electronics

MDPI

Article

Exploring Public Data Vulnerabilities in Semi-Supervised Learning Models through Gray-box Adversarial Attack

Junhyung Jo [1], Joongsu Kim [2] and Young-Joo Suh [1,*]

[1] Graduate School of Artificial Intelligence, Pohang University of Science and Technology, Pohang 37673, Republic of Korea; jjo22@postech.ac.kr
[2] Department of Computer Science and Engineering, Pohang University of Science and Technology, Pohang 37673, Republic of Korea; joongsukim@postech.ac.kr
* Correspondence: yjsuh@postech.ac.kr

Abstract: Semi-supervised learning (SSL) models, integrating labeled and unlabeled data, have gained prominence in vision-based tasks, yet their susceptibility to adversarial attacks remains underexplored. This paper unveils the vulnerability of SSL models to gray-box adversarial attacks—a scenario where the attacker has partial knowledge of the model. We introduce an efficient attack method, Gray-box Adversarial Attack on Semi-supervised learning (GAAS), which exploits the dependency of SSL models on publicly available labeled data. Our analysis demonstrates that even with limited knowledge, GAAS can significantly undermine the integrity of SSL models across various tasks, including image classification, object detection, and semantic segmentation, with minimal access to labeled data. Through extensive experiments, we exhibit the effectiveness of GAAS, comparing it to white-box attack scenarios and underscoring the critical need for robust defense mechanisms. Our findings highlight the potential risks of relying on public datasets for SSL model training and advocate for the integration of adversarial training and other defense strategies to safeguard against such vulnerabilities.

Keywords: adversarial attack; gray-box attack; semi-supervised learning; deep neural networks

Citation: Jo, J.; Kim, J.; Suh, Y.-J. Exploring Public Data Vulnerabilities in Semi-Supervised Learning Models through Gray-box Adversarial Attack. *Electronics* **2024**, *13*, 940. https://doi.org/10.3390/electronics13050940

Academic Editor: Dimitra I. Kaklamani

Received: 16 January 2024
Revised: 18 February 2024
Accepted: 29 February 2024
Published: 29 February 2024

1. Introduction

Semi-supervised learning (SSL) is a machine learning paradigm that combines labeled and unlabeled data for training models. Unlike traditional supervised learning, SSL leverages unlabeled data, which is abundant but costly to annotate. SSL methods have achieved impressive results in tasks such as image classification [1–8], object detection [9–12], and semantic segmentation [13–15]. Most of the existing state-of-the-art SSL methods employ self-training techniques, specifically, pseudo-labeling methods [3,7]. The process of pseudo-labeling methods involves training an initial model on the available labeled data. Then, this trained model is used to make predictions on the unlabeled data. The predictions are used to assign pseudo-labels to the unlabeled samples, effectively treating them as if they were labeled. The augmented dataset, consisting of the labeled data and the pseudo-labeled data, is then used to train a new model or refine the existing one. In this process, the labeled data initially used for training are shared between models and utilized as a part of the augmented dataset for the subsequent model's training. To make this process more accessible, publicly available shared labeled data are often employed.

While deep learning models continue to advance in performance, their vulnerability to small perturbations, known as adversarial attacks, is widely acknowledged. Adversarial attacks can be categorized as white-box [16–18] or black-box attacks [19–21]. In a white-box attack, the attacker possesses complete knowledge of the targeted model such as the model's architecture and parameters. In contrast, a black-box attack occurs when the attacker has no knowledge of the targeted model's internal structure or parameters and can only interact through input–output queries. White-box attacks excel in performance but

lack transferability, while black-box attacks exhibit good transferability but lower overall performance. Positioned between these categories is the gray-box attack [22–24], which balances attack performance and transferability by leveraging limited knowledge about the targeted model, including partial information of its architecture and parameters.

Adversarial attacks have been extensively studied in supervised learning models [16–18,25–27], but their investigation in SSL models is comparatively limited. This discrepancy can be attributed to several factors. Firstly, SSL is commonly used in scenarios with an abundance of unlabeled data and a scarcity of labeled data, which restricts its applicability and hampers research on adversarial attacks. Secondly, the inherent complexity of SSL algorithms, compared to supervised learning algorithms, presents challenges in studying adversarial attacks within the SSL framework. Additionally, the research community often prioritizes understanding and improving the performance of SSL algorithms rather than exploring their vulnerabilities to adversarial attacks. Lastly, the lack of standardized benchmarks for assessing the robustness of SSL models impedes progress in this field.

In the context of SSL, the attacker has access to the publicly available labeled data that were used for training the SSL model. In other words, accessing complete knowledge of the model (as in a white-box scenario) is unrealistic, and not leveraging all available SSL information (as in a black-box scenario) is inefficient. A more feasible and efficient approach is the gray-box attack. This method, which bridges the gap between white-box and black-box attacks, effectively utilizes publicly available shared labeled data. In other words, if an attacker has knowledge of the limited label data used in SSL and can generate adversarial examples using this information, they can potentially cause severe damage to the SSL model. Though this scenario may seem improbable for supervised learning models, it is indeed practical and realistic for SSL models.

The use of publicly available labeled data in SSL models raises the potential for attacks. Attackers can exploit this accessibility to generate adversarial examples specifically targeting SSL models under gray-box conditions. Unlike previous studies that mainly focused on unrealistic poisoning attacks limited to white-box conditions and requiring access to the model's training phase [28], our research investigates the vulnerabilities of SSL methods and emphasizes the importance of adversarial training through evasion attacks (overall pipeline described in Figure 1). Evasion attacks can be conducted during the testing phase on pre-trained models, making them more realistic and practical. Our experiments, described in Section 4, confirm the significant impact of these adversarial examples on SSL models.

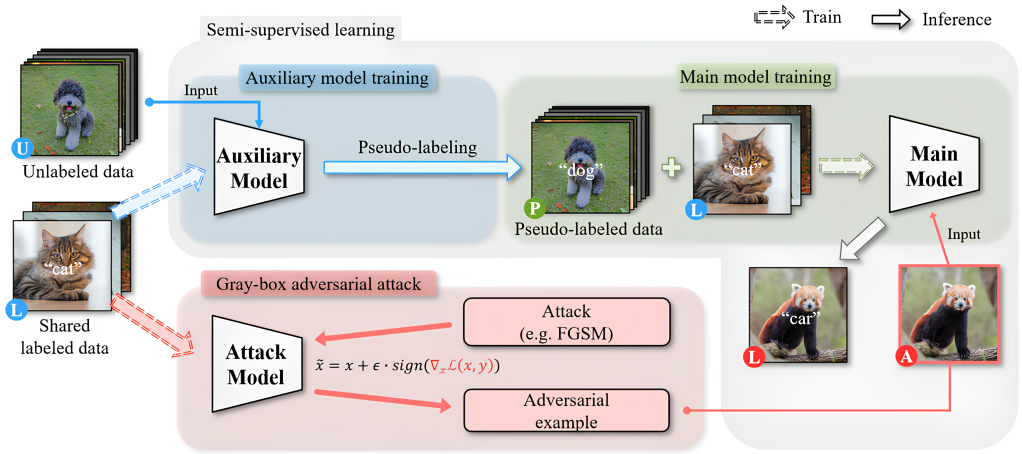

Figure 1. Overall pipeline of the proposed attack scheme. The auxiliary model and the main model are trained using a semi-supervised learning approach. The attack model is trained on shared labeled data to generate adversarial examples (the image marked as "A"), which lead to incorrect classification outcomes by the main model.

To enhance defense against attacks, SSL can employ privately trained models rather than pre-trained ones. These models are trained with customized hyperparameters or alternative initialization techniques, such as using different seed values or employing a distinct architecture. By doing so, attacks relying solely on publicly available labeled data become the only feasible option, eliminating the reliance on pre-trained models. However, experimental results in Section 4.2.1 demonstrate that attacks remain effective even under these conditions.

Consequently, this attack method has been validated to be effective across two scenarios.

1. Both the SSL model and adversarial attacker use the same pre-trained model;
2. Both the SSL model and adversarial attacker use the same labeled data but use different pre-trained models (our scenario).

Experiments were conducted to demonstrate the vulnerability of SSL models to the proposed attack methods in various scenarios. In response, defensive strategies, including adversarial training, were explored to counter these attacks. Adversarial training is recognized as an effective defense technique in machine learning that enhances model robustness against adversarial attacks. This technique involves the model being trained on both original data and intentionally crafted adversarial examples, thereby exposing it to challenging inputs and improving its ability to accurately classify perturbed inputs. Through this research, insights into potential defense strategies, such as adversarial training with unlabeled data and the utilization of an early stopping technique, are provided.

