Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Reversible data hiding using a transformer predictor and an adaptive embedding strategy*

Linna ZHOU†1, Zhigao LU†2, Weike YOU†‡1, Xiaofei FANG2

[1]*School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100084, China*
[2]*School of Information Science & Technology, University of International Relations, Beijing 100091, China*
†E-mail: zhoulinna@bupt.edu.cn; luchen@uir.edu.cn; ywk@bupt.edu.cn

Received Jan. 20, 2023; Revision accepted Feb. 28, 2023; Crosschecked June 6, 2023

Abstract: In the field of reversible data hiding (RDH), designing a high-precision predictor to reduce the embedding distortion and developing an effective embedding strategy to minimize the distortion caused by embedding information are the two most critical aspects. In this paper, we propose a new RDH method, including a predictor based on a transformer and a novel embedding strategy with multiple embedding rules. In the predictor part, we first design a transformer-based predictor. Then, we propose an image division method to divide the image into four parts, which can use more pixels as context. Compared with other predictors, the transformer-based predictor can extend the range of pixels for prediction from neighboring pixels to global ones, making it more accurate in reducing the embedding distortion. In the embedding strategy part, we first propose a complexity measurement with pixels in the target blocks. Then, we develop an improved prediction error ordering rule. Finally, we provide an embedding strategy including multiple embedding rules for the first time. The proposed RDH method can effectively reduce the distortion and provide satisfactory results in improving the visual quality of data-hidden images, and experimental results show that the performance of our RDH method is leading the field.

Key words: Reversible data hiding; Transformer; Adaptive embedding strategy
https://doi.org/10.1631/FITEE.2300041 **CLC number:** TP309

1 Introduction

Reversible data hiding (RDH) has been widely used in scenarios requiring high-quality cover signals, e.g., military communication and health care, as secret data can be embedded into the cover signal without loss (Cox et al., 2002). Prediction error expansion (PEE) (Thodi and Rodriguez, 2007) is one of the most widely used techniques to achieve RDH. PEE achieves prediction of the target pixel using the context pixels to generate the prediction error at first. Then, the result obtained is expanded according to the predefined strategy to achieve data embedding.

So far, one approach to improve the PEE is designing a high-precision prediction method to reduce embedding distortion, since, in PEE, the smaller the prediction error, the better the visual quality after embedding the pixels. Most of the prediction methods focus on the improvement of the predictor, e.g., difference predictor (Tian, 2003), median edge direction predictor (Thodi and Rodriguez, 2007), bilinear interpolation predictor (Sachnev et al., 2009; Luo et al., 2010), rhombus predictor (Chen et al., 2010), gradient adaptive predictor (Coltuc, 2011, 2012), and others using multiple predictors (Jafar et al., 2016). The above-mentioned predictors use the similarity between the target pixel and neighboring pixels. However, they all use only one or a few neighboring pixels for prediction, which

‡ Corresponding author
* Project supported by the National Natural Science Foundation of China (No. 62172053) and the National Key Research and Development Program of China (Nos. 2021YFC3340701 and 2021YFC3340602)
ORCID: Zhigao LU, https://orcid.org/0000-0002-2215-9843; Weike YOU, https://orcid.org/0000-0002-2642-6005

limits the similarity between the target pixel and the global pixels. Targeting the flaw, Weng et al. (2017) designed a prediction pattern in which each to-be-embedded pixel can be predicted by n neighboring pixels surrounding it. The larger the value of n, the more accurate the prediction, and the better the embedding performance achieved at low embedding capacity, and vice versa. Hu and Xiang (2022) proposed a global nonlinear predictor based on convolutional neural networks (CNNs). Although the predictor would effectively expand the range of pixels for prediction, the method could use only more local pixels instead of the pixels of the whole image for prediction because of the disadvantages of CNNs in processing global information. Therefore, exploring a predictor that can use every pixel of the entire image is necessary. In recent years, transformers have emerged in computer vision. Because a transformer can handle long-distance information better than CNNs, the transformer technology, which combined with CNNs or completely replaced CNNs, has developed rapidly in image generation, image division, image detection, etc. Although transformers have not been used in the field of RDH, the introduction of transformers into the RDH community makes it possible to further improve the prediction performance given its excellent performance in the image field.

In addition, another type of approach to improve the PEE is designing an effective embedding strategy to reduce the distortion caused by embedding information. Some embedding strategies focus on the improvement of embedding rules, e.g., histogram shifting (HS) (Thodi and Rodriguez, 2007), pixel value ordering (PVO)-k (Ou et al., 2014), and improved PVO (IPVO) (Weng et al., 2019). However, all existing embedding strategies include only one embedding rule. Since a complete image can be seen as being composed of complex textured blocks and simple smooth blocks, it may be more appropriate to use different embedding rules for different blocks. Therefore, designing an embedding method that can combine multiple embedding rules is necessary. Meanwhile, existing embedding strategies focus on complexity measurements, distinguishing smooth and textured blocks, and determining the embedding areas (Wang X et al., 2015; Hu and Xiang, 2022); the complexity measurements adopted usually use the neighboring pixels or pixel blocks to calculate the complexity of the target pixel (i.e., pixel block). For example, the complexity of He and Cai (2021) was calculated as the sum of the absolute differences between two consecutive pixels in the horizontal or vertical direction. However, the pixels in target blocks may affect the complexity, and these pixels cannot be ignored.

In this paper, both prediction and embedding techniques are considered; that is, our RDH method consists of a transformer predictor and an adaptive embedding strategy with multiple embedding rules. In the predictor part, we use the advantages of transformers to establish global pixel correlation (Goodfellow et al., 2014; Vaswani et al., 2017; Esser et al., 2021; Zheng et al., 2022) and design a predictor based on the transformer. The transformer predictor extends the range of pixels for prediction from neighboring pixels to global ones for the first time. An image division method is also proposed for our predictor, which can use twice the pixels of the rhombus predictor as context. Through our predictor, we obtain many minor prediction errors and then, while embedding secret data, the adaptive embedding strategy expands or secondary extends these prediction errors and embeds secret data. The smaller the prediction error, the smaller the distortion caused by the expanded image. This guarantees the effectiveness of our predictor.

In the embedding strategy part, we first present a complexity measurement step for pixels in target blocks. Then, an effective embedding rule called improved prediction error ordering (IPEO) is proposed by redesigning IPVO (Weng et al., 2019), which sorts the prediction error instead of the pixel value. Next, we offer an embedding strategy including multiple embedding rules for the first time, which can select different rules adaptively according to the complexity of the image. Our adaptive embedding strategy selects the appropriate embedding strategy for the embedded blocks with different complexities, which can improve the embedding capacity and the embedding performance.

Extensive experimental results show that the proposed RDH method provides satisfactory results in embedding information and exceeds the performance of existing methods.

