Ungeneralizable Examples

Jingwen Ye **Xinchao Wang**[†]
National University of Singapore
jingweny@nus.edu.sg, xinchao@nus.edu.sg

Abstract

*The training of contemporary deep learning models heavily relies on publicly available data, posing a risk of unauthorized access to online data and raising concerns about data privacy. Current approaches to creating unlearnable data involve incorporating small, specially designed noises, but these methods strictly limit data usability, overlooking its potential usage in authorized scenarios. In this paper, we extend the concept of unlearnable data to conditional data learnability and introduce **UnGeneralizable Examples** (UGEs). UGEs exhibit learnability for authorized users while maintaining unlearnability for potential hackers. The protector defines the authorized network and optimizes UGEs to match the gradients of the original data and its ungeneralizable version, ensuring learnability. To prevent unauthorized learning, UGEs are trained by maximizing a designated distance loss in a common feature space. Additionally, to further safeguard the authorized side from potential attacks, we introduce additional undistillation optimization. Experimental results on multiple datasets and various networks demonstrate that the proposed UGEs framework preserves data usability while reducing training performance on hacker networks, even under different types of attacks.*

1. Introduction

The widespread availability of 'free' internet data has played a pivotal role in advancing deep learning and computer vision models. However, a notable concern arises from the collection of datasets without explicit consent, with personal data often gathered unknowingly from the internet. This practice has raised public concerns about the potential unauthorized and, in some cases, potentially illegal exploitation of personal information. These issues have gained even greater significance with the introduction of the General Data Protection Regulation (GDPR) by the European Union, placing a renewed emphasis on data protection within the AI community.

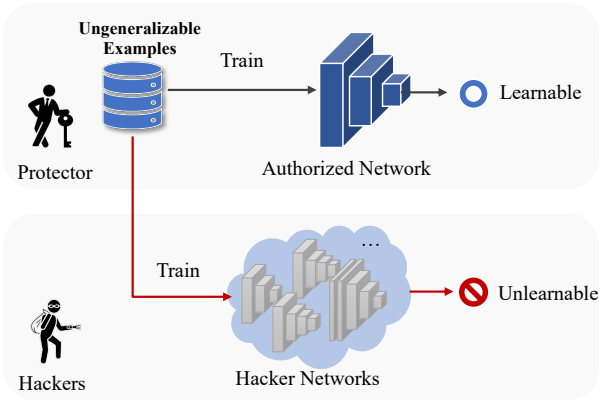


Figure 1. The threat model of ungeneralizable examples involves generating UnGeneralizable Examples. Once created, both the protector and the hacker gain access to the UGEs rather than the original data. While the UGEs can effectively train the protector's network, they result in a performance drop on hacker networks.

To address the risk of machine learning models capturing private data, recent developments have focused on the concept of unlearnable examples (ULE) [6, 9, 24]. Unlearnable examples represent data types that deep learning models struggle to effectively learn useful information from. A common method for generating unlearnable examples involves a min-min bilevel optimization framework, deceiving the model into learning a false connection between noise and labels. Consequently, models trained on such unlearnable examples exhibit significantly reduced performance, emphasizing the importance of robust data protection in machine learning. It's crucial to note that, oftentimes, the data itself is not inherently problematic; instead, *it is the manner in which they are utilized that demands careful consideration*. Therefore, we argue that such an across-the-board data protection rule could address some stringent privacy issues but might impede the shareable community under normal service conditions.

In our study, we broaden the conventional assumption in existing ULE methods, introducing a more adaptable and

[†] Corresponding author.

pragmatic data protection paradigm referred to as ungeneralizable examples. In contrast to the conventional ULE framework, we posit that the data can be learnable by networks pre-defined by the protector. This approach enables the protector to maintain authorized usage of the collected data, addressing the inflexible concern of unlearnable examples. Moreover, it offers an alternative for the protector when they need to share their data for specific legitimate purposes. The fundamental concept of UGEs is visually represented in Figure 1.

In UGE, the protector pre-defines the authorized network before generating the ungeneralizable version of the data. We approximate the training trajectories of the original data and the ungeneralizable data to ensure that the data's learnability remains unchanged. To prevent the data from being learned by hackers, we maximize the feature distance in the common feature space, where unlearnability can transfer to multiple hacker networks. Additionally, to further enhance the confidentiality of the ungeneralizable examples, we introduce the undistill loss, aiming to prevent hackers from recovering the original data from the protector network.

In summary, the contributions of this paper can be outlined as follows:

- ***Introduction of Ungeneralizable Examples Paradigm***: We propose a versatile data protection paradigm termed ungeneralizable examples. This paradigm enables the legitimate use of data by the protector while preventing unauthorized usage by potential hackers. It introduces a pragmatic scenario, challenging the unauthorized training of machine learning models.
- ***Innovative Solution for Learnability and Unlearnability Switchover***: We introduce a novel approach to address the switch between data learnability and unlearnability using three distinct losses. This is the first and only method, to our knowledge, that achieves this switchover effectively.
- ***Empirical Verification of Effectiveness and Robustness***: We empirically verify the effectiveness of our proposed approach with different network backbones on diverse datasets. Furthermore, we assess its robustness under various network architectures and multiple types of attacks.

2. Related Work

2.1. Data Privacy Protection

Ensuring data privacy protection is crucial for safeguarding individuals' sensitive information, preserving autonomy, and fostering trust in the digital landscape. This commitment is instrumental in the ethical and responsible development of technology. Here we group the data privacy protection into visual information protection and data protection from machine learning.

