Self-Recoverable Adversarial Examples: A New Effective Protection Mechanism in Social Networks

Jiawei Zhang, Jinwei Wang, Member, IEEE, Hao Wang, and Xiangyang Luo

Abstract—Nowadays, users upload numerous photos to social network platforms to share their daily lives. These photos contain numerous personal information, which can be easily captured by intelligent algorithms. To improve privacy security, we aim to form a protection mechanism by exploiting adversarial examples, which can mislead and disrupt intelligent algorithms. However, the existing adversarial attack lacks the study on recoverability, attack ability, and reversibility, which makes them unable to serve as an effective protection mechanism. To address this issue, we propose a recoverable generative adversarial network to generate self-recoverable adversarial examples. By modeling the adversarial attack and recovery as a united task, our method can minimize the error of the recovered examples while maximizing the attack ability, resulting in better recoverability of adversarial examples. To further boost the recoverability of these examples, we exploit a dimension reducer to optimize the distribution of adversarial perturbation. The experimental results prove that the adversarial examples generated by the proposed method present superior recoverability, attack ability, and robustness on different datasets and network architectures, which ensure its effectiveness as a protection mechanism in social networks.

Index Terms—Social networks, deep learning, adversarial attack and recover, privacy protection.

I. INTRODUCTION

Deep neural networks (DNNs) have achieved excellent performance in many tasks, such as steganography [1] and facial expression recognition [2]. However, recent works show that DNNs are vulnerable to adversarial examples. The adversarial examples can be generated by adding some special and imperceptible noise to the normal examples, and it can make the target DNNs output the wrong predictions.

The generation manner of adversarial examples can be categorized into white-box and black-box. With the knowledge of the structure and parameters of the targeted DNNs, the adversarial examples [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13] can be generated in a white-box manner, including the optimization-based method L-BFGS [3], the gradient-based methods FGSM [5] and various iterative variants [6], [8], [14], [15]. Besides, due to the transferability [16], the adversarial examples can perform attacks in a black-box manner. For example, the adversarial examples generated according to model A can also mislead model B, even if the structure and parameters of A are different from B. Although the existing adversarial defense methods [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27] can improve DNNs’ robustness, some new adversarial examples can always break these defense methods. As a result, the adversarial examples are practical in real-world conditions and pose a great threat to the reliability of DNNs.

Although adversarial examples bring destruction and threat to DNNs, we try to exploit these negative impacts to fulfill the positive potential of these examples serving as a new protection mechanism for privacy security in social networks. Specifically, nowadays, users upload numerous photos to social network platforms for sharing their daily lives. These photos contain personal private information, including users’ social relationships, properties, and identities. This information can be easily detected and collected by malicious intelligent algorithms (e.g., DNNs), which greatly threatens the security of social users’ privacy [28], [29], [30]. Therefore, we aim to form a protection mechanism by exploiting adversarial examples. Compared with other protection methods [31], crafting an image as an adversarial example is imperceptible and can effectively prevent the malicious DNNs from detecting, classifying, and further analyzing the image content [32]. However, the existing adversarial attack lacks the study on recoverability and reversibility, which makes them unable to
serve as an effective protection mechanism. Therefore, we consider crafting a self-recoverable adversarial example (SRAE), which owns high attack ability under various cases (e.g., disturbance, adversarial defense) and can only be recovered near losslessly by ourselves. As shown in Fig. 1, the error between the original image and the recovered image is still imperceptible despite being magnified by 10 times. Based on this, SRAE can serve as a new protection mechanism (shown in Fig. 2) and avoid the data being identified, collected, and analyzed by malicious DNNs while remaining harmless to ourselves.

In this scenario, the authorized user or enterprise (e.g., Taobao, Amazon) can remove the protection of these images to gather the fresh interests of users and further provide better individual service. At the same time, the unauthorized user or enterprise can not remove the protection and obtain any valid information. Moreover, when delivering information through social networks, the sender can safely deliver the information to the receiver, avoiding intelligent algorithms intercepting and analyzing the information. More important, the transferability of adversarial examples can give a great generalization ability to this protection mechanism, making it still effective in the unaware adversarial environment.

In this paper, we propose a recoverable generative adversarial network (RGAN) to generate the proposed SRAE in an end-to-end way. Unlike the existing white-box attacks, which constantly need the gradient information of target DNNs, the proposed RGAN is free from the structure and parameters. More importantly, instead of treating the adversarial attack and recovery as two separate and independent tasks, we attempt to model the attack and recovery as a united task by the proposed framework. For the purpose of learning the distribution of adversarial perturbation better, the recovery part is specially designed and trained jointly and dynamically with the generation part in a pipeline. The experimental results show that SRAE has better recoverability on different datasets and network architectures. SRAE minimizes the error of the recovered example while maximizing the attack ability, and outperforms the combinations of existing adversarial attack and defense methods.

To further boost the recoverability of SRAE, we study the relationship between the recovered error and the distribution of adversarial perturbation. Given network architecture with certain simulation capabilities, we observe that adversarial perturbations with lower intensity or simpler structures are easier to be recovered. Therefore, we design a dimension reducer to optimize the distribution of adversarial perturbation. We show that SRAE optimized with the proposed dimension reducer can be recovered to original examples with negligible error, further satisfying the harmless requirement of recovered examples. To summarize, the main contributions of this paper are:

- Our proposed RGAN is the first attempt to model adversarial attack and recovery, a pair of mutually-inverse challenges, as a united task. Powered by joint dynamic training, the recoverable adversarial examples maximize the attack ability and can be recovered near losslessly by our RGAN.
- We study the effects of adversarial perturbation distribution on recoverability, demonstrating that perturbation with lower intensity or simpler structures is easier to recover. Therefore, we design a dimension reducer to optimize the perturbation distribution, further boosting the recoverability.
- The experimental results show that the proposed method presents superior recoverability than the combinations of state-of-the-art attack and defense methods on different datasets and network architectures, which prove it can serve as a new effective protection mechanism for privacy security in social networks.

