Guided Erasable Adversarial Attack (GEAA) Toward Shared Data Protection

Mengnan Zhao, Bo Wang, *Member, IEEE*, Wei Wang, *Member, IEEE*, Yuqiu Kong, Tianhang Zheng, and Kui Ren, *Fellow, IEEE*

***Abstract*—In recent years, there has been increasing interest in studying the adversarial attack, which poses potential risks to deep learning applications and has stimulated numerous researches, e.g. improving the robustness of deep neural networks. In this work, we propose a novel double-stream architecture – Guided Erasable Adversarial Attack (GEAA) – for protecting high-quality labeled data with high commercial values under data-sharing scenarios. GEAA contains three phases, the double-stream adversarial attack, denoising reconstruction, and watermark extraction. Specifically, the double-stream adversarial attack injects erasable perturbations into the training data to avoid database abuse. The denoising reconstruction rebuilds the traceable denoising data from adversarial examples. The watermark extraction recovers identity information from the denoised data for copyright protection. Additionally, we introduce the annealing optimization strategy to balance these phases and a boundary constraint to degrade the availability of adversarial examples. Through extensive experiments, we demonstrate the effectiveness of the proposed framework in data protection. The Pytorch® implementations of GEAA can be downloaded from an open-source Github project https://github.com/Dlut-lab-zmn/GEAA-for-data-protection.**

***Index Terms*—Data protection, guided erasable adversarial attack, double-stream adversarial attack, denoising reconstruction, watermark extraction.**

I. Introduction

As a wide range of artificial intelligence fields such as image recognition [1], [2], semantic segmentation [3], and natural language processing [4] deploy machine learning to automatically make decisions, researchers have repeatedly highlighted the significance of understanding potential vulnerabilities in machine learning, including model vulnerabilities, data credibilities, and data missing [5], [6]. Additionally, the powerful autonomous learning ability of Deep Neural Networks (DNNs) induces data dependence of DNNs. Therefore, researchers take amounts of time to construct high-quality labeled data. However, not all datasets are freely available to all researchers like ImageNet [7]. Databases that are highly beneficial for business applications are facing data protection issues.

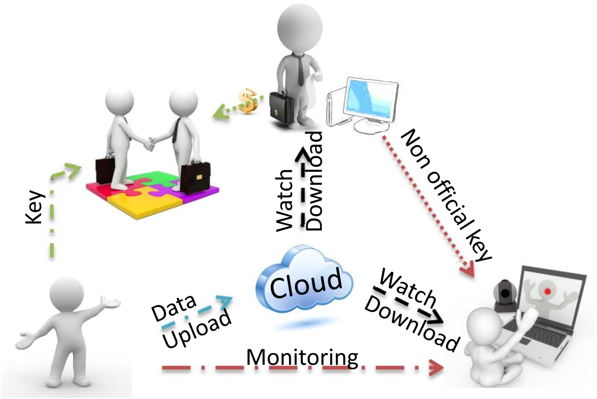

Fig. 1. Data sharing mode: Data owners upload data to an untrustworthy third party, and then, downloaders purchase the key to reconstruct data. The unauthorized data sharing should be under the data owners' monitoring.

This work focuses on protecting data in the practical data-sharing scene. We denote this scene as a Protect-Reconstruct-Identify (PRI) process. **P**: Data owners upload data to the internet, but meanwhile they want to protect data from unauthorized use. **R**: The data owners transmit the key to downloaders when two parties reach an agreement to reconstruct the data distribution. During this process, the identification information is embedded in the reconstruction data to prevent unauthorized sharing. Noteworthily, downloaders will not actively embed identification information to reconstructed data. **I**: The extracted identification information is used to verify leakers when illegal data sharing occurs. Related figure descriptions are represented in Fig. 1. Such a process faces two crucial security issues, (1) In the view of data owners, they upload data to third parties for the convenience of transmissions, such as google drive or Kaggle. However, unknown third parties may induce data leakage issues, data tamper-

Manuscript received 29 November 2021; revised 25 March 2022 and 11 June 2022; accepted 16 June 2022. Date of publication 27 June 2022; date of current version 11 July 2022. This work was supported in part by the National Natural Science Foundation of China under Grant U1936117, Grant 62106037, and Grant 62076052; in part by the Science and Technology Innovation Foundation of Dalian under Grant 2021JJ12GX018; in part by the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) under Grant 202100032; and in part by the Fundamental Research Funds for the Central Universities under Grant DUT21GF303, Grant DUT20TD110, and Grant DUT20RC(3)088. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Pedro Comesana. *(Corresponding author: Bo Wang.)*

Mengnan Zhao and Bo Wang are with the School of Information and Communication Engineering, Dalian University of Technology, Dalian 116000, China (e-mail: bowang@dlut.edu.cn).

Wei Wang is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China.

Yuqiu Kong is with the School of Innovation and Entrepreneurship, Dalian University of Technology, Dalian 116000, China.

Tianhang Zheng is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 2E4, Canada.

Kui Ren is with the Institute of Cyberspace Research, Zhejiang University, Hangzhou 310027, China.

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TIFS.2022.3186791.

Digital Object Identifier 10.1109/TIFS.2022.3186791

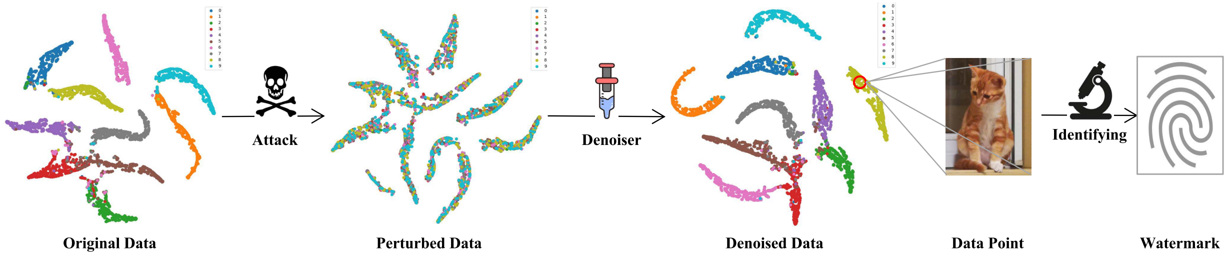

Fig. 2. Proposed GEAA framework, including the double-stream adversarial attack, denoising reconstruction, and watermark extraction. Each sub-task solves one problem mentioned in Fig. 1, namely, the threat of untrustworthy third parties, the network underfitting when training network on the denoised data, and the malicious data sharing.

ing issues, and unsanctioned data misuse issues. (2) Also, a potential issue – non-authoritative key sharing by authorized downloaders, should be concerned, including the case of leaking parts of reconstructed data. Induced by these issues, the core competitiveness of data protection under the data-sharing scene includes, (1) Uploaded data (after processing) should be well protected, observable but not available for training. (2) The original data could be reconstructed from uploaded data for normal use with authorization. (3) The reconstruction process could embed identification information for tracing the source of the leaked data. In this way, the data owners and authorized downloaders only need to share the reconstruction module (unique for each authorized downloader) as a key in advance.

Since there are no fully efficient solutions for mentioned problems, only previous inductive works are introduced. For instance, recent efforts consider injecting perturbations into the training dataset [8], in which trained networks predict bias for inputs [9]. Such technologies, strictly speaking, are harmful to users but are beneficial tools for data owners to protect their core competitiveness. The prerequisite is that the problems of data obfuscation (see details in Fig. 2) and model under-fitting can be processed without access to the original data distribution. In addition, Jia et.al [10] attack DNNs by adding watermarks into clean images. Zhang et.al [11] take advantage of reversible data hiding to construct reversible adversarial examples. Meanwhile, they propose the reversible adversarial attack based on reversible image transformation [12]. These methods are based on existing adversarial attacks and reversible data hiding technologies. However, we have verified by experiments that networks trained with adversarial examples (generated by existing attacks, e.g. PGD [13], BIM [14]) show a high generalization performance, which indicates a poor data protection performance.

To tackle these problems, we propose a newly designed double-stream architecture – Guided Erasable Adversarial Attack (GEAA) – for protecting data [15] under data-sharing scenarios. The detailed procedures of GEAA are illustrated in Fig. 2. GEAA contains several core competitive elements. The double-stream adversarial attacks take into account unknown threats from untrusted third parties. Denoising reconstruction avoids network underfitting when reconstructing the data distribution. Watermark extraction solves the problem of malicious data sharing by users who have purchased the data.

Specifically, the double-stream adversarial attack module aims to degrade the availability of databases by disrupting the initial training data distribution. Data owners inject ‘non-extractable diverse perturbations’ into all samples in the training set and upload these adversarial examples to third parties. In this way, downloaders cannot deduce injected perturbations from the perturbed data and manually eliminate them. Uploading perturbed data is for viewing only, addressing issues that third parties cannot trust, namely, data leakage and unsanctioned data misuse. Visualizing data helps users understand the specific information of the dataset. The data tampering issue could be detected by comparing the local data with downloaded data.

The denoising reconstruction module consists of two actions, data distribution reconstruction, and watermark embedding. The annealing optimization strategy is introduced for reconstructing the training data distribution while maintaining the mismatching ratio of the training-testing dataset distribution. Besides, a series of distinctive watermark features are embedded into the extracted transfer features as identifying information. After the training process, data owners transmit the denoising reconstruction modules (unique for each user) to authorized data users. The watermark features embedded in each denoising reconstruction module are unique. Noteworthily, since the denoising reconstruction module is adopted as the key, only perturbed data is available in constructing the original data distribution.

The watermark extraction module deduces watermark features from denoised data by a unified network, namely, multiple embedding-single extraction. In this way, data owners could trace users who maliciously spread data according to leaked data (or download links) on the website.

In summary, the contributions of this paper are as follows: (1) To the best of our knowledge, we are the first to consider shared data protection scenarios under the adversarial attack and the multiple embedding-single extraction algorithm. (2) An elaborate double-stream architecture – the guided erasable adversarial attack, is designed for shared data protection, including double-streamed adversarial attack, denoising reconstruction, and watermark extraction. (3) A boundary constraint strategy and an annealing optimization strategy are proposed for training the GEAA. (4) Extensive experiments on several benchmark datasets and excellent classification networks demonstrate the effectiveness of GEAA in data protection.

II. Related Work

In this section, we will first introduce security related technologies and then summarize existing adversarial attacks.

Security issues are always related to cyberspace, and media copyright. Media copyright protection such as watermark embedding technology has been highlighted [16], [17] for its importance and has attracted the attention of researchers for many years. Watermark-based media copyright protection methods can be split into two diverse classes according to the watermark property: visible watermark and invisible watermark [18]. The visible watermarks cause data availability to decline since they impact the valid message. In contrast, invisible watermark embedding technologies only modify a small amount of data information and belong to the passive forensic, that is, to guard the copyright by leaked data. For cyberspace security, Addesso et.al [19] designed the ADVoIP to detect the manipulated Voice-over-IP (VoIP) traffic streams from an adversarial perspective. Then, they [20] also regarded propagation of cyber-threats over networks under an adversarial formulation as zero-sum games involving two adversaries.