This paper examines the vulnerabilities of SSL methods and investigates the importance of incorporating adversarial training as a crucial component in SSL. Guidelines and insights are provided through experimental analysis on effectively defending against GAAS. We make the following contributions:

- By leveraging a realistic condition, it becomes possible to launch attacks on SSL models using only a small amount of publicly disclosed label data. Providing empirical validation demonstrates the effectiveness and practical applicability of such attacks.
- In response to identified vulnerabilities within semi-supervised learning (SSL) models, the Gray-box Adversarial Attack on Semi-supervised learning (GAAS) has developed. This scheme enhances attack efficiency, notably reducing the model's classification accuracy by up to 95%, thereby advancing the state-of-the-art adversarial attack methodologies.
- The proposed attack scheme can be successfully applied to the current state-of-the-art SSL methods across three main tasks—classification [1,3,5,6,8,29], object detection [9,11,12], and semantic segmentation [13,14,30]—using multiple datasets.
- Extensive experiments were conducted to investigate the defense method against GAAS.

These contributions indicate potentially harmful consequences that may arise from exploiting publicly available datasets for the training of SSL models.

2. Related Work

2.1. Adversarial Attacks

Machine learning models, despite their impressive performance, are vulnerable to adversarial attacks where the input data are perturbed to mislead the model into making incorrect predictions. These attacks can be classified into three categories: white-box, black-box, and gray-box attacks. All three types of attacks aim to mislead machine learning models by exploiting their vulnerabilities. However, they differ in the level of knowledge the attacker has about the model and the techniques used to craft adversarial examples. White-box attacks are the most powerful but least realistic, black-box attacks are the most realistic but least powerful, and gray-box attacks strike a balance between the two.

2.1.1. White-box Attack

White-box attacks, in the context of adversarial machine learning, are scenarios where the attacker has complete knowledge of the target model, including its architecture, param-

eters, and even the training data. The attacker uses this information to craft adversarial examples that can fool the model. The advantage of white-box attacks is that they can be highly effective due to the complete knowledge of the model. However, they are less realistic in real-world scenarios, as obtaining such detailed information about a model is often not feasible.

This section explores the evolution, methodologies, and significant contributions in the field of white-box adversarial attacks.

The concept of white-box attacks was first brought to light by Szegedy et al. [31], who demonstrated that neural networks are vulnerable to imperceptible perturbations in their inputs. This discovery led to a surge of interest in understanding and exploiting the vulnerabilities of machine learning models.

One of the most fundamental approaches in white-box attacks is the Fast Gradient Sign Method (FGSM) [16]. FGSM is designed to create adversarial examples rapidly by using the gradients of the neural network which is a kind of one-step gradient-based method. The one-step gradient-based method involves computing the gradient of the loss function with respect to the input data and then making a single step in the direction of the gradient to create an adversarial example. The adversarial example x_{adv} is generated as follows:

$$x_{\text{adv}} = x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}_{\text{sup}}(\theta, x, y)) \tag{1}$$

where, x is the original input, y is the true label for x, ϵ is a small constant, and $\mathcal{L}_{\text{sup}}(\theta, x, y)$ is the gradient of $\mathcal{L}$ with respect to x. The function $\text{sign}(\cdot)$ takes the sign of the gradient, resulting in a perturbation that is small but a direction that will increase the loss.

Following FGSM, Kurakin et al. [32] introduced the Basic Iterative Method (BIM), an extension of FGSM, applying the perturbation multiple times with a small step size. BIM adjusts the adversarial example iteratively:

$$x_{\text{adv}}^{(N+1)} = \text{Clip}_{x,\epsilon}\left\{ x_{\text{adv}}^{(N)} + \alpha \cdot \text{sign}(\nabla_x \mathcal{L}_{\text{sup}}(\theta, x_{\text{adv}}^{(N)}, y)) \right\} \tag{2}$$

where N is the iteration step, α is the step size, and $\text{Clip}_{x,\epsilon}$ ensures that the perturbations stay within the ϵ-neighborhood of the original input.

Iterative methods such as BIM involve making multiple small steps in the direction of the gradient, each time slightly modifying the adversarial example. The most well-known method in this category is Projected Gradient Descent (PGD) [17]. The primary difference between BIM and PGD lies in their approach to generating adversarial examples. BIM, an extension of the FGSM, applies the perturbation iteratively with a small step size. It adjusts the adversarial example in small increments, ensuring that each step stays within the ϵ-ball of the original image. The main goal of BIM is to find adversarial examples closer to the original image while maximizing the loss. On the other hand, PGD is considered a more general and robust method. It also iteratively applies perturbations but includes a random start within the allowed perturbation range before the iterations. This random start makes PGD more effective in finding adversarial examples that can fool the model, as it explores a wider range of potential perturbations. PGD is often referred to as the strongest first-order adversarial attack due to this comprehensive exploration strategy.

Carlini and Wagner [25] later developed a more sophisticated attack, known as the C&W attack, which further refined the approach to generating adversarial examples. Their method minimizes a different objective function that incorporates a term to measure the distance between the original and adversarial images, along with a term that encourages misclassification. The optimization problem for C&W attack can be formulated as:

$$\text{minimize} \parallel x - x_{\text{adv}} \parallel_p + c \cdot f(x_{\text{adv}}) \tag{3}$$

where $\parallel x - x_{\text{adv}} \parallel_p$ is the p-norm distance between the original and adversarial image, c is a constant found through binary search, and $f(x_{\text{adv}})$ is a function designed to produce high values for incorrectly classified examples.

2.1.2. Black-box Attack

Contrary to white-box attacks, black-box attacks assume that the attacker has no knowledge of the target model's internal workings. The attacker only has access to the model's inputs and outputs. Despite this lack of information, black-box attacks can still be effective by using techniques such as trial-and-error or gradient estimation. These attacks are more realistic in real-world scenarios, as attackers often do not have access to the inner workings of a model. These attacks are generally categorized into query-based and transfer-based attacks, each relying on different strategies to generate adversarial examples.

Query-based attacks involve probing the target model with inputs and using the model's outputs to craft adversarial examples. This approach typically requires numerous queries to the model, making it more challenging yet practical in real-world scenarios.

A notable example of a query-based attack is the Boundary Attack, introduced by Brendel et al. [33]. This attack starts with an adversarial example already misclassified by the model and iteratively refines it to decrease the distance to the original, correctly classified example. The update rule for the adversarial example in the Boundary Attack can be expressed as:

$$x_{\text{adv}}^{(N+1)} = x_{\text{adv}}^{(N)} + \eta \cdot \delta_N \tag{4}$$

where $x_{\text{adv}}^{(N)}$ is the adversarial example at iteration N, η is the step size, and δ_N is the perturbation vector, chosen to keep the example within the decision boundary.

Another important technique in query-based attacks is the Zeroth Order Optimization (ZOO) method proposed by Chen et al. [21]. ZOO approximates the gradient of the loss function with respect to the input using finite differences, a technique suitable for models where only output scores are accessible. The approximate gradient computation in ZOO is given by:

$$g_i \approx \frac{f(x + \beta_{e_i}) - f(x - \beta_{e_i})}{2\beta} \tag{5}$$

where $f(x)$ is the output score for input x, β is a small constant, and e_i is the unit basis vector along the i-th dimension.

Transfer-based attacks exploit the fact that adversarial examples are often transferable between different models. These attacks generate adversarial examples using a substitute model, which are then applied to the target black-box model.

Papernot et al. [34] pioneered this approach by training a local model to mimic the target model's behavior and then generating adversarial examples against this local model. The FGSM method, as discussed previously, is often used to generate these transferable adversarial examples.

An enhanced form of transfer-based attacks is the momentum iterative method introduced by Dong et al. [18]. This method integrates momentum into the iterative process, stabilizing update directions and improving transferability. The update rule for the momentum iterative method is:

$$g_{N+1} = \mu \cdot g_N + \frac{\nabla_x \mathcal{L}(\theta, x_{\text{adv}}^{(N)}, y)}{\|\nabla_x \mathcal{L}_{\text{sup}}(\theta, x_{\text{adv}}^{(N)}, y)\|_1} \tag{6}$$

$$x_{\text{adv}}^{N+1} = x_{\text{adv}}^{N} + \alpha \cdot \text{sign}(g_{N+1}) \tag{7}$$

where g_N is the accumulated gradient at iteration N, μ is the momentum factor, and α is the step size.

2.1.3. Gray-box Attack

Gray-box attacks in adversarial machine learning represent a scenario where the attacker has limited knowledge about the target model. Unlike black-box attacks, where the attacker has no information about the model's internals, or white-box attacks, where complete knowledge is available, gray-box attacks operate under partial information. This

typically includes knowledge about the model's architecture or access to some of the training data, but not the model's parameters or training process. Gray-box attacks balance the realism of black-box attacks with the effectiveness of white-box attacks, making them a significant threat in real-world scenarios.

In the previous paper, the authors introduce the concept of "extended black-box" attacks, which they also refer to as gray-box attacks [22]. In these attacks, the source model used to generate the adversarial examples can be a partially trained model that has a different network architecture than the target model. This is a slight variation on the traditional black-box attack, where the source model is typically a fully trained model that may or may not have the same architecture as the target model.

