted images, EViT uses these features to train retrieval models. EViT proposes unsupervised-learning and supervised learning retrieval models based on deep learning. Deep image retrieval is a typical deep metric learning [1] whose aim is to learn the representations of images. Given an image, we first use learnable deep neural networks $f(\cdot)$ to learn its representation h, and $f(\cdot)$ is generally called backbone. Then the images' representations are used to calculate the similarity such as cosine or Euclidean distances between the query and targets.

1) Loss: The unsupervised framework uses MoCo (see Section III) due to its simple and effective property [20], [21], [23], which does not use target labels to learn the representations of cipher-images. As shown in Fig. 5, given a cipher-image, EViT extracts features from it and denotes these features as x. Using random data augmentations t' and t, we can obtain x_i and x_j respectively. Through backbone f_θ, EViT can learn the representation h_i of x_i. For x_j, the process is like x_i, but the backbone f'_θ is stop-gradient which is updated by momentum with f_θ [21]. The process of forward propagation can be defined as:

$$x_i = t'(x), \quad x_j = t(x), \tag{9}$$

$$h_i = f_\theta(x_i), \quad h_j = f'_\theta(x_j). \tag{10}$$

The backbone f'_θ is updated with momentum manner [21] which is described as:

$$f'_\theta = m f'_\theta + (1 - m) f_\theta, \tag{11}$$

where m is momentum factor and is set to be 0.99 following MoCo [21]. The structures of f'_θ and f_θ are same, but with different parameters.

The loss function is like MoCo which is called InfoNCE [68]. MoCo proposed momentum contrast to solve the problem of large batch size by building a dynamic dictionary with a queue and momentum updating. The dynamic dictionary is a queue where the current batch enqueued and the oldest batch dequeued. For one sample x_i in the current batch, the x_j is positive sample, and other samples in the current batch and the queue are negative samples. The loss can be defined as:

$$\ell = -log \frac{exp(h_i \cdot h_j/\tau)}{exp(h_i \cdot h_j/\tau) + \sum_{k^-} exp(h_i \cdot h_{k^-}/\tau)}, \tag{12}$$

where τ is a temperature hyper-parameter proposed by [42], which we set to be 0.1 like MoCo. h_i and h_j are representations of x_i and x_j respectively (Fig. 5). h_{k^-} are representations of negative samples, and "$\cdot$" is dot product.

2) Backbone: Our backbone f_θ adopts the structure of ViT (see Section III). EViT extracts two parts features from encrypted images, local length sequence features and global Huffman-Code frequency features. As shown in Fig. 6, suppose the number of blocks of a cipher-image is $\frac{H\times W}{8\times 8}$ (H is height, W is width). These local features, through linear projection [27], produce corresponding block embeddings. Original Cls_Token (mentioned in Section III) of ViT are all ones for each image, which fails to express specific information for different images. Hence, different from standard ViT, EViT uses Huffman embedding to replace Cls_Token, which is helpful for retrieval performance in experiments. The Huffman embedding (He) is learned from global Huffman-Code frequency features ($gHff$), which can be defined as:

$$He = FC(\sigma(LN(FC(gHff)))), \tag{13}$$

where FC is fully-connected layer, LN is layer normalization [60], σ is activation function ReLU [69]. In order to keep position information, we also add position embedding [27] with block embedding and Huffman embedding. The result of these embeddings is denoted as v_0, then through L stacked Transformer encoder [27], EViT can learn the representations v_L^0 of cipher-image. The l-th Transformer encoder can be

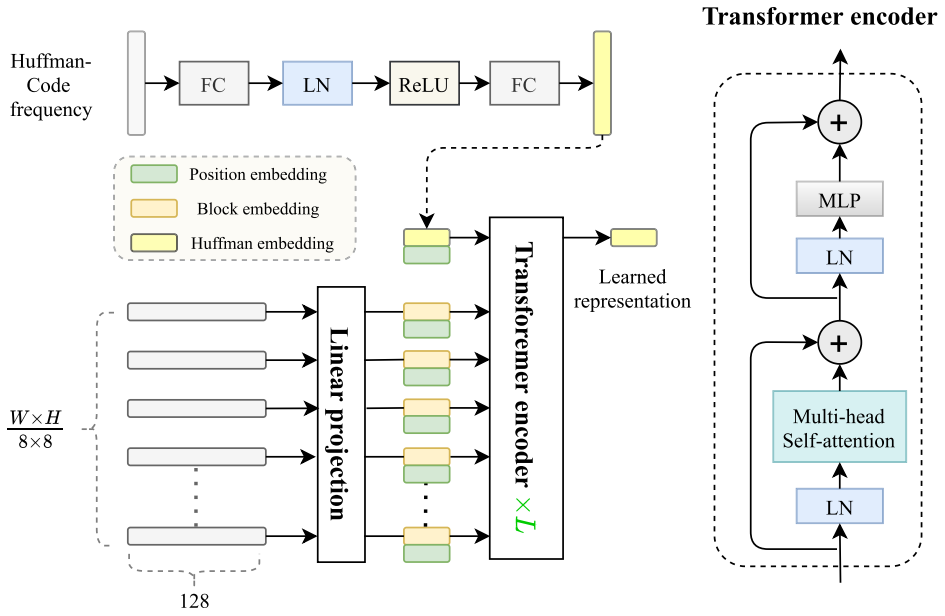


Fig. 6. Overview of our backbone.

defined as:

$$v_l' = MSA(LN(v_{l-1})) + v_{l-1}, \quad l = 1, 2, \ldots L \tag{14}$$

$$v_l = MLP(LN(v_l')) + v_l, \quad l = 1, 2, \ldots L \tag{15}$$

where MSA is multi-head self-attention [28] (Eq. 3), MLP is multi-layer perceptron block [27]. The output of L stacked is v_L, and the representation is the first embedding v_L^0.

ViT uses spatial pixels to build a block of plain-images, but spatial pixels are randomly encrypted in cipher-images. EViT uses length sequence feature to replace spatial pixels in a block. Different from plain-image retrieval which uses pre-trained ViT on ImageNet [70] as model's backbone, our task is specific retrieval on cipher-images and there is no pre-trained model as the backbone, so the retrieval model needs to be trained from scratch. Generally, small learning rate and warm-up [61] are necessary for training a new model with the structure of ViT [71]. EViT uses cosine warm-up [72] with learning rate, and the experimental results also present that it is helpful for retrieval performance.

3) Data Augmentation: EViT incorporates random data augmentations t and t' for the encrypted features in its unsupervised-learning retrieval model (Fig. 5). Cipher-images contain substantial noise and undergo complete random alterations and distortions during encryption. Traditional plain-image data augmentations (e.g., random cropping) focus on pixel-level image content, which is unsuitable for our cipher-images. Consequently, EViT directly applies data augmentations to the length sequence features that extracted from the encrypted images.