2 Related works

2.1 Rhombus predictor

The rhombus predictor is the first and most classic PEE predictor. Chen et al. (2010) designed a rhombus predictor, which includes the following two steps: feature selection and feature modification based on difference expansion. The rhombus predictor could reconstruct the process of embedding features and transform the original pixel distribution into the predicted error (predicted residual) distribution. Because the residual distribution is similar to the Laplace distribution, the image information expression is more compact, and effective entropy subtraction is realized. The rhombus predictor divides an image into dot sets and cross sets (Fig. 1), and takes the mean of four neighboring pixels of the target pixel as its prediction value because these neighboring pixels can have a significant impact on the target pixel. The subsequent modifications to the rhombus predictor focus mainly on designing an efficient predictor and using more pixels as the prediction context.

Though the rhombus predictor and the rhombus predictor based improved predictors use the similarity between the target pixel and the neighboring pixels, they are local and linear, and are not accurate enough for complex distributed images (Hu and Xiang, 2021). Therefore, it is necessary to explore a global nonlinear predictor that can improve prediction accuracy.

2.2 Complexity measurement

Complexity measurement is a selection strategy using the correlation between neighboring pixels, e.g., the local variance (Sachnev et al., 2009), forward variance (Li et al., 2011), error energy estimation (Hong, 2012), local absolute difference (Ou et al., 2013), and full-enclosing context strategy (He et al., 2017). It determines the embedding position by calculating the complexity of pixels (pixel blocks) and it is important to reduce the embedding distortion. He and Cai (2021) calculated the complexity of pixel blocks as the sum of the absolute differences between two consecutive pixels in the horizontal or vertical direction, which first determined pixel blocks to be embedded and then embedded the data according to the embedding strategy. Hu and Xiang (2022) directly calculated the complexity of each pixel composing pixel blocks by adopting an $L \times L$ block centered at the to-be-embedded pixel.

In Section 3.4.1, we will consider the work of He and Cai (2021) as an example to discuss the defects of complexity measurement based on pixel blocks, namely, the absence of available pixels in the target block in calculation.

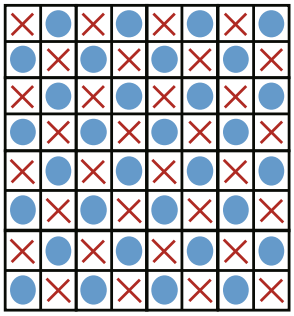

Fig. 1 Image division of a rhombus predictor

2.3 CNN predictor

Hu and Xiang (2021) proposed a CNN predictor (CNNP), which introduced deep learning into the RDH field for the first time. The feature extraction step comprises several convolution blocks arranged parallel to exploit multiple receptive fields. The kernel size of these convolution blocks is $K \times K$, where K is set to 3, 5, or 7. The features extracted from different convolution blocks are added together and fed into two convolution blocks with kernel size of $K = 3$.

Based on the CNNP, Hu and Xiang (2022) designed a new image division (Fig. 2), to enable the CNNP to use more image pixels as context. The cover image is first divided into dot sets and cross sets. Subsequently, the cover image is divided by the 2×2 gray blocks and white blocks.

Although the CNNP effectively expands the range of pixels for prediction, the method can use only more local pixels instead of the pixels of the

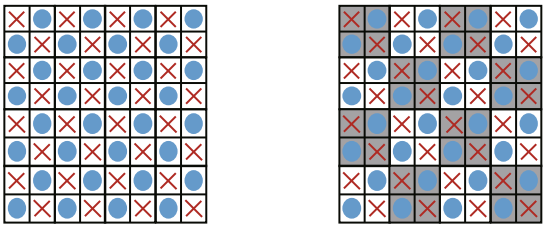

Fig. 2 Hu and Xiang (2022)'s image division method

whole image for prediction because of the disadvantages of CNNs in processing global information (Dosovitskiy et al., 2021).

2.4 PVO and PEO strategies

Li et al. (2013) proposed PVO to sort the pixel values in the blocks divided for embedding. Ou et al. (2014) proposed PVO-k by adopting more pixels in the divided block. Peng et al. (2014) proposed IPVO by exploiting image redundancy. He and Cai (2021) proposed a dual pairwise PEE strategy to fully exploit the potential of pairwise PEE. In pairwise PEE (Ou et al., 2016), instead of two bits of data, one of the combinations of the bits 00, 01, and 10 is embedded into a pair of expandable errors. In this way, the embedding distortion can be reduced at the cost of embedding $\log_2 3$ instead of two bits. Qu and Kim (2015) proposed a novel pixel-based PVO (PPVO). In their method, each pixel is predicted using its sorted context pixels. In this way, almost all the pixels can be predicted, and hence, the number of prediction errors is largely increased. Correspondingly, the number of prediction errors capable of carrying data bits is increased.

The main idea of PEO (Zhang et al., 2020b) is to exploit the intercorrelations of prediction errors by combining PEE with the recent RDH technique of PVO. Specifically, the prediction errors within an image block are first sorted. Then, the block's maximum and minimum prediction errors are predicted and modified for data embedding. By the proposed approach, image redundancy is better exploited, and promising embedding performance is achieved.

3 The proposed method

In this section, we begin with the framework of our method. Then, we give the design of the transformer predictor. Finally, we introduce the adaptive embedding strategy. The embedding process is shown in Fig. 3.

3.1 Framework: the embedding and extraction processes

In this study, we propose a new RDH method (Fig. 3). The embedding framework consists of two modules: a transformer predictor and an adaptive embedding strategy. It should be noted that the predictor mentioned in the framework has been trained.

In the prediction process, we refer to our image division method and divide the image into four sets (dot set I_1, cross set I_2, triangle set I_3, and star set I_4). At the same time, we divide the secret data into four sets as W_1, W_2, W_3, and W_4. Take set I_4 as

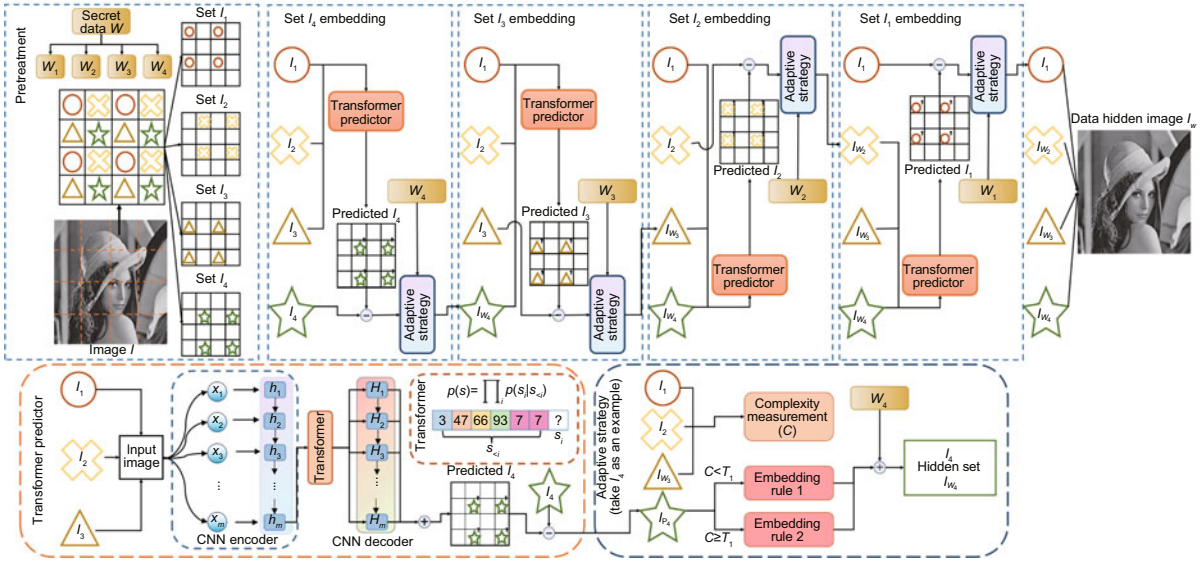

Fig. 3 Framework of the proposed reversible data hiding (RDH) method. The framework consists of a predictor based on a transformer and an embedding strategy including multiple embedding methods. The detailed processes of the predictor and embedding strategy are listed separately (CNN: convolutional neural network)