The authors argue that these gray-box adversarial attacks are a more realistic threat model, as attackers in the real world often have some level of knowledge about the model they are attacking. They propose the Gray-box Adversarial Training (GAT) algorithm as a way to make models more robust against these types of attacks.

Gray-box attacks are significant, as they reflect more realistic attack scenarios. One of the common approaches in gray-box attacks is leveraging the transferability of adversarial examples created from a known model to target an unknown model with a similar architecture or data domain.

A seminal approach in gray-box attacks was presented by Papernot et al. [35], who demonstrated the practicality of crafting adversarial samples on one model and successfully deploying them against another. This method utilizes the knowledge about the model's type or training data to create more effective attacks compared to completely blind black-box approaches.

Another important method in gray-box attacks is the use of surrogate models. In this approach, an attacker trains a surrogate model on publicly available datasets or a subset of the target model's training data. The adversarial examples generated against this surrogate model are then used to attack the target model as demonstrated by Tramèr et al. [36]. This method leverages the transferability of adversarial examples across different but related models.

Gray-box attacks are particularly relevant in scenarios where attackers can have some level of access or knowledge about the target system, such as in cloud-based machine learning services or when model architectures are public. This makes gray-box attacks a critical area of study for developing robust machine learning systems.

One practical application of gray-box attacks is in the evaluation of model robustness in environments where some information leakage is possible, such as shared cloud services or open-source machine learning platforms. Understanding the vulnerabilities exposed by gray-box attacks helps in designing more secure and resilient machine learning systems.

2.2. Adversarial Training

Adversarial training (AT) is a defensive strategy against adversarial attacks in machine learning. It involves training models on adversarial examples to improve their resilience to such attacks. This method not only enhances the robustness of models against specific attack methods but also aims to improve their generalization against a variety of adversarial manipulations.

Development and Techniques

The foundation of adversarial training was laid by Goodfellow et al. [16], who proposed incorporating adversarial examples into the training process. This initial method involved generating adversarial examples using the Fast Gradient Sign Method (FGSM) and then retraining the model with a mixture of normal and adversarial examples. The idea is to make the model learn from the adversarial perturbations, thereby reducing its susceptibility to such manipulations during inference.

A significant advancement in adversarial training was proposed by Madry et al. [17] with the introduction of Projected Gradient Descent (PGD) for generating adversarial

examples. PGD is an iterative method that takes multiple small steps in the direction of the gradient of the loss function, making it a more powerful adversarial example generator compared to FGSM. The adversarial training process using PGD can be formulated as:

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}}[\max_{\delta\in S}\mathcal{L}(\theta, x+\delta, y)] \tag{8}$$

where θ are the model parameters, $\mathcal{D}$ is the data distribution, $\mathcal{L}$ is the loss function, x and y are the input and its true label, respectively, and S is the set of allowed perturbations.

Since Madry et al. [17] defined AT as a min-max problem, its variants showed various ways to solve this problem. Some works like [37–40] add a regularization term to optimize the outer minimization problem. Ding et al. [41] focused on the model robustness being directly connected with margin maximization and improved the robust performance by adaptive ϵ and modifying the loss function. In contrast, Rade et al. [42] discussed that adversarial training leads to an unintentional increase in the margin, which harms the natural accuracy, and reduced this effect by using additional crafted images called helper examples.

Meanwhile, following the study of Schmidt et al. [43] wherein more data are required for robustness rather than for natural accuracy, many subsequent studies [44–46] have improved robustness through SSL settings using unlabeled data. These studies were conducted by using additional unlabeled data to improve robustness when the amount of labeled data for natural accuracy was sufficient. However, when using SSL, the labeled data are often insufficient to achieve natural accuracy, so these works are closer to omni-supervised learning rather than SSL. In this work, we aim to make an AT that is compatible with existing SSL methods and can maintain robustness and clean accuracy, even with limited labeled data.

2.3. Semi-Supervised Learning

Semi-supervised learning (SSL) combines the strengths of supervised and unsupervised learning, utilizing both labeled and unlabeled data for training models. This approach is particularly valuable in scenarios where acquiring a large amount of labeled data is impractical. SSL has been making significant strides in various domains such as classification and object detection, with notable methodologies emerging in recent years.

2.3.1. Classification

In SSL for classification, algorithms aim to improve classification accuracy by leveraging the vast amount of unlabeled data alongside the limited labeled data. Pioneering work in this area is the Self-Training or Pseudo-Labeling technique, where a model initially trained on labeled data generates labels for the unlabeled data, which are then used for further training [3]. This iterative process allows the model to gradually improve its understanding and adapt to the broader data distribution.

Another significant approach is the MixMatch algorithm by Berthelot et al. [47], which blends labeled and unlabeled data using a technique called data augmentation. It generates guesses for unlabeled data and mixes them with labeled data to create a more robust training set. The MixMatch approach can be formulated as:

$$\hat{x} = \lambda x_i + (1-\lambda)x_j \tag{9}$$

where x_i and x_j are instances from labeled and unlabeled datasets, respectively, and λ is a parameter controlling the mixing ratio.

Building upon this, FixMatch, introduced by Sohn et al. [5], has further refined the approach. FixMatch employs a simple yet effective mechanism that balances the use of weak and strong augmentations to enforce consistency in the model's predictions. By applying a confidence threshold to the predictions on strongly augmented unlabeled data, FixMatch ensures that only reliable pseudo-labels are used for training, significantly enhancing the model's performance in classification tasks.

2.3.2. Object Detection

SSL has been extended to object detection, a task involving locating and classifying objects in images. A notable contribution is Self-Training with Noisy Student (STAC) by Sohn et al. [10]. In STAC, a teacher model trained on labeled data generates pseudo-labels for unlabeled data, which are then refined and used to train a student model. This approach significantly enhances object detection performance in low-data regimes.

Another breakthrough in SSL for object detection techniques is UnbiasedTeacher and SoftTeacher, which have set new benchmarks in this area.UnbiasedTeacher, proposed by Liu et al. [9], tackles the challenge of confirmation bias in SSL. It employs a teacher–student framework, where the teacher model generates pseudo-labels for training the student model. To prevent the student from inheriting biases, the teacher is periodically updated with the student's weights, promoting a more balanced learning process. On the other hand, SoftTeacher, developed by Xu et al. [12], introduces a soft-labeling approach. It assigns confidence scores to pseudo-labels, allowing the model to express uncertainty and receive more informative training signals. This method is complemented by a dynamic thresholding mechanism, which adaptively selects the most reliable pseudo-labels based on the model's evolving confidence. Both UnbiasedTeacher and SoftTeacher have shown remarkable effectiveness in improving the accuracy of object detection models under limited labeled data conditions.

2.3.3. Semantic Segmentation

Semantic segmentation in the context of semi-supervised learning (SSL) is particularly challenging due to the high cost of acquiring pixel-level annotations. SSL techniques in semantic segmentation aim to utilize unlabeled data effectively to overcome the scarcity of labeled data. Two prominent approaches in this domain are the adaptation of the CutMix [14] technique and the development of the Region Contrast (ReCo) [13] method.

Originally a data augmentation strategy for classification tasks, CutMix has been adapted for semantic segmentation to leverage unlabeled data. This method involves cutting and pasting patches between training images and their corresponding labels, effectively creating new training samples. The fundamental idea is to make the model learn to predict the correct classes for pixels in mixed regions, enhancing its ability to generalize across diverse contexts.

The CutMix technique can be mathematically represented as follows:

$$\hat{X} = M \odot X_A + (1 - M) \odot X_B \tag{10}$$

$$\hat{Y} = M \odot Y_A + (1 - M) \odot Y_B \tag{11}$$

where X_A and X_B are randomly selected training images, Y_A and Y_B are their corresponding segmentation masks, M is a binary mask indicating where to cut and paste, and $\odot$ denotes element-wise multiplication. The new training sample $\hat{X}$ and its segmentation mask $\hat{Y}$ combine features and labels from both images. This process encourages the model to handle partial objects and mixed scenes, thus improving its segmentation capability.

ReCo extends SSL to semantic segmentation by employing region-level contrastive learning. It is designed to enhance the discriminative ability of models by encouraging them to learn distinct features for different image regions. The core concept is to compare regions within an image or across images to enforce consistent predictions for similar regions and diverse predictions for different regions.

In ReCo, a region-level contrastive loss is introduced, formulated as follows:

$$\mathcal{L}_{contrast} = -\log \frac{exp(\text{sim}(f(r_i), f(r_j))/\tau)}{\Sigma_{k=1}^{N} \mathbb{1}_{[k \neq i]} exp(\text{sim}(f(r_i), f(r_k))/\tau)} \tag{12}$$

where $f(r_i)$ and $f(r_j)$ are feature representations of regions r_i and r_j, respectively, sim$(\cdot)$ is a similarity function (e.g., cosine similarity), τ is a temperature-scaling parameter, and N is

the number of regions in the contrastive loss computation. This loss function encourages the model to learn representations that are similar for corresponding regions and distinct for different regions, enhancing the quality of semantic segmentation.

