UMD: Unsupervised Model Detection for X2X Backdoor Attacks

Zhen Xiang [1] **Zidi Xiong** [1] **Bo Li** [1]

Abstract

Backdoor (Trojan) attack is a common threat to deep neural networks, where samples from one or more *source classes* embedded with a backdoor trigger will be misclassified to adversarial *target classes*. Existing methods for detecting whether a classifier is backdoor attacked are mostly designed for attacks with a single adversarial target (e.g., all-to-one attack). To the best of our knowledge, without supervision, no existing methods can effectively address the more general X2X attack with an arbitrary number of source classes, each paired with an arbitrary target class. In this paper, we propose UMD, the *first* Unsupervised Model Detection method that effectively detects X2X backdoor attacks via a joint inference of the adversarial (source, target) class pairs. In particular, we first define a novel *transferability* statistic to measure and select a subset of putative backdoor class pairs based on a proposed clustering approach. Then, these selected class pairs are jointly assessed based on an aggregation of their reverse-engineered trigger size for detection inference, using a robust and unsupervised anomaly detector we proposed. We conduct comprehensive evaluations on CIFAR-10, GTSRB, and Imagenette dataset, and show that our *unsupervised* UMD outperforms SOTA detectors (even with supervision) by 17%, 4%, and 8%, respectively, in terms of the detection accuracy against diverse X2X attacks. We also show the strong detection performance of UMD against several strong adaptive attacks.

1. Introduction

Despite the success of deep neural networks in many applications, they are vulnerable to adversarial attacks such as backdoor (Trojan) attacks (Miller et al., 2020; Li et al., 2022a). A classical backdoor attack is usually specified by one or more source classes, a target class, and a backdoor trigger, such that test samples from any source class embedded with the trigger will be misclassified to the target class; while samples without the trigger will be correctly classified (Gu et al., 2019). Typically, a backdoor attack is launched by poisoning the training set of the classifier (Chen et al., 2017; Turner et al., 2019; Zhong et al., 2020; Liu et al., 2020; Nguyen & Tran, 2021; Li et al., 2021b).

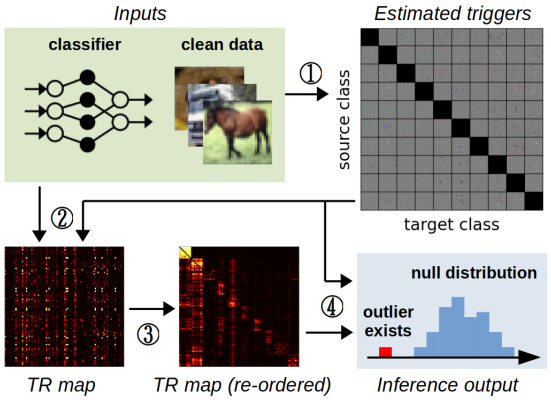

Figure 1. Outline of UMD: ① reverse-engineer a trigger for each class pair (Sec. 4.1.1); ② compute TR for all ordered pair of class pairs (Sec. 4.1.1); ③ select a subset of putative backdoor class pairs based on TR (Sec. 4.2.1); ④ inspect the selected pairs by unsupervised anomaly detection on trigger statistics (Sec. 4.2.2).

Recently, many approaches have been proposed to detect whether a trained classifier is backdoor attacked *without access to* the training set or any benign models for supervision (e.g. to set a detection threshold) (Chen et al., 2019; Guo et al., 2019; Xiang et al., 2020; Wang et al., 2020; Dong et al., 2021; NeurIPS, 2022). These methods mainly fall into either a family of reverse-engineering-based detectors (**RED**s) or a category of meta-classification-based detectors (**MCD**s). Typically, REDs reverse-engineer putative triggers for anomaly detection (Wang et al., 2019; Liu et al., 2019), while MCDs train a binary meta classifier on a large number of shadow classifiers with and without attack for detection (Xu et al., 2021). These detectors are effective against the classical backdoor attack, but they may be bypassed by more advanced backdoor attacks recently proposed to defeat them (Nguyen & Tran, 2020; Peng et al., 2022).

In this paper, we consider *X2X backdoor attacks*, which refer to a broad family of backdoor attacks with an arbitrary number of source classes, each assigned with an arbitrary target class. Thus, the X2X attack includes many popular attack types such as the "all-to-one" attack, "X-to-one" attack (Shen et al., 2021), "one-to-one" attack (Tran et al., 2018), and "all-to-all" attack (Gu et al., 2019). Unlike other

[1]University of Illinois at Urbana-Champaign. Correspondence to: Zhen Xiang <zxiangaa@illinois.edu>, Bo Li <lbo@illinois.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

advanced attacks that mostly rely on additional assumptions such as full control of the training process (Zhao et al., 2022; Wang et al., 2022), X2X attacks can be easily launched by poisoning the training set just like classical backdoor attacks. Moreover, to the best of our knowledge, X2X attacks are *not detectable* by existing methods – REDs are mostly designed for "all-to-one" attacks only, while MCDs need to assume access to the attack setting to train the shadow classifiers.

To bridge this gap, we propose an Unsupervised Model Detection approach (UMD) to detect X2X attacks and infer all the class pairs involved in the attack, without any assumptions about the number of source classes or the target class assignment rules. UMD first reverse-engineers a putative trigger for each class pair. Unlike existing detectors that directly use trigger statistics (e.g., the perturbation size of the triggers) for anomaly detection, we calculate a *transferability* (TR) statistic for each ordered pair of class pairs. The TR statistics are then used to select a subset of class pairs that are most likely involved in an X2X attack by solving a novel clustering problem. Finally, an *unsupervised*, bias-reduced anomaly detector is designed to robustly assess the atypicality of the trigger statistic aggregated over the selected class pairs – these class pairs are deemed the backdoor class pairs when an attack is detected. Our contributions in this paper are summarized as the following:

- We propose UMD, the *first* unsupervised model detector against X2X backdoor attacks with arbitrary numbers of source classes and arbitrary target class assignments.
- We propose a statistic – TR – to identify backdoor class pairs. In particular, we prove that in ideal cases, TR from one backdoor class pair to another backdoor class pair is guaranteed to be no less than TR from a backdoor class pair to a non-backdoor class pair.
- We propose a two-step inference procedure for UMD. First, a set of putative backdoor class pairs is selected based on the TR statistics by solving a novel clustering problem using an agglomerative algorithm. Second, an aggregated trigger statistic based on these selected class pairs is evaluated for inference using our robust, unsupervised anomaly detector, with a confidence threshold adapted to the number of classes in the domain.
- We conduct extensive experiments to show the strong detection capability of UMD against diverse X2X attacks and several strong advanced adaptive attacks. We show that our unsupervised UMD outperforms SOTA baselines, including the ones with supervision by Liu et al. (2019) and Shen et al. (2021) by 17%, 4%, and 8% in the average model inference accuracy against various X2X attacks on CIFAR-10, GTSRB, and Imagenette, respectively.

2. Related Work

Backdoor attacks While we focus on image classification in this paper like most existing works, backdoor attacks have also been extended to other data domains and/or learning paradigms (Li et al., 2021a; Chen et al., 2021; Li et al., 2022b; Xie et al., 2020; Yao et al., 2019; Jia et al., 2022). For the image domain, apart from the X2X attack focused on in this paper, advanced backdoor attacks, such as clean-label attacks (Turner et al., 2019; A. Saha, 2020), invisible-trigger attacks (Zhong et al., 2020; Nguyen & Tran, 2021; Zhao et al., 2022; Wang et al., 2022), and physical attacks (Liu et al., 2020), are also proposed to achieve better stealthiness against possible human inspection of either the training set or test instances. Moreover, some advanced backdoor attacks are proposed, e.g., by Nguyen & Tran (2020), Li et al. (2021), and Xue et al. (2022), to evade particular backdoor defenses. In addition to the X2X attack, we will show the effectiveness of our UMD against some of these advanced attacks (including their X2X extensions) as well.

Backdoor model detection Existing methods that detect whether a trained classifier is backdoor attacked mainly fall into two categories. Reverse-engineering-based detectors (REDs) estimate putative triggers for anomaly detection (Wang et al., 2019; Chen et al., 2019; Xiang et al., 2020; Wang et al., 2020; Shen et al., 2021; Tao et al., 2022; Hu et al., 2022). Meta-classification-based detectors (MCDs) train a meta classifier using shadow classifiers trained with and without attacks (Xu et al., 2021; Kolouri et al., 2020). Unlike our UMD, these methods *cannot* effectively detect X2X attacks since they all rely on assumptions about the target class assignment. Except for our UMD, several prior model detectors also involved the concept of "*transferability*". For example, transferability is defined at the instance level by Xiang et al. (2022) and Huster & Ekwedike (2021), or for each putative target class to be inspected by Liu et al. (2019). Differently, the TR statistic used by our UMD is defined for each *ordered pair of class pairs*, which enables UMD to identify backdoor class pairs regardless of the target class assignment.