Adversarial attacks [5], [21] aimed at deceiving well-trained DNNs, both specified models in the white-box attack and unknown models in the black-box attack [22]. Researchers crafted adversarial examples by adding tiny adversarial perturbations. Initial adversarial attacks were based on the gradient-backward. For instance, the Fast Gradient Sign Method (FGSM) [23] generated adversarial examples by a single-step gradient update. The Basic Iteration Method (BIM) [14] reduced the single-step perturbation and generated adversarial examples by multiple iterations. Additionally, researchers introduced several special attack methods. The universal attack [24] intentionally designed a universal perturbation that was available for all inputs, which strongly saved the time consumption in generating adversarial examples. One Pixel Attack [25] can be seen as an extreme attack situation, in which only one pixel in each sample was modified to deceive the target model. Specifically, researchers adopted differential evolution to find optimal pixel locations and values. [26] proposed to add bigger perturbations to realize the adversarial attack. Researchers introduced the texture transfer model and the colorization model. It is novel that changing the color of clean samples leads to misclassification. To improve the generalization performance of the attack, Sarkar *et al.* [27] designed the *UPSET* and *ANGRI* networks to generate adversarial examples by attacking multiple classifiers. Subsequently, the Carlini and Wagner (C&W) attack [28] solved the problem of obfuscated gradients by optimization strategies. Yin et.al [12] proposed the reversible adversarial example by embedding reversible watermarks into the benign sample. Experiments illustrated that reversible watermark attacks were comparable with common adversarial attacks.

Additionally, adversarial attacks were introduced to various fields such as the attack for semantic segmentation and object detection [29], the black-box attack for audio systems [30] and the attack for image-captioning models [31], [32]. For audio attack, [33] generated selective audio adversarial examples that will be misclassified as the target phrase by the victim classifier but correctly classified as the original phrase by the protected classifier. Sonal et.al [34] introduced pre-processing defenses against adversarial attacks on speaker recognition systems. [35], [36] enhanced image steganography based on the adversarial embedding.

TABLE I

Definitions of Several Symbols That Are Used to Describe GEAA

Notation	Descriptions
$\mathcal{I}$; $\mathcal{I}^+$; $\mathcal{I}^*$	Benign samples; Perturbed samples; Denoised samples.
$\mathcal{D}$; $\mathcal{D}^+$; $\mathcal{D}^*$	Clean database; Perturbed database; Denoised database.
$\mathcal{L}$; $\mathcal{O}$	The ground truth label; The logical probability.
$\mathbf{M}_{net}$	The data distribution understanding model.
$\mathbf{F}_{net}$; $\mathbf{J}_{net}$	The feature extraction network; The feature aggregation network.
$\theta_{\mathbf{M}_{net}}$; $\theta_{\mathbf{F}}$; $\theta_{\mathbf{J}}$	Network weights for $\mathbf{M}_{net}$; $\mathbf{F}_{net}$; $\mathbf{J}_{net}$.
$\mathcal{T}_{\mathcal{I}}$; $\mathcal{T}_{\mathcal{I}}^*$	Embedded perturbations; Reconstructed perturbations.
w_1; w_2	Watermarks for denoised data; Watermarks for clean data.

III. Proposed Approach

In this section, we first define the algorithm context and the target problem formally in Section III-A, and then provide the detailed theoretical analysis, with descriptions shown in Fig. 3, of how the data distribution is disrupted in Section III-B and how the denoising module reconstructs the original data distribution in Section III-C. Additionally, Section III-D includes the research for extracting invisible inserted watermarks through the watermark extraction module. Finally, we show the joint optimization constraint and how to construct an annealing optimization strategy to facilitate the module training in Section III-E. Definitions of several symbols are provided in Table I.

A. Problem Definition

We have illustrated core security concerns of the work in Section I, including data leakage issues, data tampering issues, and unsanctioned data misuse issues. Towards these issues, we propose a Guided Erasable Adversarial Attack (GEAA) under data sharing scenarios.

Given a newly collected database $\mathcal{D} = \{(\mathcal{I}_i, \mathcal{L}_i) \mid i \in [1, n]\}$ to be protected, n is the the number of samples in $\mathcal{D}$, GEAA is split into following substeps. (1) Obtain perturbed dataset $\mathcal{D}^+ = \{(\mathcal{I}_i^+, \mathcal{L}_i) \mid i \in [1, n]\}$ by injecting mixed perturbations into training data based on the double-stream adversarial attack. (2) Train various denoisers for denoising reconstruction. The denoised database is named as $\mathcal{D}^* = \{(\mathcal{I}_i^*, \mathcal{L}_i) \mid i \in [1, n]\}$ (3) Recover particular watermark information from the leaked data (denoised data) for data protection using a universal reconstruction module. In particular, perturbed dataset $\mathcal{D}^+$ is uploaded into third parties for the convenience of transmissions. The trained denoisers are transmitted to downloaders who have purchased the key (specific to each user).

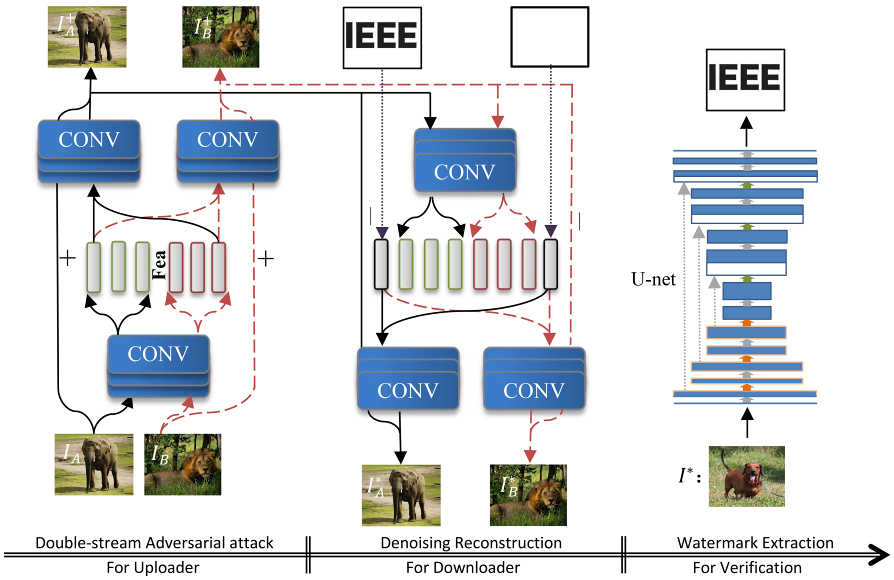
$\mathcal{I}_{\mathcal{A}}^{+}$
$\mathcal{I}_{\mathcal{B}}^{+}$
IEEE
CONV
CONV
+
Fea
+
CONV
$\mathcal{I}_{\mathcal{A}}$
$\mathcal{I}_{\mathcal{B}}$
CONV
CONV
CONV
$\mathcal{I}_{\mathcal{A}}^{*}$
$\mathcal{I}_{\mathcal{B}}^{*}$
IEEE
U-net
I^*:
Double-stream Adversarial attack
For Uploader
Denoising Reconstruction
For Downloader
Watermark Extraction
For Verification

Fig. 3. Proposed double-stream architecture. GEAA includes the double-streamed adversarial attack module to generate perturbed data for the uploader to transmit, the denoising reconstruction module to build denoised data for the downloader to use, and the watermark extraction module to verify leaked data. $\mathcal{I}_{\mathcal{A}}$ and $\mathcal{I}_{\mathcal{B}}$ represent two types of benign samples from different categories. $\mathcal{I}_{\mathcal{A}}^{+}$ and $\mathcal{I}_{\mathcal{B}}^{+}$ denote the generated perturbed samples under the supervision of the target of $\mathcal{I}_{\mathcal{B}}$ or $\mathcal{I}_{\mathcal{A}}$. $\mathcal{I}_{\mathcal{A}}^{*}$ ($\mathcal{I}_{\mathcal{B}}^{*}$) means the denoised samples with embedded watermarks based on the perturbed data $\mathcal{I}_{\mathcal{A}}^{+}$ ($\mathcal{I}_{\mathcal{B}}^{+}$).

B. Double-Streamed Adversarial Attack

This subsection first introduces previous adversarial attacks and then describe the detail of the double-stream adversarial attack.

The typical adversarial attacks are closely correlated with the specific machine learning model. However, the limited knowledge derived from the dataset cannot cover all scenarios of the real world, inducing the forecast bias. Researchers take advantage of these ‘knowledge blind zones’ to mislead well-trained models by adding subtle invisible perturbations to benign samples under the guidance of the constraint gradient. Specifically, given a set of clean samples $\mathcal{I}$ and the ground truth labels $\mathcal{L}$, attackers feed $\mathcal{I}$ into the trained model to obtain corresponding logical probabilities $\mathcal{O}$. The specific formula for the adversarial attack is as follows,

$$\mathcal{I}_{i+1}{}^{+} = \mathcal{I}_{i}{}^{+} + \varepsilon \cdot sign(\nabla_{\mathcal{I}_i^{+}} \mathcal{L}oss(\mathcal{O}, \mathcal{L})) \tag{1}$$

where i means ith attack, $\mathcal{I}_i{}^{+}$ represents generated adversarial examples in ith attack, $\mathcal{L}oss$ denotes the cross entropy loss function, and $\nabla_{\mathcal{I}_i^{+}} \mathcal{L}oss(\mathcal{O}, \mathcal{L})$ calculates the gradient update direction.

We demonstrate in Fig. 2 that the attack in GEAA aims at interrupting the identical distribution between the training set and testing set. In this way, DNNs cannot distinguish various categories after training on perturbed training data. Therefore, similar to the common adversarial attacks, we first construct initial data distributions based on the proven efficient network structure $\mathbf{M}_{net}$ and clean samples $(\mathcal{I}_i, \mathcal{L}_i)$. Specifically, we optimize the network weights $\theta_{\mathbf{M}_{net}}$ based on the following formula.

$$\min_{\theta_{\mathbf{M}_{net}}} - \ln \vec{P}\left(\mathcal{L}_i \mid \mathbf{M}_{net}(\mathcal{I}_i, \theta_{\mathbf{M}_{net}})\right) \tag{2}$$

where $\mathbf{M}_{net}(\mathcal{I}_i, \theta_{\mathbf{M}_{net}})$ represents the predicted probability distribution of the $\mathbf{M}_{net}$ for the input $\mathcal{I}_i$. $\vec{P}(\mathcal{L}_i \mid \mathbf{M}_{net}(\mathcal{I}_i, \theta_{\mathbf{M}_{net}})$ stands for the probability that $\mathcal{I}_i$ is predicted as $\mathcal{L}_i$.

Then, we intentionally disrupt the training data distribution by transferring the sample into other data distribution domains. Towards this end, we separate the training set into two categories, $\mathcal{A}$ and $\mathcal{B}$, and present the double-stream adversarial attack network. Due to the paired inputs of the double-stream adversarial attack network, we expect an equal number of samples in $\mathcal{A}_i$ ($\mathcal{A}_i \in \mathcal{A}$) and $\mathcal{B}_i$ ($\mathcal{B}_i \in \mathcal{B}$). For unbalanced datasets, we use $min \sum_{i, i\in[1,N/2]} abs(len(\mathcal{A}_i) - len(\mathcal{B}_i))$ to select categories to $\mathcal{A}$ and $\mathcal{B}$. N and $len(\cdot)$ calculate the number of categories and samples for each category. For balanced datasets, we select top-$\frac{N}{2}$ categories as $\mathcal{A}$ and the rest as $\mathcal{B}$. We do not further study the division of $\mathcal{A}$ and $\mathcal{B}$, since the inter-embedding performance of various categories is different among different datasets. The challenge that allocates best-matching category pairs to the dataset should consider the semantic similarity, visual features similarity, effective visual pixel ratio, and so on.

