Indirect Invisible Poisoning Attacks on Domain Adaptation

Jun Wu
University of Illinois at Urbana-Champaign
junwu3@illinois.edu

Jingrui He
University of Illinois at Urbana-Champaign
jingrui@illinois.edu

ABSTRACT

Unsupervised domain adaptation has been successfully applied across multiple high-impact applications, since it improves the generalization performance of a learning algorithm when the source and target domains are related. However, the adversarial vulnerability of domain adaptation models has largely been neglected. Most existing unsupervised domain adaptation algorithms might be easily fooled by an adversary, resulting in deteriorated prediction performance on the target domain, when transferring the knowledge from a maliciously manipulated source domain.

To demonstrate the adversarial vulnerability of existing domain adaptation techniques, in this paper, we propose a generic data poisoning attack framework named **I2Attack** for domain adaptation with the following properties: (1) *perceptibly unnoticeable:* all the poisoned inputs are natural-looking; (2) *adversarially indirect:* only source examples are maliciously manipulated; (3) *algorithmically invisible:* both source classification error and marginal domain discrepancy between source and target domains will not increase. Specifically, it aims to degrade the overall prediction performance on the target domain by maximizing the label-informed domain discrepancy over both input feature space and class-label space between source and target domains. Within this framework, a family of practical poisoning attacks are presented to fool the existing domain adaptation algorithms associated with different discrepancy measures. Extensive experiments on various domain adaptation benchmarks confirm the effectiveness and computational efficiency of our proposed **I2Attack** framework.

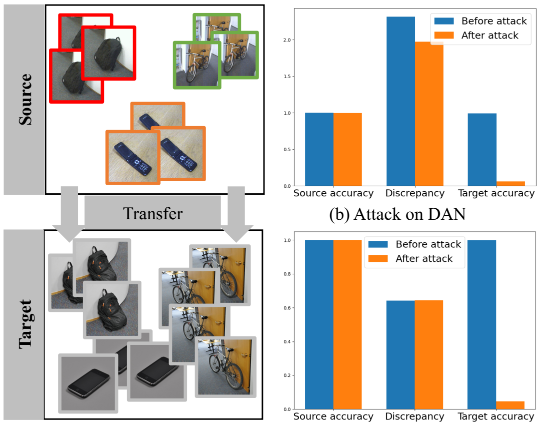
CCS CONCEPTS

• **Theory of computation** → **Adversarial learning**; • **Computing methodologies** → **Transfer learning**.

KEYWORDS

Domain Adaptation, Domain Discrepancy, Poisoning Attack

ACM Reference Format:
Jun Wu and Jingrui He. 2021. Indirect Invisible Poisoning Attacks on Domain Adaptation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3447548.3467214



(a) Images on Office-31

(b) Attack on DAN

(c) Attack on DANN

Figure 1: Illustration of indirect invisible poisoning attacks on DAN [16] and DANN [7] adapting from labeled source domain (i.e., Webcam on Office-31) to unlabeled target domain (i.e., DSLR on Office-31)

1 INTRODUCTION

Unsupervised domain adaptation [21] has been successfully applied across many high-impact applications when the source and the target domains follow similar data distributions. It improves the generalization performance of a learning algorithm under mild assumptions, e.g., the covariate shift assumption [14]. Specifically, conventional domain adaptation theory [1, 18] shows that the classification error on the target domain could be bounded in terms of source classification error and marginal domain discrepancy. This has motivated a line of practical unsupervised domain adaptation algorithms [7, 26] with the objective of minimizing the source classification error and empirical marginal discrepancy across domains (see Subsection 3.2 for a unified view of domain adaptation).

Nevertheless, very little (if any) effort has been devoted to exploring the adversarial vulnerability of existing domain adaptation algorithms [7, 16, 33], especially in the cases where (1) source data is usually publicly available for any potential adversary [2, 32]; (2) recent work [35] argued that under mild conditions, exact marginal distribution matching across domains might lead to negative transfer [30] with undesirable predictive performance on the target domain.

To demonstrate the adversarial vulnerability of existing domain adaptation algorithms [7, 16, 27, 33], in this paper we propose a generic indirect invisible poisoning framework named **I2Attack** for generating the poisoned source data such that existing domain adaptation algorithms could be easily fooled when predicting the target examples. Figure 1 provides an example to show the impact of poisoned source examples learned by our **I2Attack** framework on unsupervised domain adaptation algorithms, e.g., DAN [16] and DANN [7]. It is observed that the classification performance on

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDD '21, August 14–18, 2021, Virtual Event, Singapore
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467214

the target domain deteriorates dramatically without degrading the overall source classification error and empirical marginal domain discrepancy (e.g., multi-kernel maximum mean discrepancy [12] in DAN and $\mathcal{H}$-divergence [1] in DANN). It is worth noting that the empirical domain discrepancy becomes even smaller with poisoned source examples, which implies that the marginal distribution of source and target domains can be better matched after poisoning the source domain.

In particular, we would like to point out that the following properties of our **I2Attack** framework would make the poisoning attacks more destructive in real scenarios. ***(P1) Perceptibly unnoticeable:*** all the poisoned inputs are perceptibly indistinguishable from real inputs by adding the carefully chosen adversarial noise [11]. ***(P2) Adversarially indirect:*** only the source domain is maliciously manipulated because source data tends to be publicly available to the adversary in real scenarios; ***(P3) Algorithmically invisible:*** the loss terms of a domain adaptation algorithm (e.g., source classification error and marginal discrepancy across domains) under **I2Attack** would not increase significantly compared to learning from the clean source and target data, thus making the adversarial attacks difficult to notice during model training.

The most similar line of work is the adversarial robustness on fine-tuning based transfer learning algorithms [22, 25, 29, 34]. However, our problem setting fundamentally differs from them in the following aspects. (1) We study the unsupervised domain adaptation without labeled training examples from the target domain, while previous works require some labeled target examples for fine-tuning during model training; (2) We aim to explore the adversarial vulnerability with poisoning attacks by manipulating the source training examples in the training phase, whereas previous ones focus on performing the evasion attacks by generating adversarial examples for a pre-trained model in the test phase; (3) We constrain our attacks to be indirect and invisible, whereas this is not taken into consideration in previous works. The main contributions of this paper are summarized as follows:

- We formulate a novel indirect invisible poisoning attack problem for analyzing the adversarial vulnerability of existing unsupervised domain adaptation algorithms.
- A generic framework **I2Attack** is proposed for degrading the overall performance on the target domain, followed by a family of instantiated poisoning attack algorithms.
- Extensive experiments on publicly accessible domain adaptation benchmarks demonstrate the effectiveness and efficiency of our proposed **I2Attack**[1] framework.

The rest of the paper is organized as follows. We review the related work in Section 2. In Section 3, we present our problem definition on the adversarial vulnerability of domain adaptation. We propose a generic indirect invisible poisoning attack framework **I2Attack** in Section 4, followed by a family of instantiated poisoning attacks in Section 5. The experiments are provided in Section 6. Finally, we conclude the paper in Section 7.

2 RELATED WORK

In this section, we briefly introduce the related work on adversarial machine learning and domain adaptation.

2.1 Adversarial Machine Learning

It has been observed that modern neural network models can be easily fooled by the adversarial examples that are perceptibly indistinguishable with respect to the clean inputs [11]. The adversarial robustness of machine learning models [3, 15, 24] has been explored with the assumption that training and test data follow the same distribution. In particular, poisoning adversarial attacks aim to manipulate the training process by injecting carefully crafted examples, with the goals of either reducing the overall predictive performance of a learning algorithm [2, 19] or controlling the model behavior on some specific test examples without degrading the overall predictive performance [23, 40]. However, our problem setting is fundamentally different in the following aspects: (1) our poisoning attacks are explored under the distribution shift across domains; (2) the goal of our poisoning attacks is to degrade the overall predictive performance for test examples (from the target domain) without affecting the training process (e.g., training loss).

2.2 Domain Adaptation

Unsupervised domain adaptation [4, 21, 38] aims to improve the predictive performance on the target domain with only unlabeled training examples by transferring the knowledge from a related source domain with adequate labeled training examples. The domain adaptation theory [1, 18, 33] argues that the target error is bounded in terms of source error and discrepancy across domains. This has motivated a line of practical algorithms [7, 17, 26, 28] by minimizing the marginal domain discrepancy and source classification error. However, recent work [35] demonstrated that exact marginal distribution matching across domains might lead to negative transfer with undesirable performance on the target domain. This might allow the adversary to fool the existing unsupervised domain adaptation algorithms by maliciously manipulating the source data. A similar line of work to us is the adversarial robustness of fine-tuning based transfer learning [22, 25, 29, 34] with adequate labeled source examples and limited labeled target examples. To the best of our knowledge, this is the first work aiming at studying the adversarial vulnerability of unsupervised domain adaptation with no labeled training examples from the target domain.

3 PRELIMINARIES

In this section, we derive a unified view of unsupervised domain adaptation, followed by our problem definition on data poisoning attacks to domain adaptation.

