Information Fusion 127 (2026) 103864

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

GMM: Efficient information-containing adversarial perturbation based on gradient masking method

Hanbin Lin[a,b], Wenxing Liao[b], Zhaogang Shu[b], Xiaolong Liu[a,b,c,*]

[a] College of Computer and Information Science, Fujian Agriculture and Forestry University, Fuzhou, 35000, China
[b] School of Future Technology, Fujian Agriculture and Forestry University, Fuzhou, 35000, China
[c] Center for Agroforestry Mega Data Science, Fujian Agriculture and Forestry University, Fuzhou, 35000, China

ARTICLE INFO

Keywords:
Deep neural networks
Adversarial attacks
Meaningful adversarial example

ABSTRACT

Adversarial examples have been a significant research focus since their discovery. Recent studies have applied watermarking and data hiding techniques to generate meaningful adversarial perturbations that carry specific information, further enriching the functionality of adversarial examples. However, these methods struggle to balance time complexity with attack efficacy. To address this issue, we propose the Gradient Masking Method (GMM), introducing a new perspective on generating meaningful perturbations. Unlike previous techniques that directly embed watermarks or data as adversarial distortions, GMM embeds information into adversarial perturbations by selectively blocking the updating noise at specific positions using a message mask encoded from the information. The resulting perturbations represent the binary sequence of the embedded message. This method enables processed images to exhibit adversarial properties while simultaneously serving as carriers of information. Experimental results demonstrate the efficacy of our approach. In terms of computational cost, our method significantly outperforms previous techniques without compromising attack effectiveness. We evaluated the attack success rate of the proposed method across seven widely used classifier models, comparing it with baseline and black-box attack methods. Results confirm that our method performs effectively in common attack scenarios influenced by the message mask.

The code of GMM can be found at https://github.com/Abin110/Gradient-Masking_.

1. Introduction

Deep Neural Network (DNN) models have emerged as powerful tools for various tasks. These models consist of multiple layers of interconnected nodes that learn to recognize patterns and make predictions from data. However, DNN models are vulnerable to adversarial attacks, where small, carefully crafted perturbations to the input data generate adversarial examples that can cause the model to make incorrect predictions. Originating from pioneering research by Szegedy et al. [1] in 2013, which demonstrated that imperceptible perturbations to input data could lead to misclassification, adversarial examples have since captivated the attention of researchers and practitioners alike. Adversarial attacks can broadly be categorized into white-box [1,2] and black-box [3,4] attacks according to the sufficiency of the model parameter and structure information the adversary holds. In white-box attacks, the adversary has full access to the target model, including its architecture, parameters, and gradients, enabling them to craft precise perturbations to maximize the likelihood of misclassification. Conversely, black-box attacks occur in scenarios where the adversary has limited or no access to the target model and must resort to alternative methods such as query-based or transfer-based approaches to generate adversarial examples.

Recent studies have explored the diversity of perturbation forms, including single-pixel attacks and physical-world attacks [5]. These techniques involve applying adversarial perturbations to sensor inputs in the physical world to trick systems such as self-driving cars [6] and IoT devices [7]. However, these methods primarily focus on generating perturbations that are effective in causing misclassification, without considering additional functionalities.

In recent years, researchers have started developing adversarial perturbations with extra functionalities by combining them with data hiding techniques. Meaningful Adversarial Perturbations are defined as perturbations that can function as carriers of messages, such as images or texts, instead of being pure noise. This dual functionality can be particularly useful in scenarios where the perturbation not only fools the DNN model but also conveys hidden information. For example, in a security context, meaningful perturbations could be used to embed secret messages that are imperceptible to humans but can be recovered by authorized parties. In addition, the adversarial property can be specifically

* Corresponding author.
E-mail address: xlliu@fafu.edu.cn (X. Liu).

https://doi.org/10.1016/j.inffus.2025.103864
Received 13 November 2024; Received in revised form 29 July 2025; Accepted 14 October 2025
Available online 17 October 2025

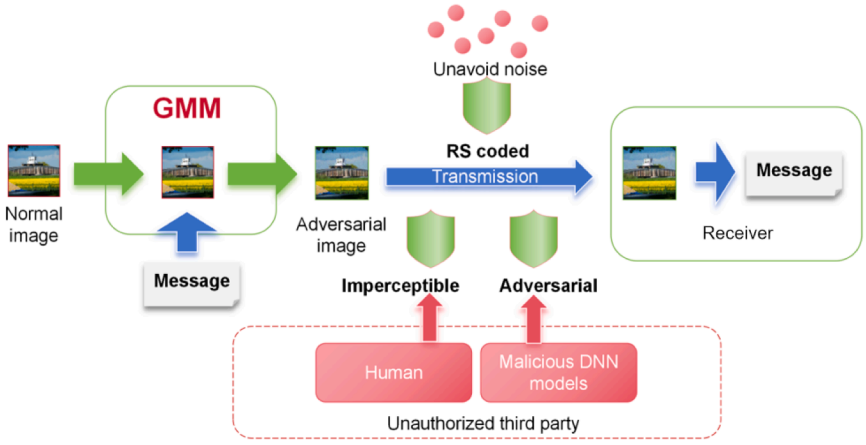

Fig. 1. Further application scenario of GMM, the image processed by GMM is able to carry information, the imperceptibility and the adversarial feature reduce the risk of awareness and DNN based tempering by other people.

designed to counter malicious DNN models that are capable of stealing secret information.

However, combining adversarial attacks with information embedded is challenging because both processes require altering the image's pixel values, and their objectives often conflict. For instance, embedding information typically requires adding consistent pixel values to the host image, while adversarial perturbations need changeable values to find the best distortion that can fool the DNN model. To address this issue, joint processing methods have been proposed. For example, Jia et al. [8] introduced the Basin Hopping Evolution (BHE) algorithm, which uses visible watermarks as meaningful adversarial perturbations by optimizing their placement on images. Building on this work, Jiang et al. proposed FAWA [9], which uses a fast differential evolution algorithm to find the optimal combination of watermark position, size, transparency, and rotation. More recently, Liang et al. [10] proposed the Basin Hopping Improvement (BHI) algorithm, which uses the Least Significant Bits (LSB) technique to embed invisible information while minimizing classification confidence. However, these approaches have limitations. The BHE and FAWA methods require significant computational resources due to the large search space, and visible watermarks can degrade image quality, especially when they obscure important content. The BHI method, while improving visual subtlety, is more time-consuming due to the increased number of iterations required. More recently, Liang et al. [11] extended their prior work by introducing ISWP (Invisible and Secure Watermark Perturbation), a more secure and robust variant of BHI.

However, these approaches require significant computational resources due to the searching space is limited within the scale of the host image and the visible watermark can degrade image quality, especially when watermarks obscure important content. In this paper, we focus on the trade off between the dual functionality and economical time complexities, and proposed Gradient Mask Method (GMM), an efficient and effective way to generate meaningful adversarial perturbations. The gradient descent-based method have been proved to be the most efficient selection for adversarial attack, we implemented Projected Gradient Descent (PGD) attack in GMM as the fundamental framework, to ensure the efficiency and efficacy. To achieve the dual-functionality of meaningful adversarial perturbations, in GMM, the ready-to-embedded information is preprocessed to a message mask array that matches the size of the image. And the message mask array is involved in each updating iteration, selectively blocking the upgraded gradients of the perturbations. The obtained adversarial perturbations are applied only on the pixel positions that refers to positive on the message mask. In addition to that, in the preprocess stage for the ready-to-embedded message, we implement Reed-Solomon (RS-coding) coding [12], an error correction code, to enhance the robustness of the embedded information. This allows the

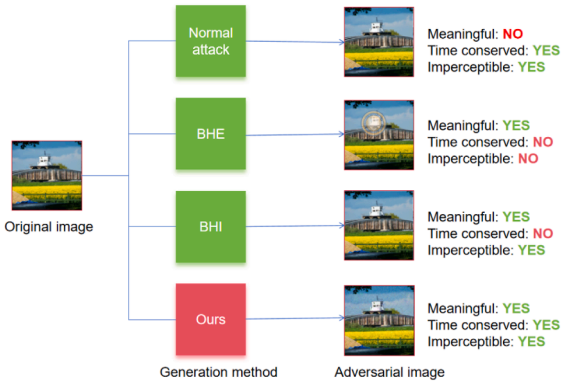

Fig. 2. Advantages of the proposed GMM compare to other methods. Comparing to normal adversarial attack, the adversarial perturbations generated by our method contains specific meaning, and requires less time complexitys compare to BHE and BHI.

embedded message to propagate without being affected by noise, ensuring its readability even when the image is distorted during transmission. The potential application scenario of the proposed method is shown in Fig. 1. The adversarial image serves as a container for the message, with its adversarial properties protecting against malicious use of DNN models, and its imperceptibility ensuring that the hidden message remains undetected. Meanwhile, the receiver can recover the message even when the image is distorted by unavoidable noise during transmission.