Next, randomly selected samples $\mathcal{I}_{\mathcal{A}}$, $\mathcal{I}_{\mathcal{B}}$ from categories $\mathcal{A}$ and $\mathcal{B}$ are adopted as inputs of the feature extraction network $\mathbf{F}_{net}$. The feature aggregation network $\mathbf{J}_{net}$ extracts distribution transfer perturbations by combining features specific to samples $\mathcal{I}_{\mathcal{A}}$, $\mathcal{I}_{\mathcal{B}}$. Finally, $\mathcal{L}_{\mathcal{I}_{\mathcal{B}}}$ and $\mathcal{L}_{\mathcal{I}_{\mathcal{A}}}$ are used as the target mapping of $\mathcal{I}_{\mathcal{A}}$ and $\mathcal{I}_{\mathcal{B}}$. The formula description is as follows:

$$\begin{aligned} \mathcal{T}_{\mathcal{I}_{\mathcal{A}}} &= \mathbf{J}_{net-\mathcal{A}}(<\mathbf{F}_{net}(\mathcal{I}_{\mathcal{A}}, \theta_{\mathcal{F}}); \mathbf{F}_{net}(\mathcal{I}_{\mathcal{B}}, \theta_{\mathcal{F}})>, \theta_{J}) \\ \min_{\theta_{\mathcal{F}}, \theta_{\mathcal{J}}} \mathcal{L}oss_m &= \mathcal{L}oss\{\mathcal{L}_{\mathcal{I}_{\mathcal{A}}} \mid \mathbf{M}_{net}(\mathcal{I}_{\mathcal{B}} + \mathcal{T}_{\mathcal{I}_{\mathcal{A}}}), \theta_{\mathcal{F}}, \theta_{\mathcal{J}}, \theta_{\mathbf{M}_{net}}\} \end{aligned} \tag{3}$$

where $\theta_{\mathcal{F}}$, $\theta_{\mathcal{J}}$ are network weights of $\mathbf{F}_{net}$ and $\mathbf{J}_{net}$ in the double-stream adversarial attack module, $\mathcal{T}_{\mathcal{I}_{\mathcal{A}}}$ and $\mathcal{T}_{\mathcal{I}_{\mathcal{B}}}$ are the distribution transfer perturbations for $\mathcal{I}_{\mathcal{B}}$ and $\mathcal{I}_{\mathcal{A}}$, and

$<;>$ denotes the feature concatenation. $\theta_{\mathbf{M}_{net}}$ is fixed in the double-stream adversarial attack process. The perturbed samples deviate from the original samples as the model optimization, and as a result, the identical distribution of the training-testing dataset is broken.

To better disrupt the consistency of the training-testing distribution and maintain the inter-dataset distribution balance, we introduce the boundary constraint. This constraint helps the double-stream adversarial attacks generate more effective disturbances than conventional attacks.

$$\max_{\theta_{\mathcal{F}},\theta_{\mathcal{J}}} \mathcal{L}oss_b = \min\{\min\{\|\mathcal{T}_{\mathcal{I}_\mathcal{A}}\|_1, \|\mathcal{T}_{\mathcal{I}_\mathcal{B}}\|_1\}, \mathrm{MIN}\} \\ - \max\{abs(\|\mathcal{T}_{\mathcal{I}_\mathcal{A}}\|_1 - \|\mathcal{T}_{\mathcal{I}_\mathcal{B}}\|_1), \mathrm{MAX}\} \quad (4)$$

where MIN represents the lower boundary constraint for transfer perturbations, MAX limits the differences in the transfer perturbations between datasets.

C. Denoising Reconstruction

In contrast to the double-streamed adversarial attack, the denoising reconstruction task aims to remove perturbations from the perturbed data $\mathcal{I}^+$. The denoising reconstruction contains two actions, (1) Reconstruct the data distribution (denoised samples, named $\mathcal{I}^*$) from the perturbed dataset $\mathcal{I}^+$; (2) Embed the designated watermarks into denoised samples, where the designated watermarks are utilized to guard data copyright.

We also utilize the double-stream network to extract perturbations embedded in the perturbed data. The related figure descriptions are given in the second section of Fig. 3. The formula description is given as,

$$\mathcal{T}^*_{\mathcal{I}_\mathcal{A}/\mathcal{I}_\mathcal{B}} = \mathbf{J}^*_{net-\mathcal{A}/\mathcal{B}}(< \mathbf{F}^*_{net}(\mathcal{I}^+_\mathcal{A}, \theta^*_\mathcal{F}); \mathbf{F}^*_{net}(\mathcal{I}^+_\mathcal{B}, \theta^*_\mathcal{F}) >, \theta^*_J) \quad (5)$$

where $\mathcal{I}^+_\mathcal{A} = \mathcal{I}_\mathcal{A} + \mathcal{T}_{\mathcal{I}_\mathcal{B}}$ and $\mathcal{I}^+_\mathcal{B} = \mathcal{I}_\mathcal{B} + \mathcal{T}_{\mathcal{I}_\mathcal{A}}$, $\mathbf{F}^*_{net}$ and $\mathbf{J}^*_{net}$ denotes the feature extraction network and feature aggregation network in the denoising reconstruction module. Next, we select the constraint for updating the $\theta^*_\mathcal{F}$ and $\theta^*_\mathcal{J}$.

Different from conventional image reconstruction tasks [37], we find that the surrogate perceptual distance – learned perceptual image patch similarity (LPIPS) [38] – is not effective in the data protection mode even it correlates well with human perception. The denoised data generated by the LPIPS-supervised reconstruction module contains recognition information, causing the trained model to underfit. That is, the remaining information in the denoised samples leads to a fast training convergence, and as a result, the denoised data-trained network exhibits poor generalization performance. Therefore, we adopt the L2-norm constraint between the perturbations $< \mathcal{T}^*_{\mathcal{I}_\mathcal{A}}; \mathcal{T}^*_{\mathcal{I}_\mathcal{B}} >$ extracted by the denoising reconstruction module and the embedded perturbations $< \mathcal{T}_{\mathcal{I}_\mathcal{A}}; \mathcal{T}_{\mathcal{I}_\mathcal{B}} >$ as the reconstruction loss.

$$\min_{\theta^*_F,\theta^*_J} \mathcal{L}oss_r = \|\mathcal{T}_{\mathcal{I}_\mathcal{A}} - \mathcal{T}^*_{\mathcal{I}_\mathcal{A}}\|_2 + \|\mathcal{T}_{\mathcal{I}_\mathcal{B}} - \mathcal{T}^*_{\mathcal{I}_\mathcal{B}}\|_2 \quad (6)$$

Additionally, the designated watermark features are embedded in the denoised samples. Thus, Eq. (5) is modified as Eq. (7).

$$F = < \mathbf{F}^*_{net}(\mathcal{I}^+_\mathcal{A}, \theta^*_\mathcal{F}); \mathbf{F}^*_{net}(\mathcal{I}^+_\mathcal{B}, \theta^*_\mathcal{F}); F_{w1}; F_{w2} > \\ \mathcal{T}^*_{\mathcal{I}_\mathcal{A}/\mathcal{I}_\mathcal{B}} = \mathbf{J}^*_{net-\mathcal{A}/\mathcal{B}}(F, \theta^*_J) \quad (7)$$

where F_{w1} and F_{w2} denote the embedded watermark features extracted by a double convolution layer (without the batch normalization layer) for the denoised data and the original data, respectively.

However, the single embedding-extraction method cannot distinguish the concrete user identity, and thus, we introduce the multiple embedding-single extraction mode. For the multiple-embbeding, identity card F_{w1} is randomly selected from a set of recognizable watermarks ID in the module training process but is fixed in the validation process.

$$\min_{\theta^*_F,\theta^*_J} \mathcal{L}oss_r = \|\mathcal{T}_{\mathcal{I}_\mathcal{A}} - \mathcal{T}^*_{\mathcal{I}_\mathcal{A}}\|_2 + \|\mathcal{T}_{\mathcal{I}_\mathcal{B}} - \mathcal{T}^*_{\mathcal{I}_\mathcal{B}}\|_2, \quad F_{w1} \in \mathrm{ID} \quad (8)$$

In this way, the data owners could construct a series of denoising reconstruction modules but only transmit a specific denoising reconstruction module to a potential user. For the single-extraction, we extract all embedded identification information from the generic model and single input.

Here we explain why we insert watermarks in denoised data instead of perturbed data. In fact, GEAA contains four steps, including the double-stream adversarial attack, denoising reconstruction, watermark embedding, and watermark extraction. Considering that the watermark embedding affects the performance of the double-stream adversarial attack and denoising reconstruction, we insert watermarks into data after the denoising reconstruction module. After that, we incorporate the denoising reconstruction and the watermark embedding as a unified network. The integrated denoising reconstruction module more robustly embeds identifying information than separate denoising reconstruction and watermarking, e.g. manually removing the watermarking process.

D. Watermark Extraction

We experimentally discovered that the simple U-net [39] is efficient in accurately extracting embedded watermarks. All embedded identity information are extracted through a trained U-net, namely, single extraction. To obtain recovered watermarks, we feed outputs of the U-net into the sigmoid function $\sigma(\cdot)$. The binary outputs of the sigmoid function and the standard watermarks are adopted to measure the reconstruction difference.

$$\min_{\theta_U} \mathcal{L}oss_w = \|\sigma(\text{U-net}(\mathcal{S}, \theta_U)) - w\|_2 \quad (9)$$

where $\mathcal{S}$ includes both benign samples $\mathcal{I}$ and denoised samples $\mathcal{I}^*$, and w contains w_1 and w_2. w_1 and w_2 are the pre-defined watermarks for the denoised samples and benign samples. We only select a series of simple watermarks, as they are sufficient to identify the source of the leaked data. Fig. 6 illustrates several watermark examples. We select words, letters of an alphabet, digits, or signs as w_1. For w_2, we adopt the blank watermark as it.

E. Optimization Goal

In above sections, we have explained in detail the newly designed GEAA framework. Next, we will show how to combine the proposed submodules and jointly optimize them.

As shown in Fig. 3, GEAA contains three modules, and we feed the outputs of the current module as the inputs of the subsequent module. Following, we optimize all network parameters together. To balance the double-streamed adversarial attack and the denoising reconstruction, we introduce the annealing optimization technique. That is, we first set a high temperature **T** in the regression process so that perturbed data can effectively map into the randomly selected target data distribution region. Then, the annealing optimization gradually reduces the constraint degree on the double-stream adversarial attack and pays more attention to the denoising reconstruction. In this way, GEAA successfully achieves both high adversarial attack performance and denoising reconstruction performance.

$$\min_{\theta_F,\theta_J,\theta_F^*,\theta_J^*} \mathcal{L}oss_r + \min\{-\mathcal{L}oss_{m,s}, \mathbf{T}_s\}, \quad \mathbf{T}_s \in \mathbf{T} \tag{10}$$

T is the set of temperatures, and s is the optimization stride.