The essence of visual information protection lies in rendering data visually unrecognizable or inaccessible to third parties. A direct approach involves employing basic techniques like pixelization, blurring, or scrambling to obscure facial features in images. Alternatively, recent advancements explore the use of encryption [11, 32] directly applied to an image, followed by inpainting [20, 31, 33], making it challenging to recover the original content. As another illustration, the concept of dataset condensation [3, 17–19, 37] is introduced to distill the essence of data into a compact synset. This approach aims to safeguard the original data's integrity while retaining its ability to effectively train a neural network. Regarding data privacy, federated learning [14, 28, 40] emphasizes a distributed model training paradigm that prioritizes keeping sensitive information localized on individual devices, thereby mitigating privacy risks associated with centralized data storage or sharing.

Data protection from machine learning primarily focuses on the control and management of learnable features extracted from networks. For exmaple, machine unlearning [2, 16, 29] exemplifies the recalibration of machine learning models through the selective discarding of specific data points, patterns, or predictions. This process involves the removal of sensitive data information from the network, effectively eliminating the risk of unintended data exposure. An additional aspect of data protection involves preventing data from being learned by machine unlearning models. Huang et al. [9] have made a significant contribution to safeguarding image data from unauthorized machine learning exploitation. They introduced a method focused on generating error-minimizing noise with the primary goal of intentionally degrading images uploaded to the internet. This degradation aims to impede the training process of neural networks. As a result, images incorporating this introduced noise are classified as unlearnable examples [6, 24].

We posit our proposed ungeneralizable examples as an expanded version of unlearnable examples, offering enhanced flexibility in data management. In this approach, the data remains unlearnable by the defender while remaining learnable by the protector, thereby providing a more nuanced control over the learning dynamics.

2.2. Model Privacy Protection

In the realm of model privacy protection, our focus centers on Intellectual Property (IP) safeguarding. The escalating commercial significance of deep networks has garnered heightened attention from both academia and industry, emphasizing the imperative for robust IP protection.

As a conventional technique, network watermarking [12, 13, 27] involves embedding identification information into the target network, enabling copyright claims without compromising the network's predictive capabilities. Numerous recent studies [10, 15, 35] have investigated defensive strategies against model stealing, aiming to safeguard the intellectual property of the network. As an additional mea-

sure for intellectual property (IP) protection, knowledge undistillation [21, 34] is introduced to prevent knowledge theft by other networks. This entails maintaining the network's fundamental prediction performance while inducing a performance drop when attempting to distill knowledge.

Our proposed ungeneralizable examples have something common with knowledge undistillaion, which are designed to introduce modifications to the original images, leading to suboptimal performance on unauthorized networks while preserving their efficacy in training the protector's network.

2.3. Adversarial and Data Poisoning Attacks

Adversarial attacks [1, 4, 8, 22, 36, 39] are designed to deceive machine learning models by adding small, imperceptible perturbations to input data, causing the model to generate incorrect outputs or misclassify inputs. One of the traditional attack methods [7] is to use gradient information to update the adversarial example in a single step along the direction of maximum classification loss.

Data poisoning [26, 30, 41] is a type of adversarial attack that involves manipulating the training data used to train machine learning models. The goal of these attacks is to introduce malicious or misleading data into the training set, with the intention of influencing the performance of the trained model.

However, such methods don't affect the model's performance on clean data, which makes them unsuitable for data privacy pretection.

3. Proposed Method

Assumptions on Protector's Capability: We assume that the protector has unrestricted access to the specific dataset they intend to make ungeneralizable. However, it's crucial to clarify that the protector lacks the capacity to interfere with the training process and does not have access to the entire training dataset. In simpler terms, the protector's influence is confined to transforming their designated data portion into ungeneralizable examples. Furthermore, it's essential to underscore that once the ungeneralizable examples are generated, the protector is prohibited from making further modifications to their data. Importantly, these modifications are irreversible. In other words, once the alterations are applied, the original data is replaced by the modified versions.

3.1. Problem Formulation

Following the previous setting on unlearnable examples [9], we focus on image classification tasks in this paper.

Suppose $\mathcal{D} = \{(x, y)\}^{(n)}$ is a clean training dataset with K-class, where images can be denoted as $x \in \mathcal{X} \subset \mathbb{R}^d$, the corresponding groundtruth labels are denoted as $y \in \mathcal{Y} = \{1, 2, ..., K\}$. Two distinct networks are introduced: the authorized network, denoted as f_θ, and the hacker's network, denoted as f'_{θ_A}. The network f_θ is predetermined by the protector, where the network's architecture and initial parameters are set. Alongside the original data $\mathcal{D}$, the protector utilizes f_θ to generate the ungeneralizable version of the dataset, denoted as $\mathcal{D}_u = \{(x_u, y_u)\}^{(n)}$. This process is defined as follows:

$$x_u \leftarrow x + \delta(f_\theta), \quad y_u \leftarrow y; \quad \{(x, y)\} \in \mathcal{D}. \tag{1}$$

Here $\delta(f_\theta) \subset \mathbb{R}^d$ is the generated ungeneralizable noise that is related to the authorized network f_θ. The ungeneralizable noise is typically regulated to be imperceptible. We omit f_θ from the ungeneralizable noise in the rest of the paper. The ungeneralizable dataset $\mathcal{D}_u$ is assumed to be the shareable dataset collected by both the hackers and the protector, which will be utilized to train both the authorized network f_θ and the hacker network f'_{θ_A}.