II. RELATED WORK

In this section, we briefly describe the notion used in this paper. Furthermore, due to the lack of recoverable adversarial examples, we review some related works about adversarial attack and defense. In Section IV, these attacks and defense will be combined to serve as competitive solutions with our SRAE.

We denote $x$ as the clean image from the dataset and $y$ as the ground-truth label of $x$. A target deep neural network is represented by model $f(\cdot)$, which can achieve $f(x) = y$. The goal of adversarial attack is to find a perturbation $r$, which meet $f(x + r) \neq y$. Let $x^{adv}$ represent adversarial example, which means $x^{adv} = x + r$.

A. Existing Methods for Adversarial Attack

Existing methods for generating adversarial examples can be categorized into three groups: optimization-based, gradient-based, and generation-based.
Optimization-based methods (e.g., L-BFGS [3], C&W [4]) solve the generation of adversarial example as an optimization problem, which combines the magnitude of perturbation $r$ with the attack ability of adversarial example $x^{adv}$ as the optimization goal [3]. C&W further [4] introduces a variable $w$ using the tanh function to eliminate the box constraint. C&W can be formulated as:

$$
\min_w ||x^{adv} - x||_2^2 + \lambda f(x^{adv}),
$$

where  
$$
f(x^{adv}) = \max_{i \neq y} \{Z(x^{adv})_i - Z(x^{adv})_y, -\kappa\},
$$

and  
$$
x^{adv} = \frac{1}{2}(\tanh(\text{arctanh}(x) + w) + 1). \quad (1)
$$

$|| \cdot ||_2$ is the $L_2$ norm, $\lambda$ is the hyper parameter to balance the loss, $\kappa$ controls the confidence level, $Z(\cdot)_i$ is the logit to the $i$-th class, and $Z(\cdot)_y$ is the logit to the ground-truth class.

Gradient-based methods (e.g., FGSM [5]) maximize the loss function $J(x, y)$ by a chosen perturbation step size $\epsilon$ according to the gradient direction $\nabla_x J$, which can be formulated as:

$$
x^{adv} = x + \epsilon \text{sign}(\nabla_x J(x, y)). \quad (2)
$$

sign(\cdot) represents the sign function. Although the gradient-based methods have certain drawbacks in attack ability, they are much faster than the optimization-based method. Furthermore, the vanilla FGSM [5] was extended and optimized with iteration (I-FGSM [6], PGD [7], DDN [9]), momentum (MI-FGSM [8]), which highly improved the attack ability, visual effect, and transferability.

Generation-based method (e.g., AdvGAN [10], Rob-GAN [19]) train another model to generate perturbation for the target model. These methods have similar flexibility as the optimization-based method but cost less time. Moreover, these methods can generate adversarial examples without the parameters and structure of the target model.

B. Existing Methods for Adversarial Defense

Existing methods for adversarial defense can also be categorized into three main directions: adversarial training, adversarial denoising, and adversarial detection.

Adversarial training [5], [7], [17], [26], [33] augment the training dataset with adversarial examples to train a robust model. The trained model has a higher classification accuracy on adversarial examples. Due to this intriguing property, various adversarial training strategies [7], [17], [26], [33] have surged since the original one [5] was proposed. However, adversarial training inevitably reduces the accuracy of the model on clean images, and the cost of adversarial training is also very high, which make the adversarial training on large-scale dataset hard to perform.

Adversarial denoising [18], [19], [20] exploit a denoiser to pre-process the input, which can erase the aggressive of adversarial examples. For example, Jin et al. [18] proposed APEGAN, attempting to eliminate the adversarial perturbation by a generative adversarial network. Liao et al. [19] proposed a high-level representation denoiser (HGD) to suppress the influence of the adversarial perturbation. Zhou et al. [20] proposed an adversarial noise removing network (ARN) to remove adversarial perturbation by exploiting attack-invariant features (AIF).

Adversarial detection [21], [22], [23], [24], [25] aim at determine whether the input is adversarial or not, rather than eliminating the adversarial property. For example, Liu et al. [21] used steganalysis features for adversarial detection, which is sensitive to statistics changes. Wang et al. [22] proposed a generalized network for adversarial detection, which is independent of the target network attack by adversaries.

III. PROPOSED METHOD

A. Overview

Our goal is to develop a learnable, end-to-end model for self-recoverable adversarial examples (SRAE) that can form a protection mechanism. Due to the combination of adversarial property and high recoverability, SRAE can be aggressive to the attacker while being harmless to ourselves. Note that the recoverability doesn’t mean SRAE are fragile and can be easily destroyed. In opposite, SRAE are robust to various transformations $T(\cdot)$ existing in social networks and other adversarial defense methods (e.g., JPEG compression, Gaussian noise, denoising filter, APE [18], ARN [20]). In our scenario, SRAE can only be precisely recovered by the proposed recoverable generative adversarial network (RGAN). Definitions of several symbols are provided in Table I.
Fig. 3. The overall framework of our recoverable generative adversarial network (RGAN). Our main idea is to model the adversarial attack and recovery as a united task. By optimizing the perturbation distribution with the dimension reducer, SRAE can achieve high attack ability and recoverability to form a powerful protection mechanism.

| Symbols | Descriptions |
|---------|--------------|
| $x$     | Original examples. |
| $r$     | Adversarial perturbation. |
| $r^*$   | Recovered perturbation. |
| $r'$    | Coarse-grained perturbation. |
| $G$     | Generator. |
| $DR$    | Dimension reducer. |
| $D$     | Discriminator. |
| $C$     | Target classifier. |
| $R$     | Recover. |
| $G();$  | The output of $G$; $DR();$ $D();$ $C();$ $R();$. |
| $L_{G, adv}$ | The adversarial loss of $r$. |
| $L_{G, mse}$ | The intensity loss of $r$. |
| $L_{G, dis}$ | The indistinguishability loss of $r$. |
| $L_{R, adv}$ | The adversarial loss of $r^*$. |
| $L_{R, mae}$ | The error loss of $r^*$. |
| $\lambda_1; \lambda_2$ | The weight of $L_{G, mae}; L_{G, dis}$. |

The calculation details of each loss are listed in Section III C.