Moreover, considering the uncertainty of the multi-embedding, we dynamically separate the watermark extraction constraint $\mathcal{L}oss_w$ from the double-stream adversarial attack $\mathcal{L}oss_m$ and denoising reconstruction $\mathcal{L}oss_r$. Towards this end, we adopt the sigmoid output of the watermark extraction module and standard watermarks w to calculate the average value of the Watermark Extraction Accuracy (WEA).

$$\mathrm{WEA}(\mathcal{S}_i) = |\sum abs(\sigma(\text{U-net}(\mathcal{S}_i, \theta_U))) - \sum abs(w)| < 1$$

$$\overline{\mathrm{WEA}} = \forall_{\mathcal{S}_i \in \mathcal{S}} \frac{\sum \mathrm{WEA}(\mathcal{S}_i)}{len(\mathcal{S})} \tag{11}$$

where $\mathcal{S}$ includes both benign samples $\mathcal{I}$ and denoised samples $\mathcal{I}^*$, and w contains w_1 and w_2. On this basis, the watermark extraction difference is modified as

$$\min_{\theta_U} \mathcal{L}oss_{w,\overline{\mathrm{WEA}}} = (1 - \overline{\mathrm{WEA}}) \cdot \mathcal{L}oss_w \tag{12}$$

Namely, the watermark extraction constraint decreases with the increase of $\overline{\mathrm{WEA}}$.

Combining the boundary constraint $\mathcal{L}oss_b$ that disrupts the data distribution and the watermark extraction loss $\mathcal{L}oss_{w,\overline{\mathrm{WEA}}}$, the ultimate optimization loss is given by

$$\min_{\theta_F,\theta_J,\theta_F^*,\theta_J^*,\theta_U} \mathcal{L}oss_{Sum} = \mathcal{L}oss_r + \mathcal{L}oss_b + \mathcal{L}oss_{w,\overline{\mathrm{WEA}}} + \min\{-\mathcal{L}oss_{m,s}, \mathrm{T}_s\}, \mathrm{T}_s \in \mathbf{T} \tag{13}$$

IV. Experiments

To demonstrate the effectiveness of the proposed system, the results of both qualitative and quantitative experiments are given in this section. We first illustrate the implementation details and evaluation metrics of GEAA in Section IV-A. Next, we present the data protection performance of GEAA on several standard datasets quantitatively and qualitatively in Section IV-B. Section IV-C and Section IV-D display the influence of $\mathbf{M}_{net}$ and the distribution accuracy on GEAA. Similarly, we compare the variation in GEAA under different hyperparameter settings, including various lower boundary constraints and temperatures, in Section IV-E. Section IV-F gives the comparison of GEAA with existing attacks. Finally, ablation analyses for multi embedding - single extraction are provided in Section IV-G.

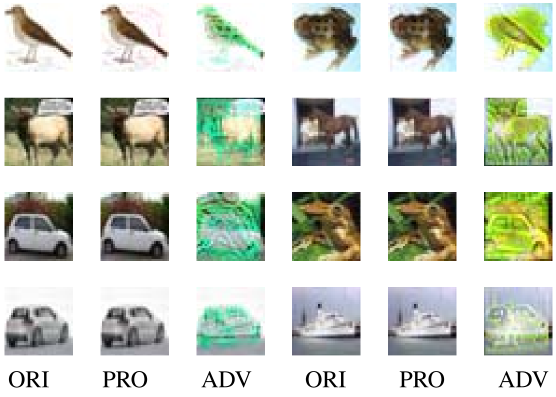

Fig. 4. Visual results of the generated perturbed data and denoised data (MIN = 30, temperature = 0.9). ORI, PRO and ADV denote the original image, the denoised sample and the perturbed sample. The left and right samples come from $\mathcal{A}$ and $\mathcal{B}$ respectively.

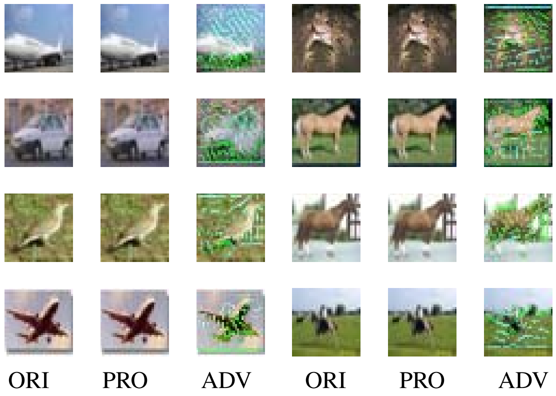

Fig. 5. Visual results of the generated perturbed data and denoised data (MIN = 10, temperature = 0.9). ORI, PRO and ADV denote the original image, the denoised sample and the perturbed sample. The left and right samples come from $\mathcal{A}$ and $\mathcal{B}$ respectively.

A. Experimental Setup

1) Models and Dataset Setup: In this section, we evaluate the data protection performance of the proposed GEAA on several popular classification models, including ResNet [40], VGG [41], MobileNet [42], and DenseNet [43]. Three benchmark datasets are applied in our experiments: (1). CIFAR10 [44]: The CIFAR10 dataset contains 60000 RGB images with a size of 32 × 32 and is classified into 10 categories. A total of 50000 images are selected as the training set, and the remaining 10000 images are used as the testing set. (2). Fashion MNIST [45]: The Fashion-MNIST dataset is an open set that was proposed to replace the MNIST handwritten digit set. This set covers 70000 different images from 10 categories, with a 60000/10000 training and test data division and 28 × 28 greyscale images. (3). SVHN [46]: The Street View House Numbers Dataset is a real-world image

TABLE II

Quantitative Results of the Proposed GEAA Algorithm (MIN = 10, **T** = 0.9). The First Line Denotes the Classification Accuracy of the Models Trained on the Normal Training Set. Perturbation Is Calculated by $Noise(\mathcal{I}^+, \mathcal{I})$, and Reconstruction Is Calculated by $Noise(\mathcal{I}^*, \mathcal{I})$. PSNR Denotes Peak Signal to Noise Ratio

Dataset		PSNR (Noise)		Acc {Perturbation↑/Reconstruction↓}				DPR ↑	$\overline{\text{WEA}}$↑
	Phase	Perturbation ↑	Reconstruction ↓	ResNet	VGG	MobileNet	DenseNet		
CIFAR10	-	-	-	88.89	86.78	86.34	92.12	-	-
	1	18.99(28.64)	25.59(13.40)	45.64/45.83	49.20/47.99	46.72/43.19	45.00/52.19	0.104	0.436
	5	20.82(23.19)	42.74(1.859)	44.83/70.50	41.50/73.06	42.13/70.90	42.62/80.36	0.320	98.73
	10	24.52(15.16)	49.99(0.807)	56.61/78.63	55.97/86.38	57.03/77.56	57.48/88.55	0.301	98.93
	15	25.43(13.64)	52.44(0.609)	61.35/88.53	54.10/86.41	60.93/86.34	58.48/91.29	**0.332**	**99.12**
FASHION	-	-	-	94.07	93.62	93.64	94.41	-	-
	1	11.22(70.10)	31.87(6.502)	47.59/91.50	52.59/91.46	61.27/91.99	60.21/91.58	**0.383**	0.892
	5	15.21(44.28)	43.45(1.715)	50.11/92.77	68.52/93.17	69.02/92.35	66.14/92.63	0.313	89.21
	10	18.80(29.28)	47.72(1.049)	60.56/92.79	78.15/92.39	67.93/91.78	69.85/92.77	0.252	95.89
	15	24.34(15.48)	52.28(0.620)	65.10/93.49	72.28/93.61	72.20/93.32	71.64/93.89	0.249	**96.48**
SVHN	-	-	-	95.31	94.68	94.42	95.98	-	-
	1	14.99(45.39)	34.27(4.933)	40.83/83.77	52.79/84.89	52.57/84.99	52.63/85.06	0.349	2.049
	5	18.38(30.73)	48.98(0.907)	59.31/94.64	59.13/94.53	59.83/93.66	58.99/94.45	**0.368**	94.19
	10	25.84(13.02)	57.48(0.341)	80.67/95.06	70.53/94.46	67.08/93.88	74.25/94.76	0.227	94.19
	15	27.31(10.99)	55.30(0.438)	80.24/95.02	80.86/94.46	68.45/94.24	78.43/95.05	0.188	**99.74**

dataset that contains 73257 digits for training, 26032 digits for testing, and 531131 additional digits. All of the digits are RGB images with a size of 32 × 32. It is important to note that many images in the dataset do contain some distractors on the sides.

2) Training Details: By default, we train the classification network $\mathbf{M}_{net}$ for 200 epochs with a batch size of 128. Adam optimizer is adopted with an initial learning rate of 0.1. We decay the learning rate by 90% after 50 epochs. Meanwhile, for the GEAA framework, we train it for 15 phases, and each phase contains 30 epochs with a batch size of 100. At the beginning of each phase, we reset the initial learning rate to 0.0001 and decay it by 0.5 after 10 epochs. All of the experiments are performed on a Nvidia GTX 1080 Ti GPU. Unless stated otherwise, we set MIN to 10 and **T** to 0.9.

3) Evaluation Metrics: To evaluate the performance of GEAA, we introduce the Data Protection Rate (DPR). We use $\frac{\vec{P}-\vec{P}^+}{\vec{P}}$ to measure the degree of data disturbance and $\frac{\vec{P}^*}{\vec{P}}$ to measure the degree of data fidelity, where $\vec{P}$, $\vec{P}^+$ and $\vec{P}^*$ denote the classification accuracy trained on the original data, the perturbed data, and the denoised data. Based on these indicators, we define the DPR as follows:

$$\text{DPR} = \frac{1}{len(\Omega)} \sum_{i \in \Omega} \frac{\vec{P}_i - \vec{P}_i^+}{\vec{P}_i} \cdot (2^{(\frac{\vec{P}_i^*}{\vec{P}_i})^2} - 1) \tag{14}$$

where $len(\Omega)$ denotes the number of models used in the evaluation process. In fact, DPR deliberately increases the proportion of data fidelity, which means that we pay more attention to data reconstruction tasks rather than data disturbance degrees. Namely, the ultimate intention of data protection is to decrease the classification accuracy trained on the perturbed data while not affecting the normal classification accuracy as much as possible. In addition, we adopt the average value $\overline{\text{WEA}}$ to measure the source-identifying performance.

The $Noise$ is calculated by the L2-norm function, namely, $Noise(\mathcal{I}_1, \mathcal{I}_2) = \| \mathcal{I}_1 - \mathcal{I}_2 \|_2$.

Runtime We calculate the runtime on the CIFAR10 dataset. The training stage takes 4.5 hours (15 phases and each phase contains 30 epochs, batch_size = 100) and the prediction stage takes 8.76s (50000 samples, batch_size = 100). Furthermore, we also provide the runtimes of each submodule of GEAA in the prediction stage. We only calculate the forward-propagation time of each submodule. The double-stream adversarial attack, denoising reconstruction, and watermark extraction modules take 1.224s, 1.622s, and 2.361s, respectively. Therefore, a single forward propagation for the three submodules takes 0.0049, 0.0065, and 0.0095s.