The generation of ungeneralizable examples serves two main objectives: firstly, they are designed to remain learnable for the authorized network f_θ; secondly, they are intended to become unlearnable for the malicious networks f'_{θ_A} employed by the hackers. Thus, the objective could be formulated as:

$$\min_\theta \frac{1}{n} \sum_{(x,y)\in\mathcal{D}}^{n} \min_{\|\delta\|\le\rho} \Big[\mathcal{L}\big(f'_{\theta_A}(x+\delta), y\big) + \big\| \mathcal{L}\big(f_\theta(x+\delta), y\big) - \mathcal{L}\big(f_\theta(x), y\big) \big\| \Big], \tag{2}$$

where $\mathcal{L}(\cdot)$ denotes the loss function for network training, and ρ is the radius represents the radius of the applied ungeneralizable noise. The formal section of the objective function introduces error-minimizing noise to render the data unlearnable by diminishing the associated training loss, making it challenging for hackers using f'_{θ_A} to acquire the knowledge. The latter part of the objective function aims to reconstruct this knowledge on the authorized network f_θ by minimizing the training loss between the clean input and the ungeneralizable version input.

Design Goals of UGEs: We aim to generate the ungeneralizable examples with the following characteristics:

- **Visual Integrity**: The ungeneralizable version of the images should remain visually recognizable to human observers, meaning that the ungeneralizable noise should be confined to a small norm.
- **Effectiveness**. UGEs facilitate authorized training on the authorized networks while preventing unauthorized training by hackers, demonstrating conditional learnability.
- **Robustness**. The unlearnability of UGEs should be stable and resistant to attacks by hackers; their safety should be verified under various types of attacks. Additionally, it should be transferable to different network architectures.
- **User-friendliness**. It should be convenient for authorized usage. That is, it shouldn't affect the training process on

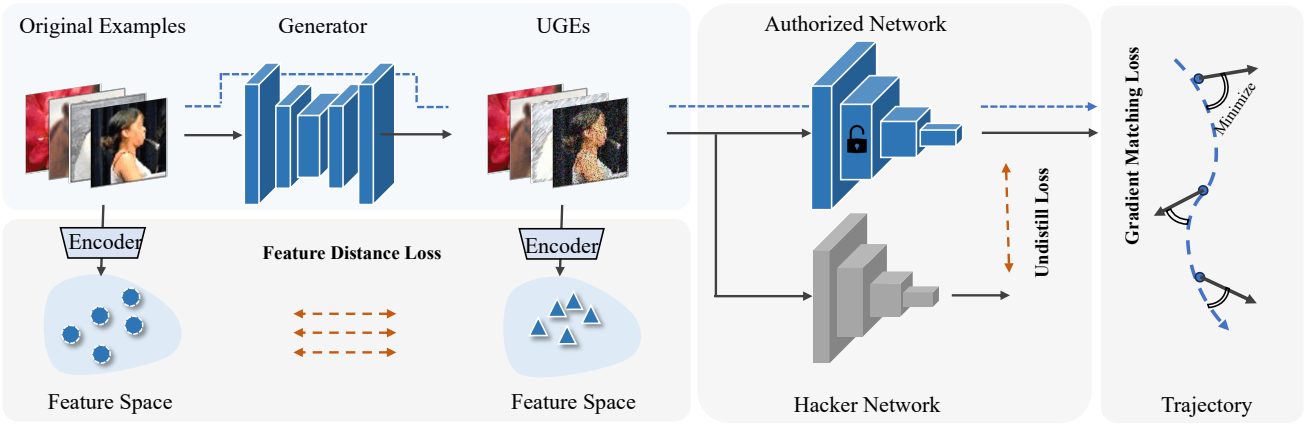


Figure 2. The comprehensive workflow of UGEs involves the protector training a generator to produce the ungeneralizable version of the original examples. Three distinct loss functions are employed in training the generator: gradient matching loss, feature distance loss, and undistill loss. Upon completion of the training process, the UGEs are published, and both the protector and hackers no longer have access to the original examples.

the authorized network. No new losses or components are introduced for training on UGEs, and it shouldn't increase the computational load of the training process.

3.2. Ungeneralizable Examples

As is shown in Fig. 2, the framework of obtaining the ungeneralizable version of the original data is depicted, where we train a generator to synthesize the UGEs:

$$x_u \leftarrow Clamp\big(\mathcal{G}(x), x-\rho, x+\rho\big) \quad x \in \mathcal{D}, \tag{3}$$

where the $Clamp()$ operation is to constrain the ungeneralizable noise's norm within ρ. A total of three loss functions are utilized to train the generator $\mathcal{G}$:

$$\mathcal{L}_{all} = \mathcal{L}_{gm} + \lambda_{fd} \cdot \mathcal{L}_{fd} + \lambda_{ud} \cdot \mathcal{L}_{ud}, \tag{4}$$

where $\mathcal{L}_{gm}$ is the gradient matching loss to ensure UGEs learnanle on the authorized network, $\mathcal{L}_{fd}$ is the feature distance loss to make UGEs unlearnable on the hacker networks, and $\mathcal{L}_{ud}$ is the undistill loss to make the original examples inreversible on the authorized network. λ_{fd} and λ_{ud} are the weights to balance each loss item.

Learnable on the authorized network. As is stated in Eq. 2, the latter loss item which tries to minimize the training loss between the inputs of x and x_u. Note the architecture and the initial parameters θ_0 of the authorized network are confirmed, the training process of f_θ on the original dataset $\mathcal{D}$ could be determined:

$$f: \theta_{t+1} \leftarrow \theta_t - \eta \frac{1}{n} \sum_{(x,y)\in\mathcal{D}} \nabla\mathcal{L}\big(f_{\theta_t}(x), y\big), \tag{5}$$

where η is the learning rate, and $t = \{0, 1, ..., T-1\}$ is the training epoch number, T is the total training epoch number. $\nabla\mathcal{L}$ is the gradients while in each training epoch.