3.1 Notation

Let $\mathcal{X}$ and $\mathcal{Y}$ denote the input feature space and output label space. We denote $\mathbb{Q}, \mathbb{P}$ to be the source and target domains associated with data distributions $\mathbb{Q}_{XY}, \mathbb{P}_{XY}$ over $\mathcal{X} \times \mathcal{Y}$, respectively. The source and target marginal distributions over $\mathcal{X}$ are denoted as $\mathbb{Q}_X$ and $\mathbb{P}_X$, respectively. We let $l_{\mathbb{Q}}$ and $l_{\mathbb{P}}$ denote the labeling functions of the source and target domains. In this paper, we consider the unsupervised domain adaptation setting where there are n_s labeled training examples $\{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ from the source domain and n_t unlabeled training examples $\{x_j^t\}_{j=1}^{n_t}$ from the target domain. Let $\mathcal{H}$ be the hypothesis class on $\mathcal{X}$ where a hypothesis is a function $h : \mathcal{X} \to \mathcal{Y}$. $L(\cdot, \cdot)$ is the loss function such that $L : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. The distribution shift between the source and target domains can be

[1] The source code is available at: **https://github.com/jwu4sml/I2Attack**.

measured by the domain discrepancy $d(\cdot,\cdot)$, e.g., $\mathcal{H}$-divergence [1], discrepancy distance [18], etc.

3.2 A Unified View of Domain Adaptation

Unsupervised domain adaptation [21] refers to the knowledge transfer from the source domain with adequate labeled training data to the target domain with no labeled training data. The following theorem [1] argued that under the covariate shift assumption (i.e., $l_{\mathbb{Q}}(x) = l_{\mathbb{P}}(x)$ for any $x \in X$), the target error is bounded by the expected source error and marginal domain discrepancy between source and target domains.

Theorem 3.1. *Let $\mathcal{H}$ be the hypothesis space and $\epsilon_s(h), \epsilon_t(h)$ be the expected classification error of a hypothesis $h \in \mathcal{H}$ on the source and target domains, respectively. Then for any hypothesis $h \in \mathcal{H}$,*

$$\epsilon_t(h) \le \epsilon_s(h) + d_1(\mathbb{Q}_X, \mathbb{P}_X) + \min\left\{\mathbb{E}_{x\sim\mathbb{Q}_X}\left[\left|l_{\mathbb{Q}}(x) - l_{\mathbb{P}}(x)\right|\right], \mathbb{E}_{x\sim\mathbb{P}_X}\left[\left|l_{\mathbb{Q}}(x) - l_{\mathbb{P}}(x)\right|\right]\right\}$$

where $\epsilon_s(h) = \mathbb{E}_{(x,y)\sim Q}[L(h(x), y)]$ and $d_1(\mathbb{Q}_X, \mathbb{P}_X)$ is the variation divergence between source and target domains[2].

Remark. *It is observed that the variation divergence $d_1(\mathbb{Q}_X, \mathbb{P}_X)$ has the following limitations [1, 18]: (1) it cannot be accurately estimated from finite samples; (2) it provides the relatively loose error bound when considering all the measurable subsets in the feature space. To address these problems, various domain discrepancy measures have been proposed, including $\mathcal{H}$-divergence [1, 7], discrepancy distance [18], Maximum Mean Discrepancy (MMD) [16, 17], Wasserstein distance [26], covariances distance [27], Margin Disparity Discrepancy (MDD) [33], etc.*

Following Theorem 3.1, we provide a simple unified view of unsupervised domain adaptation as follows.

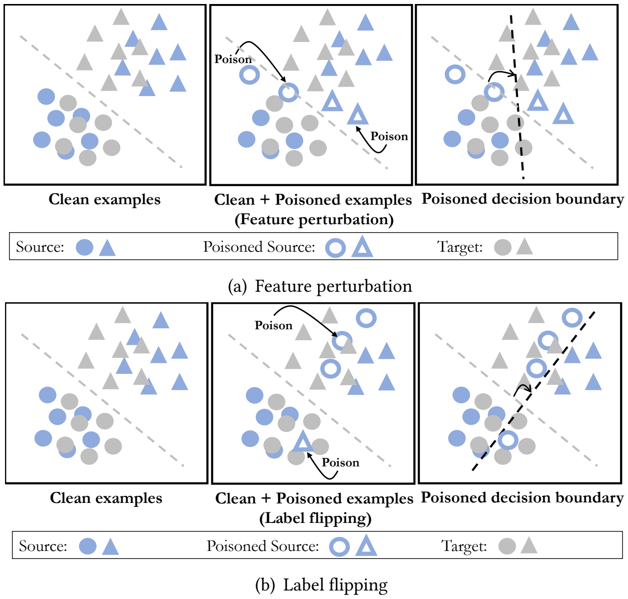
$$\min_{\theta,\phi} \frac{1}{n_s} \sum_{i=1}^{n_s} L\left(h_\phi\left(f_\theta(x_i^s)\right), y_i^s\right) + d\left(\mathbb{Q}_X, \mathbb{P}_X; \theta\right) \tag{1}$$

where $f_\theta(\cdot)$ is the feature extractor function parameterized by θ and $h_\phi(\cdot)$ is the classifier function parameterized by ϕ, and $d(\cdot,\cdot;\theta)$ denotes a generic hypothesis-dependent domain discrepancy measured in the feature space. It aims to empirically minimize the upper error bound in Theorem 3.1 associated with the classification error on the source domain and the marginal domain discrepancy across domains, under the strong assumption that both domains share the same labeling function. Many popular domain adaptation algorithms could be fitted into the objective function in Eq. (1), e.g., CORAL [27, 28], DAN [16], DANN [7], MDD [33], etc.

3.3 Problem Definition

Formally, our problem setting could be defined as follows.

Definition 3.2. *Given a source domain with labeled examples $\{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ and a target domain with unlabeled examples $\{x_j^t\}_{j=1}^{n_t}$, the* ***indirect and invisible data poisoning attack*** *aims to degrade the overall classification performance of a domain adaptation algorithm on the target domain, and meanwhile, it satisfies the following three conditions: (i)* ***imperceptible:*** *poisoned inputs are perceptibly indistinguishable from real inputs; (ii)* ***indirect:*** *only source domain examples are manipulated; (iii)* ***invisible:*** *both source classification error and marginal domain discrepancy will not increase.*



(a) Feature perturbation

(b) Label flipping

Figure 2: Two examples of poisoning attacks on unsupervised domain adaptation where the decision boundary of a learning algorithm becomes much worse after perturbing raw features or flipping class-labels on source examples

As shown in Figure 2, we provide two motivating scenarios to explain how existing unsupervised domain adaptation algorithms could be attacked: (i) perturb the source examples only by adding the adversarial noise to their raw feature (see Figure 3(a)); (ii) carefully flip the labels of some source examples (see Figure 3(b)). Notice that both of them focus on manipulating the source examples only (***indirect attacks***) while preserving the discrimination of source examples and marginal domain discrepancy across domains (***invisible attacks***). In this paper, we will focus on the first scenario on adding the adversarial noise to raw inputs, and leave the analysis of the second scenario regarding label flipping as our future work[3].

4 A GENERIC FRAMEWORK

We have derived a unified view of unsupervised domain adaptation (see Eq. (1)) based on domain adaptation theory [1]. The intuition is that it would learn a common feature space such that source and target distributions could be well matched and source examples are class-separable in the feature space. However, the exact matching of marginal data distribution across domains might lead to negative transfer with undesirable performance on the target domain [35]. This motivates us to develop the data poisoning attacks on existing unsupervised domain adaptation algorithms by maliciously manipulating the relatedness between source and target domains.

4.1 Overall Goal

The overall goal of our poisoning attacks is to inject the adversarial noise into the source data in the training phase such that the overall prediction performance of most existing unsupervised domain

[2] $d_1(\mathbb{Q}_X, \mathbb{P}_X) = 2\sup_{B\in\mathcal{B}} |\mathbb{Q}_X[B] - \mathbb{P}_X[B]|$ where $\mathcal{B}$ is the set of measurable subsets under $\mathbb{Q}_X$ and $\mathbb{P}_X$.

[3] Note that label flipping is also powerful for generating poisoned source examples, but much more challenging due to discrete representations of data class-labels.

adaptation algorithms [7, 16, 27, 33] on the target domain could be largely deteriorated. Conventional poisoning attacks [2, 19] on single-domain classification could be applied to degrade the domain adaptation performance by enforcing the source examples to be non-separable in the feature space. But in this case, the training loss (e.g., source classification error) would significantly increase, and thus such attacks can be easily noticed in the training phase.

To solve this problem, we develop the indirect and invisible poisoning attacks such that both source classification error and marginal domain discrepancy across domains would not increase significantly during model training. In our work, we assume that the adversary has the full knowledge about the source training data and the learning algorithm (e.g., model architecture, hyper-parameters, etc.) for domain adaptation. The adversary might have either full or no knowledge of unlabeled training data from the target domain[4]. Besides, in order to enforce the adversarial attacks to be perceptibly unnoticeable, it requires to produce the poisoned example $\hat{x}$ with respect to the input x under a reasonable perturbation constraint $\Omega(x)$, i.e., $\hat{x} \in \Omega(x)$. In this paper, we will consider the commonly used constraint $\Omega(x) := \{\hat{x} | ||\hat{x} - x||_\infty \leq \epsilon\}$ for a perturbation magnitude ϵ in image classification [11].