an example. We use I_1, I_2, and I_3 as the context for our transformer predictor to generate the prediction set of I_4. The specific generation process is as follows: first, mask I_4; then, convert the remaining three sets into codes through the encoder and input the sequence of codes into the transformer for content reasoning of the masked region; finally, generate an image of the masked area through the generator. The prediction error of I_4, I_{P_4}, is calculated from I_4 and the generated prediction set of I_4. Our adaptive embedding strategy then uses the prediction error to embed hidden data. In the embedding process, we first divide I_{P_4} into several non-overlapping blocks and calculate each block's complexity. Finally, according to the complexity measurement, we decide the embedding region and embedding order, select different embedding rules to embed hidden data W_4, and gain the secret part I_{W_4}. Similarly, I_{W_3}, I_{W_2}, and I_{W_1} are obtained by the same method. The final secret image I_W can be obtained by combining I_{W_1}, I_{W_2}, I_{W_3}, and I_{W_4}.

The extraction process of secret data is similar to the embedding process. We first use I_{W_2}, I_{W_3}, and I_{W_4} to generate $\hat{I}_1$, which is in cooperation with I_{W_1}, to extract the information W_1 and recover the original dot set image I_1. The same process recovers the other sets. It should be noted that the extraction process must be reversed to the embedding process. For example, if the embedded part starts from I_1, the extraction must start from I_{W_4}. In the processes of embedding and extraction, the parts used for prediction and complexity calculation are not changed. Therefore, reversibility can be guaranteed.

3.2 Image division and predictor mask

We provide an image division method for our predictor because our predictor can use global pixels for prediction. The more pixels the predictor can use, the better the prediction performance. The rhombus predictor divides an image into dot sets and cross sets, and each pixel can be predicted using four neighboring pixels. Hu and Xiang (2022) adopted a new division method (Fig. 2) to improve the rhombus predictor, which can provide nearly three quarters of the image information as the context. First, the image was divided into dot sets and cross sets. Subsequently, the image was divided by the 2×2 gray blocks and white blocks. From the local information, the image division method of Hu and Xiang (2022) can use four neighboring pixels as context. From the global information, the image division method can use nearly three quarters of the image.

Referring to Wang XY et al. (2021), we propose a method to divide an image into four parts. Fig. 4 demonstrates the proposed division method using a 6×6 image. The image is divided into four parts, including dot set I_1, cross set I_2, triangle set I_3, and star set I_4. From the local information, our image division method can use eight neighboring pixels as context, which is twice as much as those used in Hu and Xiang (2022). From the global information, the image division method can use three quarters of the image.

In image generation, the mask is used to shield the area to be generated, e.g., zeroing the area or turning it into random noise. Image generation aims to reconstruct the masked part to the original image. Usually, the masked area is a continuous block of pixels, e.g., a dog or an airplane in the image.

However, since we divide the image into four sets, the pixels in each set are discontinuous, and continuing to use the usual mask method will weaken the correlation between pixels. Therefore, we set the mask to a set of pixels with a step size of 2.0. Fig. 5a shows the original image, and Fig. 5b shows our mask method. It is not difficult to see that this mask is the same as that in our image division method. Specifically, we use all the pixels with a step size of 2.0, starting from $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$ to obtain four masks, corresponding to I_1, I_2, I_3, and I_4, respectively, obtained by image division. In other words, we can directly reconstruct target pixels by using the modified mask.

3.3 Transformer predictor

3.3.1 Design of transformer predictor

As stated in Section 2.3, a high-performance predictor should be designed to break the limitation

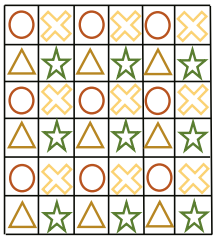

Fig. 4 Our image division method

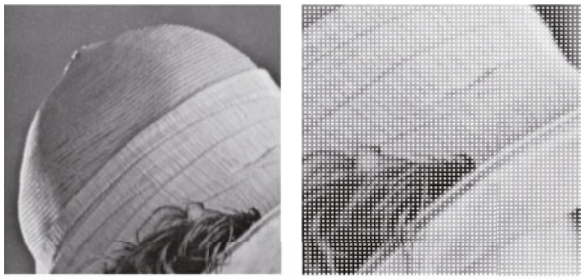

Fig. 5 Parts of the test image Lena: (a) unmasked original image; (b) masked image after using our proposed mask method. The mask in (b) is used for final training and testing, which is the collection of pixels starting from (0,0) with a step size of 2.0. The white area is the masked pixels. In addition, there are three other masks starting from (0,1), (1, 0), and (1,1), respectively

of neighboring pixels. Unlike CNNP, we use the image generation method to predict the target pixels and introduce transformers into RDH.

We need a lightweight image generation algorithm that does not require large datasets to design our predictor because RDH has only a small number of datasets. We design our predictor by following a vector quantized generative adversarial network (VQGAN) (Esser et al., 2021), which expresses the consideration of an image in the form of a sequence, instead of as pixels, and learns the image composition according to the index of the code. The structure of our predictor is composed of two parts: a convolutional network consisting of an encoder E and a decoder G, and a content reasoning module with transformers. The former is used to build codebook and generate images, and the latter is used to reason about the contents of the mask. Our predictor is designed to generate more minor prediction errors, which can achieve data expansion and embedding more effectively, thus improving the embedding performance.