To sustain the learning trajectory on the original data x, we introduce the gradient matching loss during the training process between x and x_u. Specifically, we randomly sample several intermediate training epochs from $\{0, 1, ..., T-1\}$, denoted as τ. The gradient matching loss is then calculated in these sampled epochs, thus the gradient matching loss can be finally expressed as:

$$\mathcal{L}_{gm} = \frac{1}{|\tau| \times n} \sum_{t\in\tau} \sum_{(x,y)\in\mathcal{D}} \mathcal{D}ist\big[\nabla\mathcal{L}\big(f_{\theta_t}(x), y\big), \nabla\mathcal{L}\big(f_{\theta_t}(x_u), y\big)\big], \tag{6}$$

where $\mathcal{D}ist(\cdot)$ represents the cosine distance, and we employ it to distill the gradient information from x to x_u. Minimizing the gradient matching loss $\mathcal{L}_{gm}$ effectively aligns the training trajectory of the original data with that of the ungeneralizable data. This optimization ensures the preservation of data learnability.

Unlearnable on the hacker network. As outlined in Eq. 2, the first loss renders the data unlearnable on hacker networks. Traditional unlearnable methods address this optimization challenge by introducing error-minimizing noise, typically through bi-level optimization, which is considered less efficient.

In this approach, we leverage a shared feature space to apply perturbations, thereby achieving the unlearnable characteristic in the data. As depicted in Fig. 2, a common image encoder $\mathcal{E}_i$ extracts features from both the original data $\mathcal{D}$ and its ungeneralizable version $\mathcal{D}_u$. To ensure that the feature perturbation designed in this feature space remains effective across diverse networks, a robust and powerful encoder selection becomes crucial.

Considering this perspective, we opt to utilize the pretrained image encoder of the CLIP model [23]. As a lead-

ing Vision-and-Language (VL) model, CLIP learns state-of-the-art image representations from scratch on a dataset containing 400 million image-text pairs collected from the internet. This training allows it to excel in various tasks, including zero-shot classification. Alongside the powerful image encoder $\mathcal{E}_i$ offered by CLIP, an additional textual encoder $\mathcal{E}_t$ is available to provide supplementary guidance.

To be specific, the feature distance loss $\mathcal{L}_{fd}$ can be computed as follows:

$$\mathcal{L}_{fd} = \frac{1}{n} \sum_{x_u \in \mathcal{D}_u} \ell_{feat}(x_u, \mathcal{E}_i, \mathcal{D}) + \ell_{tri}(x_u, \mathcal{E}_i, \mathcal{E}_t, \mathcal{D}), \quad (7)$$

which contains two main loss items ℓ_{feat} and ℓ_{tri}. The former loss ℓ_{feat} pushes the features of the ungeneralizable x_u away from the original x:

$$\begin{aligned} \ell_{feat}(x_u, \mathcal{E}_i, \mathcal{D}) &= -\|\mathcal{F}_i - \mathcal{F}_i'\|^2, \\ where \quad \mathcal{F}_i &= \frac{\mathcal{E}_i(x)}{\|\mathcal{E}_i(x)\|}, \mathcal{F}_i' = \frac{\mathcal{E}_i(x_u)}{\|\mathcal{E}_i(x_u)\|}, \end{aligned} \quad (8)$$

where $\mathcal{F}_i/\mathcal{F}_i'$ is the normalized features and ℓ_{feat} is calculated based on the MSE loss.

In addition to maximizing the similarity between the features of the original input and the ungeneralizable input, we introduce an additional triplet loss. This triplet loss ensures that the features of $\mathcal{F}_i'$ in the ungeneralizable input can be effectively transferred to various hacker networks.

$$\begin{aligned} \ell_{tri}(x_u, \mathcal{E}_i, \mathcal{E}_t, \mathcal{D}) &= \|\mathcal{F}_i' - \mathcal{F}_t'\|^2 + \max\left(0, \alpha - \|\mathcal{F}_i' - \mathcal{F}_t\|^2\right), \\ where \quad \mathcal{F}_t &= \frac{\mathcal{E}_t(y)}{\|\mathcal{E}_t(y)\|}, \mathcal{F}_t' = \arg\min_{\mathcal{F}_t^c} Sim(\mathcal{F}_i, \mathcal{F}_t^c). \end{aligned} \quad (9)$$

Here, α is the margin of the triplet loss and $\mathcal{F}_t$ is the textual features with the groundtruth label y as input and $Sim(\cdot)$ is the similarity function measuring the distance between the textual features and image features. $\mathcal{F}_t^c$ is the textual input with label c ($c \in \{1, 2, ..., K\}$ and $c \neq y$). Consequently, $\mathcal{F}_t'$ refers to the textual features with the least similarity to the original image encoder features $\mathcal{F}_i$. The term ℓ_{tri} encourages the features of ungeneralizable examples to move away from their associated textual features towards those of the least similar textual features.

Untransferable on the authorized network. After the ungeneralizable version of the data $\mathcal{D}_u$ is published, both the protector and hackers gain access to UGEs. UGEs show to be unlearable on the hacker networks when standard training with $\min_{\theta_A} \mathcal{L}(f'_{\theta_A}(x_u), y_u)$. On the contrary, UGEs can be employed for normal training on the network f_θ authorized by the protector. By minimizing the gradient matching loss $\mathcal{L}_{gm}$ as defined in Eq. 6, f_θ attains similar performance to when trained with the original data $\mathcal{D}$. The protector doesn't constrain the authorized network to be confidential, which means the hackers also have access to the authorized network f_θ (including architecture and parameters). In this way, the hackers have another alternative to train their networks, with both the ungeneralizable examples $\mathcal{D}_u$ and f_θ available.

To be concrete, the protector has authorized the data learning on the authorized network f_θ, and there exists a potential risk for hackers to exploit distillation-based learning process, expressed as:

$$f' : \min_{\theta_A} \frac{1}{n} \sum_{x_u \in \mathcal{D}_u} \mathcal{L}_{kd}\big(f_\theta(x_u), f'_{\theta_A}(x_u)\big), \quad (10)$$

where $\mathcal{L}_{kd}$ represents the KL-divergence loss for distilling knowledge directly from the authorized network f_θ. This poses a significant security risk as it could expose the confidentiality of UGEs through the authorized network f_θ.