4.2 I2Attack

In this paper, we propose a generic indirect and invisible poisoning attack framework named **I2Attack** on unsupervised domain adaptation. For notation brevity, we denote $X_s = \{x_i^s\}_{i=1}^{n_s}$ be the raw source examples associated with class labels $Y_s = \{y_i^s\}_{i=1}^{n_s}$, and $\hat{X}_s = \{\hat{x}_i^s\}_{i=1}^{n_s}$ be the poisoned source examples (associated with unchanged class labels $Y_s = \{y_i^s\}_{i=1}^{n_s}$). The overall objective function could be mathematically formulated as the following bi-level optimization problem:

$$\begin{aligned}
&\max_{\hat{X}_s} O(\hat{X}_s, X_s, Y_s; \theta^*, \phi^*) \\
\text{s.t.,}\quad & \theta^*, \phi^* = \arg\min_{\theta,\phi} L\left(h_\phi\left(f_\theta(\hat{X}_s)\right), Y_s\right) + d\left(f_\theta(\hat{X}_s), f_\theta(X_t)\right) \\
\text{s.t.,}\quad & d(f_{\theta^*}(\hat{X}_s), f_{\theta^*}(X_s)) \leq \delta_1 \\
\text{s.t.,}\quad & L\left(h_{\phi^*}\left(f_{\theta^*}(\hat{X}_s)\right), Y_s\right) \leq \delta_2 \\
\text{s.t.,}\quad & \hat{X}_s \in \Omega(X_s)
\end{aligned} \tag{2}$$

where $d\left(f_\theta(\hat{X}_s), f_\theta(X_t)\right)$ is the domain discrepancy across domains in the feature space learned by $f_\theta(\cdot)$. $O(\cdot)$ is the attacking function (see Subsection 4.3) for learning the poisoned source examples. Here $\delta_1 \geq 0$ constraints the marginal domain discrepancy between poisoned source domain and clean source domain, and $\delta_2 \geq 0$ constraints the classification error on the poisoned source domain. In this case, the adversary poisons the source data under the following conditions: (i) the model parameters θ and ϕ are optimal with respect to the poisoned source domain and raw target domain; (ii) the last three constraints guarantee that the poisoning attacks are perceptibly unnoticeable and algorithmically invisible.

If the discrepancy measure $d(\cdot, \cdot)$ satisfies the triangle inequality property, it holds that $d(f_\theta(\hat{X}_s), f_\theta(X_t)) \leq d(f_\theta(\hat{X}_s), f_\theta(X_s)) +$ $d(f_\theta(X_s), f_\theta(X_t)) \leq \delta_1 + d(f_\theta(X_s), f_\theta(X_t))$. That is, the minimization of $d(f_{\theta^*}(\hat{X}_s), f_{\theta^*}(X_s))$ on the source domain encourages the poisoning attacks to preserve the marginal domain discrepancy between source and target domains. Therefore, the constraints $d(f_{\theta^*}(\hat{X}_s), f_{\theta^*}(X_s)) \leq \delta_1$ and $L(h_{\phi^*}(f_{\theta^*}(\hat{X}_s)), Y_s) \leq \delta_2$ ensure that the source classification error and domain discrepancy will not be significantly affected, thus leading to the algorithmically invisible adversarial attacks.

4.3 Attacking Function

It is observed that minimizing the marginal data distribution and source classification error in unsupervised domain adaptation algorithms could lead to negative transfer with undesirable predictive performance on the target domain [35]. Moreover, the following theorem shows that for any target domain, there exists a source domain satisfying that it is class-separable and has identical marginal data distribution with the target domain over the input space $\mathcal{X}$, such that the target error of the optimal hypothesis $h \in \mathcal{H}$ is large.

Theorem 4.1. *Let ϵ_s and ϵ_t denote the expected source and target classification error. Given any class-separable target domain $\mathbb{P}$, there exist at least one source domain $\mathbb{Q}$ and $h \in \mathcal{H}$ satisfying $\epsilon_s(h) = 0$ and $d(\mathbb{Q}_X, \mathbb{P}_X) = 0$, such that the target classification error $\epsilon_t(h) = 1$.*

Furthermore, the following corollary provides the insight into designing the source domain for maximizing the target error while minimizing the source error and marginal domain discrepancy.

Corollary 4.2. *Let ϵ_s and ϵ_t denote the expected source and target classification error. For any class-separable target domain $\mathbb{P}$, there exists a source domain $\mathbb{Q}$ and $h \in \mathcal{H}$ such that $\epsilon_s(h) = 0$, $d(\mathbb{Q}_X, \mathbb{P}_X) = 0$ and $\epsilon_t(h) = 1$ if it satisfies one of the following conditions: (i) $\mathbb{Q}_{XY}(x, y = i) = \mathbb{P}_{XY}(x, y = j)$; (ii) $\mathbb{Q}_{Y|X}(y = i|x) = \mathbb{P}_{Y|X}(y = j|x)$; (iii) $\mathbb{Q}_{X|Y}(x|y = i) = \mathbb{P}_{X|Y}(x|y = j)$, for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and $i \neq j$.*

Corollary 4.2 indicates that the malicious poisoned source domain can be learned by maximizing the label-informed domain discrepancy [31], e.g., joint distribution over $\mathcal{X} \times \mathcal{Y}$, feature-conditional distribution over $\mathcal{Y}|\mathcal{X}$ and class-conditional distribution over $\mathcal{X}|\mathcal{Y}$. Therefore, we have three options to design the attacking function $O(\hat{X}_s, X_s, Y_s; \theta^*, \phi^*)$ in Eq. (2) as follows.

$$O(\hat{X}_s, X_s, Y_s; \theta^*, \phi^*) = d\left(\hat{X}_s \circ Y_s, X_s \circ Y_s; \theta^*, \phi^*\right) \tag{3}$$

$$O(\hat{X}_s, X_s, Y_s; \theta^*, \phi^*) = d\left(Y_s|\hat{X}_s, Y_s|X_s; \theta^*, \phi^*\right) \tag{4}$$

$$O(\hat{X}_s, X_s, Y_s; \theta^*, \phi^*) = d\left(\hat{X}_s|Y_s, X_s|Y_s; \theta^*, \phi^*\right) \tag{5}$$

where $\circ$ is the combination of input feature and output class label over $\mathcal{X} \times \mathcal{Y}$. Please note that the feature-conditional distribution might not be tractable to be estimated explicitly from finite examples. Thus, in this paper, we focus on the attacking functions based on joint distribution and class-conditional distribution. We will instantiate the model-specific attacking functions in Section 5.

4.4 Discussion

We see that the proposed framework **I2Attack** has the following advantages in analyzing the adversarial vulnerability of unsupervised domain adaptation algorithms. (1) **Flexible:** it is flexible to be instantiated for attacking any discrepancy-based domain adaptation

[4] Note that when the adversary has no knowledge of target domain, it might require an auxiliary target domain for generating the poisoned source data (see Subsection 5.2 for model analysis and Subsection 6.3 for empirical evaluation of **I2Attack** framework).

Table 1: Summary of domain adaptation algorithms

Algorithm	Feature extractor	Classifier	Discrepancy
CORAL [27]	Linear mapping	SVM	Covariance
DAN [16]	CNN	MLP	MK-MMD
DANN [7]	CNN	MLP	$\mathcal{H}$-divergence
MDD [33]	CNN	MLP	Margin disparity

algorithm (see next section), especially when there are no available labeled training examples in the target domain; (2) **Unnoticeable:** the properties of indirect and invisible attacks would make it difficult to be noticed in the training phase, thus posing a significant threat to public source domain data in real scenarios; (3) **Interpretable:** the label-informed data distribution over either $\mathcal{X} \times \mathcal{Y}$ or $\mathcal{X}|\mathcal{Y}$ between source and target domains is maximized in **I2Attack**, such that the existing domain adaptation algorithms [7, 16, 27, 33] with matched marginal distributions over $\mathcal{X}$ across domains leads to negative transfer [21] on the target domain without affecting the training loss.

5 THE PROPOSED ALGORITHMS

In this section, we instantiate our framework **I2Attack** for attacking the unsupervised domain adaptation algorithm, followed by model analysis from various aspects.

5.1 Poisoning Attack Algorithms

As shown in Eq. (1), a typical domain adaptation algorithm aims to minimize the source classification error and marginal domain discrepancy. Specifically, most of the existing domain adaptation algorithms can be divided into the following two categories: (i) two-stage framework [27], which first learns a domain-invariant feature space to minimize the marginal domain discrepancy over $\mathcal{X}$ and then trains the classifier in the learned feature space; (ii) unified framework [7, 16, 33] that minimizes both source classification error and marginal domain discrepancy in an end-to-end manner. Table 1 summarizes how those works fit into the objective function of Eq. (1) with different feature extractors, classifiers and domain discrepancy measures. Note that here we design the attacking function $O(\cdot)$ by maximizing the joint domain discrepancy across domains over $\mathcal{X} \times \mathcal{Y}$, but it can be naturally substituted with class-conditional domain discrepancy over $\mathcal{X}|\mathcal{Y}$.

5.1.1 **I2Attack-CORAL**.