As shown in Fig. 2, compared to normal attack methods, the adversarial perturbations generated by our method are meaningful rather than chaotic noise. When compared to BHE, FAWA, and BHI, the time complexity of GMM is significantly lower.

In summary, this paper introduces an improved approach to generate meaningful adversarial perturbations. The key contributions of our research are outlined as follows:

(1) Compared to previous studies, we are the first to consider the computational costs of crafting meaningful adversarial perturbations. The proposed Gradient Mask Method (GMM) requires much less computation resources than baseline methods

(2) The proposed method outperformed other baseline methods in fooling deep neural network (DNN) models in experimental scenarios, with notably higher attack success rate.

(3) The generated adversarial examples exhibit dual functionality, retaining their adversarial properties while embedding the information within the host image.

2. Related works

2.1. Adversarial attack and defense

Adversarial examples–inputs subtly perturbed to mislead machine learning models–have attracted considerable attention in the deep learning community. First introduced by Szegedy et al. [1], these examples have spurred extensive research into both attack and defense strategies.

Gradient-based methods are among the most common approaches for generating adversarial examples. FGSM [2] computes the gradient of the loss with respect to the input and applies a single-step perturbation in the direction of the gradient sign. Iterative methods such as I-FGSM [5] and PGD [13] improve attack strength by applying multiple small updates, constrained within an ϵ-ball around the input to maintain imperceptibility. These methods are computationally efficient and widely adopted due to their balance between effectiveness and scalability.

In addition to gradient-based approaches, several attack methods explore optimization or feature-saliency strategies. JSMA [14] perturbs input features deemed most influential to the model's prediction, minimizing perturbation size while maximizing misclassification probability. The CW attack [15] formulates the problem as an optimization task with a tailored loss function that minimizes perturbation while suppressing the correct class confidence. DeepFool [16] iteratively shifts input points toward the nearest decision boundary, achieving high attack success with minimal distortion.

In recent years, several studies [17,18] have shifted attention to the frequency domain to extract more general and transferable image features. By selectively manipulating specific frequency bands, these methods suppress high-frequency noise while enhancing low-frequency structures, thereby improving the transferability of adversarial examples across different models and datasets.

In practice, physical-world attacks [19–21] have adapted adversarial perturbations to real-world environments, where inputs are subject to changes in lighting, perspective, and scale. Techniques such as robust printable patterns and adversarial accessories have been successfully deployed to mislead systems like autonomous vehicles and surveillance cameras, significantly extending the practical applicability of adversarial examples.

In response to the growing threat of adversarial attacks, researchers have proposed various defense strategies aimed at either detecting adversarial examples or enhancing model robustness. Input preprocessing techniques, such as JPEG compression [22] and feature squeezing [23], attempt to eliminate adversarial perturbations while preserving the semantic integrity of the original inputs. Although these approaches are computationally efficient, they often fail against adaptive attacks specifically designed to circumvent them.

To address these shortcomings, more principled defense mechanisms have been developed. Among them, adversarial training [24–26] has proven particularly effective. By training models on a mixture of clean and adversarial examples, deep neural networks can build intrinsic robustness without relying on additional preprocessing steps at inference time. This approach has demonstrated substantial gains in defending against adversarial inputs while maintaining strong performance on clean data. Complementing adversarial training, anomaly detection techniques have emerged as scalable solutions for identifying malicious or out-of-distribution inputs in real-world machine learning systems [27–30]. These methods enhance system resilience by leveraging statistical deviations, inconsistencies in latent representations, or reconstruction errors to detect potential threats that may evade conventional defenses.

Nevertheless, most existing efforts remain focused on attacking or protecting deep learning models themselves, with limited attention paid to safeguarding private data from misuse by malicious models. As the demand for large-scale datasets continues to grow, unauthorized access to sensitive information has become a critical privacy concern, necessitating defense strategies that protect both model performance and data security.

2.2. Meaningful adversarial examples

To mitigate unauthorized data access, recent researches have begun to investigate the potential for adversarial perturbations to carry meaningful information. These meaningful information typically serve as digital watermark or hidden message, which is protected by the adversarial property from DNN models. This emerging direction investigates how adversarial examples can serve not only as a tool for evasion but also as a medium for secure data embedding, which has been proven to have significant potential in digital art authentication [31].

Ding et al. [32] proposed AdMarks, an intriguing solution that embeds adversarial perturbations–generated for object detection models–into watermarks encoding false outputs. This approach leverages the attacked model itself as a decoder to extract hidden watermarks from adversarial images. However, the method's watermark decoding performance is tightly coupled with its attack success rate, which may lead to significant failures in both functionalities.

A straightforward approach to achieving dual functionality is to separate the tasks of information embedding and adversarial perturbation generation. However, these objectives often conflict, as both processes involve modifying pixel values, leading to interference and suboptimal performance.

Jia et al. [8] pioneered the concept of embedding watermarks into input images via Basin Hopping Evolution (BHE) to generate adversarial examples. Their method maximizes the discrepancy between the model's prediction on the clean image and that on the watermarked image by optimizing the watermark's position. Utilizing Basin Hopping and Evolution strategies, BHE explores a diverse solution space to discover globally optimal adversarial watermarks, thereby extending the scope of adversarial examples through watermarking techniques.

Building upon this, Jiang et al. [9] introduced FAWA, which employs a fast differential evolution algorithm to optimize watermark parameters such as position, size, transparency, and rotation, further enhancing the adversarial efficacy. Liang et al. [10] proposed the Basin Hopping Improvement (BHI) method, which embeds invisible information into host images using the Least Significant Bit (LSB) technique, achieving greater visual subtlety.

More recently, Liang et al. [11] presented the Invisible and Secure Watermark Perturbation (ISWP) method, which unifies adversarial perturbation generation and watermark embedding into a single framework. ISWP maintains imperceptibility of perturbations while preserving watermark functionality, thereby improving both security and attack effectiveness. However, the increased computational complexity results in longer processing times and higher resource demands, which may limit applicability in real-time or resource-constrained environments.

The method proposed in this work advances the field by employing a more computationally efficient algorithm that maintains high attack performance while minimizing degradation of the original image quality. The generated perturbations effectively deceive DNN models and concurrently function as robust digital watermarks.

3. Preliminaries

3.1. Reed-Solomon code

Reed-Solomon codes [12] are a class of error-correcting codes widely used in digital communication and storage systems. They are particularly effective for correcting multiple errors, including burst errors, making them essential for applications requiring high data integrity, such as satellite communication and disk drives.