In the retrieval model, we propose two adaptive data augmentations on the length sequence features: random swapping and splicing. Fig. 7 illustrates examples of these augmentations. The motivation behind these approaches stems from two observations: a) the length sequence features can be likened to words in a sentence, where swapping two words

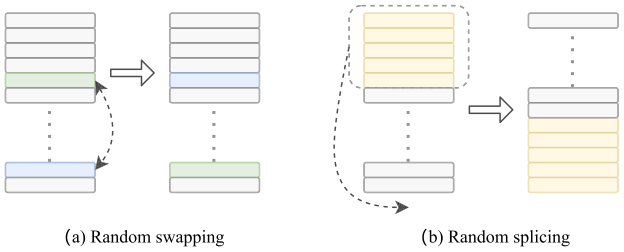


Fig. 7. Examples of the data augmentations on length sequence features.

is a common augmentation technique in natural language processing [73]; b) ViT has demonstrated decent performance with simple block permutation [29], thus EViT shuffles the length sequence features using random block feature splicing. Additionally, the model benefits from random dropout [74] as an additional form of data augmentation [75]. Through applying random dropout twice to an image during training, different embeddings can be obtained. ViT itself incorporates a random dropout function, and EViT adopts the same dropout setting as ViT for its backbone. Experimental results confirm the crucial role of these two adaptive data augmentations in significantly improving retrieval performance.

D. Supervised-Learning Retrieval Model

EViT also provides a simple supervised-learning retrieval model, which uses the same structures of backbone as the unsupervised-learning model. The supervised model can obtain representations h of cipher-images after backbone f_θ (Fig. 6). Here we can use the pre-trained backbone from unsupervised-learning model. The supervised loss function ArcFace [76] is used to train the model. The supervised model is defined as:

$$h = f_\theta(x), \quad h' = l_2(h), \tag{16}$$

where l_2 is short for l_2 normalisation [76].

Algorithm 2 EViT's Main Algorithm

input: plain-images, secret keys, labels=None

```
// first module: image encryption
algorithm (Section IV-A) ;
cipher-images ← Image encryption (plain-images,
 secret keys) ;
// second module: feature extraction
method (Section IV-B) ;
features ← feature extraction (cipher-images) ;
// third module: retrieval models ;
unsupervised ← retrieval model(features)
 // Section IV-C;
if labels is Not None then
    supervised ← Fine-Tuning (unsupervised,
     features, labels) // Section IV-D;
    return supervised
else
    return unsupervised
end
```

ArcFace is a common deep metric learning loss function, which has been widely used in retrieval tasks [77], [78], [79]. Compared with Triplet loss [80], ArcFace is more easy and effective [76] which gets rid of the disadvantages such as hard sample mining and combinatorial explosion in the number of triplets. ArcFace adds an additive angular margin within softmax loss [76], which can be defined as:

$$\ell_{ArcFace} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i}+\alpha))}}{e^{s(\cos(\theta_{y_i}+\alpha))}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}}, \tag{17}$$

where N and n are the batch size and the class number, and θ_j is the angle between the representation of i-th sample and j-th class center. y_i is ground-truth of i-th sample, and θ_{y_i} represents the angle between i-th sample and ground-truth class center. s and α are hyper-parameters that represent feature re-scale and angular margin parameters, respectively. For more details please refer to [76].

In the inference stage, we just need to calculate cosine distances of the representations h of cipher-images, and then rank these distances and return top-K results. On the basis of unsupervised-learning model, the supervised-learning model seems straightforward, and we will further explore it in the future such as combining Triplet loss [80] and Center loss [81] to learn more discriminative representations. As shown in Algorithm 2, we present the main processes of EViT.

V. Experiments and Analysis

In this section, the experimental results of the proposed method are presented. We evaluate the performance on Corel10K [82] dataset, which is widely used by many related researches. Corel10K dataset contains 10000 images in 100 categories, with 100 images in each category. The programming language is Python. In the following section, we first describe the retrieval performance, then present the encryption performance of EViT.

TABLE I
Descriptions of Datasets

Datasets	Corel10K-a	Corel10K-b
Training set	7000	7000
Testing set	3000	3000
Classes of Training set	70	100
Classes of Testing set	30	100
Type	Open-set	Close-set

A. Retrieval Performance

EViT respectively proposes the unsupervised-learning and supervised-learning retrieval models. We compare the retrieval performance of EViT with current image-encryption-based schemes whose retrieval models are divided into unsupervised and supervised models. In order to better compare retrieval performance, we split training set and testing set into two different types: open-set and close-set [76]. Open-set dataset means that the testing set has no same classes as the training set, and the close-set dataset is on the contrary. Specifically, training and testing sets are divided according to the ratio of 7 : 3, and Corel10K dataset is transformed into Corel10K-a and Corel10K-b datasets. For Corel10K-a dataset, we train the retrieval model on 70 classes, so the testing set has no same classes as training set (open-set classification). For Corel10K-b, we train the retrieval model on 100 classes, and for each class, we select 70 images, and the remaining 30 images of each class are in the testing set (close-set classification). The descriptions of the two datasets are shown in Tab. I, Corel10K-a dataset is open-set and Corel10K-b dataset is close-set. We use stochastic gradient descent (SGD) as an optimizer. The weight decay is $5e^{-5}$, and SGD momentum is 0.9. We set batch size to be 32 and 64 for the unsupervised-leaning and supervised-learning models respectively. The retrieval models are trained by the PyTorch framework [83] on a machine with NVIDIA Tesla T4 16G GPU.

The evaluation metric of retrieval performance we use is mean Average Precision (mAP) which is widely used in many retrieval tasks. When returning top-K results, mAP is calculated as follows:

$$mAP@K = \frac{1}{Q}\sum_{q=1}^{Q}AP@K(q), \tag{18}$$

$$AP@K(q) = \frac{1}{R_q}\sum_{k=1}^{K}p_q(k)\ rel_q(k), \tag{19}$$

where Q is the number of query images, R_q is the number of similar images for the query q, $p_q(k)$ is precision at rank k for the query q, and $rel_q(k)$ is 1 if the rank k result is similar to q, 0 otherwise. In this paper, we use $mAP@100$ to evaluate retrieval performance. The higher $mAP@100$, the better retrieval performance.

TABLE II
COMPARISONS OF RETRIEVAL PERFORMANCE WITH CURRENT SCHEMES

Schemes	Corel10K-a	Corel10K-b	Unsupervised/Supervised
Xia [7]	0.378	0.230	Unsupervised
Liang [8]	0.321	0.217	Unsupervised
Xia [3]	0.383	0.235	Unsupervised
Xia [6]	0.301	0.206	Unsupervised
Zhang [5]	0.396	0.269	Unsupervised
Li [4]	0.410	0.269	Unsupervised
EViT	**0.476**	**0.323**	Unsupervised
Cheng [11]	—	0.407	Supervised
Feng [15]	0.423	0.528	Supervised
Lu [16]	0.327	0.412	Supervised
Ma [14]	0.201	0.299	Supervised
Feng [12]	0.513	0.625	Supervised
EViT	**0.568**	**0.750**	Supervised

TABLE III
UNSUPERVISED LEARNING RETRIEVAL PERFORMANCE WITH DIFFERENT L VALUES ON COREL10K-A AND COREL10K-B DATASETS

L	4	5	6	7
Corel10K-a (mAP@100)	0.474	0.476	0.473	0.471
Corel10K-b (mAP@100)	0.319	0.323	0.320	0.317

TABLE IV
ABLATION EXPERIMENTS WITH DATA AUGMENTATIONS, WARM-UP, AND HUFFMAN EMBEDDING FOR UNSUPERVISED-LEARNING RETRIEVAL MODEL

Data augmentations		✓	✓	✓
Warm up			✓	✓
Huffman embedding				✓
Corel10K-a (mAP@100)	0.405	0.419	0.431	0.476
Corel10K-b (mAP@100)	0.268	0.282	0.301	0.323