B. Data Protection Performance of GEAA

In this section, we provide both qualitative and quantitative results to show the performance of the proposed GEAA. By default, we adopt ResNet as the target model $\mathbf{M}_{net}$ to understand the data distribution and verify the network performance on the CIFAR10 dataset. Five categories are selected from the dataset, denoted as $\mathcal{A}$, and the remaining categories are denoted as $\mathcal{B}$. Considering that users may manually remove perturbed data from the dataset, we inject perturbations into all samples in the training set. The perturbed data and the denoised data generated by the trained GEAA are gathered into local servers. Then, we train classification models mentioned above on stored data. Quantitative results, including the DPR and WEA, are provided in Table II. Table II also contains several lists of experiment results in different phases, helping potential readers understand how the double-stream adversarial

TABLE III

QUANTITATIVE RESULTS OF THE PROPOSED GEAA ALGORITHM WITH VARIOUS DATA DISTRIBUTION MODELS ON THE CIFAR10 DATASET ($\text{MIN} = 10$, $\mathbf{T} = 0.9$). THE DIGITS (*) DENOTE THE CLASSIFICATION ACCURACY OF THE MODELS TRAINED ON THE NORMAL TRAINING SET. PERTURBATION IS CALCULATED BY $Noise(\mathcal{I}^{+}, \mathcal{I})$, AND RECONSTRUCTION IS CALCULATED BY $Noise(\mathcal{I}^{*}, \mathcal{I})$. PSNR DENOTES PEAK SIGNAL TO NOISE RATIO

Data Distribution Model	PSNR(Noise)		Acc {Perturbation↑/Reconstruction↓}				DPR ↑	$\overline{\text{WEA}}$↑
	Perturbation ↑	Reconstruction ↓	ResNet(88.89)	VGG(86.78)	MobileNet(86.34)	DenseNet(92.12)		
ResNet	25.43(13.64)	52.44(0.609)	61.35/88.53	54.10/86.41	60.93/**86.34**	58.48/91.29	0.332	99.12
VGG	26.40(12.21)	64.38(0.154)	**49.66/88.57**	**49.55/87.04**	**47.25**/86.25	**48.62/91.55**	**0.446**	98.97
Mobilenet	25.55(13.46)	50.72(0.742)	64.9/88.54	72.57/86.93	73.64/86.20	70.35/91.10	0.202	**99.22**

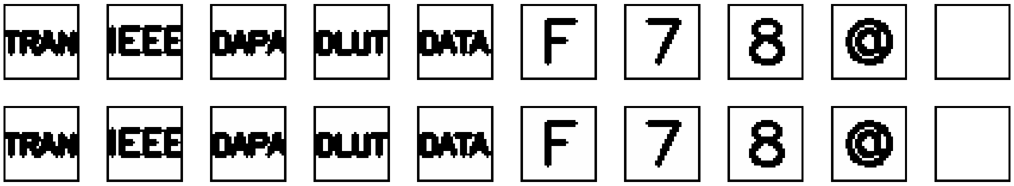

Fig. 6. Visual results of the watermark. The first line represents the original watermarks, and the second line is the reconstructed watermarks. We adopt the blank watermark as w_2 and other watermarks as w_1. w_1 includes words, letters of an alphabet, digits, or signs. w_1 and w_2 are the pre-defined watermarks for the denoised samples and benign samples.

attack and denoising reconstruction affect each other. Moreover, not all these datasets obtain the optimal performance by setting the final training phase to 15.

It is observed that the proposed GEAA not only inserts perturbations into clean samples, thereby preventing the shared data from training the model but also successfully recovers the initial data distribution from the perturbed data in the watermark embedding way. For instance, the classification accuracy of ResNet trained on the perturbed data decreases by 31% than trained on normal data, from 88.89% to 61.35%, whereas the classification accuracy of the ResNet model trained on the reconstructed data only decreases by 0.4%. Both the denoising reconstruction noise and double-stream adversarial attack noise decrease as the training phase advances, leading to classification accuracy improvement of the model trained on both kinds of data.

Additionally, we observe that the average difference between the watermarked images $\mathcal{I}^*$ and watermark-free images $\mathcal{I}$ in the last phase is less than 0.005 (normalized image), confirming that the high image quality (The average PSNR between reconstructed images and benign images is greater than 52dB.) is well preserved in the reconstruction process. Meanwhile, results demonstrate that even tiny reconstruction difference can hide enough watermark identification information, i.e. the WEA is greater than 95%. We provide several visual results of the reconstructed watermark in Fig. 6.

To better illustrate how the double-stream adversarial attack affects the network transferability, we provide several perturbed examples in Fig. 4 and Fig. 5 by setting MIN to 30, 10. Each perturbed image ADV in Fig. 4 and Fig. 5 contains two kinds of object information. We observe that transfer disturbances are greatly affected by the boundary constraint value. For instance, in the last row, the third column of Fig. 4, a 'ship' is visible in the perturbed image, while the source category 'car' is invisible. However, the source category of perturbed images in Fig. 5 are human distinguishable. Therefore, we set MIN to 10 by default.

Next, we maintain the same experimental settings to prove that GEAA is also effective for other datasets. Experimental results on Fashion-MNIST and SVHN datasets are provided in Table II. Similar experimental phenomena exist for these two benchmark datasets, i.e. the classification accuracy of models trained on both perturbed data and reconstructed data increases with the increment of the training phase. More visualization images are given in the supplement for the reader to view.

C. Influence of $\mathbf{M}_{net}$ *on GEAA*

To prove the effectiveness of $\mathbf{M}_{net}$ on the performance of GEAA, we first add various degrees of Gaussian noise to the initial training set and then train MobileNet (for clear comparison) using the noise-added training sets. The experimental results are represented in Fig. 7. By comparing the classification accuracies of models trained on the noise-added data with that of trained on the perturbed data, we observe that the data distribution model $\mathbf{M}_{net}$ is conducive to the generation of effective adversarial perturbations.

Under the same experimental settings, we evaluate the influence of various data distribution understanding models on the proposed GEAA performance. Experimental results are shown in Table III. The performance of GEAA with the VGG as the $\mathbf{M}_{net}$ (about a 44% classification accuracy decrease) is better than that of GEAA with ResNet and MobileNet. This indirectly verifies that VGG is more robust than ResNet and MobileNet; as a result, the GEAA with VGG as the data distribution understanding model spends amounts of time on fitting the double-stream adversarial attack task.

TABLE IV

RESEARCH ON THE INFLUENCE OF THE MODEL DISTRIBUTION ACCURACY ON GEAA (MIN = 10, **T** = 0.9). PERTURBATION IS CALCULATED BY $Noise(\mathcal{I}^{+}, \mathcal{I})$, AND RECONSTRUCTION IS CALCULATED BY $Noise(\mathcal{I}^{*}, \mathcal{I})$. PSNR DENOTES PEAK SIGNAL TO NOISE RATIO

CIFAR10		PSNR(Noise)		Acc {Perturbation↑/Reconstruction↓}				DPR ↑	$\overline{\text{WEA}}$↑
		Perturbation ↑	Reconstruction ↓	ResNet(88.89)	VGG(86.78)	MobileNet(86.34)	DenseNet(92.12)		
Phase15	Init	29.22(8.82)	37.10(3.56)	59.85/79.25	63.82/75.76	56.86/74.25	61.28/83.86	0.228	50
	Low	26.53(12.03)	58.02(0.32)	68.99/88.36	64.70/**87.11**	64.95/86.29	68.20/**91.33**	0.245	**99.33**
	Mid	25.43(13.64)	52.28(0.62)	61.35/**88.53**	**54.10**/86.41	60.93/**86.34**	58.48/91.29	**0.332**	99.12
	High	22.41(19.32)	48.04(1.01)	**55.53**/75.46	59.13/72.31	**51.33**/73.29	**54.28**/84.34	0.252	99.12

TABLE V

RESEARCH ON THE INFLUENCE OF LOWER BOUNDARY CONSTRAINT VALUES ON THE GEAA PERFORMANCE (**T** = 0.9). PERTURBATION IS CALCULATED BY $Noise(\mathcal{I}^{+}, \mathcal{I})$, AND RECONSTRUCTION IS CALCULATED BY $Noise(\mathcal{I}^{*}, \mathcal{I})$. PSNR DENOTES PEAK SIGNAL TO NOISE RATIO

CIFAR10		PSNR(Noise)		Acc {Perturbation↑/Reconstruction↓}			DPR ↑	$\overline{\text{WEA}}$↑
ResNet(79.38)		Perturbation ↑	Reconstruction ↓	ResNet(88.89)	VGG(86.78)	MobileNet(86.34)		
MIN	0	25.89(12.94)	30.60(7.528)	51.06/51.65	52.32/51.85	48.62/55.47	0.123	55.54
	1	22.38(19.39)	51.20(0.702)	61.43/88.85	68.57/86.60	59.68/85.94	0.274	**99.41**
	10	25.43(13.64)	52.44(0.609)	61.35/88.53	54.10/86.41	60.93/86.34	**0.324**	99.12
	30	17.68(33.3)	50.46(0.765)	70.61/88.52	67.55/86.17	63.85/86.95	0.229	99.24
	60	15.77(41.52)	34.68(4.703)	45.47/73.98	48.12/64.41	40.36/63.05	0.249	99.23

TABLE VI

RESEARCH ON THE INFLUENCE OF THE TEMPERATURE VALUES ON THE GEAA PERFORMANCE (MIN = 10). PERTURBATION IS CALCULATED BY $Noise(\mathcal{I}^{+}, \mathcal{I})$, AND RECONSTRUCTION IS CALCULATED BY $Noise(\mathcal{I}^{*}, \mathcal{I})$. PSNR DENOTES PEAK SIGNAL TO NOISE RATIO

CIFAR10		PSNR(Noise)		Acc {Perturbation↑/Reconstruction↓}			DPR ↑	$\overline{\text{WEA}}$↑
ResNet(79.38)		Perturbation ↑	Reconstruction ↓	ResNet(88.89)	VGG(86.78)	MobileNet(86.34)		
Temperatures	0	19.00(28.60)	43.03(1.80)	53.25/80.59	55.77/86.69	52.36/74.78	0.311	99.36
	0.3	23.99(16.10)	49.54(0.85)	62.72/79.93	54.29/**87.44**	55.52/76.53	0.287	99.40
	0.6	24.00(15.51)	49.85(0.82)	**44.57**/88.44	**46.46**/87.11	**46.72**/86.24	**0.473**	99.33
	0.9	25.43(13.64)	52.42(0.61)	61.35/88.53	54.10/86.41	60.93/ **86.34**	0.324	99.12
	1.2	22.49(19.14)	51.87(0.65)	49.70/88.23	54.73/87.03	54.73/86.32	0.390	**99.48**
	1.5	26.48(12.09)	48.31(0.98)	54.31/**88.64**	52.81/86.55	65.22/85.90	0.339	99.44

D. Distribution Accuracy Influence on GEAA

In this subsection, we mainly study the impact of the data distribution accuracies on data protection performance.