Other types of backdoor defense Backdoor mitigation approaches aim to remove the learned backdoor mapping from a trained classifier (Liu et al., 2018; Wu & Wang, 2021; Li et al., 2021c; Guan et al., 2022; Zheng et al., 2022; Zeng et al., 2022). They usually require a large number of samples and may degrade the clean accuracy of the classifier. Training-phase defenses aim to obtain a backdoor-free classifier from the possibly poisoned training set (Tran et al., 2018; Chen et al., 2018; Xiang et al., 2019; Du et al., 2020; Huang et al., 2022). They can not be deployed at the user end where the classifier is already trained. Inference-stage defenses detect whether a test sample is embedded with a backdoor trigger (Gao et al., 2019; Doan et al., 2020; Chou et al., 2020). They require test samples with the actual backdoor trigger, which are unavailable for our detection problem. Thus, we will not further discuss these methods.

3. Threat Model

X2X backdoor attacks refer to a family of backdoor attacks with arbitrary numbers of source classes each assigned with an arbitrary target class. It covers many popular attacks with different settings including the "all-to-one" (A2O) attack (Chen et al., 2017), "X-to-one" (X2O) attack (Shen et al., 2021) (a.k.a. a "partial backdoor" (Wang et al., 2019)), "one-to-one" (O2O) attack (Tran et al., 2018), and "all-to-all" (A2A) attack (Gu et al., 2019). The complete taxonomy of X2X backdoor attacks is shown in Fig. 2. Formally, for a classification task with sample space $\mathcal{X}$ and label space $\mathcal{Y}$, an X2X backdoor attack can be defined as the following:

Definition 3.1. (**X2X Backdoor Attack**) *An X2X backdoor attack against a victim classifier $f : \mathcal{X} \to \mathcal{Y}$ is specified by a trigger embedding function $\delta : \mathcal{X} \to \mathcal{X}$ and a subset $\mathcal{A} \subset \mathcal{Y} \times \mathcal{Y}$ of backdoor class pairs, satisfying:* **(1)** $\forall a = (s,t) \in \mathcal{A}$, $s \neq t$, **(2)** *if* $|\mathcal{A}| > 1$[1], *for any* $a_i = (s_i, t_i) \in \mathcal{A}$ *and* $a_j = (s_j, t_j) \in \mathcal{A}$, $s_i \neq s_j$ *if* $a_i \neq a_j$. *A (perfectly) successful X2X attack will:* **(a)** *jointly minimize* $\mathbb{E}_{P_{XY|a}}[l(Y, f(\delta(X)))]$ *over both* δ *and* f, $\forall a \in \mathcal{A}$, *and* **(b)** *jointly minimize* $\mathbb{E}_{P_{XY|a}}[l(Y, f(X))]$ *over* f *for all class pairs* $a = (s,t)$ *with* $s = t$ *(i.e., high accuracy on clean samples), where* $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ *is the loss function of classifier* f.

Notes: In Def. 3.1, $P_{XY|a}$ is the joint distribution of (source class) sample $X \in \mathcal{X}$ and (target) label $Y \in \mathcal{Y}$ conditioned on class pair $a \in \mathcal{Y} \times \mathcal{Y}$. In particular, for any $a = (s,t)$, the marginal distribution $P_{Y|a}$ is a singleton at $Y = t$, and X only depends on s, i.e., $P_{XY|a}(x,y) = P_{X|s}(x) \cdot \mathbb{1}[y = t]$ for any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ where $\mathbb{1}[\cdot]$ is the indicator function. Thus, goal (a) can be achieved *only if* condition (2) holds; otherwise, there will be at least two class pairs in $\mathcal{A}$ with conflict minimization objectives. Moreover, although l can be any legitimate loss function for classification, for simplicity, in this paper, we consider the *0-1 loss* with $l(Y_1, Y_2) = 0$ if $Y_1 = Y_2$ and $l(Y_1, Y_2) = 1$ otherwise. Finally, we do not specify the form of δ here, since our UMD is applicable to a variety of trigger types – (e.g.) (1) *image perturbation trigger* embedded by $\delta(X) = [X + v]_c$, where v is a small perturbation and $[\cdot]_c$ is a clipping function, and (2) a *patch trigger* embedded by $\delta(X) = (1-m) \odot X + m \odot u$, where u is a small image patch, m is a binary mask, and $\odot$ represents element-wise multiplication.

By definition, X2X attacks are different from the N2N attacks proposed by Xue et al. (2022). The latter refers to backdoor attacks with multiple triggers, each associated with a unique target class, which can be viewed as the joint deployment of multiple A2O attacks (Xue et al., 2022a). By contrast, X2X attacks use a single trigger, with the main focus on different configurations of the (source, target) class pairs. In Sec. 5.4, we show that UMD (with trivial generalization) can easily detect N2N attacks.

[1]For attacks with only one backdoor class pair, i.e. $|\mathcal{A}| = 1$, our method is still effective empirically (see Sec. 5.4) due to a "collateral damage" phenomenon observed by Xiang et al. (2020).

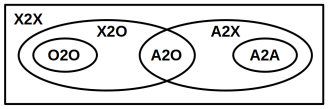

Figure 2. Venn map for the family of X2X attacks, with 'X' for "arbitrary", 'A' for "all", and 'O' for "one".

In practice, X2X attacks can be easily launched by poisoning the training set of f, with δ prescribed by the attacker. For many choices of δ (even without optimization), both (a) and (b) in Def. 3.1 can be achieved by only optimizing over f during the training. Thus, the attacker does not need access to the training process, which is required by many other advanced attacks. Moreover, X2X attacks are not detectable by existing methods without supervision. REDs mostly assume that the attack is A2O (Wang et al., 2019). MCDs need to train shadow models for a variety of attack settings, which cannot effectively cover all possible backdoor class pair configurations for X2X attacks (Xu et al., 2021). Thus, we propose UMD (introduced next) to close this gap.

4. Method

Next, we will first provide a formal problem statement for model detection against backdoor attacks, and then provide an overview of our proposed detection approach UMD, followed by a detailed introduction of UMD procedures.

Model detection problem For any potentially backdoor attacked classifier $f : \mathcal{X} \to \mathcal{Y}$, a defender aims to detect (without supervision) whether f is backdoor attacked and infer all the backdoor class pairs (i.e. set $\mathcal{A}$). Similar to the importance of the target class inference for A2O attacks (NeurIPS, 2022), for X2X attacks, the detected class pairs can be used to "fix" the classifier by "unlearning" the backdoor on these class pairs (Wang et al., 2019). The defender is assumed with the following *constraints*: (1) does not know *a priori* if f is attacked or not; (2) no access to the training set or any samples embedded with the backdoor trigger; (3) no access to any benign classifiers for reference (otherwise, one can use the benign classifier for the task); (4) no prior knowledge about the number of backdoor class pairs or the assignment rules for the target classes. Thus, the detection problem is *unsupervised* due to the unavailability of both models with and without a backdoor. Commonly, the defender is allowed to possess a small dataset $\mathcal{D}_c$ containing clean samples for detection (Wang et al., 2019).

Overview of UMD To address the unavailability of the true backdoor trigger, UMD first reverse-engineers a putative trigger for each class pair using samples in $\mathcal{D}_c$. Different from prior works (e.g. Wang et al. (2019)) that assume an A2O attack and perform trigger reverse-engineering for each

putative backdoor *target class*, our design makes class-pair-wise inference possible. However, as a result, the premise behind those prior works – the (image) trigger estimated for the backdoor target class will have a much smaller perturbation size than for all the other classes – cannot be extended to our method with class-pair-wise trigger reverse-engineering. Indeed, when there is an attack, the estimated trigger for the backdoor class pairs in $\mathcal{A}$ will have a small perturbation size due to the nature of the attacks. But when there is no attack, the estimated trigger for some non-backdoor class pairs may also have a small perturbation size – this is called an "*intrinsic backdoor*" (a.k.a. natural backdoor) (Xiang et al., 2022b; Tao et al., 2022), which easily causes a false detection if the class-pair-wise perturbation size statistics are directly used for inference (as will be shown by our experiments in Sec. 5.3). To avoid such false detection, we propose a statistic "*transferability*" (TR), which is defined for each *ordered pair of class pairs* based on the reverse-engineered trigger (Sec. 4.1.1). We show that in ideal cases, TR from a backdoor class pair to another backdoor class pair is guaranteed to be no less than TR from a backdoor class pair to a non-backdoor class pair (Sec. 4.1.2). Based on this property, UMD selects a subset of putative backdoor class pairs using the TR estimated for all ordered class pairs, by solving a proposed optimization problem (Sec. 4.2.1) Then, an aggregation of the perturbation size statistics over all the selected class pairs is assessed by an *unsupervised*, bias-reduced anomaly detector (Sec. 4.2.2). In summary, the set of class pairs being detected should have: (a) a large TR to any other class pair in the set and a small TR to any class pair not in the set, and (b) a small perturbation size for the reverse-engineered trigger. The pipeline of our UMD is illustrated in Fig. 1 and summarized by Alg. 1.