3.3.2 Learning an effective codebook of images

In this part, we learn a convolutional network consisting of an encoder E and a decoder G to represent an image $\boldsymbol{x} \in \mathbb{R}^{H\times W\times 3}$ to a spatial collection of codebook entries $\boldsymbol{z}_q \in \mathbb{R}^{h\times w\times n_z}$, where n_z is the dimensionality of codes, and H and W represent the height and width, respectively (h and w are the corresponding values in the codebook). An equivalent representation is a sequence of hw indices, which specify the respective entries in the learned codebook (Esser et al., 2021). More precisely, we follow Esser et al. (2021) and approximate a given image $\boldsymbol{x}$ by $\hat{\boldsymbol{x}} = G(\boldsymbol{z}_q)$. We obtain $\boldsymbol{z}_q$ using the encoding $\hat{\boldsymbol{z}} = E(\boldsymbol{x}) \in \mathbb{R}^{h\times w\times n_z}$ and a subsequent element-wise quantization $q(\cdot)$ of each spatial code $\hat{\boldsymbol{z}}_{ij} \in \mathbb{R}^{n_z}$ onto its closest codebook entry $\boldsymbol{z}_k$:

$$\boldsymbol{z}_q = q(\hat{\boldsymbol{z}}) := \underset{\boldsymbol{z}_k\in Z}{\arg\min} \|\hat{\boldsymbol{z}}_{ij} - \boldsymbol{z}_k\| \in \mathbb{R}^{h\times w\times n_z}. \quad (1)$$

$\boldsymbol{z}_q$ is readily recovered and decoded to an image $\hat{\boldsymbol{x}}$ using the following expression:

$$\hat{\boldsymbol{x}} = G(\boldsymbol{z}_q) = G(q(E(\boldsymbol{x}))). \quad (2)$$

Due to our image division, we need to modify the part of feature extraction to adapt discontinuous mask regions. To avoid problems such as a VQGAN (Esser et al., 2021) being gradually influenced in the first place by neighboring pixels in the deep CNN layers or an image generative pre-trained transformer (iGPT) (Chen et al., 2020) that loses important context details due to the large-scale down-sampling, we use a stacked ($\times 4$) CNN embedding as our encoder. In each block, the 1×1 filter and layer norm are applied for nonlinear projection, followed by a partial convolutional layer (Zheng et al., 2022) that uses a 2×2 filter with stride two to extract visible information. The original partial convolution operation is done as follows:

$$E(\boldsymbol{x}) = W_p\left(x_p m_p \frac{1}{\sum m_p}\right) + b, \quad (3)$$

where W_p contains the convolution filter weights, b is the corresponding bias, and x_p and m_p are the feature values and mask values, respectively, in the current convolution window.

We continue to use the loss function in Esser et al. (2021):

$$L = L_{\mathrm{Re}} + L_{\mathrm{sgo}} + L_{\mathrm{commitment}}, \quad (4)$$

$$L_{\mathrm{R}} = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2, \quad (5)$$

$$L_{\mathrm{sgo}} = \|\mathrm{sg}[E(\boldsymbol{x})] - \boldsymbol{z}_q\|_2^2, \quad (6)$$

$$L_{\mathrm{commitment}} = \|\mathrm{sg}[\boldsymbol{z}_q] - E(\boldsymbol{x})\|_2^2, \quad (7)$$

where L_{Re} is the reconstruction loss, L_{sgo} is the stop-gradient operation loss, $\mathrm{sg}[\cdot]$ denotes the stop-gradient operation, and $L_{\mathrm{commitment}}$ is the so-called commitment loss (van den Oord et al., 2017).

3.3.3 Reasoning content with transformer

Our transformer encoder is built on the standard qkv self-attention (SA) model (Vaswani et al., 2017). With encoder E and decoder G, we can represent images in terms of a sequence of indices ($s \in \{0, 1, \ldots, |Z|-1\}^{hw}$) from the codebook, where s is equivalent to $\boldsymbol{z}_q = q(E(\boldsymbol{x})) \in \mathbb{R}^{h\times w\times n_z}$ given by image $\boldsymbol{x}$, and Z is the codebook. The input of the transformer is as follows:

$$\mathbf{MSA}(s) = [\mathrm{SA}_1(s); \mathrm{SA}_2(s); \ldots; \mathrm{SA}_h(s)], \tag{8}$$

$$[\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}] = \boldsymbol{W}_{qkv} s, \tag{9}$$

$$\boldsymbol{A} = \mathrm{softmax}\left(\boldsymbol{q}\boldsymbol{k}^{\mathrm{T}}/\sqrt{C_h}\right), \tag{10}$$

where $\boldsymbol{W}_{qkv}$ is the learned parameter to refine feature s for query $\boldsymbol{q}$, key $\boldsymbol{k}$, and value $\boldsymbol{v}$, and $\boldsymbol{A}$ is the dot similarity which is scaled by the square root of feature dimension C_h. Then, we compute a weighted sum over all values $\boldsymbol{v}$ via

$$\mathrm{SA}(s) = \boldsymbol{A}\boldsymbol{v}. \tag{11}$$

With indices $s_{<i}$, the transformer learns to predict the distribution of the possible next indices, i.e., $p(s_i|s_{<i})$ to compute the likelihood of the full representation as $p(s) = \prod_i p(s_i|s_{<i})$. This allows us to directly maximize the log-likelihood of the data representations:

$$L_{\mathrm{transformer}} = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}[-\log_2 p(s)]. \tag{12}$$

3.4 Adaptive embedding strategy

Our adaptive embedding strategy includes a complexity measurement and multiple embedding rules. The strategy matches appropriate embedding rules for different areas of embedding according to target requirements and reduces embedding distortion.

3.4.1 Complexity measurement

The complexity (He and Cai, 2021) shown in Fig. 6 is calculated as the sum of the absolute differences between two consecutive pixels in the horizontal or vertical direction.

However, only one quarter of the image is used for embedding, with one-fourth pixels used for prediction every time. Hence, we propose a complexity measurement to use the remaining invariant pixels. According to the proposed image division method, more context can be provided to undertake the complexity measurement.

Fig. 7 shows our proposed complexity measurement. Since the image is divided into four sets, we take the dot set as an example. We use the block complexity measurement method in Section 2.2, and a complexity measurement Com_i is computed for each block. As shown in Fig. 7, we first divide the image into the dot set and non-dot set. To avoid affecting reversibility, we use a non-dot set to calculate the complexity of the dot set. Moreover, we use a method similar to that proposed by He and Cai (2021) to use the neighboring pixel blocks. Com_i is calculated as the sum of the absolute differences between two pixels of the same set in the horizontal or vertical direction. For boundary blocks, we simply double the value derived from the nearest full-enclosing pixels. We follow He and Cai (2021) and mark the pixels of the cross set as $c_1, c_2, \ldots, c_n$ from top to bottom and from left to right. Similarly, the pixels in the triangle set I_3 and star set I_4 are marked as $t_1, t_2, \ldots, t_n$ and $s_1, s_2, \ldots, s_n$, respectively. The complexity measurement formula of the neighboring pixel blocks $\mathrm{Com}_{\mathrm{out}}$ is expressed as follows:

$$\mathrm{Com}_{\mathrm{out}} = \sum_{i}^{n-1} |x_{i+1} - x_i|, \tag{13}$$

where n is the number of pixels in each set, and $x \in \{c, t, s\}$. The complexity of pixels in the target