As shown in Fig. 3, the proposed RGAN consists of five parts: a generator $G$, a dimension reducer $DR$, a discriminator $D$, a target classifier $C$, and a recover $R$. Specifically, the generator $G$ takes the original examples $x$ as the input and outputs the perturbations $G(x)$. Then, the perturbations $G(x)$ are optimized with the dimension reducer $DR$. Afterward, the adversarial examples can be obtained by $x + DR(G(x))$. Next, the adversarial examples are sent to the discriminator $D$ and the target classifier $C$ for indistinguishability and aggressiveness optimization. Meanwhile, the adversarial examples are sent to the recover $R$, which aims to recover these examples back to the original examples. It is worth noting that the proposed framework is different from the existing denoising methods. The proposed RGAN jointly trains the generator $G$ and the recover $R$, rather than separating the recovery from the attack, resulting in a better recoverability. This allows the generated SRAE to serve as a protection mechanism for privacy security in social networks.

B. Recoverable Generative Adversarial Network (RGAN)

The generator $G$ is the starting point and aims to generate perturbation according to the features of the input $x$. It consists of an encoder, a bottleneck module, and a decoder. The encoder, consisting of three convolution layers, normalization, and ReLU activation function, extracts features from the clean images. Correspondingly, the decoder exploits transposed convolution layers, normalization, and activation functions to map the features to perturbation with the same size as the image. To increase the representative capacity of the generator $G$, we add the bottleneck module between the encoder and the decoder, which consists of residual blocks [34].

The recover $R$ aims to recover the adversarial examples. We study the effect of different structures of the recover $R$ towards the recoverability. Suppose the recover $R$ is more complex and deeper than the generator $G$, whether it can better recover the adversarial examples.

Fig. 4 shows the transferability heatmap of adversarial examples generated by FGSM [5] ($\epsilon = 8/255$). FGSM is one of the fastest and most effective gradient-based attacks. Therefore, we perform it on various network architectures to study the transferability of adversarial examples. As shown in Fig. 4, diving into the transferability of adversarial examples, we find that the examples generated for a specific network transfer better to its homologous networks. For example, the adversarial examples generated for one ResNet are more likely to be adversarial to another network within the ResNet family. Moreover, within the network family, the adversarial examples generated for networks with similar depth tend to have better transferability. For example, the adversarial examples generated for ResNet-50 can transfer better to ResNet-101.
Fig. 4. Transferability heatmap of adversarial examples generated by FGSM ($\epsilon = 8/255$). Values represent attack success rates (%) of the examples against a target model (column), which is generated according to a source model (row).

Fig. 5. Adversarial examples and perturbations generated by the optimization-based C&W, the gradient-based PGD, the generation-based AdvGAN. From the left: original examples, adversarial examples, adversarial perturbation.

Fig. 6. Adversarial perturbation generated by setting different $\lambda_2$. $\lambda_2$ is the hyper parameter to control the perturbation intensity.

Fig. 7. Line graph of $L_{G, mse}$ and $L_{G, adv}$. $L_{G, mse}$ reflect the magnitude of perturbation and $L_{G, adv}$ reflect the attack ability (lower is better).

method. As shown in Fig. 5, the perturbation generated by the generation-based method AdvGAN [10] is larger and messier than the gradient-based method PGD [7] and optimization-based method C&W [4]. This observation pushes us to consider how to reduce the redundancy of perturbation generated by the generation-based method.

As shown in Fig. 6, magnifying the weight $\lambda_2$ of $L_{G, mse}$ does help to reduce the perturbation intensity. However, as shown in Fig. 7, the larger $\lambda_2$ brings lower $L_{G, mse}$ also brings the increase of $L_{G, adv}$, which leads to the decrease of attack ability. Note that the generation-based method is not the white-box attack. The generation-based method does not require parameters and structure of the target model after training. This difference makes the perturbation intensity important in ensuring the attack ability of the generation-based method. The smaller intensity is harder to optimize (e.g., when $\lambda_2 = 14$, the $L_{G, adv}$ increases dramatically, which indicates the perturbation almost loses the adversarial property.)

Therefore, we focus on reducing the complexity of the perturbation structure rather than perturbation intensity. To reduce the perturbation complexity, we implement a down-sample and up-sample operation within the dimension reducer $DR$ to obtain a coarse-grained perturbation.

As shown in Fig. 8, by setting larger kernel sizes of down-sample and up-sample operation, the perturbation structure gradually changes from fine-grained to coarse-grained. Compared with fine-grained perturbation, the coarse-grained perturbation has lower structural complexity, which improves
the recoverability. In addition, the coarse-grained perturbation is more robust, which is hard to be destroyed by various transformations (e.g., JPEG compression). More importantly, as shown in Fig. 8, there exist significant differences between the distribution of coarse-grained and fine-grained perturbations. The differences in distribution improve the resistance of our SRAE against existing adversarial defense methods (e.g., ARN [20]), which ensures the SRAE can only be recovered by ourselves.