4.1. Transferability

4.1.1. Definition

As motivated above, the TR statistic is defined for each ordered pair of class pairs based on the reverse-engineered trigger. Since neither TR nor any part of the UMD pipeline is limited to any objective function or algorithm for trigger reverse-engineering, we define a general form for the trigger reverse-engineering problem as the following. That is, for each $a = (s, t) \in \mathcal{Y} \times \mathcal{Y}$ $(s \neq t)$, we solve:

$$\begin{aligned} \underset{\delta}{\text{minimize}} \quad & \mathbb{E}_{P_{X|a}}[d(X, \delta(X))] \\ \text{s.t.} \quad & \delta \in \arg\min_{\delta'} \mathbb{E}_{P_{XY|a}}[l(Y, f(\delta'(X)))] \end{aligned} \tag{1}$$

Here $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a distance metric with respect to the trigger type, e.g. the ℓ_2 norm $d(X, \delta(X)) = ||X - \delta(X)||_2$ for image perturbation triggers (Xiang et al., 2020). The distance is minimized since image triggers are typically designed to be human-imperceptible. Moreover, if a is a backdoor class pair, the set of δ satisfying the constraint of (1) will include the true backdoor trigger due to the goal (a) of the attacker in Def. 3.1. Empirically, for each $a = (s, t)$, problem (1) can be solved on clean samples in $\mathcal{D}_\text{c}$ from class s (Wang et al., 2019). Denoting the reverse-engineered trigger (i.e. the optimal solution to (1)) for each class pair a by δ_a, we define the TR statistic as the following:

Definition 4.1. (**Transferability (TR)**) *For any class pair $a_i = (s_i, t_i)$, $s_i \neq t_i$, with a reverse-engineered trigger δ_{a_i}, and 0-1 loss $l(\cdot, \cdot)$, TR from a_i to any other class pair $a_j = (s_j, t_j)$, $a_j \neq a_i$ and $s_j \neq t_j$, is defined by:*

$$T_{a_i a_j} \triangleq 1 - \mathbb{E}_{P_{XY|a_j}}[l(Y, f(\delta_{a_i}(X)))]. \tag{2}$$

Based on the notes below Def. 3.1, the expectation in Eq. (2) is equivalent to $\mathbb{E}_{P_{X|s_j}}[l(t_j, f(\delta_{a_i}(X)))]$. Thus, empirically, $T_{a_i a_j}$ can be estimated using the clean samples from class s_j in $\mathcal{D}_\text{c}$ and the trigger reverse-engineered for class pair a_i. The form of Eq. (2) is chosen for l being 0-1 loss with the value of TR scaled to $[0, 1]$ for simplicity, though other forms can be adopted for different choices of the loss function. In plain language, $T_{a_i a_j}$ represents the misclassification rate to class t_j when the trigger δ_{a_i} (estimated for class pair a_i) is applied to examples from class s_j.

4.1.2. Property

Next, we show that TR is intrinsically suitable for identifying backdoor class pairs. Consider an *arbitrary* set of class pairs $\mathcal{A}' = \{a_1, \cdots, a_k\}$ satisfying both conditions (1) and (2) in Def. 3.1, and with $P_A(a) > 0$ for $\forall a \in \mathcal{A}'$. For any trigger embedding function δ, we denote $X^\delta \triangleq \delta(X)$ as the random variable for samples with a trigger embedded by δ. Then, the set of Bayes classifiers (Devroye et al., 1996) for (optimal) estimation of Y from X^δ can be written as:

$$\mathcal{F}^\delta = \{f \in \mathcal{F} | \mathbb{E}_{P_{X^\delta Y}}[l(Y, f(X^\delta))] = R^{\mathcal{F}}(Y|X^\delta)\} \tag{3}$$

where $P_{X^\delta Y} = P_{XY} \cdot P_{X^\delta|X}$ is the joint distribution of X^δ and Y, $l(\cdot, \cdot)$ is the classification loss (i.e. 0-1 loss here), $\mathcal{F}$ is the set of all legitimate classifiers[2], and

$$R^{\mathcal{F}}(Y|X^\delta) = \min_{f \in \mathcal{F}} \mathbb{E}_{P_{X^\delta Y}}[l(Y, f(X^\delta))] \tag{4}$$

is the Bayes risk over all classifiers in $\mathcal{F}$ for estimating Y from X^δ. Here, we assume the minimum always exists for simplicity. Similarly, for each class pair $a \in \mathcal{A}'$, we denote the set of "*class-pair-conditional*" Bayes classifiers as $\mathcal{F}_a^\delta$ and the associated Bayes risk as $R_a^{\mathcal{F}}(Y|X^\delta)$, by replacing $P_{X^\delta Y}$ in both Eq. (3) and (4) with $P_{X^\delta Y|a} = P_{XY|a} \cdot P_{X^\delta|X}$. These classifiers in $\mathcal{F}_a^\delta$ are optimal for predicting Y from X^δ, with X^δ and Y both conditioned on the class pair a. Then, we have the following theorem for the transferability of reverse-engineered triggers:

[2]For example, all classifiers with the same architecture as the one to be inspected but with different parameter values.

Theorem 4.2. *(**Optimal Transferability Condition**) For any class pair $a \in \mathcal{A}'$, consider a trigger embedding function δ that minimizes $R_a^{\mathcal{F}}(Y|X^\delta)$. Then, δ minimizes:*

$$\min_{f \in \mathcal{F}_a^\delta} \sum_{a' \in \mathcal{A}' \setminus a} P_{A|A \neq a}(a') \mathbb{E}_{P_{X^\delta Y|a'}}[l(Y, f(X^\delta))] \tag{5}$$

if and only if δ also minimizes $R^{\mathcal{F}}(Y|X^\delta)$.

Proof (sketch). First, we derive the lower bound of (5) over δ. Then, sufficiency is proved by showing that the lower bound will be achieved if δ minimizes $R^{\mathcal{F}}(Y|X^\delta)$, while necessity is proved by showing that the lower bound cannot be achieved if δ does not minimize $R^{\mathcal{F}}(Y|X^\delta)$ via contradiction. The complete analysis is shown in Apdx. B. □

Remarks: For any class pair a and classifier f, the reverse-engineered trigger satisfying the constraint of problem (1) should also minimize $R_a^{\mathcal{F}}(Y|X^\delta)$ if f is a Bayes classifier conditioned on a. Thus, based on goal (a) in Def. 3.1, δ considered by the theorem may be a trigger reverse-engineered for some backdoor class pair a of a successful X2X attack. In this case, based on Def. 4.1, the conditional expectation in (5) for each $a' \in \mathcal{A}' \setminus a$ represents one minus the TR statistic from a to a'. Thus, the theorem shows the condition for δ maximizing the expected TR from a to all the other class pairs in $\mathcal{A}'$, which is that δ also minimizes $R^{\mathcal{F}}(Y|X^\delta)$ – the Bayes risk without any class-pair-conditioning. Apparently, this optimal transfer condition holds if $\mathcal{A}' \subset \mathcal{A}$ contains only backdoor class pairs of a successful X2X attack, with f being a Bayes classifier on $\mathcal{A}'$ and δ being the actual backdoor trigger. Thus, for a perfectly successful attack and optimal trigger reverse-engineering, if we apply Thm. 4.2 to any set $\mathcal{A}'$ of two class pairs with at least one being a backdoor class pair, we will have the guarantee that *TR from a backdoor class pair to another backdoor class pair is no less than TR from a backdoor class pair to a non-backdoor class pair*. Empirically, we will likely observe large TRs (possibly close to 1) for any ordered pair of class pairs in $\mathcal{A}'$ if the set is pure in backdoor class pairs. Otherwise, there will likely be at least two class pairs in $\mathcal{A}'$ with a small TR from either direction.

4.2. Detection Inference

4.2.1. Select Putative Backdoor Class Pairs

Due to the absence of supervision, it is hard to choose a threshold on TR to identify the backdoor class pairs directly if there is any. Moreover, a naive combination of TR with other statistics such as the perturbation size of the reverse-engineered trigger cannot effectively detect backdoor class pairs, while still causing a high false detection rate (as will be shown by our experiments in Sec. 5.3). Thus, we propose to use TR to select a set $\hat{\mathcal{A}}$ of putative backdoor class pairs for further inference. Based on our analysis for TR, if there is an attack, we expect: **(1)** a large TR for any ordered pair of class pairs in $\hat{\mathcal{A}}$, **(2)** a small TR from any class pair in $\hat{\mathcal{A}}$ to class pairs outside $\hat{\mathcal{A}}$, **(3)** $\hat{\mathcal{A}}$ satisfies the conditions in Def. 3.1 for valid X2X attacks. Accordingly, we propose to solve the following optimization problem:

$$\begin{aligned} \underset{\hat{\mathcal{A}} \subset \mathcal{Y} \times \mathcal{Y} \setminus \mathcal{B}}{\text{maximize}} \quad & H(\hat{\mathcal{A}}) = \min_{a \in \hat{\mathcal{A}}} \frac{\sum_{a' \in \hat{\mathcal{A}} \setminus a} (T_{aa'} + T_{a'a})}{2(|\hat{\mathcal{A}}| - 1)} \\ & - \max_{a \notin \hat{\mathcal{A}}} \frac{\sum_{a' \in \hat{\mathcal{A}}} T_{a'a}}{|\hat{\mathcal{A}}|} \\ \text{subject to} \quad & s \neq s', \forall a = (s,t), a' = (s',t') \in \hat{\mathcal{A}} \end{aligned} \tag{6}$$

where $\mathcal{B} = \{(s,t) \in \mathcal{Y} \times \mathcal{Y} | s = t\}$ is the set of all "identical" pairs. Clearly, for problem (6), the two terms in the objective function and the constraint are designed to satisfy the requirements (1)-(3), respectively. In particular, the second term of $H(\hat{\mathcal{A}})$ is critical in practice when the actual number of backdoor class pairs is unknown. Without this term, we will likely obtain a parsimonious set $\hat{\mathcal{A}}$ of two class pairs with the top "mutual-TR". Finally, we propose to solve problem (6) using an agglomerative algorithm *without any hyperparameter*, as detailed by lines 5-13 of Alg. 1.