Considering this concern, we introduce an undistill loss $\mathcal{L}_{ud}$ to safeguard the knowledge of the authorized network. Building upon prior work on knowledge undistillation [21] designed for network IP protection, our proposed undistill loss is expressed as:

$$\min_{\mathcal{D}_u} \mathcal{L}_{ud} = \frac{1}{n} \sum_{(x_u, y_u) \in \mathcal{D}_u} \big[\mathcal{L}(f_\theta(x_u), y_u) - \omega \mathcal{L}_{kd}\big(f_\theta(x_u), f'_{\theta_A}(x_u)\big)\big], \quad (11)$$

where $\mathcal{L}(\cdot)$ represents the standard training loss of f_θ and ω is the balancing weight. It's worth noting that in previous knowledge undistillation approaches, the undistill loss is employed to update the parameters of the network to be protected. In our case, we maintain f_θ and f'_{θ_A} fixed and optimize x_u to ensure that its learnable knowledge within f_θ cannot be transferred to f'_{θ_A}. This additional optimization step further enhances the security of our proposed ungeneralizable examples.

It's essential to highlight that in this context, *we do not restrict f'_{θ_A} to any specific networks*; it can be any arbitrarily initialized network. Our proposed UGEs not only demonstrate effectiveness on the randomly chosen f'_{θ_A} but also exhibit generalizability, extending their unlearnability characteristics to other networks.

3.3. Algorithm

The whole algorithm is depicted in Alg 1.

3.4. UGEs in Various Usages

The proposed UGEs seamlessly combine both data learnability and unlearnability within a single framework, showcasing a flexible approach to data management suitable for various applications.

***Scenario I**: Utilizing UGEs in Decentralized Model Training.* In scenarios resembling federated learning, where privacy constraints exist in individual local servers, UGEs offer a viable solution. The global server establishes the initial global model, communicates the model information

Algorithm 1 The framework of the proposed UGEs.

Require: $\mathcal{D}$: original data to be protected; f_θ: authorized network; $\{\theta_0, \theta_1, ..., \theta_\tau\}$: sampled trajectory of the authorized network. f'_{θ_A}: randomly initialized hacker network. ρ: ungeneralizable noise ℓ_∞ bound;
Ensure: $\mathcal{D}_u$: ungeneralizable examples.

```
Initialize the generator model G;
Initialize the text input as 'A photo of a < CLASS >';
Input text input to E_t to get textual features for all classes;
while not convergence do
    Input x to G and get x_u = G(x) bounded with ρ;
    Randomly choose θ_t from the input trajectory;
    Input both x and x_u to f_{θ_t} to calculate L_gm with Eq. 6;
    Input both x and x_u to encoder E_i to get F_i and F'_i ;
    Calculate L_fd with Eq. 7;
    Input x_u to both networks f_θ and f'_{θ_A};
    Calculate L_ud with Eq. 11;
    Update G by minimizing L_all;
end while
Get the ungeneralizable examples D_u with G;
Publish the final D_u.
```

to each local server, and enables the joint training of the global model. UGEs effectively address privacy concerns by selectively publishing data for specific use cases.

***Scenario II**: Enhancing Code Publication Safety with UGEs.* In open-source platforms such as GitHub, researchers are encouraged to share their code for collaborative AI development. However, instances arise where researchers collectively publish their gathered data. To mitigate the risk of malicious utilization of this data, researchers can opt to publish the ungeneralizable version of their training data, ensuring a more secure sharing environment.

***Scenario III**: Ensuring Secure Data Transmission with UGEs.* In instances where secure data transmission is required to train a downstream network, a secure process can be established. The receiver initiates the transmission by sending its information to the protector. Subsequently, only the UGEs are transmitted to the receiver, mitigating the risk of interception by hackers during the transmission process. This approach ensures a secure and protected data exchange, with UGEs playing a pivotal role in safeguarding sensitive information.

4. Experiments

In this section, we conduct comprehensive experiments to validate the effectiveness of the robust ungeneralizable examples. Additional details regarding the experiment setup can be found in the supplementary materials.

4.1. Experiment Setup

Datasets. Continuing the experimental setup from previous unlearnable methods, we present our results on CIFAR-10, CIFAR-100, and TinyImageNet datasets. The input size for CIFAR-10 and CIFAR-100 datasets is 32×32, while for the TinyImageNet dataset, we utilize an input size of 256×256.

Model Training. We employ the PyTorch framework for implementation and investigate several network backbones, including plain CNN, LeNet, ResNet, MobileNetV2, and ShuffleNetV2. The generator utilizes a ResNet backbone.

In our supposition, both the authorized network and hacker networks are optimized using standard Stochastic Gradient Descent (SGD). The authorized network f_θ is determined with given network architecture and initialization parameters. We assume networks with either different architectures or different initialization parameters are regarded as hacker networks. When training the UGEs, we randomly select a distinct network as the hacker network, exclusively including the hacker network for training.

Evaluation Metrics. We evaluate the data protection capability of the ungeneralizable noise using test accuracy. A low test accuracy on hacker networks indicates that the model has learned minimal knowledge from the training data, reflecting strong protection. Conversely, a high test accuracy on the authorized network indicates that the model has successfully learned knowledge from the training data, demonstrating data learnability for authorized usage.