However, the coarse-grained perturbation is inaccurate, which would also lead to drawbacks in attack ability and increase the training difficulty. To release these drawbacks, we add a skip connection within the dimension reducer $DR$. The effective perturbation outputted by the generator $G$ can skip the down-sample and up-sample operation through the skip connection. The preservation of this effective perturbation can improve the attack ability. In addition, the skip connection can improve the performance of DNNs, by reducing the training difficulty brought by the deepening of networks [34]. With the combination of down-sample, up-sample, and skip connection within the dimension reducer $DR$, we effectively reduce the complexity of the perturbation to boost the recoverability of SRAE further. We evaluate the performance of different combinations of down-sample, up-sample, and skip connection in Section IV-A.2.

Here we give the derivation and proof of the second observation: the less complex perturbation is easier to be recovered. (see Appendix A for more detail proof) Given a image $x$ with $n$ pixels, suppose $r = [r_1, r_2, r_3, \ldots, r_n] \sim \mathbb{R}^n$ denote the perturbation adding to the each pixel of the image and $r^* = [r^*_1, r^*_2, r^*_3, \ldots, r^*_n] \sim \mathbb{R}^n$ denote the recovered perturbation. The adversarial example can be obtained by $x+r$, and the recovered example can be obtained by $x+r-r^*$. Here we use $\Delta$ to represent the difference between $r$ and $r^*$ under $L_2$ norm, which can be calculated as:

$$\Delta = ||r^* - r||_2$$

$$= \sqrt{(r^*_1 - r_1)^2 + (r^*_2 - r_2)^2 + \ldots + (r^*_n - r_n)^2}. \quad (3)$$

Let the $r' = [r'_1, r'_2, r'_3, \ldots, r'_n] \sim \mathbb{R}^n$ denote the perturbation $r$ pass the dimension reducer $DR$. Correspondingly, the difference $\Delta'$ between $r^*$ and $r'$ can be calculated as:

$$\Delta' = ||r^* - r'||_2$$

$$= \sqrt{(r^*_1 - r'_1)^2 + (r^*_2 - r'_2)^2 + \ldots + (r^*_n - r'_n)^2}. \quad (4)$$

For clarity, we take the average pooling and unpooling with size $m$ (e.g., $m = 9$ for the kernel size is $3 \times 3$) as the down-sample and up-sample operation within the dimension reducer $DR$ for illustration, which means

$$r'_1 = r'_2 = \ldots = r'_m = \bar{r}, \quad (5)$$

where

$$\bar{r} = \frac{1}{m} \sum_{i=1}^{m} r_i. \quad (6)$$

The difference between $(\Delta'_m)^2$ and $(\Delta_m)^2$ can be calculated as

$$(\Delta'_m)^2 = \sum_{i=1}^{m} (r^*_i - r'_i)^2 = \sum_{i=1}^{m} (r^*_i - r_i)^2 = m\bar{r}^2 - m\bar{r}^2. \quad (7)$$

Meanwhile,

$$\sum_{i=1}^{m} (r_i - \bar{r})^2 \geq 0$$

$$m\bar{r}^2 \geq m\bar{r}^2. \quad (8)$$

Combine the Eq. 7 with Eq. 8, it can be obtained that

$$\Delta' \leq \Delta \quad (9)$$

which means $r^*$ is closer to $r'$. This proves the less complex perturbation is easier to be recovered. As a result, we can further boost the recoverability of SRAE by the proposed dimension reducer $DR$.

**Discriminator** $D$ is trainable and aims to tell the difference between adversarial and clean examples. Part of the loss of the Discriminator $D$ is used to guide the Generator $G$ to generate the perturbation, which can ensure that the generated adversarial examples are as similar as possible to the clean examples in terms of data distribution and high-dimensional features (e.g., statistical features).

**C. Loss Function**

The loss function for optimizing the generator $G$ can be described as:

$$L_G = L_{G_{\text{adv}}} + \lambda_1 L_{G_{\text{dis}}} + \lambda_2 L_{G_{\text{mse}}}. \quad (10)$$

where

$$L_{G_{\text{adv}}} = H(x + DR(G(x)), y),$$

$$L_{G_{\text{dis}}} = \log(1 - D(x + DR(G(x)))), \quad (12)$$

$$L_{G_{\text{mse}}} = ||D(G(x))||_2^2. \quad (13)$$

$|| \cdot ||_2$ is the $L_2$ norm. $L_{G_{\text{adv}}}$ aims to improve the attack ability by calculating the cross-entropy loss $H(.)$. $L_{G_{\text{dis}}}$ aims to make sure the indistinguishability between the generated examples and the original examples. $L_{G_{\text{mse}}}$ aims to constrain
the perturbation intensity. $\lambda_1$ and $\lambda_2$ are the weights of the corresponding losses.

We exploit the $L_{R, mse}$ to optimize the recover $R$, which can be described as:

$$L_{R, mse} = ||R(x + DR(G(x))) - DR(G(x)))||^2_2. \quad (14)$$

To comprehensively measure the recoverability, we also calculated the adversarial loss of the recovered examples, which can be described as:

$$L_{R, adv} = H(x + DR(G(x)) - R(x + DR(G(x))), y). \quad (15)$$

We also exploit $L_{R, adv}$ to optimize the recover $R$ as competitive solutions (see Section IV-A.3).

For the discriminator $D$, the loss function can be described as:

$$L_D = log(D(x)) + log(1 - D(x + DR(G(x))))), \quad (16)$$

where $log(D(x))$ and $log(1 - D(x + DR(G(x))))$ aim to calculate the loss of the original examples and adversarial examples be recognized, respectively.