In our notation, let k be the number of message symbols, n be the total number of symbols after encoding, and t be the maximum number of

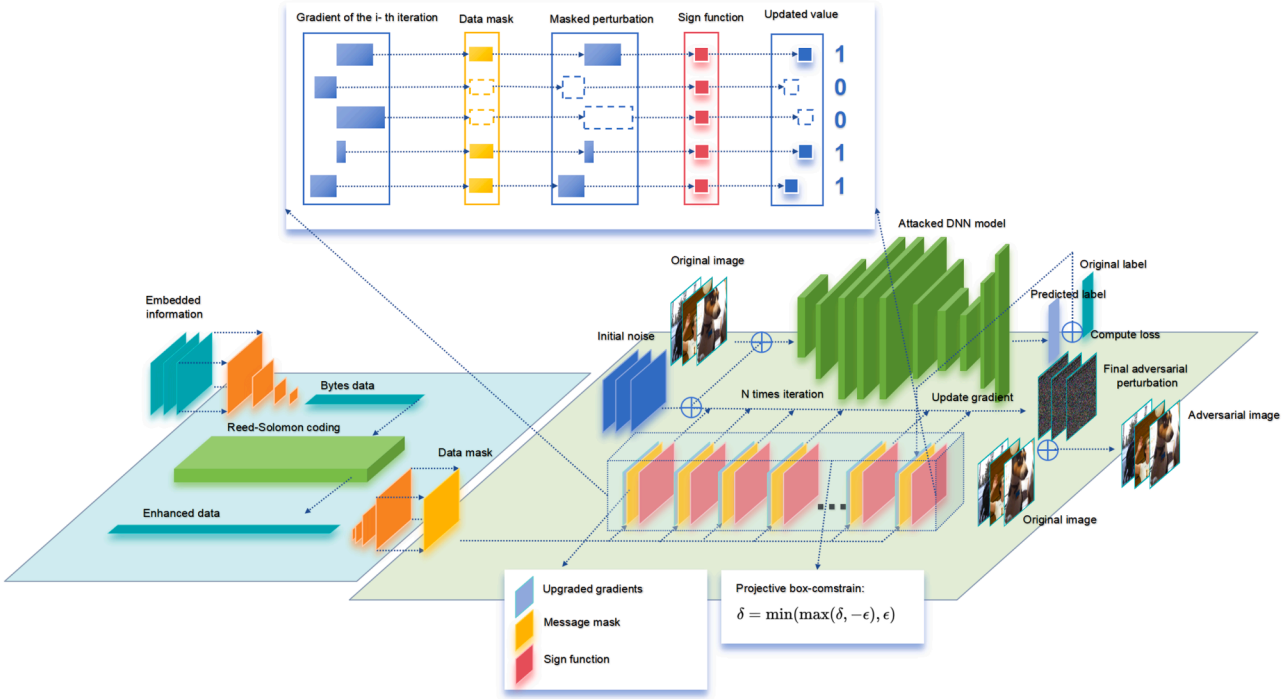

Fig. 3. Overall construction of the proposed method. It has two stages: message mask preprocessing and a gradient-based attack. First, information is encoded, Reed-Solomon [12] enhanced, and converted to a binary array message mask. And iteratively updates the perturbations using the gradient, while masked elements remain zero.

errors the code can correct. The relationship between these parameters is $n = k + 2t$.

3.1.1. Encoding process

Reed-Solomon codes use polynomial arithmetic over finite fields, specifically GF(2^m), where m is the number of bits per symbol. The encoding process involves generating n symbols from k message symbols using a systematic encoding scheme. This ensures that any k received symbols can be used to recover the original message as long as the number of errors is within the code's correction capability.

The encoding operation is represented as:

$$c(x) = m(x) \cdot g(x) \tag{1}$$

Here, $c(x)$ is the encoded codeword polynomial of degree $n-1$, $m(x)$ is the message polynomial of degree $k-1$, and $g(x)$ is the generator polynomial of degree $2t$.

The generator polynomial $g(x)$ is defined as:

$$g(x) = \prod_{i=0}^{2t-1} (x - \alpha^i) \tag{2}$$

where α is a primitive element of GF(2^m). The roots of $g(x)$ are chosen to ensure the code can correct up to t errors.

3.1.2. Decoding process

The decoding process involves identifying and correcting errors in the received codeword polynomial $r(x) = c(x) + e(x)$, where $e(x)$ is the error polynomial. The syndrome polynomial $S(x) = r(x) \bmod g(x)$ is first computed. If $S(x) = 0$, the received codeword is error-free. Otherwise, the error locator polynomial $\sigma(x)$ is derived from $S(x)$ using algorithms such as the Berlekamp-Massey algorithm or the Euclidean algorithm. The error magnitudes are then calculated using the Forney algorithm, and the errors are corrected. Finally, the message polynomial $m(x)$ is extracted from the corrected codeword $c'(x)$ by dividing it by the generator polynomial $g(x)$.

3.2. Adversarial example generation

In gradient-based attack methods, the objective is to compute the perturbations δ that maximizes the model's prediction error while keeping the perturbations imperceptible. The adversarial example can be expressed as:

$$x' = x + \delta \tag{3}$$

where x represents the input data. The optimization objective can be expressed as:

$$\begin{aligned} &\text{maximize} \;\; \mathcal{L}(\mathcal{M}(x+\delta), y) \\ &\quad \text{subject to} \|\delta\| \leq \epsilon \end{aligned} \tag{4}$$

where $\mathcal{L}(\cdot)$ represents the cost between the correct label of the clean image and the prediction of the adversarial example, and $\mathcal{M}(\cdot)$ represents the attacked model. y represents the correct label of the clean image, and ϵ limits the perturbations δ to an imperceptible range.

4. Proposed method

4.1. Problem formulation

Previous works generate meaningful adversarial perturbations by embedding visible or invisible information onto the image and processing the coordinates of the embedded information (e.g., watermark image or hidden data) on the host image. The objective goal can be expressed as follows:

$$\begin{aligned} &\text{maximize} \;\; \mathcal{L}(\mathcal{M}(x, w, i, j), y) \\ &\quad \text{subject to} \;\; i \leq W, j \leq H \end{aligned} \tag{5}$$

where i, j denote the coordinates of the embedded information, W, H denote the width and the height of the host image, and w denotes the message carried by the perturbations. The objective is to search for the best i, j that maximizes the cost between the predicted label of the attacked model $\mathcal{M}(\cdot)$ and the original label y.

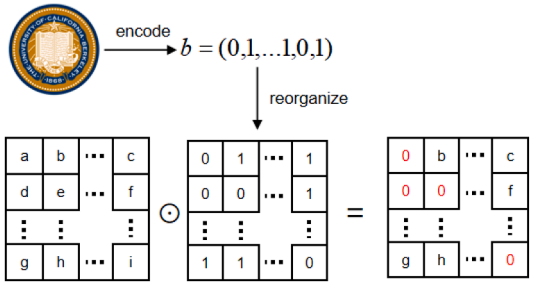

Fig. 4. Illustration of the information embedding, the logo image is encoded to binary form and reorganized to a message mask matches the size of the updated gradient, the Hadamard product of this two vector will be the updated perturbations.

These methods consistently involves evolutionary algorithm to solve the former objective goal. The time complexity are mainly depends on the population size (P) and the iterations or generations size (N). Which means the time complexity of previous approaches are at least $O(N \times P \times O(M))$, while the model evaluation takes $O(M)$ times.

Departing from former works, GMM do not embed watermarks and hidden data as the perturbations to the image. We leverage gradient-based adversarial attacks, by covering the noise by a message mask to embed the information onto the perturbations. The time complexity of GMM only relies on the iteration number N, and the overall time complexity of our method is in the maximum with $O(N \times O(M))$.

By defining the combination of the message and the perturbation as $m * \delta$ the objective goal can be presented as:

$$\begin{aligned} &maximize \;\; \mathcal{L}(\mathcal{M}(x + (m * \delta)), y) \\ &\qquad st. \|\delta\| \leq \epsilon, m = m \end{aligned} \tag{6}$$

where the x is the original image, and m is the embedded message, while δ presenting the perturbations. During the searching process, the embedded message should be consistent, while optimizing the perturbations δ to achieve dual functionality.