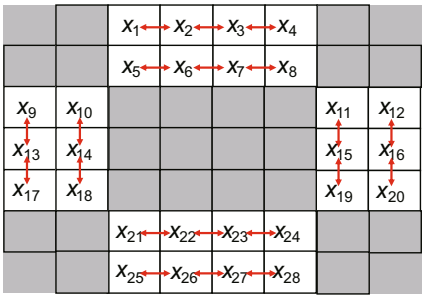

Fig. 6 He and Cai (2021)'s complexity measurement

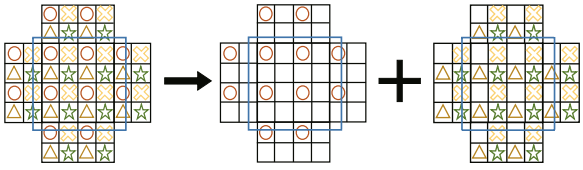

Fig. 7 Our complexity measurement

block $\mathrm{Com}_{\mathrm{in}}$ is derived using the same approach:

$$\mathrm{Com}_{\mathrm{in}} = \sum_{i}^{m-1} |x_{i+1} - x_i|, \tag{14}$$

where m is the number of pixels in each set, and $x \in \{c, t, s\}$. The complexity measurement Com_i of each block is calculated as follows:

$$\mathrm{Com}_i = \mathrm{Com}_{\mathrm{out}} + \mathrm{Com}_{\mathrm{in}}. \tag{15}$$

3.4.2 Embedding strategy of multiple embedding rules

The proposed embedding strategy is the first in the field to use multiple embedding rules, including IPEO and HS. IPEO is a new embedding rule based on IPVO, which sorts the prediction error rather than the pixel value. We first divide the image into several non-overlapping blocks. For each split block, the prediction errors are sorted by values from small to large, to obtain $e_{\sigma(1)}, e_{\sigma(2)}, \ldots, e_{\sigma(n)}$. Here, $\sigma(\cdot)$ represents the location sorted by the size of the prediction error. IPEO calculates the second-order prediction error from the prediction error and embeds the data. The embedding process is similar to IPVO. The calculation formula of the maximum error pair $(d^1_{\max}, d^2_{\max})$ is as follows:

$$\begin{cases} d^1_{\max} = e_{r_1} - e_{p_1}, \\ d^2_{\max} = e_{r_2} - e_{p_2}, \end{cases} \tag{16}$$

where

$$\begin{cases} r_1 = \min(\min(\sigma(n), \sigma(n-1)), \sigma(n-2)), \\ p_1 = \max(\min(\sigma(n), \sigma(n-1)), \sigma(n-2)), \\ r_2 = \min(\max(\sigma(n), \sigma(n-1)), \sigma(n-2)), \\ p_2 = \max(\max(\sigma(n), \sigma(n-1)), \sigma(n-2)). \end{cases}$$

According to Eq. (16), the expandable second-order errors can be determined as $d^1_{\max} \in \{0, 1\}$ and $d^2_{\max} \in \{0, 1\}$. As a result, our two-dimensional (2D) mapping is shown in Fig. 8, which describes the transformation of $(d^1_{\max}, d^2_{\max})$, where the second-order error pair $(1, 1)$ is embedded with $\log_2 3$ bits. The other three types of second-order errors are embedded with 1, 1, and 0 bits, separately.

However, if the prediction error is small enough, using IPEO will waste the embedding capacity. HS is an excellent alternative to IPEO. HS directly embeds secret data into prediction error, and the embedding

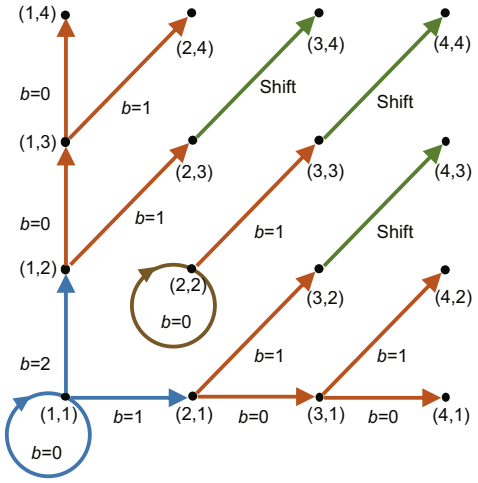

Fig. 8 Four types of second-order prediction error pairs of improved prediction error ordering (IPEO). The embedded capacities of the blue, brown, red, and green types are $\log_2 3$, 1, 1, and 0 bits, respectively. References to color refer to the online version of this figure

capacity of each pixel is 1 or 0 bits. The specific description is as follows:

$$e'_i = \begin{cases} 2e_i + b, & \text{if } e'_i \in [-T, T], \\ e_i + T, & \text{if } e'_i \in (T, +\infty), \\ e_i - T, & \text{if } e'_i \in (-\infty, -T), \end{cases} \tag{17}$$

where $b \in \{0, 1\}$ denotes the embedded data and T is an artificially set threshold.

An effective embedding strategy can ensure invisibility and increase the embedding capacity as much as possible. In addition, it can adapt to images with different complexities. The proposed IPEO can significantly improve the imperceptibility, but it will lead to the loss of embedding capacity when dealing with images with low complexity. HS can embed a large amount of data when the prediction error is small. Still, it will produce many invalid translations when the image is more complex, resulting in small embedding capacity and poor imperceptibility.

Therefore, we propose an embedding strategy consisting of IPEO and HS, and adopt complexity to select embedding rules. We define a range for smooth, textured, and non-embeddable blocks with our complexity measurement. HS embeds data into smooth blocks, while IPEO is used for textured blocks. The pixels in the non-embeddable blocks do not change. This strategy realizes the adaptive matching of embedding rules in different embedding areas.

3.5 Auxiliary information embedding

In practical applications, some auxiliary information is needed to extract the hidden information and recover the original image at the receiver end. The pixels in the cover image with values 255 and 0 are changed to 254 and 1 to avoid overflow and underflow errors, respectively. The location map (Howard and Vitter, 2016) is used to determine whether a pixel valued 1 (254) should be changed to 0 (255) or not. The auxiliary information, as in He and Cai (2021), includes block size (4 bits), thresholds (T_1, T_2) (24 bits), end position (16 bits), and length of the compressed location map (16 bits).

4 Experiment

4.1 Details of transformer predictor training

4.1.1 Training parameters

As described in Section 3.2, we modified the usual mask method to make it a set of pixels with a step size of 2.0 (as shown in Fig. 5). Considering the situation that if the mask step is set to 2.0 in the actual initial training process and discontinuous pixels are directly used as the context in the training process, the local minimum problem might occur, and the initial step size of the mask was set to N $(N > 2)$. N will gradually decrease with the training process until it finally decreases to 2. Expressly, N was initially set to 8. Compared with the direct setting of 2, it can use consecutive pixels to generate target pixels when $N = 8$. Then, we set $N = 4$ as the transition and finally fix the step size to $N = 2$. This method of decreasing the step size can avoid the local minimum problem and improve the training speed. When $N = 2$, the mask method can ensure that it is the same as the image division method described in Section 3.2. In addition, the coding stage encoding images of size $H \times W$ into discrete codes of size $(H/f) \times (W/f)$ was denoted by a factor f. We followed the parameters in Esser et al. (2021) and set $f = 16$.