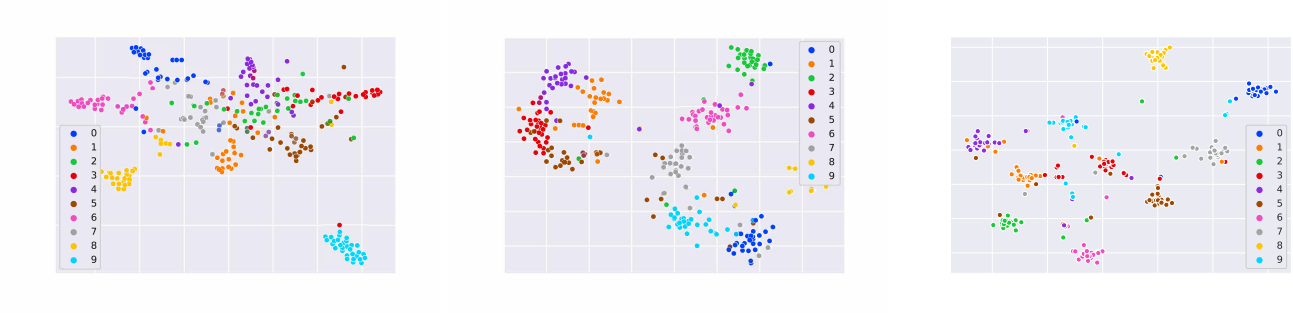
Here, we compare retrieval performance with current image-encryption-based schemes. All schemes are evaluated on the same testing set, and the results of comparison are shown in Tab. II. We can see that our retrieval performance is better than other schemes. Specifically, the unsupervised-learning model can achieve 0.476 $mAP@100$, which is higher about 6.6% than state-of-the-art retrieval performance on the open-set Corel10K-a; on the closed-set Corel10K-b, our unsupervised-learning model can achieve 0.323 $mAP@100$, which is higher about 5.4% than state-of-the-art retrieval performance. For supervised-learning model, we can significantly improve retrieval performance than other supervised schemes. The works [14], [15], [16] are end-to-end learning rather than extracting manual features, which can automatically extract features from cipher images by DNN models. However, encrypted images are extremely disordered so models are unable to learn their effective representations, leading to a negative impact on retrieval performance. It is noted that Cheng [11] used classification probability as representations of cipher-images, it is unsuitable on open-set Corel10K-a since testing set has no same classes as training set. Next, we describe more experimental details about the unsupervised-leaning and supervised-learning retrieval performance respectively.

1) Unsupervised Retrieval Performance: The backbone has L stacked Transformer encoder (Fig. 6), we use different L values to demonstrate the retrieval performances on Corel10K-a and Corel10K-b datasets. As shown in Tab. III, we can see that $L = 5$ is more suitable for our unsupervised-learning retrieval model.

We propose two adaptive data augmentations for EViT, and have mentioned that warm-up and huffman embedding are helpful for retrieval performance (Section IV-C.2). Here, we use ablation experiments to verify how data augmentations, warm-up, and huffman embedding influence the unsupervised-learning retrieval performance on Corel10K-a and Corel10K-b datasets. The learning rate is $1e^{-3}$ with cosine warm-up, as shown in Fig. 9, the "blue" line is cosine learning rate with warm-up which is linearly increased to $1e^{-3}$ in the first 20 epoch; the "red" line is cosine learning rate without warm-up. We add warm-up and huffman embedding one by one, the results of ablation experiments are shown in Tab. IV. We can see that if we do not add data augmentations, warm-up, and huffman embedding, the retrieval performance only can achieve 0.405 and 0.268 $mAP@100$ on Corel10K-a and Corel10K-b respectively. The two adaptive data augmentations are helpful to enhance retrieval performance, which can improve more than 1% $mAP@100$. Warm-up improves retrieval performance with 1.2% and 1.9% on Corel10K-a and Corel10K-b respectively. Huffman embedding significantly improves retrieval performance with 4.5% and 2.2% on Corel10K-a and Corel10K-b respectively. The Huffman embedding is learned from global Huffman-Code frequency which is one of the multi-level features. The ablation experiments prove that multi-level features express more abundant information of cipher-images, which can directly improve retrieval performance.

2) Supervised Retrieval Performance: Supervised model fine tunes on the unsupervised model. For example, when training the supervised-learning model on Corel10K-a dataset, the backbone can use unsupervised-learning model's parameters which are also trained on Corel10K-a as initial parameters. Unsupervised-learning model as pre-trained model for supervised-learning is common [20], [21], [24] which can accelerate model convergence and achieve better performance. Due to the backbone of supervised-learning being the same as the unsupervised-learning model, most strategies such as warm-up and data augmentations also are used in the supervised-learning model. Apart from loss function, the supervised-learning model is almost inspired by our unsupervised-learning model.

We mentioned that there are two hyper-parameters in ArcFace: s and α (Eq. 17), and now we use different s and α to observe their influence on retrieval performance on Corel10K-a and Corel10K-b (Corel10K-a/b) datasets.



(a) Cheng [11] (b) Feng [15] (c) Ours

Fig. 8. Comparison of semantic space visualization with t-SNE on Corel10K-b dataset.

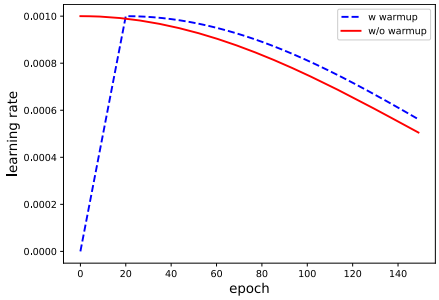


Fig. 9. Comparison of learning rate schedules.

TABLE V
RETRIEVAL PERFORMANCE WITH DIFFERENT α AND s ON COREL10K-A/B DATASETS

α \ s	8	16	32	64
0.1	0.551 / 0.726	**0.568** / 0.732	0.561 / 0.733	0.553 / 0.729
0.2	0.549 / 0.731	0.562 / 0.737	0.560 / 0.738	0.551 / 0.747
0.3	0.547 / 0.738	0.558 / 0.741	0.556 / 0.745	0.554 / **0.750**
0.4	0.544 / 0.747	0.557 / 0.748	0.549 / 0.741	0.557 / 0.746

As shown in Tab. V, we can see that on open-set dataset Corel10K-a, small α may be more fit to supervised-learning model. Because there are same classes in training set and testing set on Corel10K-b dataset, supervised-learning model is more easy to learn the representations. Here, we use t-SNE [84] to visualize the semantic space of different supervised-learning schemes [11], [15] on Corel10K-b dataset. Specifically, 10 classes are chosen from testing set, with 30 instances in each class. Fig. 8 shows that the semantic space of Cheng [11] is a few disordered. Although the semantic space of Feng [15] is separated between inter-classes, it is not compact among intra-classes. Moreover, some classes are not pulled apart. For EViT, it is not only more distinguishable in the inter-classes but also closer in the intra-classes. In summary, our supervised-learning retrieval model can significantly improve retrieval performance, and there may be still room for improvement (Section IV-D).