CORAL [27] states that domain discrepancy could be measured by the second-order statistics (covariance matrix) of source and target examples after feature normalization. Thus it proposed to learn a linear transformation A to align the source and target distributions as follows.

$$\min_{A} \left\| A^T C_s^X A - C_t^X \right\|_F^2$$

where $C_s^X = \frac{1}{n_s - 1}(X_s^T X_s - \frac{1}{n_s}(\mathbf{1}^T X_s)^T(\mathbf{1}^T X_s))$ and $C_t^X = \frac{1}{n_t - 1}(X_t^T X_t - \frac{1}{n_t}(\mathbf{1}^T X_t)^T(\mathbf{1}^T X_t))$ are covariance matrices of source and target domains over $\mathcal{X}$, respectively, and $\mathbf{1}$ is a column vector with all elements equal to 1. In this case, the input example is a m-dimensional feature vector, i.e., $X_s \in \mathbb{R}^{n_s \times m}, X_t \in \mathbb{R}^{n_t \times m}$. It only involves the first stage of domain adaptation on distribution matching. After that, the classifier (e.g., SVM, kNN) can be trained using the transformed source examples $X_s A$.

Algorithm 1 Indirect Invisible Poisoning Attack (**I2Attack**)

```
Input: Source examples $(X_s, Y_s)$ and target examples $X_t$, base domain adaptation algorithm with discrepancy measure $d(\cdot, \cdot)$, perturbation magnitude $\epsilon$.
Output: Poisoned source examples $\hat{X}_s$.
Initialize $\hat{X}_s \in \Omega(X_s)$ and base model parameters $\theta, \phi$.
while Stopping criterion is not satisfied do
    for $l = 1, \cdots, L$ do
        Update base model parameters $\theta, \phi$ using Eq. (9).
    end for
    Estimate meta-gradient $\nabla_{\hat{X}_s}^{meta} \mathcal{J}(\hat{X}_s; \theta^*, \phi^*)$ using Eq.(10).
    Update poisoned source examples $\hat{X}_s$ using Eq. (8).
end while
return Poisoned source examples $\hat{X}_s$.
```

Following Eq. (2), the data poisoning attack on CORAL can be formulated as the following bi-level optimization problem:

$$\max_{||\hat{X}_s - X_s||_\infty \leq \epsilon} \left\| A_*^T \hat{C}_s^{XY} A_* - C_s^{XY} \right\|_F^2$$

$$\text{s.t. } A_* = \arg\min_{A} \left\| A^T \hat{C}_s^X A - C_t^X \right\|_F^2 \text{ and } \left\| A_*^T \hat{C}_s^X A_* - C_s^X \right\|_F^2 \leq \delta_1$$

where $\hat{C}_s^{XY}$ and C_s^{XY} are covariance matrices over joint distribution of input feature and output label.

$$\hat{C}_s^{XY} = \frac{[\hat{X}_s \circ Y_s]^T [\hat{X}_s \circ Y_s] - \frac{1}{n_s}\left(\mathbf{1}^T [\hat{X}_s \circ Y_s]\right)^T \left(\mathbf{1}^T [\hat{X}_s \circ Y_s]\right)}{n_s - 1}$$

where $\circ$ is the vector concatenation operator over feature vector and label vector. It is shown [27] that the optimal transformation A_* of inner minimization problem could be given by $A_* = (\hat{C}_s^X + I)^{-1/2}(C_t^X + I)^{1/2}$. Therefore, it can be naturally transformed into a single-level optimization problem: $\max_{\hat{X}_s} \left\| A_*^T \hat{C}_s^{XY} A_* - C_s^{XY} \right\|_F^2 - \mu \left\| A_*^T \hat{C}_s^X A_* - C_s^X \right\|_F^2$ with box constraint $||\hat{X}_s - X_s||_\infty \leq \epsilon$ where μ is a constant hyper-parameter. This optimization problem can then be solved by stochastic gradient descent (SGD) [11].

5.1.2 **I2Attack-DAN**.

Deep Adaptation Network [16] (DAN) learns the domain-invariant feature representation in a reproducing kernel Hilbert space where the mean embeddings of different domain distributions are explicitly matched as follows.

$$\min_{\theta, \phi} L\left(h_\phi\left(f_\theta(X_s)\right), Y_s\right) + d_k(f_\theta(X_s), f_\theta(X_t)) \tag{6}$$

where $d_k(\cdot, \cdot)$ represents the empirical multi-kernel maximum mean discrepancy [12] (MK-MMD) between source and target domains in the feature space learned by $f_\theta(\cdot)$.

Following Eq. (2), we propose to learn the poisoned source examples with the following bi-level optimization problem:

$$\max_{\hat{X}_s} d_k(f_{\theta^*}(\hat{X}_s) \circ Y_s, f_{\theta^*}(X_s) \circ Y_s)$$

$$\text{s.t., } \quad \theta^*, \phi^* = \arg\min_{\theta, \phi} L\left(h_\phi\left(f_\theta(\hat{X}_s)\right), Y_s\right) + d_k\left(f_\theta(\hat{X}_s), f_\theta(X_t)\right)$$

$$\text{s.t., } \quad d_k(f_{\theta^*}(\hat{X}_s), f_{\theta^*}(X_s)) \leq \delta_1$$

$$\text{s.t., } \quad L\left(h_{\phi^*}\left(f_{\theta^*}(\hat{X}_s)\right), Y_s\right) \leq \delta_2$$

$$\text{s.t.,}\quad \hat{X}_s \in \Omega(X_s) \tag{7}$$

We tackle this bi-level optimization problem using model-agnostic meta-learning (MAML) [6] that aims to find appropriate hyper-parameter configurations (e.g., model initialization, learning rate schedules, etc.) of neural networks. In this case, we can consider the poisoned source examples $\hat{X}_s$ as the hyper-parameters of a domain adaptation algorithm and then optimize Eq. (7) using the meta-gradient of attacking function with respect to $\hat{X}_s$ as follows.

$$\hat{X}_s \leftarrow \text{Proj}_{\Omega(X_s)}\left(\hat{X}_s - \alpha\nabla^{meta}_{\hat{X}_s}\mathcal{J}(\hat{X}_s;\theta^*,\phi^*)\right) \tag{8}$$

where $\text{Proj}_{\Omega(X_s)}(\cdot)$ projects the updated poisoned input onto $\Omega(X_s)$ in every iteration, and $\mathcal{J}(\hat{X}_s;\theta^*,\phi^*) = d_k(f_{\theta^*}(\hat{X}_s)\circ Y_s, f_{\theta^*}(X_s)\circ Y_s) - \mu(d_k(f_{\theta^*}(\hat{X}_s), f_{\theta^*}(X_s)) + L(h_{\phi^*}(f_{\theta^*}(\hat{X}_s)), Y_s))$ is the meta attacking function. In particular, θ^*, ϕ^* can be learned with vanilla gradient descent as follows.

$$\begin{aligned}\theta^{l+1} &= \theta^l - \beta\nabla_{\theta^l}\left(L(h_{\phi^l}(f_{\theta^l}(\hat{X}_s)), Y_s) + d_k(f_{\theta^l}(\hat{X}_s), f_{\theta^l}(X_t))\right)\\ \phi^{l+1} &= \phi^l - \beta\nabla_{\phi^l}L(h_{\phi^l}(f_{\theta^l}(\hat{X}_s)), Y_s)\end{aligned} \tag{9}$$

where l is the iteration index. With L updates of gradient descent on θ and ϕ, following the first-order approximation of meta-gradient (FO-MAML) proposed in [6], we have

$$\begin{aligned}&\nabla^{meta}_{\hat{X}_s}\mathcal{J}(\hat{X}_s;\theta^*,\phi^*) \approx \nabla^{meta}_{\hat{X}_s}\mathcal{J}(\hat{X}_s;\theta^L,\phi^L)\\ =&\nabla_f\mathcal{J}(\hat{X}_s;\theta^L,\phi^L)[\nabla_{\hat{X}_s}f_{\theta^L}(\hat{X}_s) + \nabla_{\theta^L}f_{\theta^L}(\hat{X}_s)\nabla_{\hat{X}_s}\theta^L]\\ +&\nabla_h\mathcal{J}(\hat{X}_s;\theta^L,\phi^L)[\nabla_{\hat{X}_s}h_{\phi^L}(f_{\theta^L}(\hat{X}_s)) + \nabla_{\phi^L}h_{\phi^L}(f_{\theta^L}(\hat{X}_s))\nabla_{\hat{X}_s}\phi^L]\\ \approx&\nabla_f\mathcal{J}(\hat{X}_s;\theta^L,\phi^L)\nabla_{\hat{X}_s}f_{\theta^L}(\hat{X}_s) + \nabla_h\mathcal{J}(\hat{X}_s;\theta^L,\phi^L)\nabla_{\hat{X}_s}h_{\phi^L}(f_{\theta^L}(\hat{X}_s))\end{aligned} \tag{10}$$

Then the approximated meta-gradient $\nabla^{meta}_{\hat{X}_s}\mathcal{J}(\hat{X}_s;\theta^*,\phi^*)$ can be used to update the poisoned input $\hat{X}_s$ in Eq. (8). The overall training procedures are illustrated in Algorithm 1. The algorithm iteratively updates the poisoned source examples and stops when the user-defined stopping criterions are satisfied.