Algorithm 1 UMD against X2X backdoor attacks

1: **Input:** a classifier f; a small, clean dataset $\mathcal{D}_c$, a desired significance level β for anomaly detection.
2: **Compute TR statistics:**
3: Get δ_a by solving (1) on $\mathcal{D}_c$ for $\forall a = (s,t) \in \mathcal{Y} \times \mathcal{Y} \setminus \mathcal{B}$.
4: Compute $T_{a_i a_j}$ by Eq. (2) on $\mathcal{D}_c$ for $\forall a_i \in \mathcal{Y} \times \mathcal{Y} \setminus \mathcal{B}$ and $\forall a_j = (s_j, t_j) \in \mathcal{Y} \times \mathcal{Y} \setminus \mathcal{B}, a_j \neq a_i$.
5: **Select a set $\hat{\mathcal{A}}$ of putative backdoor class pairs:**
6: Initialize $\hat{\mathcal{A}}_2 = \arg\max_{\{a_i=(s_i,t_i), a_j=(s_j,t_j)\}, s_i \neq s_j} T_{a_i a_j}$.
7: **for** $n = 3 : |\mathcal{Y}|$ **do**
8: $\quad \hat{\mathcal{A}}_{n-1}^c = \{a = (s,t) \notin \hat{\mathcal{A}}_{n-1} | s \neq s'$ for $\forall a' = (s',t') \in \hat{\mathcal{A}}_{n-1}\}$.
9: $\quad a^* = \arg\max_{a \in \hat{\mathcal{A}}_{n-1}^c} H(\hat{\mathcal{A}}_{n-1} \cup a)$.
10: $\quad \hat{\mathcal{A}}_n = \hat{\mathcal{A}}_{n-1} \cup a^*$.
11: **end for**
12: $n^* = \arg\max_{n \in \{3, \cdots, |\mathcal{Y}|\}} H(\hat{\mathcal{A}}_n)$
13: $\hat{\mathcal{A}} = \hat{\mathcal{A}}_{n^*}$
14: **Unsupervised anomaly detection:**
15: Compute r on $\mathcal{D}_c$ using all $\{\delta_a\}$ and $\hat{\mathcal{A}}$ by Eq. (7) and (8).
16: Compute $\theta(\beta, N)$ for $N = |\mathcal{Y} \times \mathcal{Y} \setminus \mathcal{B}| - |\hat{\mathcal{A}}|$ by Eq. (9).
17: **Output:** If $r > \theta(\beta, N)$, there is an attack with backdoor class pairs $\hat{\mathcal{A}}$; otherwise, there is no attack.

4.2.2. Unsupervised Anomaly Detection

Since $\hat{\mathcal{A}}$ will always be selected regardless of the presence of attack, we still need to infer whether $\hat{\mathcal{A}}$ is indeed a set of backdoor class pairs. Inspired by previous works, we design an anomaly detector based on median absolute deviation (MAD) (Hampel, 1974). The anomaly detector uses the trigger perturbation/patch size $z_a \triangleq \mathbb{E}_{P_{X|a}}[d(X, \delta_a(X))]$

empirically estimated for each class pair $a = (s, t)$ on the clean samples $\mathcal{D}_{\text{c}}$ as the detection statistic. Under the null hypothesis of "no attack", all detection statistics are associated with non-backdoor class pairs and follow some null distribution characterized by the median statistic and MAD. Different from prior works, our estimation of MAD (denoted by σ below) is performed on $\forall a \notin \hat{\mathcal{A}}$ which are likely non-backdoor class pairs, i.e.:

$$\sigma = \text{med}_{a \notin \hat{\mathcal{A}}}(|z_a^{-1} - \text{med}_{a' \notin \hat{\mathcal{A}}} z_{a'}^{-1}|) \tag{7}$$

where med represents median. The reciprocal is taken such that the outlier statistics corresponding to small trigger sizes, if there are any, will stay at the tail of the null distribution. Compared with other detectors that use all statistics to estimate MAD (since they do not select putative backdoor class pairs like us), our estimation will not suffer from the bias caused by the possible involvement of backdoor statistics. Then, we assess the atypicality of z_a for $\forall a \in \hat{\mathcal{A}}$ through aggregation using an anomaly score computed by:

$$r = (\text{med}_{a \in \hat{\mathcal{A}}} z_a^{-1} - \text{med}_{a' \notin \hat{\mathcal{A}}} z_{a'}^{-1})/(1.4826 \cdot \sigma) \tag{8}$$

where the constant 1.4826 is a scaling factor such that the scaled MAD can be viewed as an analog to the standard deviation of the null distribution under Gaussian assumption (Rousseeuw & Croux, 1993). The aggregation, i.e. the median of z_a^{-1} for $\forall a \in \hat{\mathcal{A}}$, helps to avoid false detection caused by any $a \in \hat{\mathcal{A}}$ with an outlier statistic (e.g. for an intrinsic backdoor) when there is actually no attack. In summary, the anomaly score r describes how many "standard deviations" the aggregated statistic is away from the median.

To test whether r is an outlier to the underlying null distribution, we propose a method to determine a confidence threshold in adaption to the number of "null statistics", i.e. $N = |\mathcal{Y} \times \mathcal{Y} \setminus \mathcal{B}| - |\hat{\mathcal{A}}|$, which is largely dependent on the number of classes $|\mathcal{Y}|$. Let $R_1, \cdots, R_N$ be i.i.d. random variables following some null density form, e.g., a standard Gaussian distribution in here. It is easy to show that for any given Θ, $\text{Prob}(\max_{i=1,\cdots,N} R_i > \Theta) \to 1$ as $N \to \infty$. In other words, with a constant threshold, a false detection will be easily made when N is large. Thus, we obtain a threshold $\theta(\beta, N)$ based on both a prescribed confidence level $1 - \beta$ (e.g. $\beta = 0.05$ by convention) and N by solving θ from $\text{Prob}(\max_{i=1,\cdots,N} R_i > \theta) \leq \beta$, which gives:

$$\theta(\beta, N) = \Phi^{-1}((1 - \beta)^{1/N}) \tag{9}$$

where Φ^{-1} is the inverse of the standard Gaussian CDF. Then, if $r > \theta(\beta, N)$, we claim with confidence $1 - \beta$ (a.k.a. β-significance) that the classifier is attacked with backdoor class pairs $\hat{\mathcal{A}}$; otherwise, no backdoor attack.

5. Experiment

First, we show that our unsupervised UMD outperforms five SOTA baselines (even with supervision) by at least 17%, 4%, and 8% on CIFAR-10, GTSRB, and Imagenette, respectively, in the average model inference accuracy against various X2X attacks. Second, in our ablation study on CIFAR-10, we justify our design choices for UMD. Third, we show that UMD can even detect X2X attacks with two advanced triggers and address four different types of adaptive attacks. Finally, we show that the class pairs detected by UMD can be used to "fix" the backdoored model.

Table 1. Designed functionalities and detection capabilities of UMD compared with five SOTA baselines. UMD is the only unsupervised method against X2X attacks with pair inference. Empirically, UMD can also detect O2O attacks as shown in Tab. 6.

	NC	ABS	PT-RED	MNTD	K-Arm	UMD (ours)
A2O	✓	✓	✓	✓	✓	✓
O2O			✓		✓	△
X2O			✓		✓	✓
A2Ar						✓
A2X						✓
X2X						✓
detect pairs						✓
unsupervised	✓		✓			✓

5.1. Setup

Dataset: We consider three benchmark image datasets, CIFAR-10 (Krizhevsky, 2012), GTSRB (Stallkamp et al., 2012), and Imagenette (Deng et al., 2009), which contain color images (with resolution 32×32, 32×32 (resized), and 224×224, respectively) with 10, 43, and 10 classes, respectively. In our experiments, we follow the standard train-test split for each dataset (see Apdx. C.1 for details).

Backdoor trigger: We consider two common triggers: 1) a large, perturbation-based trigger with a big 'X' shape, and 2) a local patch trigger with a random color and a random location for each attack. Examples of these triggers are shown in Fig. 5, with more details in Apdx. C.2.