3) Why Not CNN and Non-End-to-End: Current deep plain-image retrieval works [1] are end-to-end, which directly use images as model's inputs to automatically extract features (e.g. CNN features) rather than hand-craft features. However, it is impossible for our cipher-images to extract ruled features in an end-to-end manner, since the spatial structure information of cipher-images (e.g., pixel values) is randomly changed and disordered by secret keys. Therefore, EViT adopts the non-end-to-end manner since the artificial features (e.g., local length sequence and global Huffman-Code frequency) are ruled and can be utilized to effectively learn the representations of encrypted images.

TABLE VI
RETRIEVAL PERFORMANCE (mAP@100) WITH THE DIFFERENT BACKBONES IN END-TO-END (ETE) AND NON-END-TO-END (NETE) MANNERS ON COREL10K-B DATASET, "FAILED" REPRESENTS MAP@100 LESS THAN 0.1

Backbone	ResNet50 (ETE)	ViT (ETE)	ResNet50 (NETE)	ViT (NETE)
Unsupervised	failed	failed	0.198	**0.323**
Supervised	0.101	0.128	0.512	**0.750**

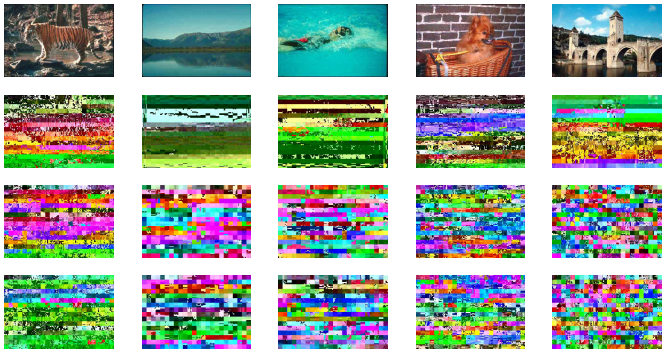
In Section III-C, we have mentioned that the backbone uses ViT rather than CNN (3 reasons), and here we compare retrieval performance with ResNet50 [61] (a classical CNN backbone) on the Corel10K-b dataset. As shown in Tab. VI, the non-end-to-end manner is far beyond the end-to-end manner for different backbones in retrieval performance, and ViT surpasses ResNet50 about 10% and 20% mAP@100 in the non-end-to-end unsupervised and supervised manner, respectively. Tab. VI presents that non-end-to-end manner and ViT are vital for improving retrieval performance.

4) Time Consumption: The current PPIR schemes for extracting features are usually divided into handcrafted features in a non-end-to-end manner and deep features in an end-to-end manner. Handcrafted feature extraction leads to extra time consumption compared with an end-to-end manner that can automatically extract features by deep learning models. Here, we test the time consumption of feature extraction and searching and average the results on the entire Corel10K dataset. As shown in Tab. VII, end-to-end schemes [15], [16] do not require separate handcrafted feature extraction (0s). In contrast, EViT and Feng [12] consume 2.976s and 3.367s respectively. When searching similar cipher-images, Feng [12] only inputs shallow global histogram features and employs a simple DNN model, resulting in shorter search times compared to [15] and [16], and EViT.

TABLE VII

TIME CONSUMPTION (S) COMPARISONS OF FEATURE EXTRACTION AND SEARCHING WITH DIFFERENT SCHEMES

Schemes	Non-end-to-end		End-to-End	
	Feng [12]	EViT	Feng [15]	Lu [16]
Feature extraction	3.367	2.976	0	0
Searching	0.091	0.158	0.162	0.173



(a) (b) (c) (d) (e)

Fig. 10. 5 encryption examples (first row: plain-images; second row: stream cipher; third row: block permutation + stream cipher; fourth row: block permutation + sign encryption + stream cipher).

Although end-to-end schemes eliminate the need for hand-crafted feature extraction and exhibit faster processing times, EViT surpasses them in terms of retrieval performance. Moreover, EViT achieves better visual security (see next Section). In the future, we will deploy a model with C++ and use Faiss [85] to further decrease searching time.

B. Security

In the proposed EViT, images are encrypted during the JPEG compression. The adopted encryption operations do not destroy the format information of JPEG, hence our encryption scheme is format-compliant to JPEG. In Fig. 10, we present five plain-images as examples to demonstrate the visual performance of the encryption algorithm. As shown in Fig. 10, we also describe the encrypted images with only stream cipher (second row). Although stream cipher achieves decent visual privacy, there remains an appearance of stripes in Fig. 10 (a) while in the last image, one can see the top of the tower in Fig. 10 (e). However, combining block permutation and stream cipher (third row), we can find that the encrypted images are disordered enough and do not disclose any visual cues about plain-images. Here, we analyze the encryption performance from four aspects: encrypted image quality, statistical attack, key security, and differential cryptanalysis.

1) Encrypted Image Quality: Generally, the Peak Signal-to-Noise Ratio (PSNR) is used to evaluate the image quality. We compare the encryption algorithm with that of current PPIR schemes and calculate the average PSNR of all images, where a smaller PSNR indicates better encrypted-image quality [67]. Because schemes [4], [5], [11], [12], [16] encrypt images during the JPEG compression (convert images into YUV spaces), and schemes [3], [6], [14], [15] encrypt images in spatial domain, we evaluate PSNR in YUV and RGB color spaces for fair comparison. As shown in Tab. VIII, we can see that EViT achieves a smaller PSNR. Additionally, we calculate the PSNRs by adding stream cipher, block permutation, and sign encryption one by one. Tab. IX shows that combining block permutation and stream cipher achieves smaller PSNR than only stream cipher, which also confirms the statement in Fig. 10. Sign encryption enhances encrypted image quality, and integrating block permutation, sign encryption, and stream cipher obtains the lowest PSNR.

TABLE VIII

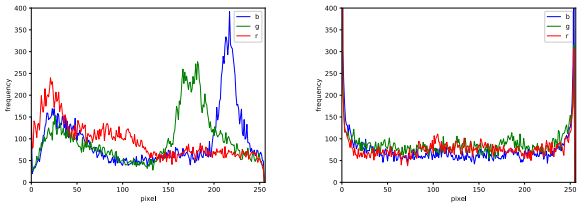
COMPARISONS OF PSNR (DB) IN YUV AND RGB COLOR SPACES FOR DIFFERENT SCHEMES (SMALLER PSNR (↓) INDICATES BETTER ENCRYPTED IMAGE QUALITY)

Schemes	Zhang [5]	Li [4]	Xia [3]	Xia [6]	Lu [16]	Feng [15]	Ma [14]	Feng [12]	Cheng [11]	EViT
YUV	16.401	13.182	13.126	13.218	13.203	13.564	13.695	11.797	11.021	10.341
RGB	9.282	7.968	8.983	8.972	7.659	9.032	9.211	7.642	6.979	6.468

TABLE IX

PSNR WITH DIFFERENT ENCRYPTION OPERATIONS IN EVIT

Encryption operation	Stream cipher	Block permutation + stream cipher	Block permutation + sign encryption + stream cipher
PSNR (RGB)	7.231	6.976	**6.468**



(a) Plain-image (b) Cipher-image

Fig. 11. Histograms of the plain and cipher images.

2) Statistical Attack: To make the statistical mode-based attack unavailable, the histograms of plain-images and that of cipher-images should be different. As shown in Fig. 11, taking the plain-image (Fig. 10 (e)) and its cipher-image as examples, it can be seen that there is no statistical correlation between the histogram of the encrypted image and that of the plain-image. Compared with the histogram of the plain-image, the frequency difference of different pixel values in the cipher-image is not so large, thus the proposed scheme has a certain resistance ability against statistical attack.