4.1.2 Datasets used for training

We trained our models on various datasets, including CelebAHQ (Liu et al., 2015; Karras et al., 2018), FFHQ (Karras et al., 2019), Places2 (Zhou et al., 2018), and ImageNet (Russakovsky et al., 2015), and saved the parameters of the models on different datasets. In addition, we used six standard test images for test. Specifically, the six standard test images were 512×512 grayscale images, including Lena, Barbara, Boat, Elaine, Lake, and Peppers. We used the models trained from four different datasets to test the pictures and recorded the mean squared error (MSE) of models with different parameters on the test set. As shown in Table 1, it is not difficult to see that the model using ImageNet as the training set is the best.

Table 1 Mean squared error (MSE) of the absolute prediction errors in the test sets under four datasets

Dataset	MSE	Dataset	MSE
CelebAHQ	41.28	Places2	45.69
FFHQ	38.77	ImageNet	32.46

CelebAHQ: Liu et al. (2015), Karras et al. (2018); FFHQ: Karras et al. (2019); Places2: Zhou et al. (2018); ImageNet: Russakovsky et al. (2015)

It was mentioned by Zhang et al. (2020b) that the prediction error determines the embedding performance, so the prediction performance of the PEO method can directly reflect the quality of the predictor. We did not compare the predictor with other predictors separately, but evaluated the whole RDH method in Section 4.3.

4.2 Use of complexity

We modified the complexity measurement of He and Cai (2021) to adapt it to our predictor, so we compared our work with only He and Cai (2021) in terms of the complexity part. We kept the predictor and embedded information unchanged, and evaluated the two methods by changing only the embedding capacity synchronously. The peak signal-to-noise ratio (PSNR) was the evaluation standard. Fig. 9 shows the complexity of our work and that of He and Cai (2021). Our method divided the image into 4×4 pixel blocks. Specifically, four pixels in each pixel block can be used to embed information, while the remaining 12 pixels can be used for complexity calculation. In addition, we used half of the neighboring pixel block to calculate the complexity, i.e., 24 pixels. Our PSNR was higher than that of He and Cai (2021) when the embedding capacity was small.

4.3 Experimental results

In this subsection, we conducted quantitative and qualitative experiments to evaluate the embedding performance of the proposed RDH method by comparing it with several state-of-the-art works, including CNN PEO (Hu and Xiang, 2022), location-based PVO (LPVO) (Zhang et al., 2020a), dual pairwise PEE (He and Cai, 2021), pairwise IPVO (Dragoi et al., 2018), and pairwise PEE (Ou et al., 2013). For the proposed RDH method, the block size that embeds data was $N = 4 \times 4$, meaning that each block was embedded using four pixels at a time. The complexity thresholds T_1 and T_2 were set to 2 and 10, respectively, and T in HS was set to 5. For other state-of-the-art works, the implementation details can be referred to the corresponding papers.

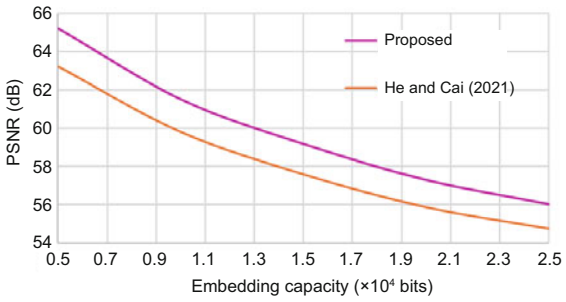

Fig. 9 Comparison of complexity measurements between our work and He and Cai (2021) (PSNR: peak signal-to-noise ratio)

By comparing the PSNR of data hiding images, the embedding performance of the proposed RDH method and the latest research results were evaluated. Under the same embedding capacity, the higher the PSNR, the better the RDH method's performance. Table 2 shows the PSNR values of our RDH method and several state-of-the-art works when the embedded capacity is 10 000 or 20 000 bits. For image Lena, when the embedding capacity is 10 000 bits, the PSNR of the proposed RDH method is as high as 62.25 dB, and when the embedding capacity is 20 000 bits, the PSNR is as high as 58.58 dB. To further compare the performance verse embedding capacity, we conducted several experiments using different images. Fig. 10 shows the changing trend of PSNR values in the six standard test images. Comparing the results in Fig. 10 and Table 2, we can find that in all the cases, the PSNR value of the proposed RDH method exceeds that of the most advanced works. This shows that the proposed RDH method can obtain satisfactory results.

Table 2 Peak signal-to-noise ratio (PSNR) values of six classic images of the test set generated by the proposed reversible data hiding (RDH) method, CNN PEO, LPVO, dual pairwise PEE, pairwise IPVO, and pairwise PEE with embedding capacity of 10 000 or 20 000 bits

Image	PSNR (dB), embedding capacity 10 000 bits					
	Ours	CNN PEO	LPVO	Dual pairwise PEE	Pairwise IPVO	Pairwise PEE
Lena	62.25	61.70	61.24	61.21	60.87	59.77
Barbara	62.79	62.23	61.35	61.17	60.71	59.49
Boat	60.36	59.81	59.32	59.14	58.41	57.56
Elaine	60.70	59.55	59.18	58.90	58.16	58.10
Lake	61.88	61.24	60.60	60.65	59.98	58.67
Peppers	61.04	60.10	59.97	59.61	59.35	56.25
Average	61.50	60.77	60.28	60.11	59.58	58.31
Image	PSNR (dB), embedding capacity 20 000 bits					
	Ours	CNN PEO	LPVO	Dual pairwise PEE	Pairwise IPVO	Pairwise PEE
Lena	58.58	57.96	57.74	57.68	57.12	56.29
Barbara	59.13	58.54	57.35	57.41	56.78	56.25
Boat	56.29	55.72	54.97	55.00	54.19	53.37
Elaine	55.70	54.93	54.33	54.18	53.43	52.92
Lake	57.10	56.47	55.63	55.85	54.84	53.76
Peppers	57.01	55.83	55.98	56.00	55.35	52.86
Average	57.30	56.58	56.00	56.02	55.29	54.24

CNN PEO: Hu and Xiang (2022); LPVO: Zhang et al. (2020a); dual pairwise PEE: He and Cai (2021); pairwise IPVO: Dragoi et al. (2018); pairwise PEE: Ou et al. (2013). CNN: convolutional neural network; PEO: prediction error ordering; LPVO: location-based pixel value ordering; PEE: prediction error expansion; IPVO: improved pixel value ordering