4.2. The proposed GMM

Based on Gradient Masking and Reed-Solomon[12] code, we propose GMM, aiming to generate meaningful adversarial perturbations, with less time complexity and more effective attack performance.

As shown in Fig. 3, our method is constructed by two stages: The first stage refers to the preprocessing of the message mask. The information to be embed is encoded into byte data first and then processed by the Reed-Solomon [12] coding to further enhance its robustness. After that, the output byte data is transformed to a binary array, and resize to a two-dimensional array fitting the size of the host image.

The second stage involves a gradient-based adversarial attack to process the image to adversarial examples, and gradient masking to embedded the information into adversarial perturbations. In each iteration the updated value will added to the perturbations according to the gradient of the objective function. Which can be formulated by:

$$\begin{aligned} \delta_\theta &= \delta_{\theta-1} + v_\theta \\ v_\theta &= \text{sign}(\nabla \mathcal{L}(x + \delta_{\theta-1})) \times \alpha \end{aligned} \tag{7}$$

where δ_θ denotes the perturbations in θ-th iteration, and v_θ denotes the upgraded value in θ-th iteration, and x denotes the original image. $\text{sign}(\cdot)$ presents the sign function that filter the gradient vector to an unit vector, and α denotes the step size of searching. While involving gradient masking, the GMM computes the Hadamard product of the updated gradient and the message mask, which would be the input of the sign function. In this case, the formulation can be presented as:

$$v_\theta = \text{sign}(\nabla \mathcal{L}(x + \delta_{\theta-1}) \odot m) \times \alpha, \;\; m_{ij} = 0, 1 \tag{8}$$

Algorithm 1: Gradient masking method.

Input: Input image $\mathbf{x} \in \mathbb{R}^{n \times m \times c}$, target labels $\mathbf{y} \in \mathbb{N}^n$, secret message $\mathbf{M}$, model $\mathcal{M}$, step size α, maximum perturbations ϵ, number of iterations N, Number of ECC symbols e

Output: Adversarial example $\mathbf{x}_{\text{adv}}$

Reed-Solomon Encoding: Encode the message $\mathbf{M}$ into byte data: $\mathbf{B} \leftarrow \{b_1, b_2 \dots b_i\}$
Enhance the byte data by RS-coding: $\mathbf{B}' \leftarrow \{b_1, b_2 \dots b_i, c_1, c_2 \dots c_e\}$
Transform the output byte data to a binary array: $\mathbf{b} \leftarrow \textbf{Transform}(\mathbf{B}')$
Resize the binary array to a two-dimensional array fitting the size of the host image: $\mathbf{m} \in \{0,1\}^{n \times m} \leftarrow \textbf{Resize}(\mathbf{b}, m, n)$
Initialize $\delta \leftarrow$ **Random**
for $i = 1$ **to** N **do**
  Compute the model output: $\mathbf{y}_{\text{pred}} \leftarrow \mathcal{M}(\mathbf{x} + \delta \odot \mathbf{m})$
  Compute the loss: $\mathcal{L} \leftarrow \mathcal{L}(\mathbf{y}_{\text{pred}}, \mathbf{y})$
  Compute the gradient of the loss with respect to the perturbations: $\nabla_\delta \mathcal{L} \leftarrow \frac{\partial \mathcal{L}}{\partial \delta}$
  GradientMasking :
  Apply the message mask to the gradient: $\nabla_\delta^{\text{masked}} \leftarrow \nabla_\delta \mathcal{L} \odot \mathbf{m}$
  Update the perturbations: $\delta \leftarrow \delta - \alpha \cdot \text{sign}(\nabla_\delta^{\text{masked}})$
  Clamp the perturbations: $\delta \leftarrow \min(\max(\delta, -\epsilon), \epsilon)$
end
Compute the adversarial example: $\mathbf{x}_{\text{adv}} \leftarrow \mathbf{x} + \delta$
return $\mathbf{x}_{\text{adv}}$

where the m denotes the message mask, the elements m_{ij} of the mask are either 1 or 0. As shown in Fig. 4, the message mask selectively blocking the updated value, constrain the perturbation in specific region. The distribution of the distorted elements of image are thus presenting the binary information, and the optimization process would not harm the information and make it unreadable.

This process is repeated for a set number of iterations N, ensuring each perturbations remains within a defined limit ϵ to maintain imperceptibility. The final obtained perturbations δ are summed by the updated value. We formulated the overall algorithm as:

$$\begin{aligned} \delta = &argmax \;\; \mathcal{L}(\mathcal{M}(x + (m * \delta)), y) \\ &st. \|\delta\| \leq \epsilon, m = m \end{aligned} \tag{9}$$

$$= \sum_{\theta}^{N} \text{sign}(\nabla \mathcal{L}(x + \delta_\theta) \odot m) \times \alpha \tag{10}$$

The algorithm is also displayed in Algorithm 1.

4.3. Discussion to the effect of masking

Instinctively, the masking operation on the gradient would affect the optimization process. In this section we will have an in-depth discussion to this problem.

4.3.1. Continual adversarial space

In the work of Li et al. [33], they first discussed the continuity of the adversarial space and concluded that the distribution of adversarial examples is not combined by discrete points, but instead is a continuous space. In our scenario, the message mask will block out some elements in the computed gradient vector, changing the searching direction.

$$\begin{aligned} &\left(\frac{\partial \mathcal{L}(e_1, e_2)}{\partial e_1}, \frac{\partial \mathcal{L}(e_1, e_2)}{\partial e_2}\right) \odot m \\ &\qquad = \left(\frac{\partial \mathcal{L}(e_1, e_2)}{\partial e_1}, 0\right) \end{aligned} \tag{11}$$

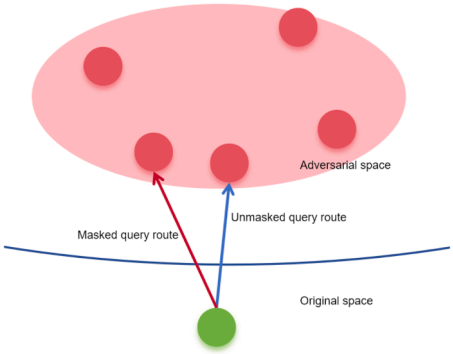

Fig. 5. The diagram of the continual adversarial space.

Fig. 6. Logos used in attack performance analysis.

Let e_1 presents the unblocked elements and e_2 presents the blocked elements, As shown in the formula above, the Hadamard product of the gradient vector and the message mask sets some elements' gradients to zero. Since the gradient functions as guidance for the query direction, as displayed in Fig. 5, the masked gradient leads to a different query route, but it still guiding the x to the adversarial space. This intriguing feature, to a certain extent, ensures the effectiveness of the outputted adversarial perturbations of GMM.

4.3.2. Optimization strategy

As mentioned earlier, we choose to involve the masking operation during the optimization process. In each iteration, predicted label $\hat{y} = \mathcal{M}(x + \delta \odot m)$ is obtained by inputting the original image plus the masked updated noise. The currently computed gradient of the unmasked elements is under the condition that the blocked elements are not perturbed at all. Therefore, the updated value of the perturbations can correctly lead the noise to the optimization goal.

Meanwhile, the masked gradient is then multiply to a sign function $sign(\cdot)$, which is widely used in gradient-based algorithms to filter out the query direction from the gradient and cooperates with a specific learning rate α. In the scenario of GMM, the sign function and learning rate also help adjust the query route in case of greater deviation and guide the perturbations correctly to the adversarial space.