4.2. Experimental Results

Ablation Study. The results of the ablation study on CIFAR-10, CIFAR-100 and TinyImageNet datasets are presented in Table 1. We compare the test accuracy on both the authorized network (Acc. (Authorized)) and the hacker networks (Acc. (Hacker)). Various network backbones are chosen to create the hacker networks, trained under two schemes: normal training with data labels (Normal) and distillation-based training using Eq. 10 (Distill). The distillation-based training can be thought as a kind of attack. For comparison, we show and compare the results with: 'Original': the unmodified original data $\mathcal{D}$; 'Unlearn': training only with unlearn loss $\mathcal{L}_{fd}$; 'UnDistill': training only with the undistillaion loss $\mathcal{L}_{ud}$; 'UGEs w/o UD': training without the undistillaion loss $\mathcal{L}_{ud}$. Form the table, conclusions could be drawn that:

- The efficacy of UGEs is evaluated based on their learnability on the authorized network (higher Acc. (Authorized)) and unlearnability on hacker networks (lower Acc. (Hacker)). Our method significantly reduces the test accuracy of hacker networks (by more than 40%) while maintaining an acceptable drop in authorized network accuracy (less than 5 %).
- UGE effectiveness is demonstrated across CIFAR-10, CIFAR-100, and TinyImageNet datasets, utilizing diverse network architectures like ResNet-18, CNN, MobileNet, and ShuffleNet. This showcases UGEs' versatility and efficacy across different scenarios;

Table 1. Experimental Results on CIFAR-10, CIFAR-100 and TinyImageNet datasets , where ResNet-18 is used as the backbone of the authorized network. Acc changes are shown in red comparing with the network normal trains on the original dataset.

Dataset	Method	Scheme	Acc. (Authorized)	Acc. (Hacker)			
				CNN	ResNetC-20	ResNetC-32	ResNet-18
CIFAR-10	Original	Normal	95.05	86.57	92.28	93.04	95.05
CIFAR-10	Original	Distill	-	88.06 (+1.49)	92.62 (+0.34)	93.24 (+0.20)	95.41 (+0.36)
CIFAR-10	Unlearn	Normal	22.59 (-72.46)	18.36 (-68.21)	20.37 (-71.91)	22.39 (-70.65)	22.44 (-72.61)
CIFAR-10	Unlearn	Distill	-	17.39 (-69.18)	21.40 (-70.88)	22.08 (-70.96)	21.48 (-73.57
CIFAR-10	UnDistill	Normal	94.52 (-0.47)	85.87 (-0.70)	85.91 (-6.37)	86.98 (-6.06)	88.07 (-6.98)
CIFAR-10	UnDistill	Distill	-	73.38 (-13.19)	78.65 (-13.63)	80.76 (-12.28)	84.07 (-10.98)
CIFAR-10	UGEs w/o UD	Normal	94.34 (-0.71)	46.97 (-39.6)	56.97 (-35.31)	75.07 (17.97)	45.95 (-49.10)
CIFAR-10	UGEs w/o UD	Distill	-	75.23 (-11.34)	69.08 (23.20)	77.45 (-15.59)	87.10 (-7.95)
CIFAR-10	UGEs	Normal	93.89 (-1.16)	26.46 (-60.11)	30.63(-61.65)	36.08 (-56.96)	26.12 (-68.93)
CIFAR-10	UGEs	Distill	-	32.08 (-54.49)	37.22(-55.06)	47.59 (-45.45)	35.23 (-59.82)

Dataset	Method	Scheme	Acc. (Authorized)	Acc. (Hacker)		
				MobileNetV2	ShuffleNetV2	ResNet-18
CIFAR-100	Original	Normal	78.24	68.92	71.26	78.24
CIFAR-100	Original	Distill	-	72.67 (+3.75)	74.39 (+3.13)	79.24 (+1.00)
CIFAR-100	UGEs w/o UD	Normal	75.26 (-2.98)	22.05 (-46.87)	21.59 (-49.67)	16.46 (-61.78)
CIFAR-100	UGEs w/o UD	Distill	-	63.52 (-5.40)	58.23 (-13.03)	40.45 (-37.79)
CIFAR-100	UGEs	Normal	74.68 (-3.56)	32.11(-36.81)	28.33(-42.93)	16.55 (-61.69)
CIFAR-100	UGEs	Distill	-	26.94 (-41.98)	25.34(-45.92)	15.50 (-62.74)
TinyImageNet	Original	Normal	63.08	56.00	59.90	63.08
TinyImageNet	Original	Distill	-	60.02 (+3.06)	63.19 (+7.19)	66.28 (+6.38)
TinyImageNet	UGEs w/o UD	Normal	60.88 (-2.20)	14.62 (-41.38)	13.97 (-45.93)	15.34 (-47.74)
TinyImageNet	UGEs w/o UD	Distill	-	36.39 (-6.18)	49.82 (-16.97,)	42.93 (-20.15)
TinyImageNet	UGEs	Normal	59.54 (-3.54)	15.79 (-40.21)	22.75 (-37.15)	17.55 (-45.53)
TinyImageNet	UGEs	Distill	-	19.69 (-36.94)	21.06 (-35.48)	24.42 (-38.66)

- Our proposed UGEs demonstrate robustness against attacks where hackers use the authorized network to acquire the learnability of UGEs (Scheme as 'Distill'). The results show that the proposed undistillation loss $\mathcal{L}_{ud}$ (comparing 'UGEs' and 'UGEs w/o UD') effectively prevents such attacks, fulfilling the robustness goals in the design.

UGEs with Multiple Authorized Networks. In the standard experimental setup, we initially configure one network as the authorized network. Here, we extend our framework to accommodate multiple authorized networks, introducing additional loss items for each newly added network in $\mathcal{L}_{all}$. Refer to the supplementary material for specific details on modifying the losses and extra experiments.