3. Methodology

This section details the attack scheme for SSL, called Gray-box Adversarial Attack on Semi-supervised learning (GAAS). The discussion begins with a review of semi-supervised learning, which is the target model of the method. Then, a comprehensive overview of the proposed method is provided, including the generation of adversarial examples. Moreover, the paper delves into how the proposed approach can be adapted to handle more complex vision tasks, such as object detection and semantic segmentation. Lastly, the justification for the method is discussed.

3.1. Review of Semi-Supervised Learning

As shown in Figure 1, the pipeline of the target model consists of two steps: auxiliary model training and main model training. This pipeline is based on pseudo-labeling methods adopted commonly by state-of-the-art SSL models [13,29,48–53]. In the auxiliary model training, the auxiliary model $f_{\text{aux}} : \mathcal{X} \mapsto \mathcal{Y}$ is trained by supervised learning on a small labeled dataset $D_L = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ for $i = 1, 2, \ldots, n$, where $\mathcal{X}$ is the input space, $\mathcal{Y}$ is the label space, $x_i \in \mathcal{X}$ represents the input, and $y_i \in \mathcal{Y}$ indicated the corresponding label. Next, pseudo-labels of the unlabeled data x'_j are then estimated by the trained auxiliary model:

$$y'_j = f_{\text{aux}}(x'_j) \tag{13}$$

The set of the unlabeled data coupled with the pseudo-labels is denoted by D_P, where $D_P = \{(x'_1, y'_1), (x'_2, y'_2), \ldots, (x'_m, y'_m)\}$ for $j = 1, 2, \ldots, m$, where $x'_j \in \mathcal{X}'$ is the input from the unlabeled dataset and $y'_j \in \mathcal{Y}'$ is the pseudo-label generated by the auxiliary model. In the main model training, the main model $f_{\text{main}} : \mathcal{X} \mapsto \mathcal{Y}$ is trained on a combination of the labeled dataset D_L and the pseudo-labeled dataset D_P. The main model improves performance through iterative learning.

3.2. Gray-box Adversarial Attack through the Shared Data

In the gray-box adversarial attack, adversarial examples for the attack model f_{atk} are generated using an attack method. The f_{atk} model serves as a surrogate model trained from the same labeled data as the target model f_{aux} but without its other knowledge. The shared labeled data between the two models are crucial for the success of this approach, ensuring they operate within the same data domain. This promotes the transferability of adversarial attacks, thereby enhancing the effectiveness of the attack on the SSL model during the inference stage.

The use of shared labeled data has two key reasons. Firstly, in real-world scenarios, SSL models often incorporate open datasets, leading to the sharing of labeled data between the attack and target models. Secondly, the shared labeled data serve as a foundation for identifying the vulnerabilities of the target model. By training both models on the same task within the same data domain, the surrogate model can uncover and exploit weaknesses in the target model. A more detailed analysis of this will be presented in Section 3.3.

Generating Adversarial Examples

Adversarial examples are generated using the model f_{atk} and attack methods designed for supervised learning models. These methods fall into two categories. The Fast Gradient Sign Method (FGSM) [16], a one-step gradient-based method, is employed, along with the momentum iterative fast gradient sign method (MI-FGSM) [18] and Projected Gradient Descent (PGD) [17], both of which are iterative methods. These attack methods are widely

recognized and established in numerous papers addressing adversarial attacks [54–58]. The FGSM attack that is a one-step gradient-based method generates an adversarial example by:

$$\tilde{x} = x + \epsilon \cdot \text{sign}\big(\nabla_x \mathcal{L}_{\text{sup}}(f_{\text{atk}}(x), y)\big) \tag{14}$$

where $\tilde{x} \in \tilde{\mathcal{X}}$ is the adversarial example, $x_i \in \mathcal{X}$ is the original input, $y_i \in \mathcal{Y}$ is the true label, $\mathcal{L}_{\text{sup}}$ is the loss function of f_{atk}, and FGSM generates adversarial examples with a small $\mathcal{L}_\infty$ norm bound, i.e., $||\tilde{x} - x||_\infty \leq \epsilon$. We also adopt iterative methods such as MI-FGSM and PGD. These methods iteratively apply multi-step variant FGSM multiple times with a small step size α:

$$\tilde{x}_{t+1} = \tilde{x}_t + \alpha \cdot \text{sign}(\nabla_x \mathcal{L}_{\text{sup}}(f_{\text{atk}}(x), y)) \tag{15}$$

By including both one-step and iterative methods, our approach ensures the versatility and effectiveness of the adversarial examples generated, enabling a thorough examination of the vulnerabilities of SSL models in different attack scenarios.

3.3. Justification for Method

The key attribute of adversarial examples that underpins the methods we have outlined above is their transferability. Essentially, different machine learning models often share similar decision boundaries surrounding data points. This similarity allows adversarial examples, initially generated for one model, to successfully manipulate other models as well [16,19,31,59].

This transferability has sparked discussions, with Papernot et al. [20] proposing that an attacker can exploit a substitute model trained on the same data to generate effective adversarial examples against the target model. By training a surrogate model using adaptive queries, this approach transforms the typically opaque black-box attack into a white-box attack, making the target model's internal mechanisms transparent. However, this approach presents challenges, requiring full access to the target model's prediction confidences and a large number of queries, especially with complex datasets like ImageNet [60]. To overcome these challenges, attackers often resort to black-box attacks, targeting remote classifiers without knowledge of the model's architecture, parameters, or training data. Nevertheless, in the context of SSL models, some labeled data are often shared between the attack model and the target model since SSL commonly employs open datasets. Hence, we believe that aligning the learning tasks and data domains of the two models could expose the target model's vulnerabilities. By synchronizing these aspects, adversarial examples can be generated to exploit the weaknesses of the target model effectively.

3.3.1. Input-gradient Alignment

Demontis et al. [59] proposed a theoretical framework predicated on the concept of 'input-gradient alignment' to account for the transferability of adversarial attacks across disparate models. The degree of alignment between input gradients of the attacking and target models primarily dictates the likelihood of adversarial examples being transferable. In other words, when the input gradients align more closely, the transferability of adversarial examples is enhanced. This degree of alignment can be quantified using several similarity metrics, such as the cosine similarity or the dot product between the input gradients of the attack model and the target model.

3.3.2. Shared vs. Non-shared Data

Changes in the data domain can impact the learned model gradient, which is particularly critical in the context of SSL models operating outside their data domain. When generating an adversarial example, the gradient employed by the attack model may not align with the target model's learning direction. To validate this hypothesis, a toy experiment was conducted using the WideResNet (WRN)-28-8 model [61] trained as both the attack and target models on standard classification datasets: CIFAR10/100 [62] and SVHN [63]. Figure 2 illustrates a shift in gradient alignment between shared and non-

shared datasets, resulting in a noticeable decrease in alignment and implying that dataset compatibility influences model congruence. This inconsistency may diminish the attack's effectiveness. Thus, it is concluded that dataset selection plays a pivotal role in determining model alignment and the efficacy of an attack.

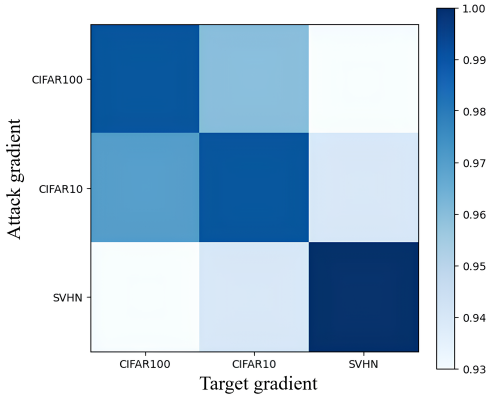

Figure 2. The cosine similarity is calculated between the gradients of the attack model and the target model for each dataset, which is then represented as a confusion matrix. Darker color signifies a higher degree of similarity.

3.4. Object Detection and Semantic Segmentation

Contrary to the previously described classification, implementing the proposed method in object detection and semantic segmentation tasks presents unique differences due to their distinctive processes. Nevertheless, it is crucial to note that the fundamental structure of the method remains consistent across all tasks, despite varying specifics.

3.4.1. Object Detection

Object detection aims to identify and locate multiple objects within an image, providing bounding boxes and class labels for each detected entity. Both the auxiliary and main models in this context can be seen as object detection models, similar to Faster R-CNN [64]. The auxiliary model predicts bounding boxes and class labels for objects in unlabeled images, generating pseudo-labels. Adversarial examples are then generated for labeled data using an attack method like FGSM, targeting the attack model. It is important to note that the loss function may require modification to accommodate both bounding box and class label predictions, known as the total-loss concept [65,66]. Finally, during inference, these adversarial examples are fed into the main model to assess the impact of adversarial attacks on object detection performance metrics, such as mean Average Precision (mAP).