5. Experiment results

5.1. Experiment setup

To thoroughly evaluate the performance of the proposed Gradient Mask Method (GMM), we conducted a series of experiments using the ILSVRC 2012 dataset [34] and a variety of pre-trained deep neural network models. The ILSVRC 2012 dataset is widely used for image classification tasks and serves as the source of clean examples. We selected 1000 images randomly from the validation set to test our method. These images were resized to 224×224 pixels to match the input requirements of the models. Additionally, we tested the generalizability of our method on the CIFAR-10 dataset [35], which contains 10,000 test images with a resolution of 32×32 pixels.

We evaluated our method on seven widely used pretrained models, all of which were trained on the ILSVRC 2012 and CIFAR-10 dataset. The pretrained weights for these models are provided by PyTorch [36]. These models include ResNet50 [37], ShuffleNetV2 [38], SqueezeNet [39], InceptionV3 [40], VGG19 [41], ResNet101 [37], and Vision Transformer (ViT) [42]. These models represent a diverse range of architectures and complexities, allowing us to assess the robustness and effectiveness of our method across different scenarios.

For the hidden messages, we used three university logos (Logo 1, Logo 2, and Logo 3) as the embedded information. Each logo was compressed to approximately 3000 bits in memory size before being encoded into binary form. Additionally, we conducted experiments using text data. For the ILSVRC 2012 dataset, we selected a short article with approximately 3000 bits, and for the CIFAR-10 dataset, we used a short sentence with 64 bits. To enhance the robustness of the embedded information, we applied Reed-Solomon (RS) coding with 85 error correction symbols [12].

The parameters for the adversarial attack were carefully chosen to ensure effective perturbations. The maximum allowed perturbation magnitude (epsilon, ϵ) was set to 0.1, and the initial step size (α) for gradient updates was set to 0.05. The number of iterations (N) for the gradient-based attack was set to 40. We used Cross-Entropy loss as the objective function to guide the optimization process.

To evaluate the performance of our method, we measured the top-1 and top-5 accuracy of the models before and after applying the adversarial perturbations. Additionally, we assessed the visual quality of the adversarial images using Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR) metrics. These metrics help quantify the imperceptibility of the perturbations.

All experiments were conducted using the PyTorch framework [36] and run on a single NVIDIA RTX 3060 Ti GPU to ensure consistent computational performance. This setup allows us to comprehensively evaluate the effectiveness, efficiency, and robustness of the proposed GMM method in generating meaningful adversarial perturbations.

5.2. Performance of the proposed method

5.2.1. The attack performance

To estimate the attack efficiency of our method, we selected three university logos (shown in Fig. 6) to be the ready-to-embedded message and tested the attack performance with each logo individually. We randomly selected 1000 images from the validation dataset to input into each model and compared the top-1 accuracy and top-5 accuracy of each model for clean images and GMM processed images.

As shown in Table 1, our method led to a substantial decrease in the classification accuracy of these models when faced with adversarial examples. Post-attack, the top-1 accuracy of all six models plummeted to under 20 %, with the highest being 19 %, and half of them dropped below 5 %, reaching as low as 1.4 %. Similarly, for top-5 accuracy, all models except InceptionV3 and ResNet101 experienced a decline to under 20 %, with the highest being 19.5 % and the lowest 3.8 %. Noteworthy is the remarkable drop in top-5 accuracy for InceptionV3 and ResNet101, reaching around 35 % and 31 % respectively, representing declines of over 55 % and 60 % from their original accuracy.

Among these DNN models, InceptionV3 and ResNet101 demonstrated greater robustness compared to the other models. As our proposed method operates as a black-box attack, fine-tuning of the attack algorithm parameters was necessary to achieve improved attack performance. When we adjusted the epsilon value (ϵ) to 0.3 while attacking InceptionV3, the top-1 accuracy decreased to approximately 7.2 % on average, and the top-5 accuracy decreased to around 20.5 % on average. However, this adjustment also resulted in a decline in the image quality of the generated adversarial examples, with the average SSIM value decreasing to around 0.97 and the PSNR value decreasing to approximately 48.0. Similarly, changing the epsilon value to 0.3 while attacking ResNet101 led to an average top-1 accuracy reduction to approximately 3.9 %, and the top-5 accuracy decreased to around 13.7 %. Additionally, the average SSIM value decreased to around 0.97, and the PSNR value decreased to approximately 48.0.

Table 1
Attack Performance in ILSVRC 2012 dataset with LOGO 1, 2 and 3, comparing the prediction accuracy of targeted model with original and adversarial images.For better comparison, the reduced classification accuracy caused by our adversarial images are noted in **bold**.

	ShuffleNetV2	SqueezeNet	ResNet50	InceptionV3	VGG19	ResNet101	ViT
Logo 1							
Top1 accuracy(%)	60.20/**1.40**	58.00/**1.60**	74.30/**6.60**	69.70/**18.10**	72.90/**3.50**	77.20/**13.70**	82.00/**6.70**
Top5 accuracy(%)	81.40/**4.00**	80.50/**4.60**	92.90/**18.90**	88.90/**34.40**	91.00/**9.60**	93.60/**30.40**	94.80/**76.9**
Logo 2							
Top1 accuracy(%)	60.20/**1.60**	58.00/**1.80**	74.30/**8.00**	69.70/**19.00**	72.90/**3.40**	77.20/**14.40**	82.00/**1.60**
Top5 accuracy(%)	81.40/**4.30**	80.50/**4.30**	92.90/**19.50**	88.90/**36.40**	91.00/**9.10**	93.60/**30.60**	94.80/**64.70**
Logo 3							
Top1 accuracy(%)	60.20/**1.40**	58.00/**1.90**	74.30/**7.70**	69.70/**18.60**	72.90/**3.30**	77.20/**14.60**	82.00/**1.10**
Top5 accuracy(%)	81.40/**3.80**	80.50/**4.70**	92.90/**18.50**	88.90/**35.30**	91.00/**9.20**	93.60/**31.20**	94.80/**64.10**

Table 2
Attack Performance in ILSVRC 2012 dataset and CIFAR-10 dataset with text hidden data, comparing the prediction accuracy of targeted model with original and adversarial images. For better comparison, the reduced classification accuracy caused by our adversarial images are noted in **bold**.

	ILSVRC 2012				CIFAR-10		
	ShuffleNetV2	SqueezeNet	ResNet50	InceptionV3	ShuffleNetV2	ResNet20	RepVGG-a1
Top1 accuracy(%)	60.20/**2.10**	58.00/**1.20**	74.30/**8.20**	69.70/**20.30**	90.10/**38.10**	92.60 /**18.20**	94.89/**46.40**
Top5 accuracy(%)	81.40/**6.10**	80.50/**7.20**	92.90/**19.90**	88.90/**33.90**	99.80/**97.30**	99.81 /**93.57**	99.83 / **94.81**

Table 3
Image quality of adversarial images with different form of embedded message.

	Average SSIM	Average PSNR
Logo 1	0.975	48.42
Logo 2	0.974	48.42
Logo 3	0.977	48.48
Text 1	0.979	48.51
Text 2(CIFAR-10)	0.996	48.88

Fig. 7. Visual quality comparison.

In addition to image logo, we also evaluate GMM with text data for generalized application. As shown in Table 2, we test GMM with text message as hidden data on two datasets: ILSVRC 2012 for various categories of image and CIFAR-10 for lower resolution images. The text messages for each dataset are different. The message we selected for ILSVRC 2012 is a short article with size of approximately 3000 bits, and for CIFAR-10, which contains images with lower capacity, we choose one short sentence with 64 bits of storage. In this set of experiment, the models we choose for ILSVRC 2012 are ShuffleNetV2, SqueezeNet, ResNet50 and InceptionV3. For CIFAR-10, we selected ShuffleNetV2, ResNet20 [37] and RepVGG-a1 [43]. The pretrained weights of these three models are provided by open-source reproducibility [44].