5.1.3 **I2Attack-DANN**. Inspired by Generative Adversarial Network (GAN) [10, 36, 37, 39], Domain-Adversarial Neural Network [7] (DANN) learns the domain-invariant latent feature space in an adversarial manner where the $\mathcal{H}$-divergence across domains and the source classification error could be minimized in the feature space. The overall objective function of DANN is given below.

$$\min_{\theta,\phi} L\left(h_\phi\left(f_\theta(X_s)\right), Y_s\right) + d_{\mathcal{H}}(f_\theta(X_s), f_\theta(X_t)) \tag{11}$$

where $d_{\mathcal{H}}(\cdot,\cdot)$ is the $\mathcal{H}$-divergence between source and target domains in the feature space learned by $f_\theta(\cdot)$. Mathematically,

$$\begin{aligned}d_{\mathcal{H}}(f_\theta(X_s), f_\theta(X_t) = \min_\theta\max_D\ &\mathbb{E}_{x^s\sim X_s}\left[D(f_\theta(x^s))\right]\\ +&\mathbb{E}_{x^t\sim X_t}\left[1 - D(f_\theta(x^t))\right]\end{aligned} \tag{12}$$

where $D(\cdot)$ is a domain discriminator for identifying which domain an example comes from.

Following Eq. (2), we propose to generate the poisoning attacks with the following quad-level optimization problem:

$$\begin{aligned}&\max_{\hat{X}_s} d_{\mathcal{H}}(f_{\theta^*}(\hat{X}_s)\circ Y_s, f_{\theta^*}(X_s)\circ Y_s)\\ &\text{s.t.,}\quad \theta^*,\phi^*,D^* = \arg\min_{\theta,\phi} L\left(h_\phi\left(f_\theta(\hat{X}_s)\right), Y_s\right) + d_{\mathcal{H}}\left(f_\theta(\hat{X}_s), f_\theta(X_t)\right)\\ &\text{s.t.,}\quad d_{\mathcal{H}}(f_{\theta^*}(\hat{X}_s), f_{\theta^*}(X_s)) \le \delta_1\\ &\text{s.t.,}\quad L\left(h_{\phi^*}\left(f_{\theta^*}(\hat{X}_s)\right), Y_s\right) \le \delta_2\\ &\text{s.t.,}\quad \hat{X}_s \in \Omega(X_s)\end{aligned} \tag{13}$$

Solving such an optimization problem is challenging due to its high-order combinatorial nature. It is observed that $\mathcal{H}$-divergence could be upper bounded by maximum mean discrepancy [9]. Thus we would like to derive an efficient approximation by substituting the domain discrepancy $d_{\mathcal{H}}(\cdot,\cdot)$ of attacking function with MK-MMD $d_k(\cdot,\cdot)$ as follows.

$$\begin{aligned}&\max_{\hat{X}_s} d_k(f_{\theta^*}(\hat{X}_s)\circ Y_s, f_{\theta^*}(X_s)\circ Y_s)\\ &\text{s.t.,}\quad \theta^*,\phi^*,D^* = \arg\min_{\theta,\phi} L\left(h_\phi\left(f_\theta(\hat{X}_s)\right), Y_s\right) + d_{\mathcal{H}}\left(f_\theta(\hat{X}_s), f_\theta(X_t)\right)\\ &\text{s.t.,}\quad d_k(f_{\theta^*}(\hat{X}_s), f_{\theta^*}(X_s)) \le \delta_1\\ &\text{s.t.,}\quad L\left(h_{\phi^*}\left(f_{\theta^*}(\hat{X}_s)\right), Y_s\right) \le \delta_2\\ &\text{s.t.,}\quad \hat{X}_s \in \Omega(X_s)\end{aligned} \tag{14}$$

Specially, the inner parameters θ^*, ϕ^*, D^* of DANN can be trained using standard backpropagation with gradient reversal layer [7]. On top of this observation, the overall optimization problem of Eq. (14) can then be efficiently solved via meta-learning [6] when adopting vanilla gradient descent to update the inner parameters.

5.1.4 **I2Attack-MDD**. Margin Disparity Discrepancy [33] (MDD) minimizes the empirical source classification error and the disparity discrepancy across domains in the feature space using the following objective function.

$$\min_{\theta,\phi} L\left(h_\phi\left(f_\theta(X_s)\right), Y_s\right) + d_{MDD}(f_\theta(X_s), f_\theta(X_t)) \tag{15}$$

where $d_{MDD}(\cdot,\cdot)$ denotes the margin-aware disparity discrepancy between source and target domains and can be empirically minimized using a minimax adversarial game in the feature space learned by $f_\theta(\cdot)$.

Similar to **I2Attack-DANN**, we design the attacking function with non-adversarial domain discrepancy $d_k(\cdot,\cdot)$ for efficiently generating the poisoning attacks on MDD algorithm as follows.

$$\begin{aligned}&\max_{\hat{X}_s} d_k(f_{\theta^*}(\hat{X}_s)\circ Y_s, f_{\theta^*}(X_s)\circ Y_s)\\ &\text{s.t.,}\theta^*,\phi^*,D^* = \arg\min_{\theta,\phi} L\left(h_\phi\left(f_\theta(\hat{X}_s)\right), Y_s\right) + d_{MDD}\left(f_\theta(\hat{X}_s), f_\theta(X_t)\right)\\ &\text{s.t.,}d_k(f_{\theta^*}(\hat{X}_s), f_{\theta^*}(X_s)) \le \delta_1\\ &\text{s.t.,}L\left(h_{\phi^*}\left(f_{\theta^*}(\hat{X}_s)\right), Y_s\right) \le \delta_2\\ &\text{s.t.,}\hat{X}_s \in \Omega(X_s)\end{aligned} \tag{16}$$

It can then be optimized with meta-gradient based updating method derived in Subsection 5.1.2.

5.2 Model Analysis

Transferability and Universalness of I2Attack: In the previous subsection, we present several poisoning attack schemes for specific unsupervised domain adaptation algorithms. It might lead to two follow-up questions: (1) whether the poisoned source examples are transferable across different domain adaptation algorithms given source and target domains? (2) does there exist universal poisoning attacks for multiple target domains?

For the first question, we argue that the poisoned source examples learned on one domain adaptation method (e.g., DAN [16]) can be directly applied to attack other methods (e.g., DANN [7] and MDD [33]). That is because the poisoning attacks maximize the discrepancy of joint (or class-conditional) data distribution between clean and poisoned source domains. Normally, the domain discrepancy measures (e.g., $\mathcal{H}$-divergence [1], margin disparity [33]) will monotonically change with respect to the relatedness across domains. Thus, maximizing one domain discrepancy measure of a domain adaptation approach implies the increase of another discrepancy measure in a new domain adaptation approach. For the second question, it might find the universal black-box poisoning attacks when the adversary has no knowledge of the potential target domain. In this case, it can simply find an auxiliary target domain and then learn the poisoned examples using the source and auxiliary target domains. Such attacks might work for any target domain when it is related to the raw source domain because it holds $d(\hat{X}_s \circ Y_s, X_t \circ Y_t) \geq d(\hat{X}_s \circ Y_s, X_s \circ Y_s) - d(X_s \circ Y_s, X_t \circ Y_t)$ if discrepancy $d(\cdot, \cdot)$ satisfies the triangle inequality property.

Convergence and Complexity of I2Attack: We first discuss the convergence of optimization methods used in our **I2Attack** algorithms. For two-stage domain adaptation methods (i.e., CORAL [27]), the poisoning attack algorithm **I2Attack-CORAL** can be optimized with stochastic gradient descent (SGD), which converges almost surely to a local minimum [8]. For deep domain adaptation methods (i.e., DAN [16], DANN [7] and MDD [33]), we show that our poisoning attack algorithms (i.e., **I2Attack-DAN**, **I2Attack-DANN** and **I2Attack-MDD**) can be optimized via model-agnostic meta-learning (MAML) [6]. Following the theoretical analysis [5] of MAML and its first-order approximation (FO-MAML), it holds that MAML finds an ϵ'-first-order stationary point for any positive $\epsilon' > 0$ after at most $O(1/\epsilon'^2)$ iterations. Furthermore, if the inner learning rate β used for updating the poisoned examples is small, then the approximation error of FO-MAML induced by ignoring the second-order term does not impact its convergence.

The computational complexity of **I2Attack** is presented as follows. For two-stage domain adaptation methods, the inputs of source and target domains are m-dimensional feature vectors, so it has a computational complexity of $O(n_S m)$ per iteration using stochastic gradient descent. On the other hand, the input of deep domain adaptation methods can be raw images. Then, the FO-MAML based optimization has a computational complexity of $O(n_s N_{pixel})$ per iteration where N_{pixel} is the average number of pixels per image and n_s is the number of source examples.

6 EXPERIMENTS

6.1 Experimental Setup

Data Sets: We use the following domain adaptation benchmarks:

- Digits: We adopt three digital image data sets: MNIST, USPS and SVHN with 70,000, 99,289 and 9,297 images of 10 categories respectively, and report the domain adaptation results on MNIST (M) → USPS (U) and SVHN (S) → MNIST (M).
- Office-31: It has 4,652 images of 31 categories from three domains: Amazon (A), Webcam (W) and DSLR (D).
- Office-Caltech10: It has 2,533 images of 10 categories from four domains: Caltech (C), Amazon (A), Webcam (W), DSLR (D).
- Office-Home: It has 15,500 images of 65 categories from four domains: Artistic images (Ar), Clip Art (Cl), Product images (Pr) and Real-World images (Rw).
- Image-CLEF: It has 2,400 images of 12 categories from four domains: Caltech-256 (C), ImageNet ILSVRC 2012 (I), Pascal VOC2012 (P) and Bing (B).
- VisDA2017: It has over 200K images of 12 categories from two domains: Synthetic (Syn) and Real.