Attack setting: We first consider the classical A2O attack addressed by most existing works for all three datasets. The target class for each A2O attack is randomly selected. Then we consider a general all-to-all (A2Ar) attack with a random bijection mapping between the source and target classes. Note that the classical A2A attack by Gu et al. (2017) uses rotational target assignment and is a special case of the A2Ar attack considered here. For each dataset, we also consider several X2X attack settings other than A2O and A2Ar. On CIFAR-10, we consider 2to2, 5to5, and 8to8 attacks; on GTSRB, we consider 20to20, 30to30, and 40to40 attacks; on Imagenette, we consider 3to3, 5to5, and 8to8 attacks. The backdoor class pairs for each X2X attack are randomly selected. Moreover, for each attack on CIFAR-10, GTSRB, and Imagenette, we create 300, 70, and 200 poisoning instances per source class, respectively.

Training: For each attack setting on each dataset, we train 10 classifiers under attack with each of the two triggers respectively. For the 8to8 and the A2Ar settings on Ima-

genette, the attacks with the patch trigger are mostly unsuccessful; thus, they are excluded from our experiments. In total, our main evaluation of the detection performance involves $((5\times3\times2-2\times1\times1)\times10=)$ 280 classifiers being attacked. For model architecture, we use ResNet-18 (He et al., 2016) for CIFAR-10 and Imagenette, and the winning model on the leaderboard (Leaderboard, 2018) for GTSRB. Detailed training configurations are shown in Apdx C.3. All the attacks we created are successful with attack success rates (ASRs) $> 78\%$ and negligible degradation in clean test accuracy (ACC) (see Tab. 8 in Apdx. C.3).

Evaluation metric: We define a model inference accuracy (**MIA**) as the proportion of *correct inference* for a group of classifiers. MIA is equivalent to the true positive rate (or one minus the false positive rate) if all classifiers in the group are attacked (or benign). For each *true positive* model inference by UMD, we also define a pair detection rate (**PDR**) which is the proportion of backdoor class pairs being successfully detected. Note that the false positive rate for pair inference (by incorrectly recognizing a non-backdoor class pair as a backdoor class pair) will always be small since UMD detects at most K (out of $K(K-1)$) class pairs, where K is the number of classes. Thus, we neglect it for brevity.

Baselines: We compare our UMD with the following SOTA baselines, including Neural Cleanse (NC) (Wang et al., 2019), ABS (Liu et al., 2019), PT-RED (Xiang et al., 2020), MNTD (Xu et al., 2021), and K-Arm (Shen et al., 2021). For a fair comparison, we set the confidence level for model inference to 95% (i.e. 5% *desired* false positive rate) for NC and PT-RED equipped with unsupervised threshold selection. For ABS, MNTD, and K-Arm which require supervision to select the detection threshold, we set the overall *actual* false positive rate (for all datasets and settings) to 5% while *maximizing* their true positive rates for model inference. The designed functionalities and detection capabilities of these methods are shown in Tab. 1, compared with UMD. More details about these methods are shown in Apdx. C.4.

Experimental Details: For our UMD, we consider the trigger reverse-engineering algorithms used by PT-RED and NC, respectively, to cover both the perturbation trigger and the patch trigger. That is, we execute Alg. 1 with both algorithms, and a classifier is deemed to be attacked if any of the two executions claim a detection. In particular, PT-RED assumes that the trigger is an additive image perturbation incorporated by $\delta(x)=[x+v]_c$ with a small $||v||_2$, where $[\cdot]_c$ is a clipping function (Xiang et al., 2020). Its reverse engineer algorithm is similar to the way to generate a universal adversarial perturbation (Moosavi-Dezfooli et al., 2017) – for any class pair (s,t), a perturbation v is initialized to zero and updated using gradient-based approaches, until a high misclassification fraction from class s to class t is achieved. NC assumes a patch trigger u embedded by $\delta(x)=(1-m)\odot x+m\odot u$ using a binary mask m with a small patch size $||m||_1$, where $\odot$ represents element-wise multiplication (Wang et al., 2019). The reverse engineering algorithm of NC also solves an optimization problem for each class pair (s,t) to achieve a high misclassification fraction from class s to class t while minimizing the patch size $||m||_1$. For all three datasets, the two algorithms consume merely 10 and 20 trigger-free images (correctly predicted by the classifier to be inspected) per class, respectively. More details about these two algorithms can be found in Apdx. C.5. Again, our UMD is not limited to any particular algorithms for trigger reverse-engineering, allowing the potential incorporation with more recent or even future techniques (Wang et al., 2023). For the selection of candidate backdoor class pairs, we repeat lines 6-13 of Alg. 1 five times, each with a different initialization, and pick the best optimal solution to avoid poor local optimum. For the anomaly detection step, we use the same confidence threshold of 95% (i.e. $\beta=0.05$) as the other detectors for a fair comparison. Results for other confidence levels are shown in Apdx. C.6.

Table 2. MIA of UMD for various X2X attacks and benign classifiers on CIFAR-10, GTSRB, and Imagenette, compared with five SOTA detectors. MIAs of ABS, MNTD, and K-Arm on benign classifiers are manually fixed to control the false positive rates; thus are "not applicable" (n.a.). UMD outperforms the five SOTA detectors (some even with supervision) on all three datasets by a clear margin in the average MIA over the X2X attacks.

(a) CIFAR-10

Setting	Benign	A2O	2to2	5to5	8to8	A2Ar	Avg
NC	0.60	0.55	0.20	0.20	0.30	0.30	0.31
ABS	n.a.	0.90	0.40	0.15	0.20	0.20	0.37
PT-RED	0.70	0.55	0.40	0.35	0.30	0.45	0.41
MNTD	n.a.	0.45	0.65	0.40	0.25	0	0.35
K-Arm	n.a.	1.0	0.90	0.70	0.65	0.45	0.74
UMD	0.90	0.90	0.90	0.95	0.85	0.95	**0.91**

(b) GTSRB

Setting	Benign	A2O	20to20	30to30	40to40	A2Ar	Avg
NC	0.90	0.85	0.30	0.25	0.35	0.35	0.42
ABS	n.a.	0.35	0.25	0.10	0.20	0.10	0.20
PT-RED	0.20	0.65	0.50	0.30	0.55	0.55	0.51
MNTD	n.a.	0.25	0.15	0.15	0.15	0	0.14
K-Arm	n.a.	1.0	0.95	0.85	0.80	0.75	0.87
UMD	0.90	0.95	0.80	0.90	0.90	1.0	**0.91**

(c) ImageNette

Setting	Benign	A2O	3to3	5to5	8to8	A2Ar	Avg
NC	0.90	0.85	0.30	0.15	0.05	0.15	0.30
ABS	n.a.	1.0	0.80	0.40	0.70	0.70	0.72
PT-RED	0.80	0.60	0.45	0.20	0.10	0	0.27
MNTD	n.a.	0.55	0.50	0.50	0.30	0.40	0.45
K-Arm	n.a.	0.90	0.60	0.65	0.90	0.80	0.77
UMD	0.80	0.90	0.75	0.80	0.80	1.0	**0.85**

5.2. Detection Performance

As shown in Tab. 2, UMD clearly outperforms the five SOTA baselines on all three datasets in terms of the average MIA over the X2X attacks on each dataset. In particular, most of these SOTA baselines exhibit some detection capability against A2O attacks they are designed for but fail

Table 3. Average PDR of UMD over successfully detected attacks for the three datasets.

CIFAR-10	Setting	A2O	2to2	5to5	8to8	A2Ar
	Avg PDR	0.93	0.92	1.0	0.88	0.98
GTSRB	Setting	A2O	20to20	30to30	40to40	A2Ar
	Avg PDR	0.90	0.72	0.83	0.79	0.86
Imagenette	Setting	A2O	3to3	5to5	8to8	A2Ar
	Avg PDR	0.96	0.87	0.75	0.70	0.65

against X2X attacks with more than one target class. In contrast, UMD performs uniformly well against all X2X attacks, with even better control of the false positive rate (reflected by the generally higher MIA on benign classifiers) compared with the other two unsupervised detectors, NC and PT-RED. We note that among the five SOTA baselines, K-Arm achieves the best average MIA against X2X attacks for all three datasets. A possible reason is that K-Arm can effectively reverse-engineer the trigger for O2O attacks, while all X2X attacks can be viewed as a joint deployment of multiple O2O attacks sharing the same trigger. However, K-Arm requires supervision to determine if a reverse-engineered trigger is associated with the backdoor, which is infeasible for practical backdoor detection problems. But even with the supervision to maximize its performance, K-Arm is still outperformed by our *unsupervised* UMD by 17%, 4%, and 8% on CIFAR-10, GTSRB, and Imagenette, respectively, in terms of the average MIA over the X2X attacks for each dataset. Finally, we show the pair inference performance of UMD in Tab. 3 since the other methods are not designed with such functionality. UMD achieves high average PDRs for most X2X settings on the three datasets. The relatively low PDRs, e.g. for A2Ar attacks on Imagenette, are likely due to the existence of intrinsic backdoor class pairs.