3) Key Security: Brute-force attack is a standard ciphertext-only attack strategy where attackers have access solely to the encrypted data. The proposed encryption method has fifteen encryption keys ($k^*_{BP}, k^*_{ODC}, k^*_{OAC}, k^*_{DC}, k^*_{AC}, * \in \{Y, U, V\}$), with each component having distinct keys. Typically, these secret keys are pseudo-random keys that generated by hash function BLAKE2 (mentioned in Section IV-A). Each original key has 256 bits and adapts to produce pseudo-random secret keys with variable lengths tailored to match specific encryption operations. For instance, in the case of AC XOR, if an image's AC coefficients bitstream has a length of 50000, the corresponding pseudo-random key, k_{AC}, is adjusted to the same length (50000). In the context of evaluating brute force attacks, our analysis encompasses two crucial facets: encryption space and key space. The encryption space is

dependent on the encryption operations, while the key space is determined by the original key length. The encryption and key spaces are explicated as follows.

a) Encryption space: Our encryption algorithm contains three operations: block shuffling, sign encryption, and stream exclusive-OR (XOR). The encryption space of block permutation (ES_{BP}) depends on the number of blocks, and the images have 384 blocks ($blknum = 384$), so ES_{BP} can be defined as:

$$ES_{BP} = \prod_{Y,U,V} (blknum!) = (384!)^3 \approx 2^{8244}. \quad (20)$$

The sign encryption operates on each 8×8 block, and its encryption space (ES_{SE}) is expressed as:

$$ES_{SE} = \prod_{Y,U,V} \prod_{j=1}^{blknum} (2^{64}) = ((2^{64})^{384})^3 = 2^{73728}. \quad (21)$$

The ES_{BP} and ES_{SE} are enough large to resist brute-force break. The pseudo-random stream-cipher keys k_{DC} and k_{AC} are generated by Hash. Since Hash is a one-way function [8], even if the encrypted JPEG bitstream and hash code are known, it is difficult to infer the secret bitstream and break k_{DC} and k_{AC}. Therefore, EViT is safe and can be secure against ciphertext-only attacks.

b) Key space: The original key length is 256. Taking an encrypted image as example, the key spaces of k_{ODC} (denoted as KS_{ODC}), k_{OAC} (denoted as KS_{OAC}), k_{DC} (denoted as KS_{DC}), and k_{AC} (denoted as KS_{AC}) in the sign encryption and stream cipher stage can be defined as:

$$KS_{ODC} = KS_{OAC} = KS_{DC} = KS_{AC} = \prod_{Y,U,V} (2^{256}) = (2^{256})^3, \quad (22)$$

It is worth noting that the secret keys k_{ODC}, k_{OAC}, k_{DC}, and k_{AC} can differ for each image since the sign encryption and stream cipher cannot change the length of VLI code. Hence, encrypting different images with various sign encryption and stream cipher keys increases the number of keys. We encrypt each image with the same block permutation secret key k_{BP} to ensure the unified feature spaces, because different block shuffling results in different orders of local block feature sequences. The key spaces of k_{BP} (KS_{BP}) is expressed as:

$$KS_{BP} = (2^{256})^3, \quad (23)$$

where 3 represents the Y/U/V color component. Thus, the total key spaces (KS) are calculated as:

$$KS = KS_{ODC} \times KS_{OAC} \times KS_{DC} \times KS_{AC} \times KS_{BP} = (2^{256})^{15}. \quad (24)$$

The size of KS is sufficiently large, making it extremely challenging to employ brute-force methods to recover the plaintext images.

The cloud server cannot be completely trusted, and there are only encrypted images without plain-images, so it also may be curious about the cipher-images. First, EViT is effective in the ciphertext-only attack and the image visual content is enough safe, so the cloud is difficult to break these cipher-images. Additionally, the cloud extracts cipher-image features, but these weak features are not enough to compromise the privacy of the images since they are not the same as that of plain-images and do not mirror the main visual information. Second, when a query-encrypted image is sent to the cloud, similar cipher-images are returned to users. Namely, the cloud can know these images may exist correlations by analyzing searching traces. However, this kind of information leakage is common for cloud servers, and most PPIR schemes hope to reduce extra computational burden through cloud servers and do not consider this type of leak [14], [62].

4) Differential Cryptanalysis: To resist differential cryptanalysis, minor alterations of plain-image such as modifying one single pixel should result in a significant change in corresponding cipher-image [86]. For instance, given a plain-image P^1, we slightly change one single pixel (e.g., select a pixel at random and decrementing its value by one) and obtain a comparable plain-image P^2. Subsequent encryption of P^1 and P^2 yields respective cipher-images C^1 and C^2. Differential cryptanalysis exploits the disparities between these cipher-images (C^1 and C^2) for attack, and therefore we anticipate substantial distinctions between these encrypted representations. To evaluate a cryptosystem's resilience against such attacks, commonly employed metrics include NPCR (Number of Pixels Change Rate) and UACI (Unified Average Changing Intensity) [87]. These metrics are defined as follows:

$$D(i,j) = \begin{cases} 0, C^1(i,j) = C^2(i,j) \\ 1, C^1(i,j) \neq C^2(i,j) \end{cases}, \quad (25)$$

$$NPCR : N\left(C^1, C^2\right) = \frac{\sum_{i,j} D(i,j)}{H \times W} \times 100\%, \quad (26)$$

$$UACI : U\left(C^1, C^2\right) = \frac{\sum_{i,j} \frac{|C^1(i,j) - C^2(i,j)|}{255}}{H \times W} \times 100\%, \quad (27)$$

where C^1 and C^2 are corresponding cipher-images of plain-images P^1 and P^2, and H and W are the width and height of the image, respectively. $C(i,j)$ is the pixel value at coordinates (i,j) $(1 \leq i \leq H, 1 \leq j \leq W)$. As mentioned in Section IV-A, the secret keys for stream cipher and sign encryption ($k^*_{ODC}, k^*_{OAC}, k^*_{DC}$, and k^*_{AC}, where $* \in Y, U, V$) may vary for each image. However, to ensure effective retrieval within a unified feature space, the block permutation key k^*_{BP} remains constant across all images. We test differential attacks by two scenarios: (a) all images employing the same fifteen encryption keys; (b) all images sharing k^*_{BP} but employing distinct $k^*_{ODC}, k^*_{OAC}, k^*_{DC}$, and k^*_{AC}. Here, we select six categories (church, girl, sky, architecture, painting, Africa) which contain 600 images from the Corel10K dataset, and a single pixel in each plain-image was modified. NPCR and UACI were calculated and averaged within each category. The closer the NPCR is to 100% and the UACI is to 33%, the more vital ability to resist differential attack [87]. The results, presented in Tab. X, reveal that when employing identical fifteen keys, NPCR is only around 1%, and UACI remains below 1%, indicating minimal disparities between the

TABLE X

NPCR AND UACI OF CIPHER-IMAGES WITH ONE-PIXEL CHANGING ('#1' MEANS ALL IMAGES USE THE SAME FIFTEEN ENCRYPTION KEYS; '#2' MEANS ALL IMAGES USE THE SAME k^*_{BP}, BUT WITH DIFFERENT $k^*_{ODC}, k^*_{OAC}, k^*_{DC}$, AND k^*_{AC})

Category	Church	Girl	Sky	Architecture	Painting	Africa
NPCR (#1)	1.02%	1.35%	0.96%	1.31%	1.29%	1.14%
NPCR (#2)	98.68%	99.09%	98.51%	98.78%	98.41%	98.84%
UACI (#1)	0.34%	0.23%	0.17%	0.15%	0.18%	0.23%
UACI (#2)	38.73%	40.12%	41.08%	40.13%	41.07%	41.12%

TABLE XI

COMPARISONS OF MAP (UNSUPERVISED) AND PSNR WITH DIFFERENT PERMUTATION SITUATIONS

	Situation 3	Situation 2	Situation 1
Corel10K-a	0.476	0.483	0.492
Corel10K-b	0.323	0.336	0.341
PSNR (RGB)	6.468	6.572	6.734

cipher-images C^1 and C^2. In contrast, when employing distinct keys $k^*_{ODC}, k^*_{OAC}, k^*_{DC}$, and k^*_{AC}, NPCR approximates 98%, and UACI reaches about 40%, showcasing a notable resilience against differential attacks.