In this extension, we establish two authorized networks with ResNet-18 with distinct initialization parameters. The experimental results on the CIFAR-10 dataset are presented in Table 2, where 'Distill-1' indicates optimizing the hacker networks with the distillation calculated on authorized net-

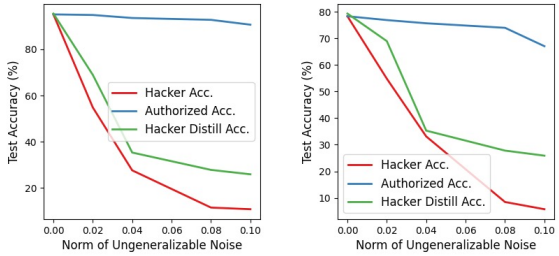
Table 2. Results on UGEs with multiple authorized networks on CIFAR-10 dataset, which are tested under three training schemes.

Method	Scheme	Authorized		Hacker	
		Net-1	Net-2	CNN	ResNetC-20
Original	Normal	95.01	94.95	86.57	93.04
Original	Distill-1	-	-	88.06	92.62
Original	Distill-2	-	-	88.09	92.55
UGEs	Normal	93.27	93.83	43.28	49.34
UGEs	Distill-1	-	-	53.60	56.32
UGEs	Distill-2	-	-	50.32	54.29

1. As observed in the table, introducing another authorized network maintains the effectiveness of our proposed framework for learning UGEs. However, with more authorized networks, the performance of UGEs experiences a slight

Table 3. Comparing the data unlearnability with the existing ULE methods on CIFAR-10 and CIFAR-100 datasets.

Method	Acc. (CIFAR-10)		Acc. (CIFAR-100)	
	Clean	ULEs	Clean	ULEs
EM [9]	94.66	13.20	76.27	1.60
TAP [5]	94.66	22.51	76.27	13.75
NTGA [38]	94.66	16.27	76.27	3.22
REM [6]	94.66	27.09	76.27	10.14
CUDA [25]	94.66	18.48	76.27	12.69
Ours ($\mathcal{L}_{fd}$)	94.66	22.59	76.27	9.35



(a) CIFAR-10 Dataset (b) CIFAR-100 Dataset

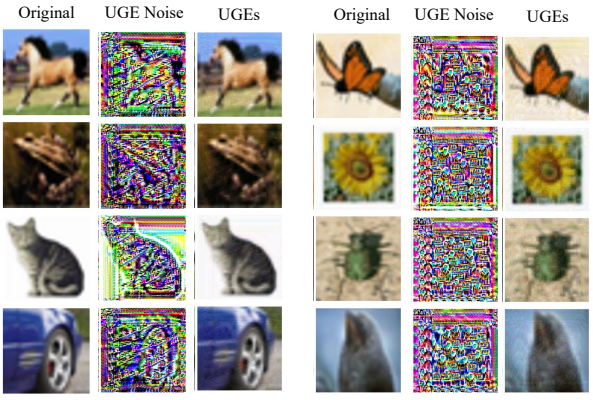
Figure 3. The performance concerning the value of ρ on CIFAR-10 and CIFAR-100 datasets.

decline. Addressing this challenge and providing a more flexible framework to include multiple authorized networks will be a focus of our future work.

How does the norm of ungeneralizable noise affect the UGEs performance. Recall that we set the norm of the ungeneralizable noise ρ as 0.04. We investigate the performance concerning the value of ρ, as illustrated in Fig. 3. From the figure, it can be observed that a larger norm of ungeneralizable noise leads to a decrease in test accuracy on the authorized network. Therefore, a properly chosen small norm of noise is essential, ensuring both the visual integrity of the protected data and maintaining acceptable authorized network performance.

Comparing with Existing ULEs. We compared the proposed method with existing ULE methods on CIFAR-10 and CIFAR-100 datasets, as shown in Table 3. The listed methods are included in the table, and the experimental setup follows previous work [25]. Lower test accuracy indicates better unlearnability. The results demonstrate that our proposed method contributes to current unlearnable example methods, achieving competitive results with existing ULE methods. The UGE framework can also be seamlessly integrated into the ULEs framework by training the generator $\mathcal{G}$ with the loss term $\mathcal{L}_{df}$.

More Analysis. In Fig. 4, we showcase the visualization results of our proposed UGE. The UGEs demonstrate visual similarity to the original images, confirming their visual integrity and aligning with the framework's design goal.



(a) CIFAR-10 Dataset (b) CIFAR-100 Dataset

Figure 4. The visualization results include the original clean images, the ungeneralizable noise (scaled by 255 for better visualization), and the resultant ungeneralizable images.

4.3. Limitations

While our method shows promise across various scenarios, it has limitations, particularly when faced with an increasing number of authorized networks (Table. 2). Addressing this, we plan to incorporate ensemble methods or knowledge amalgamation to enhance UGEs' performance in such scenarios. This underscores our commitment to ongoing improvement and adaptability. It's important to note that our UGE framework is designed for classification tasks. Looking ahead, we aim to extend its applicability to multiple tasks, enabling the seamless transition of data learnability among different tasks, thus enhancing its versatility.

5. Conlusion

In conclusion, our paper presents the ungeneralizable examples framework, a versatile paradigm for data protection. UGE allows legitimate data usage by the protector while preventing unauthorized access by potential hackers. The proposed approach, incorporating three distinct losses, successfully achieves a seamless transition between data learnability and unlearnability. Empirical verification validates the effectiveness and robustness of our method, demonstrating its potential in enhancing data security in machine learning applications.

Acknowledgements

This project is supported by the Advanced Research and Technology Innovation Centre (ARTIC), the National University of Singapore under Grant (project number: A0005947-21-00, project reference: ECT-RP2).