5.3. Ablation Study

First, we show the advantages of using the proposed TR statistic and the associated clustering approach for backdoor detection by comparing UMD with its two baseline variants. The first variant UMD† directly applies a MAD-based anomaly detector to triggers reverse-engineered for all class pairs, *without* using the TR statistic. The second variant UMD‡ uses TR simply as a secondary statistic *without* our clustering technique. More details about these two baseline variants are shown in Apdx. D.1. For a demonstration, we consider the 2to2, 5to5, 8to8, and A2Ar attacks on CIFAR-10 with the perturbation trigger (i.e. 10 backdoored classifiers per setting). We also use the 10 benign classifiers on CIFAR-10 to evaluate the false detection rate.

As shown in Tab. 4, though the desired false positive rate is set to 5%, the actual ones for the two baseline variants are very high (reflected by the low MIAs on the benign classifiers). Such high false positive rates cannot be alleviated even with alternative confidence levels, as shown in Apdx. D.2. In contrast, UMD achieves a 93% overall MIA as averaged over both attacked and benign classifiers with equal weights, showing a strong detection capability against X2X attacks with a controlled false detection rate. Moreover, UMD achieves good performance in class pair inference, which is generally better than the two baseline variants.

Table 4. MIA and average PDR of UMD, compared with the two baseline variants of UMD, against 2to2, 5to5, 8to8, and A2Ar attacks (with the perturbation trigger) on CIFAR-10. Both variants of UMD favor predicting an "attack", resulting in low MIAs on benign classifiers (i.e. high false positive rates). UMD achieves the best overall MIA (computed by adding the benign MIA with the average MIA for all attacks and then dividing by two).

	2to2		5to5		8to8		A2Ar		Benign	Overall
	MIA	PDR	MIA	PDR	MIA	PDR	MIA	PDR	MIA	MIA
UMD†	1.0	0.90	1.0	0.94	1.0	0.84	1.0	0.72	0	0.50
UMD‡	1.0	0.45	1.0	0.82	1.0	0.73	1.0	0.53	0.40	0.70
UMD	1.0	0.85	1.0	1.0	0.90	0.83	0.90	0.92	0.90	**0.93**

Next, we show the influence of the hyperparameters on UMD. Since UMD does not involve any tunable hyperparameters in the inference step, we study the influence of the hyperparameters used by the trigger reverse-engineering algorithms on our UMD. In particular, we focus on the number of images and the targeted misclassification fraction used by Xiang et al. (2020) for trigger reverse-engineering. Note that for X2X attacks, the ASR for a backdoor class pair is typically less than 100%. Thus, in principle, the defender should avoid using an overly large targeted misclassification fraction; otherwise, trigger reverse-engineering may fail to produce an accurate estimation of the actual backdoor trigger. As shown in Fig. 3, UMD performs uniformly well for targeted misclassification fractions less than 1, giving a large freedom to choose this hyperparameter.

As for the number of images, UMD prefers even fewer (though > 1) images for trigger reverse-engineering than the default setting by Xiang et al. (2020). Note that triggers reverse-engineered on a large number of images may easily contain class-discriminate features that transfer well between non-backdoor class pairs (especially those sharing the same target class) and lead to a wrong detection. In practice, the suitable number of images for trigger reverse-engineering can be easily determined as the following. Ideally, a TR map (e.g. the one in Fig. 1) is supposed to be dark almost everywhere except for a few entries that may be associated with the backdoor class pairs. Thus, we start with a relatively large number of images (e.g. 15 or even more) to compute the TR statistics. If there are more than $2(K^2 - K)$ bright entries in the TR map with TR larger than some prescribed threshold, we reduce the number of images, e.g., by dividing it by 2. Here, K is the number of classes, and $K^2 - K$ is the maximum number of entries in the TR map corresponding to a valid candidate set of backdoor class pairs. The above steps are repeated until there are at most $2(K^2 - K)$ bright entries in the TR map.

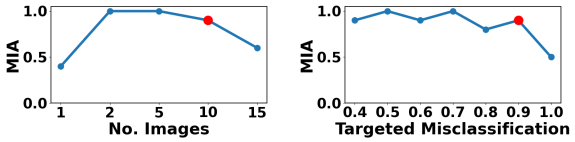

Figure 3. Influence of the number of images and the targeted misclassification fraction used by Xiang et al. (2020) for trigger reverse-engineering on our UMD. The default setting suggested by the authors, which is globally used in this work, is marked in red. UMD prefers even fewer images for trigger reverse-engineering and is insensitive to targeted misclassification fractions less than 1.

Table 5. MIA of UMD against a variety of X2X attacks with the WaNet trigger and the Blended trigger, respectively. UMD achieves generally high MIAs against all these X2X attacks for both triggers.

	2to2	5to5	8to8	A2Ar
WaNet	0.80	0.80	1.0	0.90
Blended	0.90	0.90	0.90	0.90

Table 6. UMD achieves generally high MIAs against four adaptive attacks, ISSBA, CLA, N2N, and O2O, on CIFAR-10.

	ISSBA	CLA	N2N	O2O
MIA for UMD	1.0	0.60	1.0	0.80

5.4. Performance of UMD against Adaptive Attacks

Here, we show the detection performance of UMD against two advanced trigger types, WaNet (Nguyen & Tran, 2021) and Blended (Chen et al., 2017), for a variety of X2X attack settings. We also evaluate UMD against four adaptive attacks, including the invisible sample-specific backdoor attack (ISSBA) proposed by Li et al. (2021), the clean label attack (CLA) proposed by Turner et al. (2019), the N2N attack proposed by Xue et al. (2022), and the O2O attack with one randomly selected backdoor class pair. We consider the default A2O setting for ISSBA and CLA since these two attacks cannot be easily extended to other X2X settings. For each N2N attack, we launch $N = 3$ A2O attacks together, each with a randomly selected target class and a random patch trigger. The experiments in this section are conducted on CIFAR-10. For each setting considered for each trigger or attack type, we create 10 attacks and train a model for each attack using the configurations in Sec. 5.1.

Due to the complexity of the trigger embedding functions for WaNet and Blended, we employ a more general trigger reverse-engineering algorithm proposed by Xiang et al. (2020), which estimates a *common* additive perturbation in the internal layer of the classifier (see Apdx. C.5.3 for more details). For the N2N attack, we introduce a trivial generalization of UMD by sequentially selecting multiple clusters (by repeating lines 5-13 of Alg. 1 multiple times). Each cluster is then inferred by the same anomaly detection procedure in Sec. 4.2.2, where the "null" statistics are those not belonging to any clusters. Intuitively, these clusters will either be associated with one of the N triggers or be the non-backdoor class pairs and rejected by anomaly detection.

Table 7. Using the class pairs detected by UMD to mitigate the 2to2, 5to5, 8to8, and A2Ar attacks on CIFAR-10, based on the method by Wang et al. (2019). All the backdoored classifiers are "fixed" as reflected by the low average ASR (%) with negligible degradation in the average ACC (%).

	2to2	5to5	8to8	A2Ar
ASR (Avg)	98.1→1.4	93.3→1.4	91.2→7.2	89.9→11.2
ACC (Avg)	92.4→92.2	92.7→92.3	92.8→92.3	93.7→91.9

In Tab. 5, we show the effectiveness of UMD against the WaNet trigger and the Blended trigger for a variety of X2X attacks. In Tab. 6, we show that UMD can also detect the four adaptive attacks with generally high MIA. Notably, although UMD always selects at least two putative backdoor class pairs for inference, it still detects the O2O attack (with only one backdoor class pair) well, thanks to the (almost inevitable) collateral damage which introduces additional "backdoor class pairs" (see Apdx. E.1 for more details). Moreover, for the 10 N2N attacks, the generalized UMD that selects 5 clusters of candidate backdoor class pairs correctly identifies 28 out of the 10x3 triggers, with only 2 clusters falsely recognized as associated with the backdoor.

5.5. Backdoor Mitigation

The backdoor class pairs detected by UMD can be used to "fix" the backdoored model. This process is called *backdoor mitigation* or *Trojan removal* (ICLR, 2022). Here, we use the method proposed by Wang et al. (2019) to mitigate the 2to2, 5to5, 8to8, and A2Ar attacks on CIFAR-10 that are detected by UMD. For each class pair being detected, we embed the reverse-engineered trigger into clean samples from the source class but without changing their labels. By fine-tuning using these samples, together with some clean samples without the trigger (to maintain the ACC), the model will learn to predict correctly even if a test sample is embedded with the trigger, i.e. the backdoor will be "unlearned". This is shown in Tab. 7, where for all attack settings, the average ASR drops to $\leq 11.2\%$ with negligible degradation in the average ACC – the models are fixed.

6. Conclusion

We proposed UMD, the first unsupervised backdoor model detector against X2X attacks. We defined TR and proved its intrinsic property in distinguishing backdoor class pairs from non-backdoor class pairs. Our UMD first selects a set of putative backdoor class pairs based on the TR statistics by solving a clustering problem we proposed, and then uses a robust, unsupervised anomaly detector to infer both the presence of the attack and the backdoor class pairs. Empirically, we show that UMD performs well on three datasets against X2X attacks with diverse settings.