Baselines: We compare our proposed **I2Attack** framework with the following baselines: naïve RAttack that adding random Gaussian noise to source examples, SAttack [20] that generates the poisoned source examples by maximizing the classification error; BFGSM [34] that pre-trains a source model and then generates the adversarial source examples via Fast Sign Gradient Method (FGSM) [11]. The generated poisoned source examples are then used to evaluate the adversarial vulnerability of domain adaptation algorithms.

6.2 Performance Comparison

Table 2 and Table 3 provide the results on evaluating the adversarial vulnerability of unsupervised domain adaptation algorithms under **I2Attack**. Specifically, we report the source classification accuracy (S Acc), domain discrepancy (Disc) and target classification accuracy (T Acc) to demonstrate how adversarial attacks affect the domain adaptation methods. As shown in Table 1, we adopt the covariance distance, multi-kernel maximum mean discrepancy (MK-MMD), $\mathcal{H}$-divergence and margin-aware disparity discrepancy to measure the domain discrepancy (Disc) between source and target domains on CORAL [27], DAN [16], DANN [7] and MDD [33], respectively. It is observed that: (1) the target performance could be significantly degraded (e.g., up to 90% degradation on Office-31) with the poisoned source examples learned by **I2Attack** for all domain adaptation methods; (2) the source classification accuracy and domain discrepancy could be almost unchanged and even become better (i.e., higher source accuracy or lower marginal domain discrepancy) in some cases.

Figure 3 demonstrates the effectiveness of **I2Attack** on attacking DAN and DANN, compared to baseline methods on Office-31 and Image-CLEF. More specifically, we observe that (1) naïve RAttack with random Gaussian noise would not largely degrade the domain adaptation performance; (2) the poisoned source examples generated from SAttack and BFGSM deteriorate both source and target classification performance, and thus are easily detected in the model training phase; (3) compared to baselines, our proposed **I2Attack** achieves much lower target accuracy by maximizing the discrepancy of joint data distribution over $\mathcal{X} \times \mathcal{Y}$ across domains.

6.3 Transferability and Universalness

We evaluate the transferability and universalness of **I2Attack** on Office-31 and Image-CLEF. Table 4 shows the domain adaptation results on DAN and DANN using the generated source examples from different **I2Attack** algorithms. It shows that the generated source examples by one attacking algorithm (e.g., **I2Attack**-MDD) can be used to successfully attack any other domain adaptation

Table 2: Poisoning attack on CORAL [27] ('–': almost unchanged; '↑': improved; '↓': degraded). Note that small domain discrepancy is more preferred for learning the domain-invariant representation in the feature space.

		Office-Caltech10						Office-Home					
		C→A	C→W	C→D	A→C	A→W	A→D	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw
CORAL (base model)	S Acc	0.858	0.836	0.799	0.921	0.900	0.880	0.904	0.902	0.896	0.921	0.919	0.916
	Disc	21.16	31.43	40.43	21.27	33.16	42.43	24.28	23.34	14.47	24.32	18.83	18.92
	T Acc	0.549	0.468	0.459	0.435	0.383	0.420	0.468	0.603	0.676	0.524	0.600	0.630
I2Attack-CORAL	S Acc	0.995↑	0.996↑	0.999↑	0.997↑	0.998↑	1.000↑	0.998↑	0.998↑	1.000↑	1.000↑	1.000↑	1.000↑
	Disc	20.72–	30.23–	38.55↑	20.72–	31.71↑	40.20↑	21.61↑	20.58↑	14.57–	20.82↑	17.25–	17.27–
	T Acc	0.021↓	0.031↓	0.070↓	0.126↓	0.081↓	0.121↓	0.100↓	0.099↓	0.086↓	0.138↓	0.118↓	0.167↓

Table 3: Poisoning attack of deep domain adaptation ('–': almost unchanged; '↑': improved; '↓': degraded)

		Digits		Office-31			Office-Home		Image-CLEF			VisDA2017
		M→U	S→M	W→A	W→D	D→A	Ar→Cl	Pr→Rw	B→I	C→P	P→B	Syn→Real
DAN (base model)	S Acc	0.997	0.916	1.000	1.000	1.000	1.000	0.999	1.000	1.000	1.000	0.987
	Disc	0.078	0.085	2.459	2.315	2.156	1.835	1.931	2.137	2.589	1.742	0.478
	T Acc	0.861	0.724	0.654	0.994	0.656	0.498	0.750	0.848	0.750	0.588	0.584
I2Attack-DAN	S Acc	1.000–	1.000↑	0.996–	0.998–	0.994–	0.998–	0.999–	1.000–	1.000–	1.000–	0.986–
	Disc	0.079–	0.079↑	2.304↑	1.975↑	2.152–	1.579↑	1.684↑	1.919↑	1.939↑	1.555↑	0.437↑
	T Acc	0.664↓	0.495↓	0.065↓	0.062↓	0.046↓	0.293↓	0.660↓	0.113↓	0.203↓	0.252↓	0.469↓
DANN (base model)	S Acc	0.997	0.911	1.000	1.000	1.000	1.000	0.999	1.000	1.000	1.000	0.983
	Disc	0.567	0.520	0.646	0.642	0.609	0.506	0.500	0.602	0.758	0.733	0.742
	T Acc	0.896	0.795	0.679	0.998	0.668	0.513	0.756	0.888	0.782	0.597	0.637
I2Attack-DANN	S Acc	1.000–	0.948↑	0.996–	1.000–	0.998–	0.994	0.999–	1.000–	1.000–	0.998–	0.990–
	Disc	0.569–	0.516–	0.588↑	0.643–	0.550↑	0.501–	0.500–	0.572↑	0.695↑	0.593↑	0.688↑
	T Acc	0.801↓	0.510↓	0.078↓	0.046↓	0.105↓	0.378↓	0.673↓	0.083↓	0.233↓	0.142↓	0.201↓
MDD (base model)	S Acc	0.997	0.901	1.000	1.000	1.000	1.000	0.999	1.000	1.000	1.000	0.969
	Disc	1.373	1.496	1.374	1.493	1.028	1.735	1.697	1.501	1.214	1.341	1.123
	T Acc	0.908	0.753	0.693	0.998	0.679	0.505	0.781	0.903	0.788	0.617	0.657
I2Attack-MDD	S Acc	1.000–	0.944	0.996–	0.991–	0.996–	0.993–	0.991–	1.000–	1.000–	1.000–	0.996↑
	Disc	1.317↑	1.453↑	1.056↑	1.473↑	0.938↑	1.603↑	1.645↑	1.449–	1.010↑	1.339–	0.999↑
	T Acc	0.789↓	0.585↓	0.050↓	0.024↓	0.137↓	0.382↓	0.679↓	0.163↓	0.170↓	0.217↓	0.301↓

Table 4: Transferability of I2Attack on Office-31 (W→D)

	DAN			DANN		
	S Acc	Disc	T Acc	S Acc	Disc	T Acc
Clean	1.000	2.315	0.994	1.000	0.642	0.998
I2Attack-DAN	0.998	1.975	0.062	0.996	0.622	0.020
I2Attack-DANN	0.999	2.031	0.068	1.000	0.643	0.046
I2Attack-MDD	0.991	2.156	0.092	0.994	0.649	0.032

Table 5: Universalness of I2Attack on Image-CLEF

	Clean			I2Attack		
	S Acc	Disc	T Acc	S Acc	Disc	T Acc
B→I	1.000	2.137	0.848	1.000	1.919	0.113
B→C	1.000	2.215	0.907	1.000	1.921	0.120
B→P	1.000	1.927	0.717	1.000	1.755	0.098

algorithm (e.g., DAN and DANN). This indicates that **I2Attack** allows attacking the black-box domain adaptation algorithm without knowledge of model configuration. Table 5 shows the attack results of poisoned source examples on different target domains of Image-CLEF where all poisoned examples are generated from DAN on B→I and then applied to attack other target domains, i.e., C or P. It can be seen that the generated poisoned examples on B can be directly used to attack multiple downstream target domains. This enables the black-box attacks without access to the target domain.

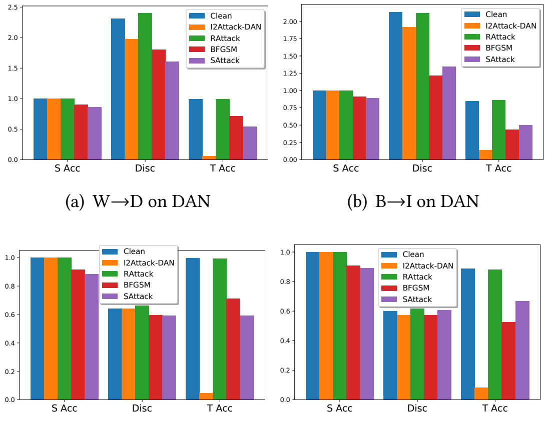
6.4 Parameter Study

We investigate the impact of perturbation magnitude ϵ on **I2Attack**. As shown in Figure 4(a), we report the results of **I2Attack**-DANN on W→D of Office-31 with ϵ increasing from 0 to 0.10. It is shown that the source classification accuracy and domain discrepancy almost keep unchanged, but the target classification accuracy decreases significantly under larger perturbation magnitude ϵ.