Owing to our distinctive encryption paradigm, EViT can encrypt diverse images utilizing unique secret keys for stream cipher and sign encryption, while maintaining uniformity with block shuffling secret keys. In the scenario where all images share identical fifteen secret keys, it is poor in resisting differential attacks due to lower NPCR and UACI. However, when all images employing twelve distinct secret keys for stream cipher and sign encryption ($k^*_{ODC}, k^*_{OAC}, k^*_{DC}, k^*_{AC}$), EViT fortifies itself against differential attacks. Fortunately, EViT accommodates the encryption of each image with distinct secret keys, including variations in stream cipher and sign encryption keys. Contrastingly, prevailing PPIR schemes [3], [6], [7], [8], [12], [14], [15], [33] falter in resisting differential attacks. To ensure compatibility with unified feature extraction spaces and optimize retrieval efficacy, these schemes [3], [6], [7], [8], [12], [14], [15], [33] mandate uniform encryption of all images with identical secret keys. Notably, differential attack is a kind of Chosen Plaintext Attack (CPA), but most PPIR schemes only store cipher-images on cloud server, devoid of plain-images and encryption oracles. So they may operate on the assumption that differential attacks are unavailable in PPIR scenarios, leading to an irrespective of it in their works.

C. Performance Impact of Block Permutation

Although block permutation changes the orders of local length sequence features, it does not guarantee enough visual security without shuffling (Fig. 10). Furthermore, stream cipher and sign encryption with VLI codes lead to the extracted features from encrypted images being the same as plain-image. Hence, EViT incorporates additional block permutation encryption to further enhance security, namely the local features are shuffled.

Here, we examine the performance impact (e.g., mAP and PSNR) of the permutation encryption, which can be divided into three situations:

- Situation 1: No block permutation.
- Situation 2: All components are shuffled using the same secret key.
- Situation 3: Different components utilize various permutation secret keys.

The results are summarized in Tab. XI. In situation 1, where no block permutation is applied, the feature spaces remain the same as the plain images because the length of the VLI code is unchanged. As a result, situation 1 achieves better retrieval performance (higher mAP) but compromises visual security (higher PSNR). Because EViT owns inherent permutation invariance and all images are shuffled with the same secret keys, the retrieval performances of situations 2 and 3 do not significantly decline. Compared with situation 2, situation 3 introduces more permutation noise since it uses three different keys to shuffle three components. Consequently, situation 2 gets higher mAP than situation 3, at the cost of lower visual security. To improve image privacy, we ultimately choose situation 3.

In accordance with Section IV-A, PPIR systems inherently strive to strike a balance between retrieval efficacy and image privacy preservation. The process of extracting features from cipher-images presents two balances that necessitate careful consideration. Firstly, the chaotic features, analogous to noise, can hinder efficient retrieval. To surmount this obstacle, the extracted features must incorporate ruled information and adhere to specific distributions conducive to the input retrieval model. Secondly, it is imperative that these extracted features do not inadvertently disclose vital image privacy information, such as DCT coefficients and color histograms. To satisfy the two balances, existing PPIR techniques generally opt for preserving certain weak feature distributions, which can provide retrieval and do not leak main image content. These weak features remain insufficient to compromise image visual privacy and exhibit distinctions when compared to plain-images, such as variations in feature order. EViT selectively retains weak features within inter-block while also introducing randomized alterations to DCT coefficients and color histograms. Notably, the cloud exclusively stores encrypted images, and EViT ensures robust visual content privacy (e.g., PSNR) and effectively resists ciphertext-only attacks. Furthermore, EViT significantly bolsters retrieval accuracy when balancing image privacy through the exploration of multi-level features and the introduction of the proposed deep learning model.

VI. CONCLUSION

This paper introduces a novel privacy-preserving image retrieval scheme named EViT, which can improve retrieval performance by large margins than current schemes and effectively ensure the security of images. First, we develop multi-level features, encompassing local length sequences and global Huffman-Code frequencies, extracted from cipher-images that undergo permutation encryption, sign bit encryption, and stre-am cipher during the JPEG compression process. Second, EViT presents an unsupervised retrieval model in a

self-supervised learning manner, leveraging the ViT structure as the backbone to integrate with the multi-level features. To improve retrieval performance, EViT employs two adaptive data augmentations within the retrieval model and advances ViT with learnable global Huffman-Code frequency. The supervised model can be easily achieved by fine-tuning the trained unsupervised retrieval model. Experimental results show that EViT not only effectively protects image privacy but also significantly improves retrieval performance than current schemes. In future work, we will try to further enhance the search efficiency without compromising retrieval accuracy.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions.

REFERENCES

[1] W. Chen et al., "Deep image retrieval: A survey," 2021, *arXiv:2101.11282*.

[2] S. R. Dubey, "A decade survey of content based image retrieval using deep learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 5, pp. 2687–2704, May 2022.

[3] Z. Xia, L. Wang, J. Tang, N. N. Xiong, and J. Weng, "A privacy-preserving image retrieval scheme using secure local binary pattern in cloud computing," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 318–330, Jan./Mar. 2021.

[4] P. Li and Z. Situ, "Encrypted JPEG image retrieval using histograms of transformed coefficients," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Nov. 2019, pp. 1140–1144.

[5] X. Zhang and H. Cheng, "Histogram-based retrieval for encrypted JPEG images," in *Proc. IEEE China Summit Int. Conf. Signal Inf. Process. (ChinaSIP)*, Jul. 2014, pp. 446–449.

[6] Z. Xia, L. Jiang, D. Liu, L. Lu, and B. Jeon, "BOEW: A content-based image retrieval scheme using bag-of-encrypted-words in cloud computing," *IEEE Trans. Services Comput.*, vol. 15, no. 1, pp. 202–214, Jan./Feb. 2022.

[7] Z. Xia, L. Lu, T. Qiu, H. J. Shim, X. Chen, and B. Jeon, "A privacy-preserving image retrieval based on AC-coefficients and color histograms in cloud environment," *Comput., Mater. Continua*, vol. 58, no. 1, pp. 27–43, 2019.

[8] H. Liang, X. Zhang, and H. Cheng, "Huffman-code based retrieval for encrypted JPEG images," *J. Vis. Commun. Image Represent.*, vol. 61, pp. 149–156, May 2019.

[9] H. Cheng, X. Zhang, and J. Yu, "AC-coefficient histogram-based retrieval for encrypted JPEG images," *Multimedia Tools Appl.*, vol. 75, no. 21, pp. 13791–13803, Nov. 2016.