IV. EXPERIMENT AND ANALYSIS

For a fair comparison, all experiments are conducted on an NVIDIA RTX 2080ti, and all methods are implemented by PyTorch. For the attack methods PGD [7], C&W [4], and DDN [9], we use the implementation from adversarialtorch [35]. For the defense methods, we use the official implementation of ARN1 [20], APE2 [18], Image Super-resolution3 [36], JPEG Compression4 [37], Pixel Deflection5 [38], Random Resizing and Padding6 [39], Image quilting + total variance minimization7 [40]. All the comparative methods mentioned above are conducted with their default setting. The learning rate for the generator $G$, the recover $R$, and the discriminator $D$ are all set to $10^{-3}$ and decrease $10^{-1}$ times for every 50 epochs (total of 150 epochs). Meanwhile, $\lambda_1 = 10, \lambda_2 = 1$ for the $L_G$, and kernel size $= 2 \times 2$.

For better measuring the generality, we conduct comparison experiments on various network architectures with different datasets. Specifically, we train LetNet-5 on the MNIST [41](the image size is $28 \times 28$), and ResNet-50, DenseNet-121, and MobileNetV3 on Caltech-2568 as the target models. Caltech-256 [42] is selected from the Google image dataset. This dataset is divided into 256 categories, with more than 80 images in each category.

As described in Section I, our SRAE aims to serve as a protection mechanism against malicious intelligent detection or classification algorithm. Therefore, we aim to ensure the target model misclassifies SRAE in various cases (e.g., disturbance, adversarial defense). More importantly, SRAE should only be able to recover near losslessly by our recover $R$.

A. Ablation Study

1) Network Structure: Here, we study the recoverability brought by different depths of networks. To prove the generator $G$ and the recover $R$ with similar depth can improve the recoverability, we train the generator $G$ and the recover $R$ with various depths. As shown in Fig. 9(a), we fix the total depth of bottleneck in the generator $G$ and the recover $R$ equal to 12. $G - R$ represents the depth difference of the generator $G$ and the recover $R$. From the changes of $L_{R, mse}$, we can observe that the similar depth of the generator $G$ and the recover $R$ can recover the adversarial examples with fewer errors.

Based on this, we set the generator $G$ and the recover $R$ with the same depth to study the effect of the total depth on the recoverability. As shown in Fig. 9(b), $G + R$ represents the sum of depth. Although the $L_{R, adv}$ does not reduce apparently, the $L_{R, mse}$ keeps reducing with the increase of total depth. It reveals the deeper network can recover the adversarial example with fewer errors. However, to make a trade-off between the parameter quantity and the recoverability, we set the total depth equal to 8 (the depth of both the generator $G$ and the recover $R$ is equal to 4) to conduct the following experiments.

2) Dimension Reducer $DR$: For the dimension reducer $DR$, we first perform an ablation experiment to prove its effectiveness. Then, we focus on the improvement brought by combinations of various down-sample, up-sample, and skip connection operations within the dimension reducer $DR$.

Table II shows the performance of RGAN without the dimension reducer $DR$, which is represented by NA. Furthermore, Table II also shows the loss of the adversarial examples and the recovered examples generated by RGAN with combinations of various down-sample, up-sample, and skip connection operations. Note that the maximum unpooling for up-sample requires the index of the down-sample operation. Thus, only the combination of maximum pooling and maximum unpooling for both down-sample and up-sample is conducted. From Table II, $L_{G, adv}$ and $L_{R, adv}$ reflect the adversarial property of adversarial examples and recovered examples. $L_{G, mse}$ reflects the error between the adversarial examples and the original examples, while $L_{R, mse}$ reflects
As shown in Table II, with the comparison of the $L_{R_{adv}}$ and $L_{R_{mse}}$, the proposed RGAN employing the dimension reducer $DR$ can recover the adversarial examples better than without employing dimension reducer $DR$. Consistent with the discussion in the proposed method, the less complex perturbation is easier to be recovered. From the observation, several combinations in Table II (e.g., convolution and transposed convolution for both up-sample and down-sample without skip connection) can not converge the loss. However, by exploiting the skip connection, each combination of up-sample and down-sample can be trained. This proves the skip connection can greatly reduce the training difficulty and make the loss converged better. Furthermore, with the combination of skip connection and convolution for down-sample, the $L_{R_{adv}}$ and $L_{R_{mse}}$ are less than other combinations. This represents the adversarial property, and the error of the recovered examples both achieve minimization. The advantages prove the convolution is more flexible, which keeps a more effective adversarial perturbation during the down-sample and further boosts the recoverability.

The column $L_{G_{adv}}$ in Table II shows the dimension reducer $DR$ brings some drawbacks in attack ability while improving the recoverability. Therefore, to evaluate the attack ability, we further explore the effect of different reducing levels by setting different kernel sizes with the corresponding stride (e.g., stride = 2 for kernel size = $2 \times 2$).
TABLE IV
THE CONVOLUTION PARAMETERS OF THE GENERATOR, RECOVER, DIMENSION REDUCER, AND DISCRIMINATOR

| Network       | Block   | Layer         | Input channel | Output channel | Kernel size | Stride |
|---------------|---------|---------------|---------------|----------------|-------------|--------|
| Encoder       | Convolution | 3 8          | 3x3 1         | 1              |
| Generator     | Convolution | 8 16         | 3x3 2         | 2              |
|               | Convolution | 16 32        | 3x3 2         | 2              |
| Bottleneck    | Convolution | 32 32        | 3x3 1         | 1              |
|               | Convolution | 32 32        | 3x3 1         | 1              |
|               | Convolution | 32 32        | 3x3 1         | 1              |
| Decoder       | Transposed convolution | 32 16 | 3x3 2 | 2 |
|               | Transposed convolution | 16 8    | 3x3 2         | 2              |
|               | Transposed convolution | 8 3     | 6x6 1         | 1              |
| Dimension reducer | Down-sample | Convolution | 3 3          | 2x2 2         |
|               | Up-sample | Average unpooling | 3 3 | 2x2 2        |
| Discriminator | Convolution | 3 8          | 4x4 2         | 2              |
|               | Convolution | 8 16         | 4x4 2         | 2              |
|               | Convolution | 16 32        | 4x4 2         | 2              |
|               | Convolution | 32 1         | 1x1 1         | 1              |

1 "=" represent the discriminator does not contain any special blocks.