Besides, we empirically evaluate the computational efficiency of **I2Attack** on VisDA2017. In this case, we randomly sample n_s examples from the source domain with n_s increasing from 1000 to 7000. The running time (measured in seconds wall-clock time) per iteration on this data set is reported in Figure 4(b). we observe that the running time of our proposed **I2Attack** is linear with respect to the number of source training examples n_s, which is consistent with our analysis in Subsection 5.2.

7 CONCLUSION

In this paper, we focus on analyzing the adversarial vulnerability of unsupervised domain adaptation. We start by identifying three properties: *perceptibly unnoticeable*, *adversarially indirect* and *algorithmically invisible*, which provide insights into designing the poisoning attacks for domain adaptation. Then we present a generic framework **I2Attack** on attacking the existing domain adaptation



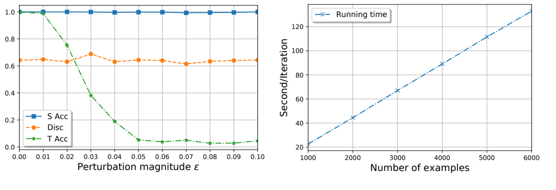
(a) W⟶D on DAN

(b) B⟶I on DAN

(c) W⟶D on DANN

(d) B⟶I on DANN

Figure 3: Comparison of I2Attack and baselines on W⟶D of Office-31 and B⟶I of Image-CLEF



(a) Effect of perturbation ϵ

(b) Running time

Figure 4: Analysis of I2Attack on (a) W⟶D of Office-31 with different perturbation magnitude ϵ, and (b) Syn⟶Real of VisDA2017 with varying number of source images

algorithms. Extensive experiments demonstrate the effectiveness and efficiency of our **I2Attack** framework.

ACKNOWLEDGMENTS

This work is supported by National Science Foundation under Award No. IIS-1947203 and IIS-2002540, and Agriculture and Food Research Initiative (AFRI) grant no. 2020-67021-32799/project accession no.1024178 from the USDA National Institute of Food and Agriculture. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

REFERENCES

[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A Theory of Learning from Different Domains. *Machine learning* 79, 1-2 (2010), 151–175.

[2] Battista Biggio, B Nelson, and P Laskov. 2012. Poisoning Attacks against Support Vector Machines. In *ICML*.

[3] Nicholas Carlini and David Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy*. 39–57.

[4] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. 2019. Transferability vs. Discriminability: Batch Spectral Penalization for Adversarial Domain Adaptation. In *ICML*.

[5] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. On the Convergence Theory of Gradient-based Model-agnostic Meta-learning Algorithms. In *AISTATS*.

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic Meta-learning for Fast Adaptation of Deep Networks. In *ICML*.

[7] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *ICML*. 1180–1189.

[8] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. 2015. Escaping from Saddle Points—online Stochastic Gradient for Tensor Decomposition. In *COLT*. 797–842.

[9] Muhammad Ghifary, David Balduzzi, W Bastiaan Kleijn, and Mengjie Zhang. 2016. Scatter Component Analysis: A Unified Framework for Domain Adaptation and Domain Generalization. *TPAMI* 39, 7 (2016), 1414–1430.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *NeurIPS*.

[11] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *ICLR*.

[12] Arthur Gretton, Dino Sejdinovic, Heiko Strathmann, Sivaraman Balakrishnan, Massimiliano Pontil, Kenji Fukumizu, and Bharath K Sriperumbudur. 2012. Optimal Kernel Choice for Large-scale Two-sample Tests. In *NeurIPS*. 1205–1213.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*. 770–778.

[14] Fredrik D Johansson, David Sontag, and Rajesh Ranganath. 2019. Support and Invertibility in Domain-Invariant Representations. In *AISTATS*.

[15] Xuanqing Liu, Si Si, Jerry Zhu, Yang Li, and Cho-Jui Hsieh. 2019. A Unified Framework for Data Poisoning Attack to Graph-based Semi-supervised Learning. In *NeurIPS*. 9780–9790.

[16] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. 2015. Learning Transferable Features with Deep Adaptation Networks. In *ICML*.

[17] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. 2017. Deep Transfer Learning with Joint Adaptation Networks. In *ICML*.

[18] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. 2009. Domain Adaptation: Learning Bounds and Algorithms. In *COLT*.

[19] Shike Mei and Xiaojin Zhu. 2015. Using Machine Teaching to Identify Optimal Training-Set Attacks on Machine Learners.. In *AAAI*. 2871–2877.

[20] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. 2017. Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization. In *AISec*. 27–38.

[21] Sinno Jialin Pan and Qiang Yang. 2009. A Survey on Transfer Learning. *TKDE* 22, 10 (2009), 1345–1359.

[22] Shahbaz Rezaei and Xin Liu. 2020. A Target-Agnostic Attack on Deep Models: Exploiting Security Vulnerabilities of Transfer Learning. In *ICLR*.

[23] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. 2018. Poison Frogs! Targeted Clean-label Poisoning Attacks on Neural Networks. In *NeurIPS*. 6103–6113.

[24] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. 2019. Adversarial Training for Free!. In *NeurIPS*. 3358–3369.

[25] Ali Shafahi, Parsa Saadatpanah, Chen Zhu, Amin Ghiasi, Christoph Studer, David Jacobs, and Tom Goldstein. 2020. Adversarially Robust Transfer Learning. In *ICLR*.

[26] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. 2018. Wasserstein Distance Guided Representation Learning for Domain Adaptation. In *AAAI*.

[27] Baochen Sun, Jiashi Feng, and Kate Saenko. 2016. Return of Frustratingly Easy Domain Adaptation. In *AAAI*.

[28] Baochen Sun and Kate Saenko. 2016. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. In *ECCV*. 443–450.

[29] Bolun Wang, Yuanshun Yao, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. 2018. With Great Training Comes Great Vulnerability: Practical Attacks against Transfer Learning. In *27th {USENIX} Security Symposium*. 1281–1297.

[30] Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. 2019. Characterizing and Avoiding Negative Transfer. In *CVPR*. 11293–11302.

[31] Jun Wu and Jingrui He. 2020. Continuous Transfer Learning with Label-informed Distribution Alignment. *arXiv preprint arXiv:2006.03230* (2020).

[32] Ban Yikun, Liu Xin, Huang Ling, Duan Yitao, Liu Xue, and Xu Wei. 2019. No place to hide: Catching fraudulent entities in tensors. In *WWW*. 83–93.

[33] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. 2019. Bridging Theory and Algorithm for Domain Adaptation. In *ICML*. 7404–7413.

[34] Yinghua Zhang, Yangqiu Song, Jian Liang, Kun Bai, and Qiang Yang. 2020. Two Sides of the Same Coin: White-box and Black-box Attacks for Transfer Learning. In *KDD*. 2989–2997.

[35] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. 2019. On Learning Invariant Representations for Domain Adaptation. In *ICML*. 7523–7532.

[36] Lecheng Zheng, Yu Cheng, Hongxia Yang, Nan Cao, and Jingrui He. 2021. Deep Co-Attention Network for Multi-View Subspace Learning. *arXiv preprint arXiv:2102.07751* (2021).

[37] Dawei Zhou, Lecheng Zheng, Jiejun Xu, and Jingrui He. 2019. Misc-GAN: A Multi-scale Generative Model for Graphs. *Frontiers in Big Data* 2 (2019), 3.

[38] Dawei Zhou, Lecheng Zheng, Yada Zhu, Jianbo Li, and Jingrui He. 2020. Domain Adaptive Multi-modality Neural Attention Network for Financial Forecasting. In *WWW*. 2230–2240.

[39] Yao Zhou, Jianpeng Xu, Jun Wu, Zeinab Taghavi, Evren Korpeoglu, Kannan Achan, and Jingrui He. 2021. PURE: Positive-Unlabeled Recommendation with Generative Adversarial Network. In *KDD*.

[40] Chen Zhu, W Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2019. Transferable Clean-Label Poisoning Attacks on Deep Neural Nets. In *ICML*. 7614–7623.

A APPENDIX

To better reproduce the experimental results, we provide additional details about the algorithms.

A.1 Notation

The notation used in this paper is summarized in Table 6.

Table 6: Notation

Notation	Definition
$\mathcal{X}, \mathcal{Y}$	Input space and output space
$\mathbb{Q}, \mathbb{P}$	Source and target domains
$\mathbb{Q}_{XY}, \mathbb{P}_{XY}$	Data distributions on source and target domains
$\mathbb{Q}_X, \mathbb{P}_X$	Marginal distributions on source and target domains
$l_{\mathbb{Q}}, l_{\mathbb{P}}$	Labeling functions on source and target domains
$(x^s, y^s) \sim \mathbb{Q}$	Labeled source example
$x^t \sim \mathbb{P}_X$	Unlabeled target example
n_s	Number of labeled source examples
n_t	Number of unlabeled target examples
$L(\cdot, \cdot)$	Loss function
$\mathcal{H}$	Hypothesis class
$d(\cdot, \cdot)$	Domain discrepancy measure

A.2 Proof of Theorem 4.1

Theorem 4.1 states that let ϵ_s and ϵ_t denote the expected source and target classification error. Given any class-separable target domain $\mathbb{P}$, there exist at least one source domain $\mathbb{Q}$ and $h \in \mathcal{H}$ satisfying $\epsilon_s(h) = 0$ and $d(\mathbb{Q}_X, \mathbb{P}_X) = 0$, such that the target classification error $\epsilon_t(h) = 1$.