We have prepared four ResNet models with different classification accuracies, 10.0%, 57.68%, 79.38%, and 88.89%. In terms of the non-trained network, we only store the initial network weights. Next, four GEAA models based on these prepared distribution models are trained simultaneously on the same training set. Similar to Section IV-B, we adopt generated perturbed images and denoised images to train four popular classification networks and only provide experimental results of the last phase in Table IV. An examination of the results presented in Table IV shows that the reconstruction loss and adversarial attack loss are positively correlated with the data distribution accuracy, implying that too high or too low distribution accuracies decrease the ultimate DPR. Meanwhile, it is difficult to generate effective adversarial perturbations with low precision $\mathbf{M}_{net}$, and reconstruct the data distribution with high precision $\mathbf{M}_{net}$.

More detailed experiment results about the influence of distribution accuracy on GEAA are further displayed in Fig. 8, which again confirm that the reconstruction loss and adversarial attack loss are closely correlated with data distribution accuracies. Meanwhile, results demonstrate the non-trained GEAA cannot understand the training data distribution.

E. Ablation Study for Hyperparameters

Next, we study the influence of the boundary constraint and the temperature on GEAA performance. In this subsection, we set the medium-precision ResNet as the data distribution understanding model, the initial temperature to 0.9, and the lower boundary constraint to a series of values of 0, 1, 10, 30, 60. The 0 boundary constraint denotes no further limitations on the perturbation degree. The constraint MIN

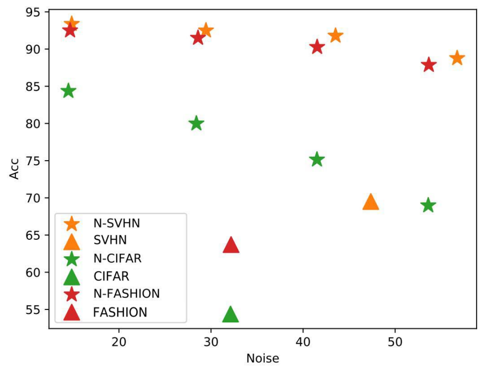

Fig. 7. Research on the influence of the data distribution model on GEAA performance. N-{} denotes the case of adding Gaussian noise to the training set. The dataset name represents the scenario of generating adversarial perturbations by the trained GEAA. Acc is the evaluation accuracy of the trained MobileNet model on the testing set.

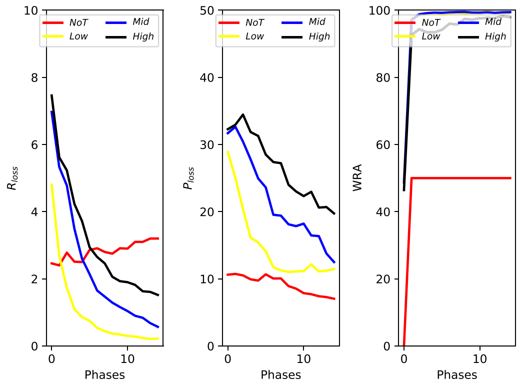

Fig. 8. Research on the influence of the model distribution accuracies on GEAA performance. R_{loss} denotes the reconstruction loss, and P_{loss} represents the average value of perturbations.

TABLE VII

THE COMPARISON OF GEAA WITH EXISTING ATTACKS. GEAA(*) DENOTES EXPERIMENT RESULTS UNDER THE SETTING OF $\mathbf{T}$ = 0.6 AND USING THE '*' AS THE TARGET MODEL. ε AND ξ REPRESENT THE SINGLE STEP DISTURBATION AND TOTAL DISTURBATIONS. AR IS THE ATTACK RATE. THE LAST THREE COLUMNS DENOTE MODEL CLASSIFICATION ACCURACIES TRAINED ON PERTURBED SAMPLES

Attacks	ε	ξ	AR	ResNet	VGG	MobileNet
PGD	0.001	0.1635	90.41	72.07	70.90	70.78
PGD	0.0055	0.1656	100.0	65.09	58.45	63.88
BIM	0.01	0.0459	92.68	75.40	74.95	75.18
BIM	0.0055	0.0310	85.03	78.12	78.06	77.23
MIFGSM	0.0055	0.0644	92.48	73.46	71.01	68.41
GEAA(Resnet)	-	0.0610	98.52	44.57	46.46	46.72
GEAA(VGG)	-	0.0484	99.11	**39.65**	**40.10**	**38.79**

TABLE VIII

EVALUATION OF THE ROBUSTNESS OF THE WATERMARK EXTRACTION PROCEDURE. ξ DENOTES THE MAXIMUM PERTURBATION ON EACH IMAGE PIXEL (NO NORMALIZE). ACCURACY MEANS THE CLASSIFICATION ACCURACY OF TRAINED MODEL ON $\mathcal{I}^{+} + \xi$

Methods/Noise	ξ(Accuracy)			
	0 (88.64)	1 (88.66)	3 (87.12)	5 (85.42)
GEAA($\overline{\text{WEA}}$)	99.82	98.22	90.17	50.00
GEAA$_{\xi=5}$($\overline{\text{WEA}}$)	99.82	99.66	99.15	98.71

TABLE IX

EVALUATION OF REVERSIBLE WATERMARKING IN ATTACK PERFORMANCE

Method	PSNR	Resnet	VGG	MobileNet	DenseNet
Default	-	88.89	86.78	86.34	92.12
GEAA	26.4	**44.57**	**46.46**	**46.60**	**46.82**
Ycocg-R [49]	35.6	88.42	87.66	85.9	91.78
Interpolation [50]	52.1	88.82	87.87	86.06	91.49

for the transfer noise $(\mathcal{T}_{\mathcal{I}_{\mathcal{A}}}, \mathcal{T}_{\mathcal{I}_{\mathcal{B}}})$ is adopted to restrict the minimum perturbation, thereby indirectly reducing the availability of the perturbed data $\mathcal{I}^{*}$. Experimental results are given in Table V and verify that the larger MIN improves the double-stream adversarial attack performance while it degrades the recoverability of the denoising reconstruction. For instance, the training classification accuracy based on the denoised data drops sharply at the large boundary constraint value (60), a decrease of approximately 21%. Moreover, the model classification accuracies for the 0 boundary constraint trained on both perturbed data and the denoised data are less than 55%, indicating the importance of the boundary constraint.

Similar to the analysis for the lower boundary constraint MIN, we next study the role of the starting temperature in the network optimization process. By default, we set MIN to 10 and the number of optimization phases to 15. The minimum value of the starting temperature is set to 0, with six groups of experiments at an interval of 0.3. Experimental results presented in Table VI show that the value of the starting temperature strongly affects the classification accuracy of the denoised data-trained model, i.e. low temperatures (T = 0, 0.3) leading to a classification accuracy decrease of 7%.

F. The Comparison of GEAA With Existing Attacks

[11], [12] proposed the reversible adversarial attacks to deceive the well-trained classification model. However, there are not available public training or testing codes. Luckily, these two methods are based on existing adversarial attacks and reversible data hiding technologies.

Therefore, to verify the data protection performance of GEAA, we set a series of comparative experiments with existing adversarial attacks. Specifically, we adopt the popular attacks including PGD [13], BIM [14], MIFGSM [47], to generate adversarial examples, and then train the proven effective classification networks using these generated adversarial examples. Both PGD, BIM, and MIFGSM are multi-step adversarial attack methods. Compared with BIM, PGD adopts randomly generated values as initial perturbations. Experimental results are shown in Table VII. By observing the decline

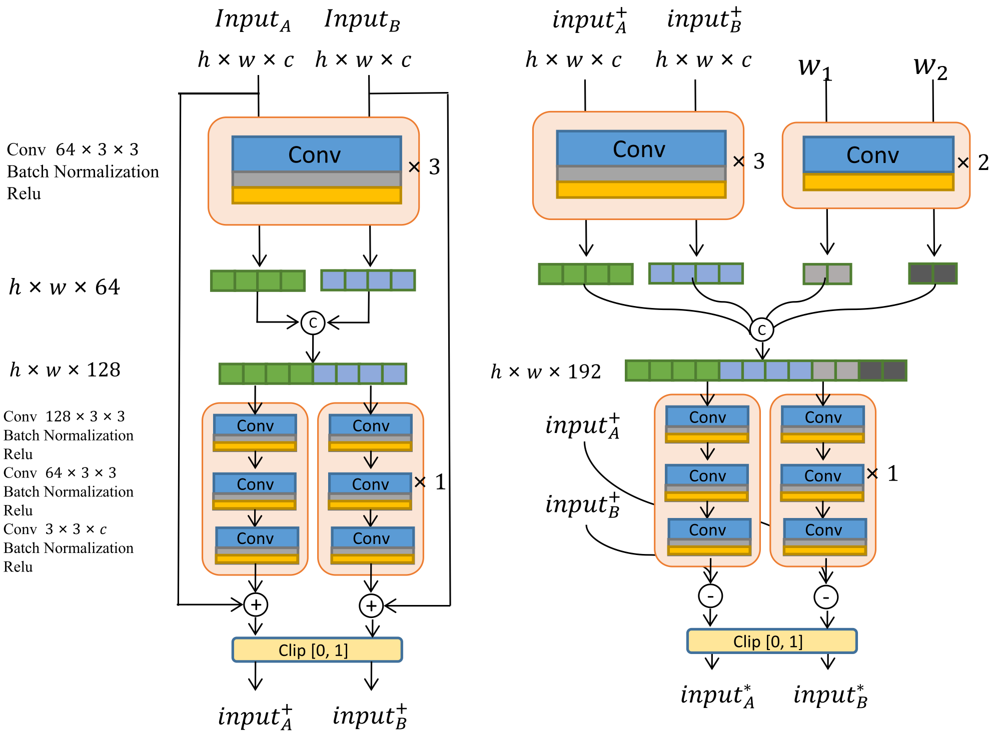

Fig. 9. The detailed network structure of GEAA. GEAA includes the double-streamed adversarial attack module, denoising reconstruction module, and watermark extraction module.

TABLE X

RESEARCH ON THE INFLUENCE OF EMBEDDED WATERMARKS ON THE GEAA PERFORMANCE (MIN = 10, $\mathbf{T} = 0.9$). PERTURBATION IS CALCULATED BY $Noise(\mathcal{I}^+, \mathcal{I})$, AND RECONSTRUCTION IS CALCULATED BY $Noise(\mathcal{I}^*, \mathcal{I})$

CIFAR10		PSNR(Noise)		Acc {Perturbation↑/Reconstruction↓}			DPR ↑	$\overline{\text{WEA}}$↑
ResNet(79.38)		Perturbation ↑	Reconstruction ↓	ResNet(88.89)	VGG(86.78)	MobileNet(86.34)		
Watermark	'TRAN'	19.68(26.47)	48.58(0.950)	67.87/88.59	69.71/86.27	59.24/86.18	0.247	99.26
	'F'	20.97(22.80)	49.88(0.818)	67.84/88.46	69.40/86.28	65.82/86.17	0.222	99.66
	'8'	25.58(13.42)	51.61(0.670)	63.42/72.82	64.31/81.86	56.63/74.91	0.209	99.99
	'@'	26.39(12.21)	52.02(0.639)	66.14/88.79	62.32/86.15	69.52/85.97	0.241	99.98
	Maskall	25.43(13.64)	52.43(0.609)	61.35/88.53	54.10/86.41	60.93/86.34	0.324	99.12

TABLE XI

RESEARCH ON THE INFLUENCE OF THE NUMBER OF EMBEDDED WATERMARKS ON THE GEAA PERFORMANCE (MIN = 10, $\mathbf{T} = 0.9$). EACH WATERMARK IS COMPOSED OF ONE TO FOUR LETTERS OR NUMBERS. PSNR DENOTES PEAK SIGNAL TO NOISE RATIO

CIFAR10		PSNR(Noise)		Acc {Perturbation↑/Reconstruction↓}			DPR ↑	$\overline{\text{WEA}}$↑
		Perturbation ↑	Reconstruction ↓	ResNet(88.89)	VGG(86.78)	MobileNet(86.34)		
Watermark	10	25.72(13.2)	52.10(0.633)	62.12/88.56	53.98/86.45	61.34/85.99	0.319	99.92
	50	25.48(13.57)	51.52(0.677)	63.51/88.42	55.20/86.07	61.55/85.64	0.306	99.20
	100	25.56(13.45)	51.79(0.656)	62.07/88.51	54.26/86.29	59.88/86.15	0.324	50.0

rate in the classification accuracy of benign samples, we can conclude that under similar perturbation degrees, the proposed GEAA shows a better data protection performance than the mentioned popular attacks.