3.4.2. Semantic Segmentation

In the semantic segmentation task, each pixel in an image is assigned a class label, resulting in a densely labeled image. Both the auxiliary and main models can be designed as semantic segmentation models like DeepLab [67]. The auxiliary model predicts class labels for pixels in unlabeled images, generating pseudo-labels in the form of segmentation masks. The loss function needs to be adapted to incorporate pixel-wise label predictions, such as using cross-entropy loss over all pixels. Similar to object detection, adversarial examples are inputted to the main model during inference to assess the impact of adversarial attacks on semantic segmentation performance metrics, such as mean Intersection over Union (mIoU) or pixel accuracy.

4. Experiments

This section provides a comprehensive and detailed analysis of the experimental setup and results for the proposed attack method, GAAS, across classification, object detection, and semantic segmentation tasks. Additionally, the attack performance is evaluated under

various conditions, including the use of private models, different datasets, and variations in model capacity.

4.1. Setup

4.1.1. Dataset

For the classification task, multiple datasets are utilized, including CIFAR-100, CIFAR-10, and SVHN. Specifically, in CIFAR-100, 10K out of 60K samples are used for training purposes, and the remaining 50K for testing. In the case of CIFAR-10 and SVHN, 4K and 1K samples are employed, respectively. Shifting the focus to the object detection task, the MS-COCO [68] dataset is implemented, where label ratios of 1% and 5% are maintained within the SSL environment. Lastly, for addressing the semantic segmentation task, the Pascal VOC 2012 [69] dataset is used, also maintaining label ratios of 1% and 5%.

4.1.2. Hyperparameter

Regarding hyperparameters, for the classification task, model configurations from the USB framework [70] are adopted with an iteration count of 500K. The evaluation process encompasses assessing the attack's performance on private models, analyzing the impact of diverse datasets on learning, and evaluating the attack's efficacy based on model capacity. In the object detection task, settings from Detectron2 [71] and MMdetection [72] with 180K iterations are utilized. For the semantic segmentation task, models are trained for 200 epochs, aligning the iteration count with the training batch length. To ensure consistency, the proposed method is directly incorporated into the original code, following the settings established in a previous paper.

To comprehensively evaluate the robustness and effectiveness of the proposed attack method, experiments are designed across various settings and configurations. The "White-box" item is included in the results to compare adversarial attack performance under white-box conditions, representing the model's performance when subjected to the FGSM attack, assuming complete knowledge of the model's architecture and parameters by the attacker. Additionally, the "Clean" item indicates the unattacked model's performance on clean data. These comparisons provide insights into the potency of the GAAS attack in relation to white-box attacks and the baseline performance of the SSL models.

4.1.3. Evaluation

Evaluation of the impact on the main model is conducted by the adversarial examples generated using the attack model, which are input to the main model f_{main} during the inference phase, potentially degrading the performance of the semi-supervised model:

$$\tilde{y} = f_{\text{atk}}(\tilde{x}) \tag{16}$$

We can then compare the predictions of the main model on clean inputs and adversarial examples to measure the impact of adversarial attacks on the model's performance:

$$\text{Robust accuracy} = \frac{\sum_{i=1}^{n} \mathbb{1}(\tilde{y}_i = f_{\text{main}}(\tilde{x}_i))}{n} \tag{17}$$

where n is the number of test examples, and $\mathbb{1}(x)$ is the indicator function, which returns 1 if the condition x is true and 0 otherwise. By comparing the accuracy and adversarial accuracy, we can evaluate the robustness of the semi-supervised learning model to adversarial attacks. A significant drop in adversarial accuracy compared to the standard accuracy would indicate the model's vulnerability to these attacks.

4.2. Classification

For the classification task, we compared the Pseudo-label model [3] with Mean-Teacher [6], ReMixMatch [1], FixMatch [5], FlexMatch [8], and SimMatch [29]. These models were selected based on their performance and popularity in SSL. We adjusted the

hyperparameters to ensure fair evaluation and optimal performance, training on the CIFAR-100 [62] dataset. Adversarial examples were generated by attacking a WRN-28-8 model.

We evaluated the effectiveness of various attack methods: FGSM [16], PGD-10/PGD-100 [17], and BIM [32], representing a wide range of extensively studied adversarial techniques [27,73–77]. PGD had an epsilon value of 8/255 and an alpha value of 2/255.

Table 1 presents the experimental results of the GAAS attack on the classification tasks. SSL models exhibited significant performance drops when subjected to adversarial attacks (FGSM, BIM, PGD-10, and PGD-100) compared to clean data. Notably, GAAS achieved performance degradation similar to white-box attacks, despite operating under less favorable conditions.

Table 1. Attack performance variation across different classification models by GAAS.

	Pseudo-Label [3]	**FixMatch [5]**	**FlexMatch [8]**	**ReMixMatch [1]**	**SimMatch [29]**
Clean	63.50	77.25	78.08	78.19	79.24
FGSM	12.92	44.94	46.15	46.15	50.07
BIM	3.52	46.74	46.24	46.63	52.94
PGD-10	3.49	47.60	47.12	46.94	53.93
PGD-100	2.57	44.36	43.79	43.14	51.55
White-box	8.78	23.28	20.22	27.33	39.33

These findings demonstrate the effectiveness of the GAAS attack and highlight the comparable vulnerability of SSL models to both GAAS and white-box attacks. It is noteworthy that GAAS operates with limited knowledge of the labeled data domain used for training, while white-box attacks possess complete knowledge of the model's architecture and parameters.

4.2.1. Private Attack Model

We investigate a scenario where the attacker possesses a private attack model instead of having access to the publicly available auxiliary model typically used in SSL attacks. This examination is vital, as it evaluates the transferability of the proposed attack method across different models. The private models employed in this scenario encompass a wide range, including models with varying hyperparameter settings, representing different complexities and distinct architectures such as WRN-28-8, ViT [78], and ResNet50 [79]. These models are chosen for their diverse architectures and widespread usage in various classification tasks.

To ensure a comprehensive and robust evaluation, we conduct cross-validation using each model. As illustrated in Figure 3, the accuracy experiences a decline ranging from 6% to 35%, depending on the model and attack method. These findings indicate that even with an undisclosed model, an attacker can still target the SSL model using only a portion of the labeled data domains used during training, significantly impacting its performance.

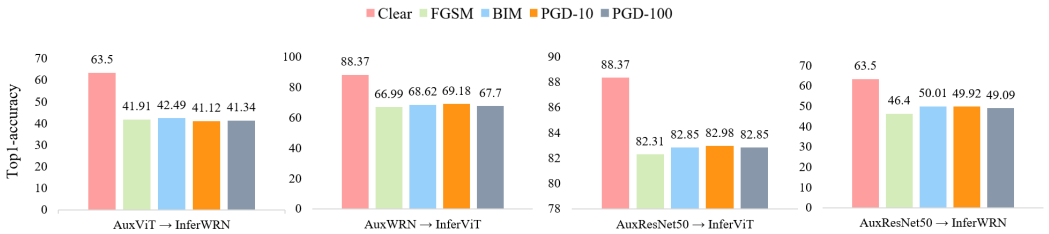

Figure 3. Impact of private model selection on attack performance in semi-supervised learning models. The notation "AuxViT → InferWRN" signifies that ViT was employed as the attack model, and the adversarial example generated through it was used to attack the main model, which was trained using WRN-28-8 as an auxiliary model.

4.2.2. Various Datasets

We evaluate the attack performance based on the dataset used for model training. The datasets include CIFAR-10/100 [62], and SVHN [63]. These datasets are selected because

they offer a diverse range of complexity and represent various domains, allowing us to assess the versatility of the proposed attack method. We use the Pseudo-label [3] model for this evaluation.

The results show that model accuracy decreases from a minimum of 68% to a maximum of 98%, depending on the dataset and attack method (Table 2). These results highlight the varied effects of the proposed attack methods across different datasets and suggest that attacks against the SSL model are possible for any particular dataset and domain.

Table 2. Attack performance variation across different datasets in SSL models.

	CIFAR-100	**CIFAR-10**	**SVHN**
Clean	63.50	86.17	90.34
FGSM	12.92	23.72	28.52
BIM	3.52	2.25	14.35
PGD-10	3.41	2.72	15.04
PGD-100	2.73	1.84	13.24

4.2.3. Model Capacity

We conduct a detailed evaluation of the attack performance by varying the capacity of the attack model. Specifically, we utilize the Pseudo-label [3] SSL model with WRN-28-8 as an auxiliary model, modifying the depth of the WRN-28 [61] attack model. This experiment is crucial for understanding the impact of model capacity on the effectiveness of our proposed attack method.

The results demonstrate a positive correlation between the attack model's capacity and the attack effect (Table 3). This indicates that higher-capacity attack models generate more accurate pseudo-labels, leading to improved attack performance. In other words, increasing the attack model's capacity results in the generation of more accurate pseudo-labels, which in turn enhances the attack performance of adversarial examples. These findings highlight the significance of considering model capacity in the design of both attacks and defenses within the realm of semi-supervised learning.