Proof. We first consider the binary classification scenario with $y \in \{-1, 1\}$. Given any target domain $\mathbb{P}$ and the loss function $L(\cdot, \cdot)$, we can construct a source domain $\mathbb{Q}$ as follows. For any data point $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we let the data distribution of source domain be either $\mathbb{Q}_{XY}(x, y = 1) = \mathbb{P}_{XY}(x, y = 1)$ or $\mathbb{Q}_{XY}(x, y = 1) = \mathbb{P}_{XY}(x, y = -1)$. Then, it is easy to show $\mathbb{Q}_X = \mathbb{P}_X$, so we have $d(\mathbb{Q}_X, \mathbb{P}_X) = 0$ for any domain discrepancy measure. Specifically, we can flip the class-labels of target examples if they are correctly classified. For example, if one point $(x, y) \in \mathcal{X} \times \mathcal{Y}$ from target domain satisfies $h^*(x) = y$, we hold $\mathbb{Q}_{XY}(x, y = 1) = \mathbb{P}_{XY}(x, y = -1)$, otherwise, $\mathbb{Q}_{XY}(x, y = 1) = \mathbb{P}_{XY}(x, y = 1)$. Then there exists $h = -h^* \in \mathcal{H}$ such that $\epsilon_s(h) = 0$. In this case, the target classification error $\epsilon_t(h) = 1 - \epsilon_t(h^*)$. Furthermore, we can see that $\epsilon_t(h) = 1$ when $\epsilon_t(h^*) = 0$.

Then we can use the similar way to construct the source domain for multi-classification scenario as follows. For any data point $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we let the data distribution of source domain be either $\mathbb{Q}_{XY}(x, y = i) = \mathbb{P}_{XY}(x, y = j)$ where i and j represents the i^{th} and j^{th} classes respectively. Suppose the optimal target hypothesis h^* can partition the space $\mathcal{X} \times \mathcal{Y}$ into C sub-regions where C is number of classes. In this case, we can simply set the class label of examples in each sub-region as one of classes such that $\mathbb{Q}_{XY}(x, y = i) = \mathbb{P}_{XY}(x, y = j) (i \neq j)$. □

Table 7: Poisoning attack of deep domain adaptation by maximizing class-conditional domain discrepancy

		Office-31		Image-CLEF	
		W→A	W→D	B→I	C→P
DAN (base model)	S Acc	1.000	1.000	1.000	1.000
	Disc	2.459	2.315	2.137	2.589
	T Acc	0.654	0.994	0.848	0.750
I2Attack-DAN	S Acc	0.994	0.998	1.000	1.000
	Disc	2.109	1.950	1.774	2.194
	T Acc	0.403	0.568	0.468	0.593
DANN (base model)	S Acc	1.000	1.000	1.000	1.000
	Disc	0.646	0.642	0.602	0.758
	T Acc	0.679	0.998	0.888	0.782
I2Attack-DANN	S Acc	0.998	0.998	1.000	1.000
	Disc	0.605	0.638	0.606	0.714
	T Acc	0.401	0.588	0.663	0.570
MDD (base model)	S Acc	1.000	1.000	1.000	1.000
	Disc	1.374	1.493	1.501	1.214
	T Acc	0.693	0.998	0.903	0.788
I2Attack-MDD	S Acc	0.994	0.996	1.000	1.000
	Disc	1.357	1.462	1.400	1.119
	T Acc	0.257	0.677	0.777	0.693

A.3 Proof of Corollary 4.2

Corollary 4.2 states that let ϵ_s and ϵ_t denote the expected source and target classification error. For any class-separable target domain $\mathbb{P}$, there exists a source domain $\mathbb{Q}$ and $h \in \mathcal{H}$ such that $\epsilon_s(h) = 0$, $d(\mathbb{Q}_X, \mathbb{P}_X) = 0$ and $\epsilon_t(h) = 1$ if it satisfies one of the following conditions: (i) $\mathbb{Q}_{XY}(x, y = i) = \mathbb{P}_{XY}(x, y = j)$; (ii) $\mathbb{Q}_{Y|X}(y = i|x) = \mathbb{P}_{Y|X}(y = j|x)$; (iii) $\mathbb{Q}_{X|Y}(x|y = i) = \mathbb{P}_{X|Y}(x|y = j)$ if $\mathbb{Q}_Y(y = i) = \mathbb{P}_Y(y = j)$, for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and $i \neq j$.

Proof. As proven in Theorem 4.1, $\mathbb{Q}_{XY}(x, y = i) = \mathbb{P}_{XY}(x, y = j)$ could lead to the negative transfer with $\epsilon_s(h) = 0$, $d(\mathbb{Q}_X, \mathbb{P}_X) = 0$ and $\epsilon_t(h) = 1$. When $d(\mathbb{Q}_X, \mathbb{P}_X) = 0$ or $\mathbb{Q}_X = \mathbb{P}_X$, it holds $\mathbb{Q}_{Y|X}(y = i|x) = \mathbb{P}_{Y|X}(y = j|x)$ using the Bayes' theorem. Similarly, if $\mathbb{Q}_Y(y = i) = \mathbb{P}_Y(y = j)$, it holds $\mathbb{Q}_{X|Y}(x|y = i) = \mathbb{P}_{X|Y}(x|y = j)$. Therefore, these conditions are equivalent on designing the source domain satisfying $\epsilon_s(h) = 0$, $d(\mathbb{Q}_X, \mathbb{P}_X) = 0$ and $\epsilon_t(h) = 1$. □

A.4 Experiments

All the experiments are performed on a Windows machine with four 3.80GHz Intel Cores, 64GB RAM and two NVIDIA Quadro RTX 5000 GPUs.

A.4.1 Data Sets. The data sets used in our experiments are publicly available as follows.

- Digits: We adopt three digital image data sets: MNIST[5], USPS[6] and SVHN[7] with 70,000, 99,289 and 9,297 images of 10 categories respectively, and report the domain adaptation results on MNIST (M) → USPS (U) and SVHN (S) → MNIST (M).

[5] http://yann.lecun.com/exdb/mnist/

[6] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

[7] http://ufldl.stanford.edu/housenumbers/

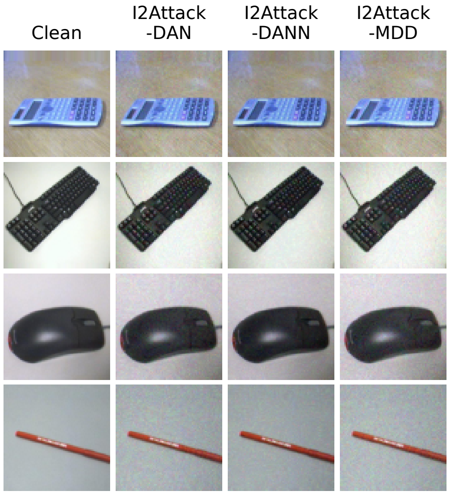


Figure 5: Illustration of clean and poisoned source examples on W→D of Office-31

- Office-31[8]: It has 4,652 images of 31 categories from three domains: Amazon (A), Webcam (W) and DSLR (D).
- Office-Caltech10[8]: It has 2,533 images of 10 categories from four domains: Caltech (C), Amazon (A), Webcam (W), DSLR (D).
- Office-Home[9]: It has 15,500 images of 65 categories from four domains: Artistic images (Ar), Clip Art (Cl), Product images (Pr) and Real-World images (Rw).
- Image-CLEF[10]: It has 2,400 images of 12 categories from four domains: Caltech-256 (C), ImageNet ILSVRC 2012 (I), Pascal VOC2012 (P) and Bing (B).
- VisDA2017[11]: It has over 200K images of 12 categories from two domains: Synthetic (Syn) and Real.

A.4.2 Model Configuration. For **I2Attack**-CORAL, we use vanilla gradient descent for optimization with learning rate $1e-5$. For **I2Attack**-DAN, **I2Attack**-DANN and **I2Attack**-MDD (see Algorithm 1), we adopt stochastic gradient descent with mini-batch of 72 for inner updates with $L=1$, $\beta=0.001$, $\alpha=0.01$ and $\epsilon=0.1$. The overall iterations are 25 in our experiments. Besides, for DAN [16], DANN [7] and MDD [33], we adopted the ResNet-50 [13] pretrained on ImageNet for feature extraction with an added 256-dimension bottleneck layer between the *res5c* and *fc* layers. It is then optimized using stochastic gradient descent with mini-batch of size 32. The learning rate η_p is adjusted as: $\eta_p = \frac{\eta_0}{(1+\omega p)^\tau}$, where p is an epoch-dependent scalar linearly varying from 0 to 1, and $\eta_0 = 0.01, \omega = 10, \tau = 0.75$.

A.4.3 Additional Results. The additional experimental results are provided below.

Attacking function based on class-conditional distribution: Table 7 shows the poisoning attack results of **I2Attack** with the attacking function on maximizing the class-conditional domain discrepancy. It can be seen that the target classification performance could be degraded without worsening the source classification error and marginal domain discrepancy.

Visualization: Figure 5 visualizes the source examples before and after the attack on W→D of Office-31. It can be seen that the poisoned source images are perceptibly indistinguishable from the raw clean images.

[8] https://people.eecs.berkeley.edu/~jhoffman/domainadapt/
[9] http://hemanthdv.org/OfficeHome-Dataset/
[10] https://www.imageclef.org/2014/adaptation
[11] http://ai.bu.edu/visda-2017/