[10] H. Cheng, X. Zhang, J. Yu, and Y. Zhang, "Encrypted JPEG image retrieval using block-wise feature comparison," *J. Vis. Commun. Image Represent.*, vol. 40, pp. 111–117, Oct. 2016.

[11] H. Cheng, X. Zhang, J. Yu, and F. Li, "Markov process based retrieval for encrypted JPEG images," in *Proc. 10th Int. Conf. Availability, Rel. Secur.*, Aug. 2015, pp. 417–421.

[12] Q. Feng et al., "DHAN: Encrypted JPEG image retrieval via DCT histograms-based attention networks," *Appl. Soft Comput.*, vol. 133, Jan. 2023, Art. no. 109935.

[13] Z. Lu, Q. Feng, and P. Li, "Encrypted JPEG image retrieval via Huffman-code based self-attention networks," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Nov. 2022, pp. 1–6.

[14] W. Ma, T. Zhou, J. Qin, X. Xiang, Y. Tan, and Z. Cai, "A privacy-preserving content-based image retrieval method based on deep learning in cloud computing," *Expert Syst. Appl.*, vol. 203, Oct. 2022, Art. no. 117508.

[15] Q. Feng, P. Li, Z. Lu, G. Liu, and F. Huang, "End-to-end learning for encrypted image retrieval," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Dec. 2021, pp. 1839–1845.

[16] Z. Lu, Q. Feng, and P. Li, "A privacy-preserving and end-to-end-based encrypted image retrieval scheme," in *Proc. IEEE Int. Conf. Vis. Commun. Image Process. (VCIP)*, Dec. 2022, pp. 1–5.

[17] J. MacQueen et al., "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, Oakland, CA, USA, 1967, vol. 1, no. 14, pp. 281–297.

[18] S. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2003, pp. 1470–1477.

[19] J. Xie, R. B. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Conf. Mach. Learn. (ICML)*, vol. 48, 2016, pp. 478–487.

[20] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, vol. 119, 2020, pp. 1597–1607.

[21] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9726–9735.

[22] X. Chen, H. Fan, R. B. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2003, *arXiv:2003.04297*.

[23] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 12356. Cham, Switzerland: Springer, 2020, pp. 776–794.

[24] W. V. Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. V. Gool, "SCAN: Learning to classify images without labels," in *Proc. Eur. Conf. Comput. Vis., (ECCV)*, vol. 12355. Cham, Switzerland: Springer, 2020, pp. 268–285.

[25] Z. Dang, C. Deng, X. Yang, K. Wei, and H. Huang, "Nearest neighbor matching for deep clustering," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13693–13702.

[26] Y. K. Jang and N. I. Cho, "Self-supervised product quantization for deep unsupervised image retrieval," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12065–12074.

[27] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[28] A. Vaswani et al., "Attention is all you need," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NeuIPS)*, 2017, pp. 5998–6008.

[29] M. Naseer, K. Ranasinghe, S. Khan, M. Hayat, F. S. Khan, and M. Yang, "Intriguing properties of vision transformers," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2105, pp. 23296–23308.

[30] L. Weng, L. Amsaleg, A. Morton, and S. Marchand-Maillet, "A privacy-preserving framework for large-scale content-based information retrieval," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 1, pp. 152–167, Jan. 2015.

[31] M. Osadchy, B. Pinkas, A. Jarrous, and B. Moskovich, "SCiFI—A system for secure face identification," in *Proc. IEEE Symp. Secur. Privacy*, May 2010, pp. 239–254.

[32] L. Weng, L. Amsaleg, and T. Furon, "Privacy-preserving outsourced media search," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 10, pp. 2738–2751, Oct. 2016.

[33] Z. Xia, Q. Ji, Q. Gu, C. Yuan, and F. Xiao, "A format-compatible searchable encryption scheme for JPEG images using bag-of-words," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 18, no. 3, pp. 1–18, Aug. 2022.

[34] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Data Compression Standard*. Springer, 1992.

[35] C. A. Christopoulos, T. Ebrahimi, and A. N. Skodras, "JPEG2000: The new still picture compression standard," in *Proc. ACM Workshops Multimedia*. New York, NY, USA: ACM Press, Nov. 2000, pp. 45–49.

[36] C. Deng, E. Yang, T. Liu, and D. Tao, "Two-stream deep hashing with class-specific centers for supervised image search," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 2189–2201, Jun. 2020.

[37] S. S. Husain and M. Bober, "REMAP: Multi-layer entropy-guided pooling of dense CNN features for image retrieval," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 5201–5213, Oct. 2019.

[38] L. Liao, M. Yang, and B. Zhang, "Deep supervised dual cycle adversarial network for cross-modal retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 2, pp. 920–934, Sep. 2023.

[39] I. Misra and L. van der Maaten, "Self-supervised learning of pretext-invariant representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6706–6716.

[40] Q. Qin, L. Huang, Z. Wei, K. Xie, and W. Zhang, "Unsupervised deep multi-similarity hashing with semantic structure for image retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 7, pp. 2852–2865, Jul. 2021.

[41] J. Huang, Y. Huang, Q. Wang, W. Yang, and H. Meng, "Self-supervised representation learning for videos by segmenting via sampling rate order prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 6, pp. 3475–3489, Jun. 2022.

[42] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3733–3742.

[43] Z. Deng, Y. Zhong, S. Guo, and W. Huang, "InsCLR: Improving instance retrieval with self-supervision," in *Proc. Conf. Artif. Intell. (AAAI), Innov. Appl. Artif. Intell. IAAI, Educ. Adv. Artif. Intell. (EAAI)*, 2022, pp. 516–524.

[44] Z. Xie et al., "Self-supervised learning with Swin Transformers," 2105, *arXiv:2105.04553*.

[45] S. A. A. Ahmed, M. Awais, and J. Kittler, "SiT: Self-supervised vision transformer," 2021, *arXiv:2104.03602*.

[46] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. Minneapolis, MI, USA: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[47] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, pp. 140:1–140:67, 2020.

[48] M. Farahani, M. Gharachorloo, M. Farahani, and M. Manthouri, "ParsBERT: Transformer-based model for Persian language understanding," *Neural Process. Lett.*, vol. 53, no. 6, pp. 3831–3847, Dec. 2021.

[49] K. Lin, L. Wang, and Z. Liu, "End-to-End human pose and mesh reconstruction with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1954–1963.

[50] B. Kim, J. Lee, J. Kang, E. Kim, and H. J. Kim, "HOTR: End-to-end human-object interaction detection with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 74–83.

[51] A. Prakash, K. Chitta, and A. Geiger, "Multi-modal fusion transformer for end-to-end autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7077–7087.

[52] K. Li, S. Wang, X. Zhang, Y. Xu, W. Xu, and Z. Tu, "Pose recognition with cascade transformers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1944–1953.

[53] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "LoFTR: Detector-free local feature matching with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8922–8931.

[54] Y. Zhou, F. Wang, J. Zhao, R. Yao, S. Chen, and H. Ma, "Spatial–temporal based multihead self-attention for remote sensing image change detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 10, pp. 6615–6626, Oct. 2022.

[55] R. Ji, J. Li, L. Zhang, J. Liu, and Y. Wu, "Dual transformer with multi-grained assembly for fine-grained visual classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 9, pp. 5009–5021, 2023.