As presented on the table, comparing with results in Table 6, the top-1 and top-5 accuracies after attack are close to each other. It reveal that GMM is able to neglect the form of embedded data, and ensure a reliable performance. At the same time, the results on CIFAR-10 are slightly worse than those on ILSVRC 2012. It is worth to notice that the top-5 accuracy appears slightly affected in CIFAR-10 dataset, which could be due to the lower resolution and complexity of the images in CIFAR-10 compared to the higher resolution and more complex images in ILSVRC 2012. The model may be able to lean more general features of the image, this property could help the model to be more robust to adversarial attack.

5.2.2. *The visual quality of adversarial images*

By setting the epsilon value to 0.1, the degradation in image quality caused by our proposed method is limited, ensuring efficient attack performance. As illustrated in Table 3, the average SSIM value of the adversarial images with the three logos and their original counterparts is consistently above 0.98. Similarly, the average PSNR of these adversarial images remains above 49.82. In comparison, the method proposed by Liang et al.(BHI) employs invisible watermarking to embed watermarks onto images for generating adversarial examples, resulting in PSNR values ranging from 33.08 to 36.64. Although our method maintains comparable SSIM values to BHI, the PSNR values of adversarial examples generated by our adversarial watermarking method are significantly higher, exceeding those of BHI by over 10 dB. The outperformed visual quality of our adversarial images are presented in Fig. 7.

5.2.3. *The impact of the hidden messages*

In Fig. 8, we examine the correlation between the number of error correction symbols in Reed-Solomon (RS) codes with the performance of attacks and the visual quality. We chose ResNet50 and VGG19 as the models under attack and assessed the attack effectiveness as we varied the quantity of error correction symbols.

As shown in Fig. 8a and b, both top-1 and top-5 accuracies decline with an increasing number of error correction symbols. Notably, a higher number of error correction symbols correlates with better attack performance. This can be explained by the enhanced robustness of the information encoded through RS-coding with more error correction symbols. This indicates that our approach can achieve superior attack performance while maintaining the robustness and security of the embedded information. However, it is crucial to recognize that a larger number of error correction symbols requires a larger host image and allows for less embedded information, presenting a trade-off that must be considered in further applications.

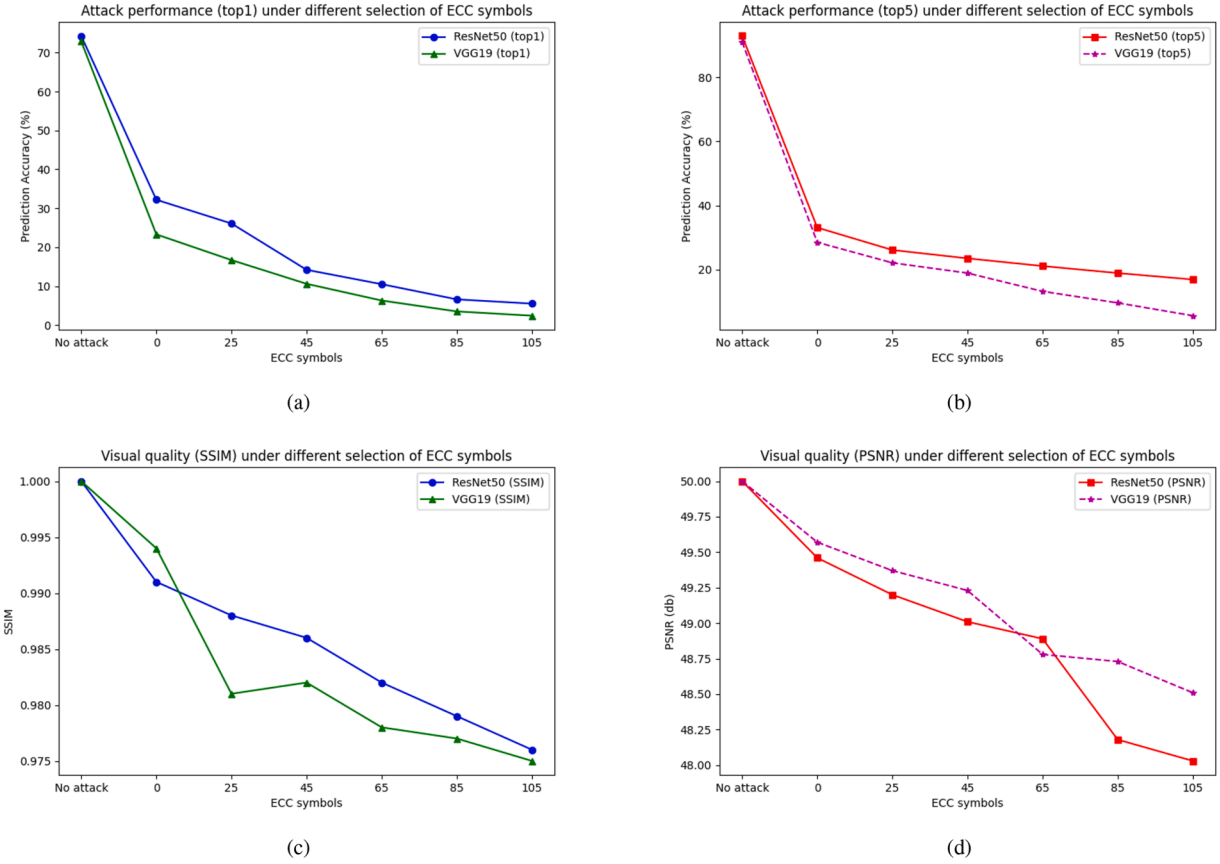

Fig. 8. Attack performance and image quality of adversarial images under different settings of ECC symbols.

Table 4
Time consumption compared to other methods with best performances highlighted with **bold**.

	SqueezeNet	ShuffleNetV2	InceptionV3	ResNet101	VGG19	Average
BHE [8]	10.768	4.069	34.434	39.263	39.916	25.690
FAWA [9]	7.819	6.971	30.523	40.437	22.766	14.682
BHI [10]	15.898	7.432	41.014	43.793	44.012	30.429
ISWP [11]	17.856	8.951	43.463	45.273	45.735	32.256
GMM	**0.254**	**0.513**	**0.956**	**1.079**	**1.381**	**0.836**

Table 5
This table compares the attack success rate of the proposed method to other black-box methods, with best performances highlighted with **bold**.

	InceptionV3(%)	ResNet101(%)	Average(%)
Spatial Attack [45]	58	52	55
Boundary Attack [46]	48	33	42
BHE [8]	73	78	75
FAWA [9]	50	48	59
BHI [10]	**81**	78	80
ISWP [11]	**81**	75	78
GMM	**81**	**85**	**83**

Table 6
Comparison of robust performance under different defensive methods with best performances highlighted with **bold**.

InceptionV3					
	BHE [8]	FAWA [9]	BHI [10]	ISWP [11]	GMM
Jpeg defence	94	96	**99**	98	94
Cropping	88	90	91	**95**	**95**
Partial-mosaic	72	78	80	99	**100**
HGD	95	96	90	**97**	83.3
ResNet101					
	BHE [8]	**FAWA** [9]	**BHI** [10]	**ISWP** [11]	**GMM**
Jpeg defence	93	96	**99**	97	98
Cropping	70	83	96	**99**	**99**
Partial-mosaic	70	77	73	99	**100**
HGD	89	92	89.3	**98**	88.2

In Fig. 8c and d, we analyze the effect of varying numbers of error correction symbols (ECC symbols) on visual quality. Again, we used ResNet50 and VGG19 as the target models and assessed visual quality while adjusting the number of error correction symbols. As Fig. 8c and d illustrate, visual quality, as measured by SSIM and PSNR, diminishes as the number of error correction symbols grows. This implies that while an increased number of error correction symbols bolsters the robustness of the encoded information, it also leads to more disturbances in the host image, thereby reducing visual quality.