We choose the combination of convolution for down-sample and average pooling for up-sample without skip connection to evaluate the performance of attack ability. As mentioned in Section I, our SRAE aims to serve as a protection mechanism on social network platforms. Thus, we evaluate the robustness of SRAE against the disturbance of widespread image manipulation (e.g., JPEG compression, Gaussian noise), which reflects the effectiveness and applicability of the proposed methods on social network platforms. As shown in Fig. 10, without disturbance, the attack success rate (ASR) of RGAN without employing the dimension reducer DR is 98%, while the ASR of RGAN is still higher than 59% in most disturbance cases.

When exploiting dimension reducer DR, the ASR is between 90% and 95%. And a larger kernel size brings lower ASR. It proves that a larger kernel size will lead to a coarser-grained perturbation, resulting in a disadvantage in attack ability. However, as shown in Fig. 10, the coarse-grained perturbation brought by the dimension reducer DR also greatly improves the robustness of SRAE against disturbances. Thanks to the coarse-grained perturbation, over 90% of the generated examples remain adversarial in most disturbance cases. Moreover, with the increase in kernel size, SRAE becomes more and more robust. The robustness against these disturbances further ensures the effectiveness and security of the proposed SRAE served as a protection mechanism in social networks.

3) Loss Function: Here, we study the effect of different loss functions for the recover R. As mentioned above, the recoverability can be measured by $L_{R_{adv}}$ and $L_{R_{mse}}$. We combine the $L_{R_{adv}}$ and $L_{R_{mse}}$ with different weights $\alpha$ and $\beta$ to form target loss function.

As shown in Table III, peak signal to noise ratio (PSNR) within the column Adversarial reflects the difference between adversarial and original example, while PSNR within the column Recovered reflects the difference between recovered and original example. ACC reflects the classification accuracy for adversarial and recovered examples. It can be observed from Table III that only taking $L_{R_{adv}} (\alpha = 0, \beta = 1)$ as the loss function can not achieve satisfactory recoverability. The difference between the recovered and original examples is even larger than the difference between the adversarial and original examples. In addition, we try to alleviate between $L_{R_{mse}}$ and $L_{R_{adv}}$ by setting different $\alpha$ and $\beta$. However, the improvement of PSNR can not be consistent with improving ACC. These phenomena partly reveal the defects of the decision boundary of the target network. Recovering the examples according to this decision boundary may enlarge the difference between the recovered and original examples. Thus, we only exploit $L_{R_{mse}}$ for the optimization, which is superior in both difference and recoverability.

Moreover, Table III shows the overall correlation between recoverability and attack ability was negative. In the first column of Table III, the highest accuracy (82.76%, higher ACC in the column Recovered means better recoverability) of recovered samples corresponds to the high (not highest but still relatively high) accuracy (2.10%, higher ACC in the column Adversarial means worse attack ability) of the adversarial attack. In the fourth column of Table III, the lowest accuracy (81.91%) of recovered samples corresponds to the lowest accuracy (1.79%) of the adversarial attack. This is because attack ability and recoverability are related to the intensity and complexity of the perturbation. The growth in perturbation intensity increases attack ability but inevitably decreases recoverability. It also further justifies improving recoverability by optimizing the perturbation structure.

After conducting all ablation studies, the detailed parameters of the generator G, recover R, dimension reducer DR, and discriminator D are summarized as Table IV unless specified.

B. Recoverability Comparison

Fig. 11 shows some samples and perturbations generated and recovered by the proposed RGAN. It can be seen in the column Error that the adversarial examples are recovered to original examples near losslessly. Even if the error magnitude
is magnified by 10 times, the difference is still imperceptible. In addition, it can be observed that the differences between both the adversarial examples and the original sample are imperceptible. The imperceptibility ensures that our users can share images on social networks without suffering from image quality degradation. More importantly, it also proves that SRAE has good concealment, which would not arouse suspicion.

Table V further shows the recoverability results on different datasets with various network architectures. Due to the lack of recoverability study of adversarial examples, we combine the classical DDN [9] with state-of-the-art adversarial recovery or defense methods to serve as competitive solutions with our SRAE. We also compare C&W [4] and PGD [7] in Table VI and Table VII. The recoverability is measured from two aspects: the difference between the recovered and original examples and the adversarial property of the recovered examples. Specifically, the difference is reflected by the two criteria: the $L_2$ norm of the error (smaller is better) and PSNR (larger is better). And the adversarial property is reflected by the classification error rate (CER, lower is better) of the target network towards these recovered examples. From Table V, minor difference and lower CER represent better recoverability of the solution. It can be observed that our RGAN consistently achieves the best recoverability on both the small size MNIST (with image size $28 \times 28$) and the large size Caltech-256 (with image size $224 \times 224$) with various network architectures. This advantage ensures the effectiveness and generality of our RGAN served as a protection mechanism.
On MNIST, the combination of DDN for attack with several defense methods (e.g., [18], [20], [36], [37]) shows competitive performance, recovering around 98% of the adversarial examples. In comparison, our RGAN recovers near 99% of the examples with much more minor errors ($L_2$ norm reduced more than halved). Besides, our RGAN can maintain the recoverability (still can recover near 99% of the examples with a small error) when transferring to larger datasets and other network structures. However, the above competitive combinations on MNIST don’t show a satisfactory transfer of recoverability.