In our opinion, this is because existing adversarial attacks generate perturbations that only disrupt the local features extracted by the target model. However, these local features are only a subset of classification features that can distinguish different categories. Therefore, the retrained network based on these adversarial examples finds other local features to classify samples. However, the proposed GEAA adopts the double-stream adversarial attack network. Each sample

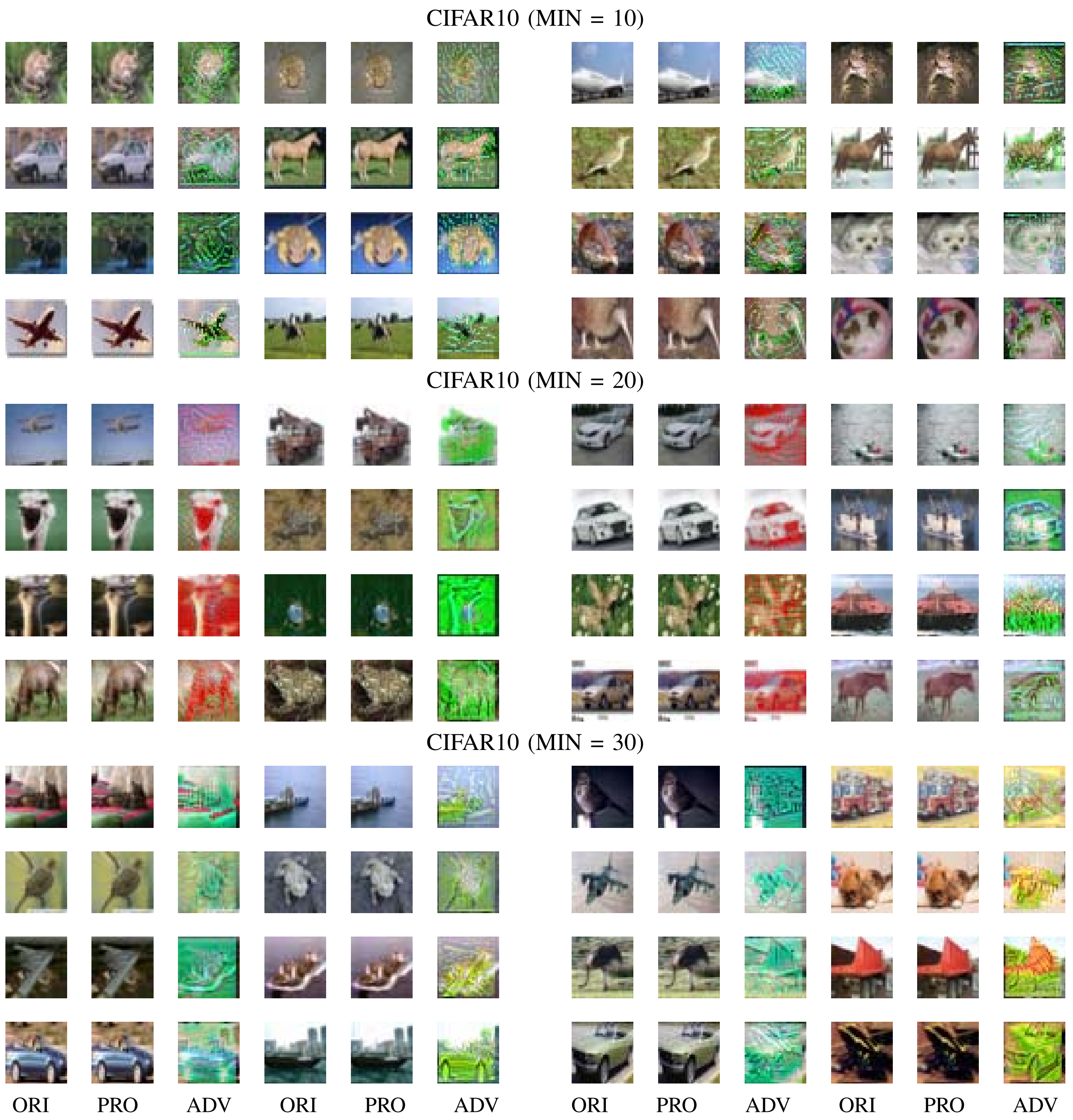

Fig. 10. Visual results of the generated perturbed data and denoised data on CIFAR10 dataset. ORI, PRO and ADV denote the orginal image, the denoised sample and the perturbed sample.

generated by GEAA contains two kinds of category features.

G. Multiple Embedding-Single Extraction

We further analyze the correlation between the GEAA performance and the embedded watermarks. Considering the potential non-authorized data sharing problem, GEAA adopts the multiple embedding-single extraction. Intuitively, complicated watermarks are difficult to embed and reconstruct. To solve this problem, we select a series of binary greyscale images as the identity card ID and embed the message into data and GEAA weights. The mutual memories of the data and model weights alleviate the problem that low-quality data cannot carry adequate identification information.

As shown in Fig. 6, we select 9 different binary greyscale images as embedded watermarks and set five groups of comparative experiments, embedding ‘TRAN’, ‘F’, ‘8’, or ‘@’ and embedding all watermarks. We observe from Table X that the single watermark embedding leads to an attack performance decrease. Perturbations under the complicated watermarks are more efficient for data protection. In our opinion, this is because the embedded watermark increases the difficulty of the denoising reconstruction, which in turn impacts the double-stream adversarial attack task.

Will the efficiency of the multiple embedding-single extraction be affected due to a large number of denoisers in practical applications? To answer such question, we conduct supplementary experiments in this section. We randomly generate 100 watermarks. Each watermark is composed of one

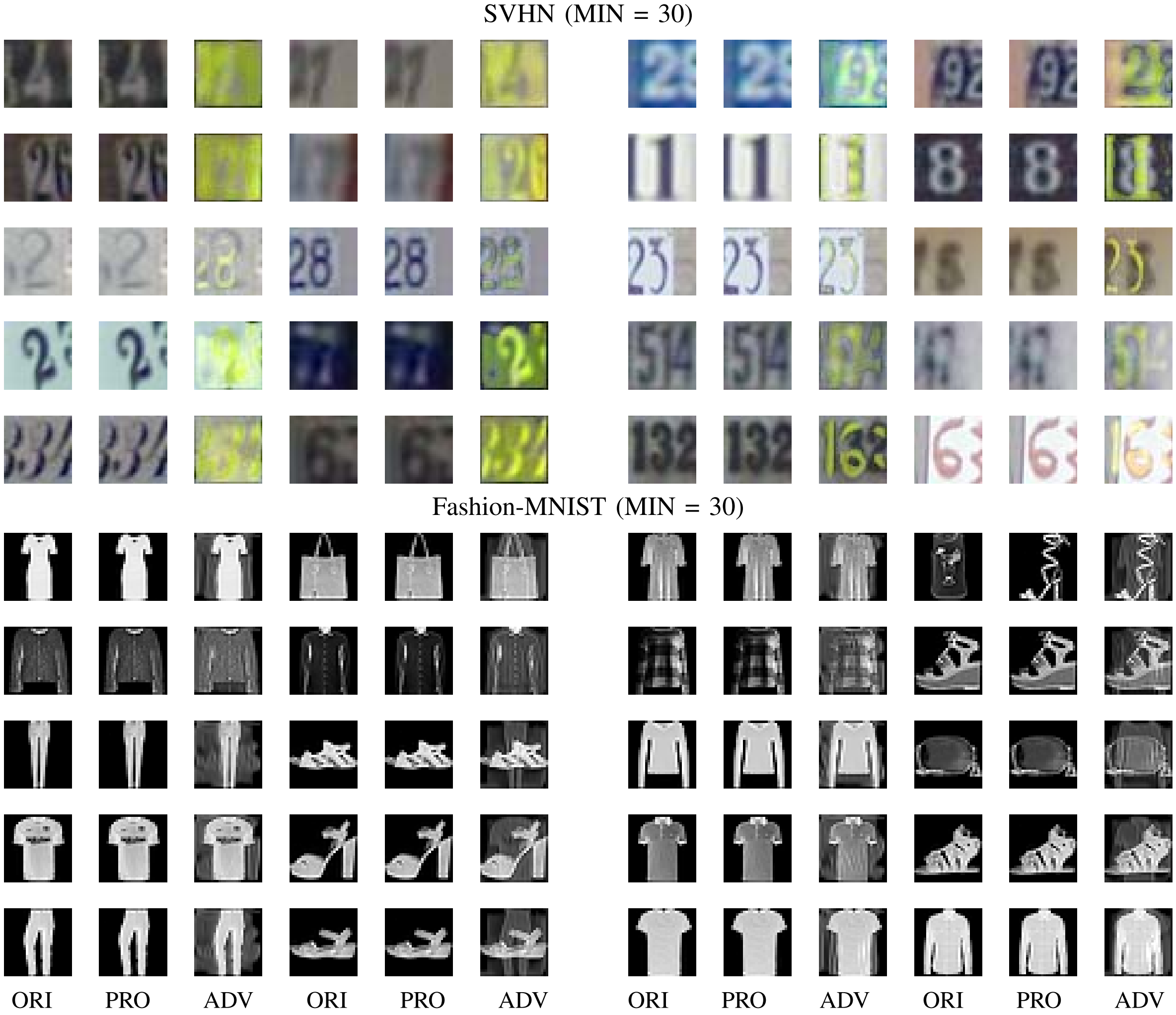

Fig. 11. Visual results of the generated perturbed data and denoised data on the SVHN and Fashion-MNIST datasets. ORI, PRO and ADV denote the original image, the denoised sample and the perturbed sample.

to four letters or numbers. For a fair comparison, we only modify the number of embedded watermarks, 10, 50, 100. Experiment results are given in Table. XI. We observe that a large number of extractors affects the efficiency of watermark extraction, but GEAA still performs robustness when embedding 50 watermarks. Therefore, how to embed enough watermarks in the sharing data protection mode is still a potential challenge.