Table 3. Effect of varying attack model capacity on GAAS performance.

	WRN_28_2	**WRN_28_4**	**WRN_28_6**	**WRN_28_8**
Clean		63.50		
FGSM	13.83	13.27	12.92	11.08
BIM	7.94	4.82	3.67	3.52
PGD-10	8.35	4.98	3.71	3.41
PGD-100	6.47	3.87	2.97	2.73

4.3. Object Detection

We examine the impact of GAAS on object detection, a challenging task in computer vision. We evaluate popular semi-supervised learning object detection models, including Unbiased Teacher [9], Pseco [11], and Soft Teacher [12]. These models are selected based on their performance, architecture, and prominence in the object detection domain.

We train these models using the widely recognized and demanding COCO-standard dataset [68]. To assess the attack performance under different degrees of supervision, we incorporate 1% and 5% labeled data for Faster-RCNN [64], which serves as the attack model. Our comprehensive analysis reveals the vulnerability of these models to the proposed attack method, particularly in real-world scenarios with limited labeled data. This emphasizes the practical significance and potential impact of our findings across various applications.

As shown in Table 4, the results show a reduction in mAP from a minimum of 50% to a maximum of 91%. These findings demonstrate the effectiveness of the proposed GAAS attack method in object detection tasks.

Table 4. Attack performance comparison for SSL object detection models with varying labeled data percentages under different attack methods.

	Unbiased Teacher [9]		**Soft Teacher [12]**		**Pseco [11]**	
	1%	**5%**	**1%**	**5%**	**1%**	**5%**
Clean	20.16	27.84	22.40	30.70	22.70	32.60
FGSM	13.83	13.27	14.12	15.33	14.15	15.89
MI-FGSM	13.27	4.82	8.02	5.21	8.24	5.66
PGD-10	8.92	5.09	8.42	5.40	8.61	5.74
PGD-100	0.69	0.87	0.88	1.01	1.71	1.73
White-box	5.98	8.16	6.23	8.42	8.33	10.58

4.4. Semantic Segmentation

We conduct further verification of the GAAS attack effect in semantic segmentation, a critical computer vision task. The experiment employs Reco [13], CutMix [14], and Cutout [30] as SSL semantic segmentation models, selected based on their performance, architecture, and relevance to the task.

The models are trained on the widely used Pascal VOC dataset [80] with 1% and 5% labeled data for Deeplabv3+ [67], which serves as the attack model. This evaluation explores the vulnerability of these models to the proposed attack method under different levels of supervision, simulating scenarios with limited labeled data.

The results reveal a decrease in mIoU ranging from 39% to 66%, highlighting the effectiveness of the GAAS attack method in semantic segmentation tasks (Table 5). These findings emphasize the need for robust defense mechanisms to safeguard semi-supervised learning models in this domain.

Table 5. Attack performance comparison for SSL semantic segmentation models with varying labeled data percentages under different attack methods.

	ReCo [13]		**CutMix [14]**		**CutOut [30]**	
	1%	**5%**	**1%**	**5%**	**1%**	**5%**
Clean	72.76	74.48	70.81	72.99	68.94	71.85
FGSM	41.47	38.75	42.83	43.70	40.46	43.70
MI-FGSM	26.99	25.61	32.28	33.57	31.71	33.27
PGD-10	28.47	26.71	34.67	34.94	33.47	34.94
PGD-100	27.84	26.07	34.96	35.27	33.95	35.91
White-box	38.66	35.12	39.12	41.31	36.72	40.75

Through extensive and detailed experiments in classification, object detection, and semantic segmentation tasks, we demonstrate the efficacy of the proposed GAAS attack method. The attack performance varies based on factors like the private model, dataset, and capacity of the attack model. These insights underline the importance of comprehending vulnerabilities in SSL models and developing resilient defense mechanisms to counter adversarial attacks across diverse domains and tasks.

5. Discussion on Defense

In this section, we present a defense strategy against the adversarial attacks proposed in this paper. We focus on adversarial training as a defense mechanism, following the method of Madry et al. [17].

5.1. Adversarial Training

We conduct experiments using the CIFAR-100 [62] dataset, with 10,000 labeled data. We apply the adversarial training method to the basic pseudo-labeling model [3], incorporating both labeled and unlabeled data.

To enhance robustness, we experiment with training from not only labeled and unlabeled data separately but also together during adversarial training. We find that the best robustness is achieved when training from the unlabeled data (see Figure 4). The adversarial training process can be represented as:

$$\min_\theta \mathbb{E}_{(x,y)\sim \tilde{D}_{\text{ulb}}}[\max_{\delta\in\Delta} \mathcal{L}(f_\theta(x+\delta), y)] \tag{18}$$

where $\tilde{D}_{\text{ulb}}$ denotes the adversarial example dataset (unlabeled data), f_θ is the model parameterized by θ, $\mathcal{L}$ is the adversarial loss function, and Δ represents the set of allowable perturbations.

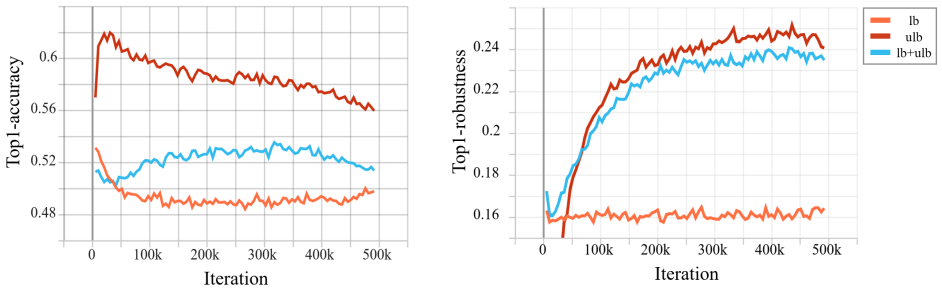

Figure 4. Adversarial training: Examining model (**a**) robustness and (**b**) accuracy with labeled (lb), unlabeled (ulb), and combined (lb+ulb) adversarial examples.

5.2. Robustness and Auxiliary Model Performance

Our experiments, depicted in Figure 5, indicate that the model's robustness is influenced by the performance of the auxiliary model, consistent with prior studies [39,40,81]. As the auxiliary model's performance improves, the SSL model's robustness also increases, but it gradually diminishes during the training process. Therefore, employing techniques like early stopping becomes valuable when conducting adversarial training with adversarial examples generated by an auxiliary model trained on limited labeled data. Early stopping can be defined as finding the model parameter θ^* that minimizes the validation loss $\mathcal{L}(f_\theta(x), y)$, where (x, y) is sampled from the validation dataset D_{val}:

$$\theta^* = \text{argmin}_\theta \mathbb{E}_{(x,y)\sim D_{\text{val}}}[\mathcal{L}(f_\theta(x), y)] \tag{19}$$

By combining adversarial training with early stopping, we observe promising results in mitigating the impact of adversarial attacks on the semi-supervised learning model. These techniques enhance the model's robustness across various tasks, including classification, object detection, and semantic segmentation. Future research can explore alternative defense mechanisms and evaluate their effectiveness against the proposed adversarial attacks.

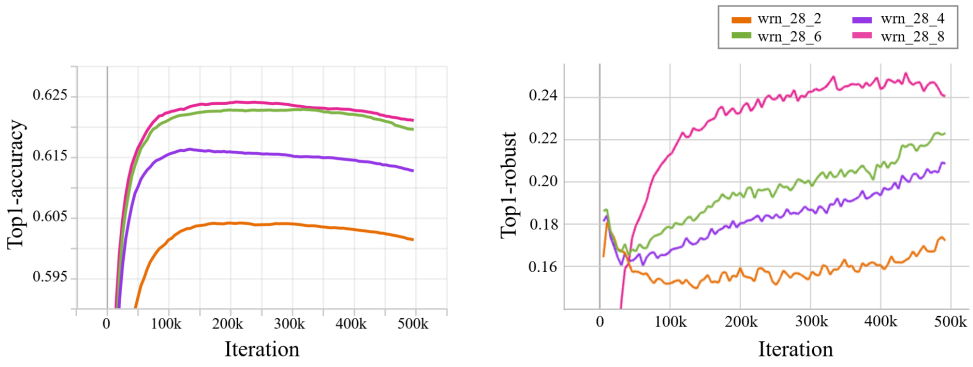

Figure 5. Impact of auxiliary model capacity on (**a**) model robustness and (**b**) accuracy in adversarial training: This figure illustrates the relationship between auxiliary model capacity and SSL model performance, with graph (**a**) demonstrating the influence on robustness and graph (**b**) depicting the effect on accuracy during adversarial training.