On Caltech-256, the combination of DDN and Das et al. [37] outperform other combinations in recoverability. In comparison, our RGAN still recovers 3% of the examples more than this combination. Note that the defense method proposed by Das et al. [37] are based on JPEG Compression. This means that Das et al. [37] reduces the CER based on the destruction rather than the recovery of adversarial perturbation, which can be reflected by the larger errors between the recovered and original examples (larger $L_2$ norm and smaller PSNR). As a result, thanks to the better recoverability, the proposed method can better serve as a protection mechanism than these combinations.

C. Adaptive Attack Evaluation

As described in Section I, SRAE aims to serve as a new effective protection mechanism against malicious intelligent detection algorithms in social network platforms. To more comprehensively measure the effectiveness of this protection mechanism, we evaluate the attack ability under different disturbance conditions in Fig. 10. Apart from disturbance conditions, if the attackers are aware of our adversarial protection mechanism, can they destroy our SRAE by exploiting the existing adversarial defense methods? Thus, we further evaluate the attack ability of SRAE under the state-of-the-art adversarial defense methods.

As shown in Fig. 12, the attack success rate of our SRAE from left to right, we evaluate the attack success rate (ASR) of our SRAE against various state-of-the-art adversarial defense methods.
high attack success rate (from 71% to 94%). It can be observed from Fig. 10 and Fig. 12 that our SRAE can maintain strong attack ability in both interference and adversarial defense conditions, which proves SRAE can serve as an effective and robust protection mechanism for users’ privacy in social network platforms.

V. CONCLUSION

Despite the destruction and threat brought by the adversarial examples, we transformed these negative effects into a positive protection mechanism in this paper. We proposed a recoverable generative adversarial network (RGAN) to generate self-recoverable adversarial examples (SRAE). Specifically, by joint dynamic training of the generator G and the recover R, the proposed model improved the recoverability while maintaining the attack ability. Besides, by studying the effects of perturbation intensity and complexity on the recoverability, we designed a dimension reducer DR to optimize the perturbation distribution and further boost the recoverability. Experimental results demonstrated that our model presented superior recoverability than the combinations of state-of-the-art attack and defense methods on different datasets and network architectures. The advantage in recoverability, attack ability, and robustness ensures the effectiveness and generality of our model served as a new effective protection mechanism for privacy security in social networks.

Since the proposed RGAN is lightweight, in future work, we will try to explore the upper limit for recoverability by training a deeper and wider network architecture with more data.

APPENDIX A

PROOF

Here we provide a proof of \( \Delta' \leq \Delta \).

Given a image \( x \) with \( n \) pixels, supposing \( r = [r_1, r_2, r_3, \ldots, r_n] \sim \mathbb{R}^n \) denote the perturbation adding to the each pixel of the image, \( r' = [r'_1, r'_2, r'_3, \ldots, r'_n] \sim \mathbb{R}^n \) denote the perturbation \( r \) pass the dimension reducer \( DR \), and \( r^* = [r^*_1, r^*_2, r^*_3, \ldots, r^*_n] \sim \mathbb{R}^n \) denote the recovered perturbation. Here we use \( \Delta \) to represent the difference between \( r \) and \( r^* \) under \( L_2 \) norm, which can be described as

\[
\Delta = ||r^* - r||_2 = \sqrt{(r^*_1 - r_1)^2 + (r^*_2 - r_2)^2 + \ldots + (r^*_n - r_n)^2}.
\]

\( \Delta' \) represent the difference between \( r' \) and \( r^* \) under \( L_2 \) norm, which can be described as

\[
\Delta' = ||r^* - r'||_2 = \sqrt{(r^*_1 - r'_1)^2 + (r^*_2 - r'_2)^2 + \ldots + (r^*_n - r'_n)^2}.
\]

We take the average pooling and unpooling with size \( m \) (e.g., \( m = 9 \) for the kernel size is \( 3 \times 3 \)) as the down-sample and up-sample operation within the dimension reducer \( DR \) for illustration, which means

\[
r'_1 = r'_2 = \ldots = r'_m = \overline{r},
\]

where

\[
\overline{r} = \frac{1}{m} \sum_{i=1}^{m} r_i. \quad (m \leq n)
\]

Then, the difference between \((\Delta'_m)^2\) and \((\Delta_m)^2\) can be calculated as

\[
(\Delta'_m)^2 - (\Delta_m)^2 = \sum_{i=1}^{m} (r^*_i - r'_i)^2 - \sum_{i=1}^{m} (r^*_i - r_i)^2
\]

\[
= \sum_{i=1}^{m} (r^*_i - r_i)^2 - \sum_{i=1}^{m} (r^*_i - r_i)^2
\]

\[
= \sum_{i=1}^{m} (r^*_i)^2 - 2\overline{r} \sum_{i=1}^{m} r^*_i + m\overline{r}^2
\]

\[
- \sum_{i=1}^{m} (r^*_i)^2 + 2 \sum_{i=1}^{m} r^*_i r_i - \sum_{i=1}^{m} (r_i)^2
\]

\[
= -2m\overline{r}^2 + m\overline{r}^2 - 2m\overline{r}^2 + m\overline{r}^2
\]

\[
= m\overline{r}^2 - m\overline{r}^2. \quad (21)
\]

Meanwhile,

\[
\sum_{i=1}^{m} (r_i - \overline{r})^2 \geq 0
\]

\[
\sum_{i=1}^{m} (r^*_i + \overline{r} - 2\overline{r}^2) \geq 0
\]

\[
\sum_{i=1}^{m} r_i^2 + m\overline{r}^2 - 2m\overline{r}^2 \geq 0
\]

\[
\sum_{i=1}^{m} r_i^2 - m\overline{r}^2 \geq 0
\]

\[
\overline{m\overline{r}^2} \geq m\overline{r}^2. \quad (22)
\]

With the combination of Eq. 21 and Eq. 22, we can obtain \((\Delta'_m)^2 \leq (\Delta_m)^2\). Because \(\Delta'_m \geq 0\) and \(\Delta_m \geq 0\), we can obtain \(\Delta'_m \leq \Delta_m\) within each block. Thus, for the whole image, we can obtain \(\Delta' \leq \Delta\).