In conclusion, Fig. 8c and d highlight the necessity to balance the count of error correction symbols, the size of the host image, and the volume of embedded information while ensuring the robustness and security of the information. In practical applications, it is essential to balance

these factors according to specific needs and conditions to achieve the best possible attack performance and visual quality.

5.3. Performance comparison with other methods

In order to verify the attack performance of our proposed method, we chose two black-box attacks, including Spatial Attack [45], Boundary Attack [46]. For these attack methods, we adopted their benchmark approaches and default parameters as recommended in Foolbox [47] to compare with our method. Additionally, we compared our method to BHE [8] and the improved method from BHI [10] as well as FAWA [9] and ISWP [11] (the best results are shown in **bold**). All compared methods are set by their best parameter settings and tested on the ILSVRC 2012 dataset for the results. The proposed method is implemented with logo 1 as the watermark image and parameter settings as former experiments.

5.3.1. Computation efficiency

Table 4 presents a comparison of the time consumption required to process one image across five models between five meaningful perturbation generation algorithms. To ensure the precision of the experiment data, all experiments were conducted on the same device (NVIDIA RTX 3060) and environment. Each method was tested 10 epochs on the ImageNet test set and take the average data as the final results. The results in Table 5 demonstrate the efficiency of the proposed GMM method compared to existing techniques (BHE, FAWA, BHH, and ISWP) in terms of time consumption per image across five models. Notably, GMM achieves significantly lower processing times, with values as low as 0.254 ms for SqueezeNet and 1.381 ms for VGG19, outperforming all other methods by orders of magnitude. The average time consumption of GMM (0.836 ms) further highlights its computational superiority, making it a highly efficient solution for real-world applications.

It is worth noting that our method exhibits slightly longer processing times on the VGG19 model compared to other models, with an increase of over 0.5 s in average time consumption. This phenomenon is not unique to our method; other attack methods also demonstrate this trend. Particularly, the Single-pixel attack takes over 100 s on average to attack VGG19, approximately twice the average processing time. Investigation into the original papers of the VGG networks reveals that this discrepancy is primarily attributed to two factors. Firstly, the VGG19 model contains a significant number of parameters, particularly concentrated in its three fully connected layers. Secondly, the extensive utilization of 3×3 convolutional layers increases the depth of the overall network. These factors collectively demand higher computational resources, resulting in increased processing times for adversarial attack methods applied to the VGG19 model.

5.3.2. Attack performance

To effectively compare the performance, we selected InceptionV3 and ResNet101 which presented greater robustness on former experiment as the target models, and computed the average attack success rate of each method, as shown in Table 5. The three attack methods with watermark perturbations exhibited excellent performance. Compared to BHE and BHI, our proposed method achieved superior results, with a 9 % higher attack success rate than BHE. and 3 % higher than BHI on average. It is not a particularly big improvement, but considering the much less computational cost of our method, our method can be said to provide a huge boost in attack efficiency. Likewise, the attack success rates of our method consistently exceeded 80 % on both models, compare to other benchmark black-box attacks, the proposed method has shown significantly effectiveness, demonstrating its effectiveness for typical attack tasks.

While GMM shows only modest attack success rate (ASR) improvements over BHI (3 % higher average) and ISWP (5 % higher), Table 4 demonstrates its computational efficiency surpasses these methods by 30–40 ×. This paradox stems from the Gradient Masking Mechanics: The message mask selectively blocks gradient updates, reducing search space while maintaining adversarial continuity. This preserves effectiveness($ASR > 80\,\%$) while eliminating evolutionary algorithms' population-based overhead.

5.3.3. Attack robustness

Meanwhile, we also compared the attack robustness under different image-transformation defenses, and the results are presented in Table 6. We selected four commonly used image-transformation defenses: JPEG defense, cropping operation, partial mosaic, and an adversarial defense method: high-level representation guided denoiser (HGD). Two groups of experiments were conducted using InceptionV3 and ResNet101 as the attacked models. In each group, we compared the five meaningful adversarial perturbation generation methods. In general, all these attack methods exhibited varying degrees of robustness, with attacks incorporating meaningful perturbations showing greater resilience than the other two black-box methods. Our method demonstrated superior robustness compared to the other four methods, with the average attack success rate after defense exceeding 90 %. This is over 5 % higher than BHE and approximately 3 % higher than the method of BHI on InceptionV3, and over 6 % higher on ResNet101.

Our method exhibits a consistently lower attack success rate against HGD, averaging over 8 % lower compared to other defense methods. The efficacy of HGD lies in its ability to process adversarial examples while minimizing the error amplification typically observed in common denoising methods. Different from other two adversarial watermarking method, our method rely on add random noise on the image, which made our scheme weak in dealing with HGD. But due to the effectiveness of our method, the attack success rate after HGD defence could still exceed 80 %.

In summary, our proposed GMM outperforms existing approaches in attack performance, operating 30 times faster than BHE and nearly 40 times faster than BHI. It achieves consistently high attack success rates, exceeding 80 %, and demonstrates superior robustness, with an average success rate surpassing 90 % under various image-transformation defenses. This combination of superior performance, efficiency, and robustness makes our method a promising solution for copyright protection against malicious DNN models.

6. Discussion

The capacity of an embedded watermark or hidden message is inherently constrained by the size of the host image and the imposed binary mask, which restricts the perturbation coordinates. For example, the ImageNet dataset using images resized to 224×224, the theoretical upper bound of the embedding capacity is $\frac{224\times224}{8}$ bits. This constraint necessitates balancing message length and error resilience, especially when using Reed-Solomon (RS) coding for error correction.

RS-coding introduces redundancy to the encoded message for error correction. The total embedded message consists of the original message (A) and redundant information (B). Greater redundancy enhances robustness but must not exceed the original message length ($B \leq A$). The combined length must satisfy $A + B \leq \frac{224\times224}{8}$. Increasing B improves error resilience but degrades image quality, while longer A reduces redundancy and error correction capability. When A exceeds half of the maximum embedding capacity, RS-coding's effectiveness diminishes due to insufficient redundant symbols for error correction.

Optimally balancing A and B is crucial for applications prioritizing imperceptibility and robustness. Future work should explore adaptive strategies to dynamically allocate redundancy based on host image characteristics and expected perturbations, ensuring an optimal trade-off between capacity, robustness, and visual quality.

In addition, even though the robustness performance of GMM in Table 6 appears reliable, the performance in front of HGD is not as promising as the others. And this degradation also appears in the domain-shift setting while testing with CIFAR-10. These results indicate

the potential weakness of GMM under similar circumstance. Normal image processing like lightning transformation or heavy distortions beyond Gaussian noise may not only influence the extraction of the hidden data, but also weaken the attack performance.

Addressing these shortcomings is essential for developing a better robustness and versatile GMM suitable for complex application scenarios. Future research should focus on enhancing GMM's resilience against such distortions and improving its adaptability to diverse image conditions.

7. Conclusion

In conclusion, this paper presents the Gradient Mask Method (GMM), a novel and efficient method for generating meaningful adversarial perturbations. Unlike traditional methodologies, our approach uses a message mask to embed information into perturbations created through a gradient-based attack algorithm, offering a computationally efficient and effective solution. To the best of our knowledge, we are the first to address the computational costs involved in crafting meaningful adversarial perturbations, demonstrating that GMM outperforms baseline methods in fooling DNN models in experimental scenarios.

Additionally, the adversarial examples produced by GMM retain their adversarial characteristics while preserving the host image's information. Moreover, we theoretically examine the side effects of the masking operation on the gradient, showing that it does not significantly affect the effectiveness of the adversarial attack.