Section III-C has explained why we inject watermark features into the denoised data, and Section III-D describes how to extract embedded watermarks. However, it is unclear whether the watermark extraction process is robust. Therefore, we randomly add perturbations ξ (Normal distribution and uniform distribution) into the reconstructed images to evaluate the robustness of the watermark extraction procedure. Experiment results are illustrated in Table VIII. We observe that the default watermark extraction procedure is vulnerable to perturbation. Next, we introduce the adversarial training, which incorporates the perturbation into the training process. Namely, inputs of the watermark extraction is $\mathcal{I}^{+} + \xi$ instead of $\mathcal{I}^{+}$. In this way, we successfully improve the robustness of the watermark extraction.

H. What Are the Benefits of GEAA?

Finally, we illustrate several core competitiveness of GEAA.

1) Higher Safety Performance: (1) The double-stream adversarial attack and denoising reconstruction modules of GEAA exist in pairs, even GEAA models trained under the same experimental settings cannot denoise each other. (2) GEAA only provides users with the generated perturbation data and denoising reconstruction module (part of GEAA). (3) As discussed in Section II, traditional adversarial attacks defeat the model by adding crafted perturbations to benign samples. To implement such attacks, the attacker needs to provide available benign samples. However, there is no need to attack the model once the benign samples are available in GEAA. Therefore, GEAA has high safety performance.

2) A Variant of Reversible Adversarial Examples: GEAA could reconstruct the original data distribution from generated adversarial examples.

3) Visualize, Attack and Identify: Visualizing data helps users better understand specific data information, e.g. understanding the sample distribution. Traditional encryption methods [48] can protect data from being leaked but with poor visualization performance. Furthermore, we demonstrate experimentally that reversible watermarking techniques are not very effective in data protection, even though they show excellent visualization performance [49], [50].

Conventional watermarking methods reconstruct embedded watermarks instead of benign samples. Reversible watermarking technologies can reconstruct original samples. Therefore, we investigate the possibility of using reversible watermarking algorithms with common watermarking methods to replace GEAA.

Here, we first illustrate several limitations of the reversible watermarking alternative to the double-stream adversarial attack module. (1) Limited embedding information: To reduce the availability of watermarked samples, we embed the maximum amount of information into each training sample. (2) Few public codes: We find two public available codes [49], [50] to detect whether watermarked samples could protect dataset information. Results in Table IX illustrate that reversible watermarking techniques are not very efficient in replacing double-stream adversarial attacks. (3) Relatively simple watermarking process: Compared with watermarking, the double stream adversarial attack is obtained by optimizing amounts of model parameters.

We then explain the limitations of common watermarking techniques alternative to the denoising reconstruction module. When researchers only consider recognition tasks, common watermarking methods are sufficient. However, the data owner should transmit the key to the downloader in the data sharing scenario. In GEAA, the denoising reconstruction module (playing the role of a key) consists of two parts, data distribution reconstruction, and watermark embedding. An available alternative to denoising reconstruction is to combine reversible watermark extraction with common watermarking methods. Such techniques make shared data under threat because they are parsable, e.g. downloaders remove the watermarking process and only use the first part. In contrast, the trained denoising reconstruction module is an indivisible whole.

V. Conclusion

In this work, we have proposed a newly designed double-stream architecture – guided erasable adversarial attack – including a double-streamed adversarial attack module, a denoising reconstruction module, and a watermark extraction module, for protecting high-quality shared data. The GEAA performance is evaluated on three benchmark datasets and four popular classification models. Qualitative and quantitative results show that the proposed GEAA can effectively resolve the shared data protection problem.

In the future, we expect to improve data protection performance, such as reducing perturbation levels while ensuring high DPR values. Assigning the best matching class pair in dataset partitioning provides a feasible direction. Additionally, we tend to strengthen a series of adversarial example processing methods and propose a metric to measure the degree of perturbation and estimate whether the inserted perturbation is robust enough for post-processing. Furthermore, combining GEAA with backdoor attacks seems to be a feasible research idea. The backdoor attack refers to the attacker implanting some backdoors in the model by modifying the training data. In this way, a model trained on data generated by a GEAA + backdoor attack has backdoor effects. By detecting various backdoor effects, data owners can verify the source of the leaked data.

References

[1] H. Zhao, J. Jia, and V. Koltun, "Exploring self-attention for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 10076–10085.

[2] A. Dosovitskiy *et al.*, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[3] Q. Hu *et al.*, "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11108–11117.

[4] T. Wolf *et al.*, "Transformers: State-of-the-art natural language processing," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*, 2020, pp. 38–45.

[5] C. Szegedy *et al.*, "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.

[6] Y. Liu *et al.*, "Trojaning attack on neural networks," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2018, pp. 1–15.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[8] P. W. Koh, J. Steinhardt, and P. Liang, "Stronger data poisoning attacks break data sanitization defenses," 2018, *arXiv:1811.00741*.

[9] Y. Liu *et al.*, "Reflection backdoor: A natural backdoor attack on deep neural networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2020, pp. 182–199.

[10] X. Jia, X. Wei, X. Cao, and X. Han, "Adv-watermark: A novel watermark perturbation for adversarial examples," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 1579–1587.

[11] J. Liu, W. Zhang, K. Fukuchi, Y. Akimoto, and J. Sakuma, "Unauthorized AI cannot recognize me: Reversible adversarial example," 2018, *arXiv:1811.00189*.

[12] Z. Yin, H. Wang, L. Chen, J. Wang, and W. Zhang, "Reversible adversarial attack based on reversible image transformation," 2019, *arXiv:1911.02360*.

[13] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," Jun. 2017, *arXiv:1706.06083*.

[14] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," 2016, *arXiv:1607.02533*.

[15] D. A. Tamburri, "Design principles for the general data protection regulation (GDPR): A formal concept analysis and its evaluation," *Inf. Syst.*, vol. 91, Jul. 2020, Art. no. 101469.

[16] H. Fang *et al.*, "Deep template-based watermarking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 4, pp. 1436–1451, Apr. 2020.

[17] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 657–672.

[18] J. Zhang *et al.*, "Deep model intellectual property protection via deep watermarking," 2021, *arXiv:2103.04980*.

[19] P. Addesso, M. Cirillo, M. Di Mauro, and V. Matta, "ADVoIP: Adversarial detection of encrypted and concealed VoIP," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 943–958, 2019.

[20] P. Addesso, M. Barni, M. Di Mauro, and V. Matta, "Adversarial Kendall's model towards containment of distributed cyber-threats," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3604–3619, 2021.

[21] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.

[22] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, Sep. 2019.

[23] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[24] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1765–1773.

[25] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Oct. 2019.

[26] A. Bhattad *et al.*, "Unrestricted adversarial examples via semantic manipulation," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–15.

[27] S. Sarkar, A. Bansal, U. Mahbub, and R. Chellappa, "UPSET and ANGRI: Breaking high performance image classifiers," 2017, *arXiv:1707.01159*.

[28] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.

[29] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1369–1378.

[30] R. Taori, A. Kamsetty, B. Chu, and N. Vemuri, "Targeted adversarial examples for black box audio systems," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2019, pp. 15–20.

[31] H. Chen, H. Zhang, P.-Y. Chen, J. Yi, and C.-J. Hsieh, "Attacking visual language grounding with adversarial examples: A case study on neural image captioning," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2018, pp. 2587–2597.

[32] Y. Xu *et al.*, "Exact adversarial attack to image captioning via structured output learning with latent variables," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4135–4144.

[33] H. Kwon, Y. Kim, H. Yoon, and D. Choi, "Selective audio adversarial example in evasion attack on speech recognition system," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 526–538, 2019.

[34] S. Joshi, J. Villalba, P. Zelasko, L. Moro-Velazquez, and N. Dehak, "Study of pre-processing defenses against adversarial attacks on state-of-the-art speaker recognition systems," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4811–4826, 2021.

[35] W. Tang, B. Li, S. Tan, M. Barni, and J. Huang, "CNN-based adversarial embedding for image steganography," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 8, pp. 2074–2087, Aug. 2019.

[36] M. Liu, W. Luo, P. Zheng, and J. Huang, "A new adversarial embedding method for enhancing image steganography," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4621–4634, 2021.

[37] V. Antun, F. Renna, C. Poon, B. Adcock, and A. C. Hansen, "On instabilities of deep learning in image reconstruction and the potential costs of AI," *Proc. Nat. Acad. Sci. India A, Phys. Sci.*, vol. 117, no. 48, pp. 30088–30095, 2020.

[38] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.

[39] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.

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

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[42] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[43] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[44] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Handbook Systemic Autoimmune Diseases*, vol. 1, no. 4, 2009.

[45] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.

[46] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," 2013, *arXiv:1312.6082*.

[47] Y. Dong *et al.*, "Boosting adversarial attacks with momentum," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9185–9193.

[48] M. Kaur and V. Kumar, "A comprehensive review on image encryption techniques," *Arch. Comput. Methods Eng.*, vol. 27, no. 1, pp. 15–43, Jan. 2020.

[49] A. Roy, R. S. Chakraborty, and R. Naskar, "Reversible color image watermarking in the YCoCg-R color space," in *Proc. Int. Conf. Inf. Syst. Secur.* Cham, Switzerland: Springer, 2015, pp. 480–498.

[50] L. Luo, Z. Chen, M. Chen, X. Zeng, and Z. Xiong, "Reversible image watermarking using interpolation technique," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 1, pp. 187–193, Mar. 2009.

Mengnan Zhao received the B.S. degree in electronic and information engineering from the Tianjin University of Technology in 2018 and the M.S. degree from the School of Information and Communication Engineering, Dalian University of Technology, in 2021. His research interests include adversarial examples and deep learning.

Bo Wang (Member, IEEE) received the B.S. degree in electronic and information engineering and the M.S. and Ph.D. degrees in signal and information processing from the Dalian University of Technology, Dalian, China, in 2003, 2005, and 2010, respectively. From 2010 to 2012, he was a Post-Doctoral Research Associate with the Faculty of Management and Economics, Dalian University of Technology. He is currently an Associate Professor with the School of Information and Communication Engineering, Dalian University of Technology. His current research interests focus on the areas of multimedia processing and security, such as digital image processing and forensics.

Wei Wang (Member, IEEE) received the B.E. degree in computer science and technology from North China Electric Power University in 2007. Since 2012, he has been with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, where he is currently an Assistant Professor. His research interests include pattern recognition, image processing, and digital image forensics, including watermarking, steganalysis, and tampering detection.

Yuqiu Kong received the B.S. and Ph.D. degrees from the School of Mathematical Sciences, Dalian University of Technology (DUT), China, in 2014 and 2019, respectively. She is currently a Faculty with the School of Innovation and Entrepreneurship, DUT.

Tianhang Zheng received the B.S. degree from Peking University, China, in 2016, and the M.S. degree from the University at Buffalo, Buffalo, NY, USA, in 2019. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Toronto, Toronto, ON, Canada. His research interests include adversarial learning, data poisoning, privacy, and fairness.

Kui Ren (Fellow, IEEE) received the Ph.D. degree from the Worcester Polytechnic Institute, Worcester, MA, USA. He is currently a Professor of computer science and technology and the Director of the Institute of Cyberspace Research, Zhejiang University, Hangzhou. China. His current research interests include cloud and outsourcing security, wireless and wearable system security, and artificial intelligence security. He is a Distinguished Scientist of the ACM. He was a recipient of the NSF CAREER Award in 2011 and the IEEE CISTC Technical Recognition Award 2017.