6. Discussion

In this paper, we have introduced the Gray-box Adversarial Attack on Semi-supervised learning (GAAS), specifically targeting Semi-Supervised Learning (SSL) models. This approach capitalizes on the characteristic of SSL models to rely on publicly available labeled data in scenarios where labeled information is scarce. GAAS trains an attack model based on public data and applies the generated adversarial examples to degrade the performance of SSL models. This strategy is particularly relevant in real-world situations where SSL models depend on public datasets.

The experiments conducted in this paper demonstrate that GAAS can effectively attack SSL models across a range of visual tasks such as image classification, object detection, and semantic segmentation. These experimental results clearly illustrate the vulnerability of SSL models to attacks leveraging public data, providing significant implications for research and applications using SSL models. Notably, the efficacy of GAAS is evident across various settings and experimental conditions of the attack model, proving its adaptability to different environments of SSL models.

Furthermore, this study emphasizes the importance of adversarial training as a defensive strategy against GAAS. Through experimentation, it has been shown that adversarial training effectively enhances the resilience of SSL models against adversarial examples. In particular, adversarial training using unlabeled data demonstrates higher defense effectiveness, offering crucial guidance for developing defensive strategies in SSL models. Additionally, the incorporation of early stopping techniques in the training process further strengthens the robustness of the models.

In future research, expanding the investigation of GAAS beyond vision tasks to other domains such as natural language processing is essential. This broadened scope would offer a comprehensive understanding of GAAS's impact across various fields, shedding light on the vulnerabilities and strengths of SSL models in different contexts. Concurrently, a detailed quantitative analysis of the transferability of adversarial attacks could deepen our understanding of the mechanisms enabling these attacks to succeed across diverse SSL models. Such insights are critical for developing more robust and secure SSL systems. Furthermore, delving into the privacy and security implications of using public datasets in SSL, especially in terms of data protection and privacy preservation, is imperative. This exploration is crucial given the increasing reliance on publicly available data and the consequent risks of privacy breaches and security threats. Lastly, expanding the research to encompass other types of attacks, such as data poisoning or model inversion, would provide a more in-depth view of SSL model vulnerabilities.

7. Conclusions

In conclusion, this study on the vulnerabilities of SSL models to gray-box adversarial attacks illuminates the inherent risks associated with using publicly available data and underscores the importance of implementing advanced defensive strategies. Through demonstrating the effectiveness of adversarial training and identifying potential areas for future research, the goal is to contribute to the development of more secure and robust SSL models. The path forward involves enhancing the resilience of these models to adversarial attacks and ensuring that privacy and security considerations are prioritized in SSL research and application.

Author Contributions: Conceptualization, J.J.; methodology, J.J.; software, J.J. and J.K.; validation, J.J. and J.K.; formal analysis, J.J. and J.K.; investigation, J.J.; resources, J.J.; data curation, J.J. and J.K.; writing—original draft preparation, J.J. and J.K.; writing—review and editing, J.J.; visualization, J.J. and J.K.; supervision, Y.-J.S.; project administration, Y.-J.S.; funding acquisition, Y.-J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-01906, Artificial Intelligence Graduate School Program(POSTECH)) and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2022R1A6A1A03052954).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SSL	Semi-Supervised Learning
GAAS	Gray-box Adversarial Attack on Semi-supervised learning
FGSM	Fast Gradient Sign Method
BIM	Basic Iterative Method
PGD	Projected Gradient Descent
ZOO	Zeroth Order Optimization
GAT	Gray-box Adversarial Training
STAC	Self-Training with Noisy Student
ReCo	Region Contrast
MI-FGSM	Momentum Iterative Fast Gradient Sign Method
WRN	WideResNet
R-CNN	Regions with Convolutional Neural Network features
mAP	mean Average Precision
mIoU	mean Intersection over Union

References

1. Berthelot, D.; Carlini, N.; Cubuk, E.D.; Kurakin, A.; Sohn, K.; Zhang, H.; Raffel, C. ReMixMatch: Semi-Supervised Learning with Distribution Matching and Augmentation Anchoring. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2020.
2. Laine, S.; Aila, T. Temporal Ensembling for Semi-Supervised Learning. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
3. Lee, D.H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In Proceedings of the Workshop on Challenges in Representation Learning, Atlanta, GA, USA, 21 June 2013.
4. Pham, H.; Dai, Z.; Xie, Q.; Le, Q.V. Meta pseudo labels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA,18–24 June 2021.
5. Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C.A.; Cubuk, E.D.; Kurakin, A.; Li, C.L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Adv. Neural Inf. Process. Syst.* **2020** , *51*, 596–608.
6. Tarvainen, A.; Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv* **2017**, arXiv:1703.01780.
7. Xie, Q.; Luong, M.T.; Hovy, E.; Le, Q.V. Self-training with noisy student improves imagenet classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
8. Zhang, B.; Wang, Y.; Hou, W.; Wu, H.; Wang, J.; Okumura, M.; Shinozaki, T. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Adv. Neural Inf. Process. Syst.* **2021** , *13*, 18408–18419.
9. Liu, Y.C.; Ma, C.Y.; He, Z.; Kuo, C.W.; Chen, K.; Zhang, P.; Wu, B.; Kira, Z.; Vajda, P. Unbiased Teacher for Semi-Supervised Object Detection. In Proceedings of the International Conference on Learning Representations, Virtual Event, Austria, 3–7 May 2021.
10. Sohn, K.; Zhang, Z.; Li, C.L.; Zhang, H.; Lee, C.Y.; Pfister, T. A simple semi-supervised learning framework for object detection. *arXiv* **2020**, arXiv:2005.04757.
11. Li, G.; Li, X.; Wang, Y.; Wu, Y.; Liang, D.; Zhang, S. Pseco: Pseudo labeling and consistency training for semi-supervised object detection. In Proceedings of the Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, 23–27 October 2022 , Part IX.
12. Xu, M.; Zhang, Z.; Hu, H.; Wang, J.; Wang, L.; Wei, F.; Bai, X.; Liu, Z. End-to-end semi-supervised object detection with soft teacher. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021.
13. Liu, S.; Zhi, S.; Johns, E.; Davison, A. Bootstrapping Semantic Segmentation with Regional Contrast. In Proceedings of the International Conference on Learning Representations, Virtual Event, 25–29 April 2022.

14. Yun, S.; Han, D.; Oh, S.J.; Chun, S.; Choe, J.; Yoo, Y. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
15. Olsson, V.; Tranheden, W.; Pinto, J.; Svensson, L. Classmix: Segmentation-based data augmentation for semi-supervised learning. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Montreal, BC, Canada, 11–17 October 2021.
16. Goodfellow, I.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
17. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
18. Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; Li, J. Boosting Adversarial Attacks with Momentum. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
19. Liu, Y.; Chen, X.; Liu, C.; Song, D. Delving into transferable adversarial examples and black-box attacks. *arXiv* **2016**, arXiv:1611.02770.
20. Papernot, N.; McDaniel, P.; Goodfellow, I. Transferability in machine learning: From phenomena to black-box attacks using adversarial samples. *arXiv* **2016**, arXiv:1605.07277.
21. Chen, P.Y.; Zhang, H.; Sharma, Y.; Yi, J.; Hsieh, C.J. ZOO: Zeroth Order Optimization Based Black-Box Attacks to Deep Neural Networks without Training Substitute Models. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, Dallas, TX, USA, 3 November 2017.
22. Vivek, B.; Mopuri, K.R.; Babu, R.V. Gray-box adversarial training. In Proceedings of the European conference on computer vision (ECCV), Munich, Germany, 8–14 September 2018.
23. Xu, Y.; Zhong, X.; Yepes, A.J.; Lau, J.H. Grey-box adversarial attack and defence for sentiment classification. *arXiv* **2021**, arXiv:2103.11576.
24. Zanella-Beguelin, S.; Tople, S.; Paverd, A.; Köpf, B. Grey-box extraction of natural language models. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual Event, 18–24 July 2021.
25. Carlini, N.; Wagner, D. Towards Evaluating the Robustness of Neural Networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–24 May 2017.
26. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. Deepfool: A simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
27. Xie, C.; Zhang, Z.; Zhou, Y.; Bai, S.; Wang, J.; Ren, Z.; Yuille, A.L. Improving transferability of adversarial examples with input diversity. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
28. Carlini, N. Poisoning the unlabeled dataset of semi-supervised learning. *arXiv* **2021**, arXiv:2105.01622.
29. Zheng, M.; You, S.; Huang, L.; Wang, F.; Qian, C.; Xu, C. Simmatch: Semi-supervised learning with similarity matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
30. French, G.; Laine, S.; Aila, T.; Mackiewicz, M.; Finlayson, G. Semi-supervised semantic segmentation needs strong, varied perturbations. *arXiv* **2019**, arXiv:1906.01916.
31. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
32. Kurakin, A.; Goodfellow, I.J.; Bengio, S. Adversarial Machine Learning at Scale. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
33. Brendel, W.; Rauber, J.; Bethge, M. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
34. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B.; Swami, A. The limitations of deep learning in adversarial settings. In Proceedings of the 2016 IEEE European symposium on security and privacy (EuroS&P), Saarbrucken, Germany, 21–24 March 2016; IEEE: Piscataway, NJ, USA, 2016.
35. Papernot, N.; McDaniel, P.; Swami, A.; Harang, R. Crafting adversarial input sequences for recurrent neural networks. In Proceedings of the MILCOM 2016—2016 IEEE Military Communications Conference, Baltimore, MD, USA, 1–3 November 2016; IEEE: Piscataway, NJ, USA, 2016.
36. Tramèr, F.; Papernot, N.; Goodfellow, I.; Boneh, D.; McDaniel, P. The space of transferable adversarial examples. *arXiv* **2017**, arXiv:1704.03453.
37. Kannan, H.; Kurakin, A.; Goodfellow, I. Adversarial logit pairing. *arXiv* **2018**, arXiv:1803.06373.
38. Miyato, T.; Maeda, S.i.; Koyama, M.; Ishii, S. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 1979–1993. [CrossRef]
39. Wang, Y.; Zou, D.; Yi, J.; Bailey, J.; Ma, X.; Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 30 April 2020.
40. Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; Jordan, M. Theoretically principled trade-off between robustness and accuracy. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019.