REFERENCES

[1] F. Peng, G. Chen, and M. Long, “A robust coverless steganography based on generative adversarial networks and gradient descent approximation,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 9, pp. 5817–5829, Sep. 2022.

[2] S. Xie, H. Hu, and Y. Chen, “Facial expression recognition with two-branch disentangled generative adversarial network,” IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 6, pp. 2359–2371, Jun. 2021.

[3] C. Szegedy et al., “Intriguing properties of neural networks,” 2014, arXiv:1312.6199.

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

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” 2015, arXiv:1412.6572.

[6] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” in Artificial Intelligence Safety and Security, London, U.K.: Chapman & Hall, 2017.

[7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” 2017, arXiv:1706.06083.

[8] Y. Dong et al., “Boosting adversarial attacks with momentum,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 9185–9193.

[9] J. Rony, L. G. Hafemann, L. S. Oliveira, I. Ben Ayed, R. Sabourin, and E. Grainger, “Decoupling direction and norm for efficient gradient-based L2 adversarial attacks and defenses,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 4322–4330.
[10] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song, “Generating adversarial examples with adversarial networks,” 2018, arXiv:1801.02610.

[11] X. Liu and C.-J. Hsieh, “Rob-GAN: Generator, discriminator, and adversarial attacker,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 11234–11243.

[12] Z. Li et al., “A simple and strong baseline for universal targeted attacks on Siamese visual tracking,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 6, pp. 3880–3894, Jun. 2022.

[13] X. Qin, B. Li, S. Tan, W. Tang, and J. Huang, “Gradually enhanced adversarial perturbations on color pixel vectors for image steganography,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 8, pp. 5110–5123, Aug. 2022.

[14] C. Xie et al., “Improving transferability of adversarial examples with input diversity,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 2730–2739.

[15] S. Cheng, Y. Dong, T. Pang, H. Su, and J. Zhu, “Improving black-box adversarial attacks with a transfer-based prior,” in Proc. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 1–11.

[16] S. Tang et al., “RobustART: Benchmarking robustness on architecture design and training techniques,” 2021, arXiv:2109.05211.

[17] T. Wu, L. Tong, and Y. Vorobeychik, “Defending against physically realizable attacks on image classification,” 2019, arXiv:1909.09552.

[18] G. Jin, S. Shen, D. Zhang, F. Dai, and Y. Zhang, “APE-GAN: Adversarial perturbation elimination with GAN,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2019, pp. 3842–3846.

[19] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, “Defense against adversarial attacks using high-level representation guided denoiser,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 1778–1787.

[20] D. Zhou, T. Liu, B. Han, N. Wang, C. Peng, and X. Gao, “Towards defending against adversarial examples via attack-invariant features,” in Proc. 38th Int. Conf. Mach. Learn. (ICML), 2021, pp. 12835–12845.

[21] J. Liu et al., “Detection based defense against adversarial examples from the steganalysis point of view,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2019, pp. 4825–4834.

[22] J. Wang et al., “SmCNN: A new deep convolutional neural network model for adversarial example detection,” IEEE Trans. Multimedia, vol. 24, pp. 230–244, 2022.

[23] B. Liang, H. Li, M. Su, X. Li, W. Shi, and X. Wang, “Detecting adversarial image examples in deep neural networks with adaptive noise reduction,” IEEE Trans. Dependable Syst. Comput., vol. 15, no. 1, pp. 22–35, Jan. 2021.

[24] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel, “On the (statistical) detection of adversarial examples,” 2017, arXiv:1702.06203.

[25] Z. Gong, W. Wang, and W.-S. Ku, “Adversarial and clean data are not twins,” 2017, arXiv:1704.04960.

[26] D. Wu, S.-T. Xia, and Y. Wang, “Adversarial weight perturbation helps robust generalization,” 2020, arXiv:2004.05884.

[27] B. Wang, M. Zhao, W. Wang, F. Wei, Z. Qin, and K. Ren, “Are you confident that you have successfully generated adversarial examples?” IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 6, pp. 2089–2099, Jun. 2021.

[28] A. Tonge and C. Caragea, “Image privacy prediction using deep neural networks,” ACM Trans. Web, vol. 14, no. 2, pp. 1–32, Apr. 2020.

[29] A. Tonge, “Identifying private content for online image sharing,” in Proc. AAAI Conf. Artif. Intell., vol. 32, 2018, pp. 1–2.

[30] H. Zhong, A. C. Squicciarini, D. J. Miller, and C. Caragea, “A group-based personalized model for image privacy classification and labeling,” in Proc. IJCAI, vol. 17, 2017, pp. 3952–3958.

[31] M. M. Hassan, M. R. Hassan, S. Huda, and V. H. C. de Albuquerque, “A robust deep-learning-enabled trust-boundary protection for adversarial industrial IoT environment,” IEEE Internet Things J., vol. 8, no. 12, pp. 9611–9621, 2020.

[32] M. Zhao, B. Wang, W. Wang, Y. Kong, T. Zheng, and K. Ren, “Guided erasable adversarial attack (GEAA) towards shared data protection,” IEEE Trans. Inf. Forensics Security, vol. 17, pp. 2468–2482, 2022.

[33] B. Han, M. Su, K. Li, and X. Liu, “Ensemble adversarial training: Attacks and defenses,” 2017, arXiv:1705.07204.

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

[35] G. Weiguang Ding, L. Wang, and X. Jin, “AdvNetorch v0.1: An adversarial robustness toolbox based on PyTorch,” 2019, arXiv:1902.07623.