CRediT authorship contribution statement

Hanbin Lin: Writing – original draft, Software, Methodology, Data curation, Conceptualization; **Wenxing Liao:** Validation, Software, Methodology, Data curation; **Zhaogang Shu:** Writing – review & editing, Validation, Formal analysis; **Xiaolong Liu:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, (2013). arXiv preprint arXiv:1312.6199

[2] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, (2014). arXiv preprint arXiv:1412.6572

[3] J. Su, D.V. Vargas, K. Sakurai, One pixel attack for fooling deep neural networks, IEEE Trans. Evol. Comput. 23 (5) (2019) 828–841.

[4] L. Schott, J. Rauber, M. Bethge, W. Brendel, Towards the first adversarially robust neural network model on MNIST, (2018). arXiv preprint arXiv:1805.09190

[5] A. Kurakin, I.J. Goodfellow, S. Bengio, Adversarial examples in the physical world, in: Artificial Intelligence Safety and Security, Chapman and Hall/CRC, 2018, pp. 99–112.

[6] G. Rossolini, F. Nesti, G. D'Amico, S. Nair, A. Biondi, G. Buttazzo, et al., On the real-world adversarial robustness of real-time semantic segmentation models for autonomous driving, IEEE Trans. Neural Netw. Learn. Syst. 35 (12) (2023) 18328–18342.

[7] J. Kotak, Y. Elovici, Adversarial attacks against IoT identification systems, IEEE Internet Things J. 10 (9) (2022) 7868–7883.

[8] X. Jia, X. Wei, X. Cao, X. Han, Adv-watermark: a novel watermark perturbation for adversarial examples, in: Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 1579–1587.

[9] H. Jiang, J. Yang, G. Hua, L. Li, Y. Wang, S. Tu, S. Xia, FAWA: fast adversarial watermark attack, IEEE Trans. Comput. 73 (2) (2021) 301–313.

[10] J. Liang, Z. Feng, R. Chen, X. Liu, BHI: Embedded invisible watermark as adversarial example based on Basin-Hopping improvement, Inf. Sci. 640 (2023) 119037.

[11] J. Liang, Y. Liu, L. Gao, Z. Zhang, X. Liu, ISWP: novel high-fidelity adversarial examples generated by incorporating invisible and secure watermark perturbations, Appl. Intell. 55 (40) (2025).

[12] I.S. Reed, G. Solomon, Polynomial codes over certain finite fields, J. Soc. Ind. Appl. Math. 8 (2) (1960) 300–304.

[13] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, (2017). arXiv preprint arXiv:1706.06083

[14] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z.B. Celik, A. Swami, The limitations of deep learning in adversarial settings, in: 2016 IEEE European Symposium on Security and Privacy (EuroS&P), IEEE, 2016, pp. 372–387.

[15] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: 2017 IEEE Symposium on Security and Privacy (Sp), IEEE, 2017, pp. 39–57.

[16] S.-M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: a simple and accurate method to fool deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2574–2582.

[17] Y. Qian, S. He, C. Zhao, J. Sha, W. Wang, B. Wang, Lea2: a lightweight ensemble adversarial attack via non-overlapping vulnerable frequency regions, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 4510–4521.

[18] Y. Qian, K. Chen, B. Wang, Z. Gu, S. Ji, W. Wang, Y. Zhang, Enhancing transferability of adversarial examples through mixed-frequency inputs, IEEE Trans. Inf. Forensics Secur. 19 (2024) 7633–7645

[19] S. Li, S. Zhang, G. Chen, D. Wang, P. Feng, J. Wang, A. Liu, X. Yi, X. Liu, Towards benchmarking and assessing visual naturalness of physical world adversarial attacks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 12324–12333.

[20] X. Zheng, Y. Fan, B. Wu, Y. Zhang, J. Wang, S. Pan, Robust physical-world attacks on face recognition, Pattern Recognit. 133 (2023) 109009.

[21] Z. Cheng, J. Liang, G. Tao, D. Liu, X. Zhang, Adversarial training of self-supervised monocular depth estimation against physical-world attacks, (2023). arXiv preprint arXiv:2301.13487

[22] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, S. Li, L. Chen, M.E. Kounavis, D.H. Chau, Keeping the bad guys out: protecting and vaccinating deep learning with jpeg compression, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 1365–1378.

[23] W. Xu, D. Evans, Y. Qi, Feature squeezing: detecting adversarial examples in deep neural networks, in: Proceedings of the 2018 Network and Distributed System Security Symposium (NDSS), 2017.

[24] F. Li, X. Du, L. Zhang, Adversarial attacks defense method based on multiple filtering and image rotation, (2024). arXiv preprint arXiv:2401.01234

[25] Z. Wang, X. Li, H. Zhu, C. Xie, Revisiting adversarial training at scale, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 24675–24685.

[26] H. Bostani, J. Cortellazzi, D. Arp, F. Pierazzi, V. Moonsamy, L. Cavallaro, On the effectiveness of adversarial training on malware classifiers, (2024), arXiv:2412.18218

[27] H. Rezaei, R. Taheri, M. Shojafar, FedLLMGuard: a federated large language model for anomaly detection in 5G networks, Comput. Netw. 269 (2025) 111473.

[28] M. Sabuhi, M. Zhou, C.-P. Bezemer, P. Musilek, Applications of generative adversarial networks in anomaly detection: a systematic literature review, IEEE Access 9 (2021) 161003–161029.

[29] J. Yu, H. Oh, Y. Lee, J. Yang, Denoising diffusion model with adversarial learning for unsupervised anomaly detection on brain MRI images, Pattern Recognit. Lett. 186 (2024) 229–235.

[30] T. Schlegl, P. Seeböck, S.M. Waldstein, G. Langs, U. Schmidt-Erfurth, f-AnoGAN: fast unsupervised anomaly detection with generative adversarial networks [J], Med. Image Anal. 54 (2019) 30–44.

[31] P. Zhu, T. Takahashi, H. Kataoka, Watermark-Embedded Adversarial Examples for Copyright Protection against Diffusion Models, 2024, 2404.09401

[32] Y. Ding, M. Shao, J. Wang, Q. Wan, AdMarks: image steganography based on adversarial perturbation, in: International Conference on Wireless Artificial Intelligent Computing Systems and Applications, Springer, 2024, pp. 186–198.

[33] Q. Li, Y. Hu, Y. Liu, D. Zhang, X. Jin, Y. Chen, Discrete point-wise attack is not enough: generalized manifold adversarial attack for face recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 20575–20584.

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (2015) 211–252.

[35] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, Technical Report (2009).

[36] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., PyTrch: an imperative style, high-Performance deep learning library, in: Advances in Neural Information Processing Systems (NeurIPS), 32, 2019.

[37] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[38] X. Zhang, X. Zhou, M. Lin, J. Sun, ShuffleNet: an extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6848–6856.

[39] F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, K. Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size, (2016). arXiv preprint arXiv:1602.07360

[40] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826.

[41] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, (2014). arXiv preprint arXiv:1409.1556
[42] D. Alexey, An image is worth 16x16 words: Transformers for image recognition at scale, (2020) arXiv preprint arXiv:2010.11929
[43] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, J. Sun, RepVGG: making VGG-style convnets great again, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13733–13742.
[44] C. Yao, PyTorch CIFAR Models: Pretrained models on CIFAR10/100 in PyTorch, 2023. https://github.com/chenyaofo/pytorch-cifar-models.
[45] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, A. Madry, Exploring the landscape of spatial robustness, in: International Conference on Machine Learning, PMLR, 2019, pp. 1802–1811.
[46] W. Brendel, J. Rauber, M. Bethge, Decision-based adversarial attacks: Reliable attacks against black-box machine learning models, (2017). arXiv preprint arXiv:1712.04248
[47] J. Rauber, R. Zimmermann, M. Bethge, W. Brendel, Foolbox native: fast adversarial attacks to benchmark the robustness of machine learning models in PyTorch, tensorflow, and JAX, J. Open Source Software 5 (53) (2020) 2607. https://doi.org/10.21105/joss.02607