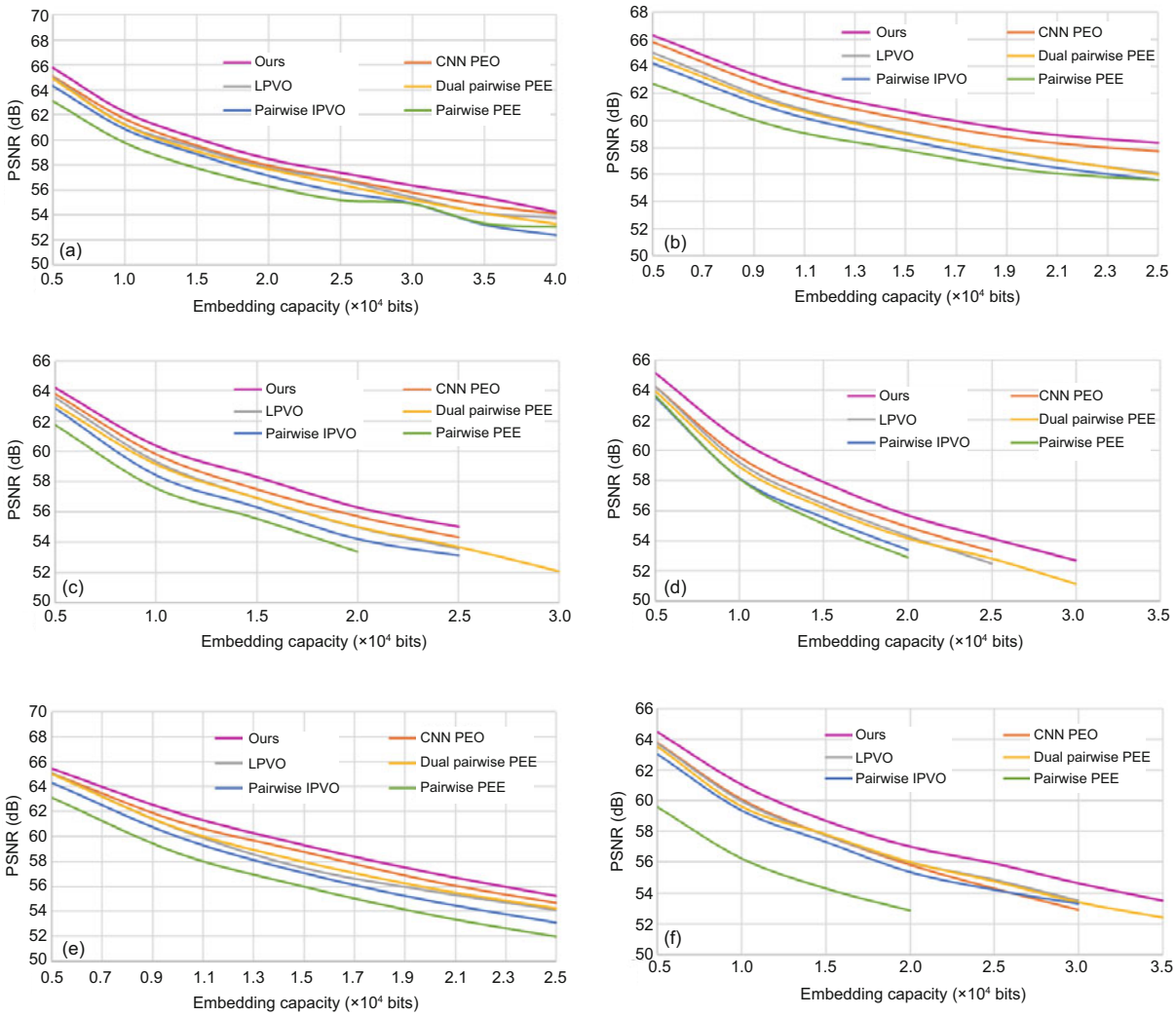

Fig. 10 Comparison of performances between the proposed RDH method and CNN PEO, LPVO, dual pairwise PEE, pairwise IPVO, and pairwise PEE on Lena (a), Barbara (b), Boat (c), Elaine (d), Lake (e), and Peppers (f). CNN: convolutional neural network; PEO: prediction error ordering; LPVO: location-based pixel value ordering; PEE: prediction error expansion; IPVO: improved pixel value ordering; RDH: reversible data hiding; PSNR: peak signal-to-noise ratio. References to color refer to the online version of this figure

5 Conclusions and future work

In this paper, we proposed a new RDH method, including a transformer predictor and an adaptive embedding strategy. In the predictor part, we first introduced a new image division method, proving that it could use more image information as the context for prediction. Besides, we confirmed that the new transformer predictor can better improve the prediction performance by comparing it with those typical predictors because of the ability to handle long-distance information. The transformer predictor successfully extends the range of pixels for prediction from neighboring pixels to global ones.

In the embedding part, we first proposed a complexity measurement with pixels in the target blocks to sort the pixel blocks. Thereafter, we developed an IPEO rule. Finally, we provided an embedding strategy including multiple embedding rules, and then more error pairs were generated for embedding data. Experimental results have shown that the embedding performance of the proposed RDH method was satisfactory and better than those of the current state-of-the-art works. For the test images, the average PSNR was 61.50 dB after hiding 10 000 bits, which exceeds those of the recently reported CNN PEO (Hu

and Xiang, 2022), LPVO (Zhang et al., 2020a), dual pairwise PEE (He and Cai, 2021), pairwise IPVO (Dragoi et al., 2018), and pairwise PEE (Ou et al., 2013) methods by 0.73, 1.22, 1.39, 1.92, and 3.19 dB, respectively.

The RDH method that we proposed is the first to use global pixels and an adaptive strategy with multiple embedding rules. In future works, we consider that there is room for improvement of the prediction performance using better deep learning methods as predictors and designing more effective strategies.

Contributors

Zhigao LU designed the research, processed the data, and drafted the paper. Weike YOU helped organize the paper. Xiaofei FANG drew the figures and checked the paper. Zhigao LU, Weike YOU, and Linna ZHOU revised and finalized the paper.

Compliance with ethics guidelines

Linna ZHOU, Zhigao LU, Weike YOU, and Xiaofei FANG declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Chen M, Chen ZY, Zeng X, et al., 2010. Model order selection in reversible image watermarking. *IEEE J Sel Top Signal Process*, 4(3):592-604. https://doi.org/10.1109/JSTSP.2010.2049222

Chen M, Radford A, Child R, et al., 2020. Generative pretraining from pixels. Proc 37th Int Conf on Machine Learning, Article 158.

Coltuc D, 2011. Improved embedding for prediction-based reversible watermarking. *IEEE Trans Inform Forens Secur*, 6(3):873-882. https://doi.org/10.1109/TIFS.2011.2145372

Coltuc D, 2012. Low distortion transform for reversible watermarking. *IEEE Trans Image Process*, 21(1):412-417. https://doi.org/10.1109/TIP.2011.2162424

Cox IJ, Miller ML, Bloom JA, 2002. Digital Watermarking. Morgan Kaufmann, San Francisco, USA.

Dosovitskiy A, Beyer L, Kolesnikov A, et al., 2021. An image is worth 16 × 16 words: transformers for image recognition at scale. Proc Int Conf on Learning Representations.

Dragoi IC, Caciula I, Coltuc D, 2018. Improved pairwise pixel-value-ordering for high-fidelity reversible data hiding. Proc 25th IEEE Int Conf on Image Processing, p.1668-1672. https://doi.org/10.1109/ICIP.2018.8451299

Esser P, Rombach R, Ommer B, 2021. Taming transformers for high-resolution image synthesis. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.12873-12883. https://doi.org/10.1109/CVPR46437.2021.01268

Goodfellow I, Pouget-Abadie J, Mirza M, et al., 2014. Generative adversarial nets. Proc 27th Int Conf on Neural Information Processing Systems, p.2672-2680.