Acknowledgements This work is partially supported by the NSF grant No.1910100, No. 2046726, Defense Advanced Research Projects Agency (DARPA) No. HR00112320012, C3.ai, and Amazon Research Award.

References

A. Saha, A. Subramanya, H. P. Hidden trigger backdoor attacks. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Borgatti, S. P. and Everett, M. G. Models of core/periphery structures. *Social Networks*, 2000.

Chen, B., Carvalho, W., Baracaldo, N., Ludwig, H., Edwards, B., Lee, T., Molloy, I., and Srivastava, B. Detecting backdoor attacks on deep neural networks by activation clustering. http://arxiv.org/abs/1811.03728, Nov 2018.

Chen, H., Fu, C., Zhao, J., and Koushanfar, F. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4658–4664, 7 2019.

Chen, X., Liu, C., Li, B., Lu, K., and Song, D. Targeted backdoor attacks on deep learning systems using data poisoning. https://arxiv.org/abs/1712.05526v1, 2017.

Chen, X., Salem, A., Chen, D., Backes, M., Ma, S., Shen, Q., Wu, Z., and Zhang, Y. Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In *Annual Computer Security Applications Conference (ACSAC)*, pp. 554–569, 2021.

Chou, E., Tramèr, F., Pellegrino, G., and Boneh, D. Sentinet: Detecting localized universal attacks against deep learning systems. In *2020 IEEE Security and Privacy Workshops (SPW)*, pp. 48–54. IEEE, 2020.

D. P. Kingma, J. B. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Li, F. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

Devroye, L., Györfi, L., and Lugosi, G. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

Doan, B. G., Abbasnejad, E., and C.Ranasinghe, D. Februus: Input purification defense against trojan attacks on deep neural network systems. In *Annual Computer Security Applications Conference (ACSAC)*, pp. 897–912, 2020.

Dong, Y., Yang, X., Deng, Z., Pang, T., Xiao, Z., Su, H., and Zhu, J. Black-box detection of backdoor attacks with limited information and data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Du, M., Jia, R., and Song, D. Robust anomaly detection and backdoor attack detection via differential privacy. In *International Conference on Learning Representations (ICLR)*, 2020.

Gao, Y., Xu, C., Wang, D., Chen, S., Ranasinghe, D. C., and Nepal, S. STRIP: A defence against trojan attacks on deep neural networks. In *Annual Computer Security Applications Conference (ACSAC)*, 2019.

Gu, T., Liu, K., Dolan-Gavitt, B., and Garg, S. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

Guan, J., Tu, Z., He, R., and Tao, D. Few-shot backdoor defense using shapley estimation. In *CVPR*, 2022.

Guo, W., Wang, L., Xing, X., Du, M., and Song, D. TABOR: A highly accurate approach to inspecting and restoring Trojan backdoors in AI systems. https://arxiv.org/abs/1908.01763, 2019.

Hampel, F. R. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69, 1974.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Hu, X., Lin, X., Cogswell, M., Yao, Y., Jha, S., and Chen, C. Trigger hunting with a topological prior for trojan detection. In *International Conference on Learning Representations*, 2022.

Huang, K., Li, Y., Wu, B., Qin, Z., and Ren, K. Backdoor defense via decoupling the training process. In *International Conference on Learning Representations (ICLR)*, 2022.

Huster, T. and Ekwedike, E. TOP: backdoor detection in neural networks via transferability of perturbation, 2021. URL https://arxiv.org/abs/2103.10274.

ICLR. IEEE Trojan Removal Competition. https://www.trojan-removal.com/, 2022.

Jia, J., Liu, Y., and Gong, N. Z. BadEncoder: Backdoor attacks to pre-trained encoders in self-supervised learning. In *IEEE Symposium on Security and Privacy (SP)*, 2022.

Kolouri, S., Saha, A., Pirsiavash, H., and Hoffmann, H. Universal litmus patterns: Revealing backdoor attacks in cnns. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 298–307, 2020.

Krizhevsky, A. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

Leaderboard. GTSRB Leaderboard. https://www.kaggle.com/c/nyu-cv-fall-2018/leaderboard, 2018.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, S., Liu, H., Dong, T., Zhao, B. Z., Xue, M., Zhu, H., and Lu, J. Hidden backdoors in human-centric language models. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 3123–3140, 2021a.

Li, Y., Li, Y., Wu, B., Li, L., He, R., and Lyu, S. Invisible backdoor attack with sample-specific triggers. In *IEEE International Conference on Computer Vision (ICCV)*, 2021b.

Li, Y., Lyu, X., Koren, N., Lyu, L., Li, B., and Ma, X. Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2021c.

Li, Y., Jiang, Y., Li, Z., and Xia, S.-T. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–18, 2022a.

Li, Y., Zhong, H., Ma, X., Jiang, Y., and Xia, S.-T. Few-shot backdoor attacks on visual object tracking. In *International Conference on Learning Representations (ICLR)*, 2022b.

Liu, K., Doan-Gavitt, B., and Garg, S. Fine-pruning: Defending against backdoor attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses (RAID)*, 2018.

Liu, Y., Lee, W., Tao, G., Ma, S., Aafer, Y., and Zhang, X. ABS: Scanning neural networks for back-doors by artificial brain stimulation. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 1265–1282, 2019.

Liu, Y., Ma, X., Bailey, J., and Lu, F. Reflection Backdoor: A Natural Backdoor Attack on Deep Neural Networks. In *European Conference on Computer Vision (ECCV)*, 2020.

Miller, D. J., Xiang, Z., and Kesidis, G. Adversarial learning in statistical classification: A comprehensive review of defenses against attacks. *Proceedings of the IEEE*, 108: 402–433, 2020.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. Universal adversarial perturbations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

NeurIPS. Trojan Detection Challenge NeurIPS 2022. https://trojandetection.ai/, 2022.

Nguyen, A. and Tran, A. Input-aware dynamic backdoor attack. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Nguyen, A. and Tran, A. Wanet - imperceptible warping-based backdoor attack. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=eEn8KTtJOx.

Peng, M., Xiong, Z., Sun, M., and Li, P. Label-Smoothed Backdoor Attack. *arXiv preprint arXiv:2202.11203*, 2022.

Rousseeuw, P. J. and Croux, C. Alternatives to the median absolute deviation. *Journal of the American Statistical Association*, 1993.

Shen, G., Liu, Y., Tao, G., An, S., Xu, Q., Cheng, S., Ma, S., and Zhang, X. Backdoor Scanning for Deep Neural Networks through K-Arm Optimization. In *International Conference on Machine Learning (ICML)*, 2021.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32: 323–332, 2012.

Tao, G., Shen, G., Liu, Y., An, S., Xu, Q., Ma, S., Li, P., and Zhang, X. Better trigger inversion optimization in backdoor scanning. In *2022 Conference on Computer Vision and Pattern Recognition (CVPR 2022)*, 2022.

Tran, B., Li, J., and Madry, A. Spectral signatures in backdoor attacks. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

Turner, A., Tsipras, D., and Madry, A. Clean-label backdoor attacks. https://people.csail.mit.edu/madry/lab/cleanlabel.pdf, 2019.

Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., and Zhao, B. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *IEEE Symposium on Security and Privacy (SP)*, 2019.

Wang, R., Zhang, G., Liu, S., Chen, P.-Y., Xiong, J., and Wang, M. Practical detection of trojan neural networks: Data-limited and data-free cases. In *European Conference on Computer Vision (ECCV)*, 2020.

Wang, Z., Zhai, J., and Ma, S. Bppattack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Wang, Z., Mei, K., Zhai, J., and Ma, S. UNICORN: A unified backdoor trigger inversion framework. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Mj7K4lglGyj.

Wu, D. and Wang, Y. Adversarial neuron pruning purifies backdoored deep models. In *NeurIPS*, 2021.

Xiang, Z., Miller, D., and Kesidis, G. A benchmark study of backdoor data poisoning defenses for deep neural network classifiers and a novel defense. In *IEEE MLSP*, Pittsburgh, 2019.

Xiang, Z., Miller, D. J., and Kesidis, G. Detection of backdoors in trained classifiers without access to the training set. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2020.

Xiang, Z., Miller, D. J., and Kesidis, G. L-RED: Efficient post-training detection of imperceptible backdoor attacks without access to the training set. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3745–3749, 2021.

Xiang, Z., Miller, D., and Kesidis, G. Post-training detection of backdoor attacks for two-class and multi-attack scenarios. In *International Conference on Learning Representations (ICLR)*, 2022a.

Xiang, Z., Miller, D. J., Chen, S., Li, X., and Kesidis, G. Detecting backdoor attacks against point cloud classifiers. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022b.

Xie, C., Huang, K., Chen, P., and Li, B. Dba: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Xu, A. and Raginsky, M. Minimum excess risk in bayesian learning. *IEEE Trans. Inf. Theory*, pp. 7935–7955, 2022.