[56] Z. Liu et al., "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[57] B. Graham et al., "LeViT: A vision transformer in ConvNet's clothing for faster inference," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12239–12249.

[58] W. Wang et al., "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 548–558.

[59] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, and C. Schmid, "ViViT: A video vision transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6816–6826.

[60] L. J. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[62] Z. Lu, Q. Feng, P. Li, K.-T. Lo, and F. Huang, "A privacy-preserving image retrieval scheme based on 16×16 DCT and deep learning," *IEEE Trans. Cloud Comput.*, vol. 11, no. 3, pp. 3314–3325, 2023.

[63] B. Ferreira, J. Rodrigues, J. Leit ao, and H. Domingos, "Practical privacy-preserving content-based retrieval in cloud image repositories," *IEEE Trans. Cloud Comput.*, vol. 7, no. 3, pp. 784–798, Jul. 2019.

[64] B. Ferreira, J. Rodrigues, J. Leit ao, and H. Domingos, "Privacy-preserving content-based image retrieval in the cloud," in *Proc. IEEE 34th Symp. Reliable Distrib. Syst. (SRDS)*, Sep. 2015, pp. 11–20.

[65] Z. Qian, X. Zhang, and S. Wang, "Reversible data hiding in encrypted JPEG bitstream," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1486–1491, Aug. 2014.

[66] J. Aumasson, S. Neves, Z. Wilcox-O'Hearn, and C. Winnerlein, "BLAKE2: Simpler, smaller, fast as MD5," in *Proc. 11th Int. Conf. Appl. Cryptogr. Netw. Secur. (ACNS)*, vol. 7954. Springer, 2013, pp. 119–135.

[67] P. Li and K.-T. Lo, "A content-adaptive joint image compression and encryption scheme," *IEEE Trans. Multimedia*, vol. 20, no. 8, pp. 1960–1972, Aug. 2018.

[68] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[69] A. F. Agarap, "Deep learning using rectified linear units (ReLU)," 2018, *arXiv:1803.08375*.

[70] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 26th Annu. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.

[71] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Lear. (ICML)*, vol. 139, 2021, pp. 10347–10357.

[72] T. Wolf et al., "Transformers: State-of-the-art natural language processing," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*. Association for Computational Linguistics, 2020, pp. 38–45.

[73] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*. Association for Computational Linguistics, 2019, pp. 6381–6387.

[74] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, Sep. 2014.

[75] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Association for Computational Linguistics, 2021, pp. 6894–6910.

[76] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4690–4699.

[77] K. Ozaki and S. Yokoo, "Large-scale landmark retrieval/recognition under a noisy and diverse dataset," 2019, *arXiv:1906.04087*.

[78] Q. Ha, B. Liu, F. Liu, and P. Liao, "Google landmark recognition 2020 competition third place solution," 2020, *arXiv:2010.05350*.

[79] J. Deng and S. Zafeririou, "ArcFace for disguised face recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 485–493.

[80] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[81] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 9911. Springer, 2016, pp. 499–515.

[82] J. Li and J. Z. Wang, "Automatic linguistic indexing of pictures by a statistical modeling approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1075–1088, Sep. 2003.

[83] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2019, pp. 8024–8035.

[84] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.

[85] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, Jul. 2021.

[86] J. He, S. Huang, S. Tang, and J. Huang, "JPEG image encryption with improved format compatibility and file size preservation," *IEEE Trans. Multimedia*, vol. 20, no. 10, pp. 2645–2658, Oct. 2018.

[87] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons Fractals*, vol. 21, no. 3, pp. 749–761, Jul. 2004.

Qihua Feng received the B.S. degree from Yangtze University, China, in 2019, and the M.S. degree in computer technology from Jinan University, Guangzhou, China, in 2022. He is currently pursuing the Ph.D. degree with Beijing Institute of Technology, Beijing, China. His research interests include image process, privacy preserving, and the Internet of Things.

Peiya Li received the B.S. degree from Anhui Normal University, China, in 2012, the M.E. degree from Zhejiang University, China, in 2014, and the Ph.D. degree in engineering from The Hong Kong Polytechnic University, Hong Kong, in 2018. She is currently a Lecturer with the College of Cyber Security, Jinan University, Guangzhou, China. Her research interests include multimedia encryption, image coding, and image retrieval.

Zhixun Lu received the B.S. degree from Guangdong University of Petrochemical Technology, Maoming, China, in 2019. He is currently pursuing the M.S. degree in computer technology with Jinan University, Guangzhou, China. His current research interests include multimedia security and applications.

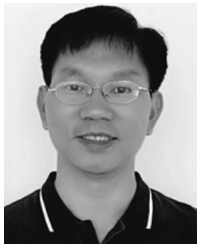
Chaozhuo Li received the B.S. degree in computer science from the School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China, in 2011. He is currently pursuing the Ph.D. degree in computer software and theory with the State Key Laboratory of Software Development Environment, School of Computer Science and Engineering, Beihang University, Beijing. He has authored more than ten papers on top conferences (e.g., AAAI, SIGIR, CIKM, and International Conference on Data Mining) and journals in these areas. He served as a reviewer for multiple top international conferences and journals in the areas of graph mining and social network analysis. His research interests include graph mining (e.g., network representation learning), social network analysis (e.g., social spammer detection), and recommender systems.

Zefan Wang received the B.S. degree from Jinan University, Guangzhou, China, in 2021, where he is currently pursuing the M.S. degree. His research interests include recommender systems.

Zhiquan Liu (Member, IEEE) is currently with the College of Information Science and Technology/College of Cyber Security, National Joint Engineering Research Center of Network Security Detection and Protection Technology, Guangdong Key Laboratory of Data Security and Privacy Preserving, Jinan University, Guangzhou, Guangdong, China.

Chunhui Duan received the B.S. and Ph.D. degrees from the School of Software, Tsinghua University, Beijing, China, in 2013 and 2018, respectively. Previously, she was a Post-Doctoral Research Fellow with Tsinghua University. She is currently an Associate Professor with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing. Her research interests include RFID, the Internet of Things, wireless sensing, and mobile computing.

Feiran Huang (Member, IEEE) received the B.S. degree from Central South University, Changsha, China, in 2011, and the Ph.D. degree from Beihang University, Beijing, China, in 2019. He is currently an Assistant Professor with the College of Cyber Security, Jinan University, Guangzhou, China. His research interests include social media analysis and multimodal learning.



Jian Weng (Member, IEEE) received the Ph.D. degree in computer science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 2008. He is currently a Professor and the Dean of the College of Information Science and Technology, Jinan University, Guangzhou, China. His research interests include public key cryptography, cloud security, and blockchain. He was the PC co-chair or a PC member of more than 30 international conferences. He also serves as an Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY.

Philip S. Yu (Life Fellow, IEEE) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA. He is currently a Distinguished Professor in computer science with the University of Illinois at Chicago, Chicago, IL, USA, and the Wexler Chair in Information Technology. His research interests include big data, data mining, data stream, database, and privacy. He was a recipient of ACM SIGKDD 2016 Innovation Award, a Research Contributions Award from the IEEE International Conference on Data Mining in 2003, and a Technical Achievement Award from the IEEE Computer Society in 2013. He is a fellow of ACM. He was the Editor-in-Chief of IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING and the *ACM Transactions on Knowledge Discovery from Data*.