He WG, Cai ZC, 2021. Reversible data hiding based on dual pairwise prediction-error expansion. *IEEE Trans Image Process*, 30:5045-5055. https://doi.org/10.1109/TIP.2021.3078088

He WG, Cai J, Zhou K, et al., 2017. Efficient PVO-based reversible data hiding using multistage blocking and prediction accuracy matrix. *J Vis Commun Image Represent*, 46:58-69. https://doi.org/10.1016/j.jvcir.2017.03.010

Hong W, 2012. Adaptive reversible data hiding method based on error energy control and histogram shifting. *Opt Commun*, 285(2):101-108. https://doi.org/10.1016/j.optcom.2011.09.005

Howard PG, Vitter JS, 2016. Arithmetic coding for data compression. In: Kao MY (Ed.), Encyclopedia of Algorithms. Springer, New York, USA, p.145-150. https://doi.org/10.1007/978-1-4939-2864-4_34

Hu RW, Xiang SJ, 2021. CNN prediction based reversible data hiding. *IEEE Signal Process Lett*, 28:464-468. https://doi.org/10.1109/LSP.2021.3059202

Hu RW, Xiang SJ, 2022. Reversible data hiding by using CNN prediction and adaptive embedding. *IEEE Trans Patt Anal Mach Intell*, 44(12):10196-10208. https://doi.org/10.1109/TPAMI.2021.3131250

Jafar IF, Darabkh KA, Al-Zubi RT, et al., 2016. Efficient reversible data hiding using multiple predictors. *Comput J*, 59(3):423-438. https://doi.org/10.1093/comjnl/bxv067

Karras T, Aila T, Laine S, et al., 2018. Progressive growing of GANs for improved quality, stability, and variation. Proc 6th Int Conf on Learning Representations.

Karras T, Laine S, Aila T, 2019. A style-based generator architecture for generative adversarial networks. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.4401-4410. https://doi.org/10.1109/CVPR.2019.00453

Li XL, Yang B, Zeng TY, 2011. Efficient reversible watermarking based on adaptive prediction error expansion and pixel selection. *IEEE Trans Image Process*, 20(12):3524-3533. https://doi.org/10.1109/TIP.2011.2150233

Li XL, Li J, Li B, et al., 2013. High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction error expansion. *Signal Process*, 93(1):198-205. https://doi.org/10.1016/j.sigpro.2012.07.025

Liu ZW, Luo P, Wang XG, et al., 2015. Deep learning face attributes in the wild. Proc IEEE Int Conf on Computer Vision, p.3730-3738. https://doi.org/10.1109/ICCV.2015.425

Luo LX, Chen ZY, Chen M, et al., 2010. Reversible image watermarking using interpolation technique. *IEEE Trans Inform Forens Secur*, 5(1):187-193. https://doi.org/10.1109/TIFS.2009.2035975

Ou B, Li XL, Zhao Y, et al., 2013. Pairwise prediction error expansion for efficient reversible data hiding. *IEEE Trans Image Process*, 22(12):5010-5021. https://doi.org/10.1109/TIP.2013.2281422

Ou B, Li XL, Zhao Y, et al., 2014. Reversible data hiding using invariant pixel-value-ordering and prediction error expansion. *Signal Process Image Commun*, 29(7):760-772. https://doi.org/10.1016/j.image.2014.05.003

Ou B, Li XL, Wang JW, 2016. High-fidelity reversible data hiding based on pixel-value-ordering and pairwise prediction error expansion. *J Vis Commun Image Represent*, 39:12-23. https://doi.org/10.1016/j.jvcir.2016.05.005

Peng F, Li XL, Yang B, 2014. Improved PVO-based reversible data hiding. *Dig Signal Process*, 25:255-265. https://doi.org/10.1016/j.dsp.2013.11.002

Qu XC, Kim HJ, 2015. Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding. *Signal Process*, 111:249-260. https://doi.org/10.1016/j.sigpro.2015.01.002

Russakovsky O, Deng J, Su H, et al., 2015. ImageNet large scale visual recognition challenge. *Int J Comput Vis*, 115(3):211-252. https://doi.org/10.1007/s11263-015-0816-y

Sachnev V, Kim HJ, Nam J, et al., 2009. Reversible watermarking algorithm using sorting and prediction. *IEEE Trans Circ Syst Video Technol*, 19(7):989-999. https://doi.org/10.1109/TCSVT.2009.2020257

Thodi DM, Rodriguez JJ, 2007. Expansion embedding techniques for reversible watermarking. *IEEE Trans Image Process*, 16(3):721-730. https://doi.org/10.1109/TIP.2006.891046

Tian J, 2003. Reversible data embedding using a difference expansion. *IEEE Trans Circ Syst Video Technol*, 13(8):890-896. https://doi.org/10.1109/TCSVT.2003.815962

van den Oord A, Vinyals O, Kavukcuoglu K, 2017. Neural discrete representation learning. Proc 31st Int Conf on Neural Information Processing Systems, p.6309-6318.

Vaswani A, Shazeer N, Parmar N, et al., 2017. Attention is all you need. Proc 31st Int Conf on Neural Information Processing Systems, p.6000-6010.

Wang X, Ding J, Pei QQ, 2015. A novel reversible image data hiding scheme based on pixel value ordering and dynamic pixel block partition. *Inform Sci*, 310:16-35. https://doi.org/10.1016/j.ins.2015.03.022

Wang XY, Wang XY, Ma B, et al., 2021. High precision error prediction algorithm based on ridge regression predictor for reversible data hiding. *IEEE Signal Process Lett*, 28:1125-1129. https://doi.org/10.1109/LSP.2021.3080181

Weng SW, Zhang GH, Pan JS, et al., 2017. Optimal PPVO-based reversible data hiding. *J Vis Commun Image Represent*, 48:317-328. https://doi.org/10.1016/j.jvcir.2017.05.005

Weng SW, Shi YQ, Hong W, et al., 2019. Dynamic improved pixel value ordering reversible data hiding. *Inform Sci*, 489:136-154. https://doi.org/10.1016/j.ins.2019.03.032

Zhang T, Li XL, Qi WF, et al., 2020a. Location-based PVO and adaptive pairwise modification for efficient reversible data hiding. *IEEE Trans Inform Forens Secur*, 15:2306-2319. https://doi.org/10.1109/TIFS.2019.2963766

Zhang T, Li XL, Qi WF, et al., 2020b. Prediction error value ordering for high-fidelity reversible data hiding. Proc 26th Int Conf on Multimedia Modeling, p.317-328. https://doi.org/10.1007/978-3-030-37731-1_26

Zheng CX, Cham TJ, Cai JF, et al., 2022. Bridging global context interactions for high-fidelity image completion. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.11512-11522. https://doi.org/10.1109/CVPR52688.2022.01122

Zhou BL, Lapedriza A, Khosla A, et al., 2018. Places: a 10 million image database for scene recognition. *IEEE Trans Patt Anal Mach Intell*, 40(6):1452-1464. https://doi.org/10.1109/TPAMI.2017.2723009