References

[1] Naveed Akhtar and Ajmal S. Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430, 2018. 3

[2] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021. 2

[3] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[4] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9185–9193, 2017. 3

[5] Liam Fowl, Micah Goldblum, Ping-yeh Chiang, Jonas Geiping, Wojciech Czaja, and Tom Goldstein. Adversarial examples make strong poisons. *Advances in Neural Information Processing Systems*, 34:30339–30351, 2021. 8

[6] Shaopeng Fu, Fengxiang He, Yang Liu, Li Shen, and Dacheng Tao. Robust unlearnable examples: Protecting data privacy against adversarial learning. In *International Conference on Learning Representations*, 2021. 1, 2, 8

[7] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning and Representations*, 2014. 3

[8] Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. Simple black-box adversarial attacks. In *International Conference on Machine Learning*, pages 2484–2493. PMLR, 2019. 3

[9] Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. In *International Conference on Learning Representations*, 2020. 1, 2, 3, 8

[10] Sanjay Kariyappa and Moinuddin K Qureshi. Defending against model stealing attacks with adaptive misinformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2020. 2

[11] Manjit Kaur and Vijay Kumar. A comprehensive review on image encryption techniques. *Archives of Computational Methods in Engineering*, 27:15–43, 2020. 2

[12] Erwan Le Merrer, Patrick Perez, and Gilles Trédan. Adversarial frontier stitching for remote neural network watermarking. *Neural Computing and Applications*, 32:9233–9244, 2020. 2

[13] Guobiao Li, Sheng Li, Zhenxing Qian, and Xinpeng Zhang. Encryption resistant deep neural network watermarking. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3064–3068. IEEE, 2022. 2

[14] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722, 2021. 2

[15] Yiming Li, Linghui Zhu, Xiaojun Jia, Yong Jiang, Shu-Tao Xia, and Xiaochun Cao. Defending against model stealing via verifying embedded external features. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1464–1472, 2022. 2

[16] Junxu Liu, Mingsheng Xue, Jian Lou, Xiaoyu Zhang, Li Xiong, and Zhan Qin. Muter: Machine unlearning on adversarially trained models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4892–4902, 2023. 2

[17] Songhua Liu and Xinchao Wang. Mgdd: A meta generator for fast dataset distillation. In *Advances in Neural Information Processing Systems*, 2023. 2

[18] Songhua Liu, Kai Wang, Xingyi Yang, Jingwen Ye, and Xinchao Wang. Dataset distillation via factorization. In *Advances in Neural Information Processing Systems*, 2022.

[19] Songhua Liu, Jingwen Ye, Runpeng Yu, and Xinchao Wang. Slimmable dataset condensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3759–3768, 2023. 2

[20] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022. 2

[21] Haoyu Ma, Tianlong Chen, Ting-Kuei Hu, Chenyu You, Xiaohui Xie, and Zhangyang Wang. Undistillable: Making a nasty teacher that cannot teach students. In *International Conference on Learning Representations*, 2020. 3, 5

[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*, 2017. 3

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 4

[24] Jie Ren, Han Xu, Yuxuan Wan, Xingjun Ma, Lichao Sun, and Jiliang Tang. Transferable unlearnable examples. In *The Eleventh International Conference on Learning Representations*, 2022. 1, 2

[25] Vinu Sankar Sadasivan, Mahdi Soltanolkotabi, and Soheil Feizi. Cuda: Convolution-based unlearnable datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3862–3871, 2023. 8

[26] Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P Dickerson, and Tom Goldstein. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. In *International Conference on Machine Learning*, pages 9389–9398. PMLR, 2021. 3

[27] Sebastian Szyller, Buse Gul Atli, Samuel Marchal, and N Asokan. Dawn: Dynamic adversarial watermarking of neural networks. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4417–4425, 2021. 2

[28] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 2

[29] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. 2

[30] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. Data poisoning attacks against federated learning systems. In *Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I 25*, pages 480–501. Springer, 2020. 3

[31] Su Wang, Chitwan Saharia, Ceslee Montgomery, Jordi Pont-Tuset, Shai Noy, Stefano Pellegrini, Yasumasa Onoe, Sarah Laszlo, David J Fleet, Radu Soricut, et al. Imagen editor and editbench: Advancing and evaluating text-guided image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18359–18369, 2023. 2

[32] Xingyuan Wang, Le Feng, and Hongyu Zhao. Fast image encryption algorithm based on parallel computing system. *Information Sciences*, 486:340–358, 2019. 2

[33] Hanyu Xiang, Qin Zou, Muhammad Ali Nawaz, Xianfeng Huang, Fan Zhang, and Hongkai Yu. Deep learning for image inpainting: A survey. *Pattern Recognition*, 134:109046, 2023. 2

[34] Jingwen Ye, Yining Mao, Jie Song, Xinchao Wang, Cheng Jin, and Mingli Song. Safe distillation box. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3117–3124, 2022. 3

[35] Jingwen Ye, Songhua Liu, and Xinchao Wang. Partial network cloning. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20137–20146, 2023. 2

[36] Jingwen Ye, Ruonan Yu, Songhua Liu, and Xinchao Wang. Mutual-modality adversarial attack with semantic perturbation. *AAAI Conference on Artificial Intelligence*, 2024. 3

[37] Ruonan Yu, Songhua Liu, and Xinchao Wang. Dataset distillation: A comprehensive review. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 2

[38] Chia-Hung Yuan and Shan-Hung Wu. Neural tangent generalization attacks. In *International Conference on Machine Learning*, pages 12230–12240. PMLR, 2021. 8

[39] Chaoning Zhang, Philipp Benz, Chenguo Lin, Adil Karjauv, Jing Wu, and In So Kweon. A survey on universal adversarial attack. *International Joint Conference on Artificial Intelligence*, 2021. 3

[40] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021. 2

[41] Xuezhou Zhang, Xiaojin Zhu, and Laurent Lessard. Online data poisoning attacks. In *Learning for Dynamics and Control*, pages 201–210. PMLR, 2020. 3