41. Ding, G.W.; Sharma, Y.; Lui, K.Y.C.; Huang, R. MMA Training: Direct Input Space Margin Maximization through Adversarial Training. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 30 April 2020.
42. Rade, R.; Moosavi-Dezfooli, S.M. Reducing Excessive Margin to Achieve a Better Accuracy vs. Robustness Trade-off. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 30 April 2020.
43. Schmidt, L.; Santurkar, S.; Tsipras, D.; Talwar, K.; Madry, A. Adversarially robust generalization requires more data. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 5019–5031.
44. Carmon, Y.; Raghunathan, A.; Schmidt, L.; Duchi, J.C.; Liang, P.S. Unlabeled data improves adversarial robustness. *Adv. Neural Inf. Process. Syst.* **2019** , *32*, 11192–11203.
45. Uesato, J.; Alayrac, J.B.; Huang, P.S.; Stanforth, R.; Fawzi, A.; Kohli, P. Are labels required for improving adversarial robustness? *arXiv* **2019**, arXiv:1905.13725.
46. Zhai, R.; Cai, T.; He, D.; Dan, C.; He, K.; Hopcroft, J.; Wang, L. Adversarially robust generalization just requires more unlabeled data. *arXiv* **2019**, arXiv:1906.00555.
47. Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; Raffel, C. MixMatch: A Holistic Approach to Semi-Supervised Learning. In Proceedings of the NeurIPS, Vancouver, BC, Canada, 8–14 December 2019.
48. Yang, F.; Wu, K.; Zhang, S.; Jiang, G.; Liu, Y.; Zheng, F.; Zhang, W.; Wang, C.; Zeng, L. Class-aware contrastive semi-supervised learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
49. Wallin, E.; Svensson, L.; Kahl, F.; Hammarstrand, L. DoubleMatch: Improving Semi-Supervised Learning with Self-Supervision. In Proceedings of the 2022 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, 21–25 August 2022; IEEE: Piscataway, NJ, USA, 2022.
50. Lee, D.; Kim, S.; Kim, I.; Cheon, Y.; Cho, M.; Han, W.S. Contrastive regularization for semi-supervised learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
51. Kwon, D.; Kwak, S. Semi-supervised semantic segmentation with error localization network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
52. Mi, P.; Lin, J.; Zhou, Y.; Shen, Y.; Luo, G.; Sun, X.; Cao, L.; Fu, R.; Xu, Q.; Ji, R. Active teacher for semi-supervised object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
53. Zhou, H.; Ge, Z.; Liu, S.; Mao, W.; Li, Z.; Yu, H.; Sun, J. Dense teacher: Dense pseudo-labels for semi-supervised object detection. In Proceedings of the Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, 23–27 October 2022 ; Part IX; Springer: Berlin/Heidelberg, Germany, 2022.
54. Li, D.; Li, Q. Adversarial deep ensemble: Evasion attacks and defenses for malware detection. *IEEE Trans. Inf. Forensics Secur.* **2020** , *15*, 3886–3900. [CrossRef]
55. Li, D.; Cui, S.; Li, Y.; Xu, J.; Xiao, F.; Xu, S. PAD: Towards Principled Adversarial Malware Detection Against Evasion Attacks. *IEEE Trans. Dependable Secur. Comput.* **2023** , *1*, 1–16. [CrossRef]
56. Jia, S.; Yin, B.; Yao, T.; Ding, S.; Shen, C.; Yang, X.; Ma, C. Adv-Attribute: Inconspicuous and Transferable Adversarial Attack on Face Recognition. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022.
57. Wang, J.; Zhang, T.; Liu, S.; Chen, P.Y.; Xu, J.; Fardad, M.; Li, B. Adversarial Attack Generation Empowered by Min-Max Optimization. In Proceedings of the Advances in Neural Information Processing Systems, Virtual Event, 6–14 December 2021.
58. Yu, Y.; Gao, X.; Xu, C.Z. MORA: Improving Ensemble Robustness Evaluation with Model Reweighing Attack. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022.
59. Demontis, A.; Melis, M.; Pintor, M.; Jagielski, M.; Biggio, B.; Oprea, A.; Nita-Rotaru, C.; Roli, F. Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks. In Proceedings of the 28th USENIX Security Symposium, Santa Clara, CA, USA, 14–16 August 2019.
60. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015** , *115*, 211–252. [CrossRef]
61. Zagoruyko, S.; Komodakis, N. Wide Residual Networks. In Proceedings of the British Machine Vision Conference (BMVC), York, UK, 19–22 September 2016.
62. Krizhevsky, A.; Hinton, G. Learning Multiple Layers of Features from Tiny Images. Available online: https://www.cs.toronto.edu/kriz/learning-features-2009-TR.pdf (accessed on 8 April 2009).
63. Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A. Reading Digits in Natural Images with Unsupervised Feature Learning. In Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning, Sierra Nevada, Spain, 16–17 December 2011 .
64. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv* **2015**, arXiv:1506.01497.
65. Chen, P.C.; Kung, B.H.; Chen, J.C. Class-aware robust adversarial training for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.
66. Wang, Y.; Wang, K.; Zhu, Z.; Wang, F.Y. Adversarial attacks on faster r-cnn object detector. *Neurocomputing* **2020** , *381*, 87–95. [CrossRef]

67. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
68. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Part V 13, 2014.
69. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012). *Pattern Anal. Stat. Model. Comput. Learn. Tech. Rep.* **2007**, *1–45*, 5.
70. Wang, Y.; Chen, H.; Fan, Y.; Sun, W.; Tao, R.; Hou, W.; Wang, R.; Yang, L.; Zhou, Z.; Guo, L.Z.; et al. USB: A Unified Semi-supervised Learning Benchmark for Classification. In Proceedings of the Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, New Orleans, LA, USA, 2–3 December 2022.
71. Wu, Y.; Kirillov, A.; Massa, F.; Lo, W.Y.; Girshick, R. Detectron2. Available online: https://github.com/facebookresearch/detectron2 (accessed on 15 December 2022).
72. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv* **2019**, arXiv:1906.07155.
73. He, X.; Liu, H.; Gong, N.Z.; Zhang, Y. Semi-Leak: Membership Inference Attacks Against Semi-supervised Learning. In Proceedings of the Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, 23–27 October 2022; Part XXXI; Springer: Berlin/Heidelberg, Germany, 2022.
74. Verma, V.; Lamb, A.; Beckham, C.; Najafi, A.; Mitliagkas, I.; Lopez-Paz, D.; Bengio, Y. Manifold mixup: Better representations by interpolating hidden states. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019.
75. Yin, M.; Li, S.; Song, C.; Asif, M.S.; Roy-Chowdhury, A.K.; Krishnamurthy, S.V. ADC: Adversarial attacks against object Detection that evade Context consistency checks. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 June 2022.
76. Croce, F.; Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual Event, 13–18 July 2020.
77. Yan, Z.; Li, G.; TIan, Y.; Wu, J.; Li, S.; Chen, M.; Poor, H.V. Dehib: Deep hidden backdoor attack on semi-supervised learning via adversarial perturbation. In Proceedings of the AAAI conference on artificial intelligence, Washington DC, USA, 7–14 February 2021.
78. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 $\times$ 16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations, Virtual Event, 3–7 May 2021.
79. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
80. Everingham, M.; Eslami, S.A.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* **2015** , *111*, 98–136. [CrossRef]
81. Zhang, J.; Zhu, J.; Niu, G.; Han, B.; Sugiyama, M.; Kankanhalli, M. Geometry-aware Instance-reweighted Adversarial Training. In Proceedings of the International Conference on Learning Representations, Virtual Event, 3–7 May 2021.