Xu, X., Wang, Q., Li, H., Borisov, N., Gunter, C., and Li, B. Detecting AI Trojans using meta neural analysis. In *IEEE Symposium on Security and Privacy (SP)*, 2021.

Xue, M., He, C., Wang, J., and Liu, W. One-to-n n-to-one: Two advanced backdoor attacks against deep learning models. *IEEE Transactions on Dependable and Secure Computing*, 19(3):1562–1578, 2022a.

Xue, M., Ni, S., Wu, Y., Zhang, Y., Wang, J., and Liu, W. Imperceptible and multi-channel backdoor attack against deep neural networks, 2022b.

Yao, Y., Li, H., Zheng, H., and Zhao, B. Y. Latent backdoor attacks on deep neural networks. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2019.

Zeng, Y., Chen, S., Park, W., Mao, Z., Jin, M., and Jia, R. Adversarial unlearning of backdoors via implicit hypergradient. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=MeeQkFYVbzW.

Zhao, Z., Chen, X., Xuan, Y., Dong, Y., Wang, D., and Liang, K. Defeat: Deep hidden feature backdoor attacks by imperceptible perturbation and latent representation constraints. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Zheng, R., Tang, R., Li, J., and Liu, L. Data-free backdoor removal based on channel lipschitzness. In *ECCV*, 2022.

Zhong, H., Liao, C., Squicciarini, A., Zhu, S., and Miller, D. Backdoor embedding in convolutional neural network models via invisible perturbation. In *CODASPY*, 2020.

A. Ethics Statement

The main purpose of this research is to understand the behavior of deep learning systems facing malicious activities and enhance their safety without degrading their utility. The X2X backdoor attack considered in this paper is the union of many well-known backdoor attacks with different settings – all these attacks are open-sourced. Thus, our work will be beneficial to the community in defending against these attacks via detection. However, we do not claim that our detector is effective against all backdoor attacks that may appear in the future. In fact, there is no published backdoor detector making such a claim, just like that there is no published backdoor attack proved to be evasive against all future detectors. The code related to this work can be found at: `https://github.com/polaris-73/MT-Detection` Finally, the paper is written by humans without the involvement of large language models.

B. Analysis of TR and Proofs

Here, we present the complete analysis showing that the TR statistic is intrinsically suitable for detecting backdoor class pairs. Such an intrinsic property of TR is not possessed by many popular statistics for backdoor model detection. For example, the (patch) size of the reverse-engineered triggers used by Wang et al. (2019) is based on the premise that the actual trigger used by the attacker is small.

Our main theoretical results are summarized in Thm. 4.2 in Sec. 4.1 (also restated as Thm. B.5 below). Intuitively, the theorem says that the trigger reverse-engineered for a backdoor class pair will likely induce a small classification loss to all the other backdoor class pairs. Thus, empirically, we will likely observe a large TR statistic (possibly close to 1) from one backdoor class pair to another. In the following, we first present the complete problem settings that will facilitate our analysis. Then we prove Thm. 4.2.

B.1. Complete Settings

Set of class pairs: We consider an *arbitrary* set of class pairs $\mathcal{A}' = \{a_1, \cdots, a_k\}$ ($k \leq |\mathcal{Y}|$) satisfying:

- For $\forall a = (s, t) \in \mathcal{A}'$, $s \neq t$ (i.e. condition (1) in Def. 3.1);
- If $|\mathcal{A}'| > 1$, for any $a_i = (s_i, t_i) \in \mathcal{A}'$ and $a_j = (s_j, t_j) \in \mathcal{A}'$, $s_i \neq s_j$ if $a_i \neq a_j$ (i.e. condition (2) in Def. 3.1);
- $P_A(a) > 0$ for $\forall a \in \mathcal{A}'$ (i.e. positive probability for all class pairs in $\mathcal{A}'$).

Note that here, we do not specify if any class pair $a \in \mathcal{A}'$ is a backdoor class pair or not.

Random variables: Following the main paper, we use $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ to denote the random variables for samples and labels respectively. $A \in \mathcal{A}'$ denotes the random variable for class pairs in $\mathcal{A}'$. Moreover, for any trigger embedding function δ, we use $X^\delta \triangleq \delta(X)$ to denote the random variable for samples generated from X by embedding a trigger using δ. Then, each δ specifies a conditional distribution $P_{X^\delta|X}$. In summary of the above, we have the following dependency:

$$(A, X, Y, X^\delta) \sim P_A \cdot P_{XY|A} \cdot P_{X^\delta|X} \tag{10}$$

Set of estimators/classifiers: Considering that TR is defined in terms of the (expected) classification loss on samples with a (reverse-engineered) trigger embedded (see Eq. (2)), we use $\mathcal{F}$ to represent the set of estimators (i.e. classifiers in our problem) for estimating Y from the trigger-embedded sample X^δ with arbitrary δ. For example, $\mathcal{F}$ may contain all classifiers with the same architecture as the one to be inspected (i.e. the classifier that will also be used for trigger reverse-engineering) but with different parameter values. For convenience, we also define Δ as the set of all trigger embedding functions. For example, for image perturbation triggers, Δ may include perturbations with different shapes and sizes. For another example, for sample-specific triggers, Δ may be the set of all autoencoders with the same architecture but different parameter values. Moreover, we define a set $\mathcal{G} \triangleq \Delta \times \mathcal{F}$ of "end-to-end" functions, such that each $g \in \mathcal{G}$ can be represented by $g = f \circ \delta$ for some $\delta \in \Delta$ and $f \in \mathcal{F}$. These sets of estimators and their relation to the random variables we have defined previously are illustrated in Fig. 4.

Bayes classifiers: Bayes classifier refers to the classifier with the minimum classification loss when predicting/estimating the label of a random input sample (Devroye et al., 1996). Typically, the Bayes classifier (usually with respect to a space of classifiers) is specified by the joint distribution of the input and the label. For example, in the main paper, given a trigger

$$\begin{tikzcd} & X^\delta \arrow[dr, "f \in \mathcal{F}"] & \\ X \arrow[ur, "\delta \in \Delta"] \arrow[rr, "g \in \mathcal{G}"'] & & Y \end{tikzcd}$$

Figure 4. Illustration of the estimators/classifiers and their relation to the random variables.

embedding function δ, we denote the (set of) Bayes classifier(s) for estimating Y from X^δ (with joint distribution $P_{X^\delta Y}$) as $\mathcal{F}^\delta \subset \mathcal{F}$ (see Eq. (3)). And we denote the associated Bayes risk as $R^{\mathcal{F}}(Y|X^\delta)$ (see Eq. (4)). Here, among all "end-to-end" classifiers in $\mathcal{G}$ for estimating Y from X (by first embedding a trigger and then classifying), where $X, Y \sim P_{XY}$, we denote the set of Bayes classifiers (i.e. with the smallest classification loss) as:

$$\mathcal{G}^* = \{g \in \mathcal{G} | \mathbb{E}_{P_{XY}}[l(Y, g(X))] = R^{\mathcal{G}}(Y|X)\} \tag{11}$$

where

$$R^{\mathcal{G}}(Y|X) = \min_{g \in \mathcal{G}} \mathbb{E}_{P_{XY}}[l(Y, g(X))] \tag{12}$$

denotes the associated Bayes risk. Similarly, for each class pair $a \in \mathcal{A}'$ with conditional joint distribution $P_{XY|a}$ for sample X and label Y, we denote the set of Bayes classifiers, with respect to the set $\mathcal{G}$, for estimating Y from X given a as:

$$\mathcal{G}_a^* = \{g \in \mathcal{G} | \mathbb{E}_{P_{XY|a}}[l(Y, g(X))] = R_a^{\mathcal{G}}(Y|X)\} \tag{13}$$

where

$$R_a^{\mathcal{G}}(Y|X) = \min_{g \in \mathcal{G}} \mathbb{E}_{P_{XY|a}}[l(Y, g(X))] \tag{14}$$

is the associated Bayes risk with conditioning on a. Finally, for each class pair $a \in \mathcal{A}'$ and any $\delta \in \Delta$, the set of Bayes classifiers, with respect to $\mathcal{F}$, for estimating Y from X^δ (both conditioned on a) can be written as:

$$\mathcal{F}_a^\delta = \{f \in \mathcal{F} | \mathbb{E}_{P_{X^\delta Y|a}}[l(Y, f(X^\delta))] = R_a^{\mathcal{F}}(Y|X^\delta)\}. \tag{15}$$

where

$$R_a^{\mathcal{F}}(Y|X^\delta) = \min_{f \in \mathcal{F}} \mathbb{E}_{P_{X^\delta Y|a}}[l(Y, f(X^\delta))] \tag{16}$$

is the associated Bayes risk conditioned on a.

B.2. Proof of Thm. 4.2

To begin with, we show a mild assumption required by the theorem:

Assumption B.1. *$\exists g \in \mathcal{G}^*$ satisfying $g \in \mathcal{G}_a^*$ for $\forall a \in \mathcal{A}'$.*

Remarks: The assumption basically says that there exists a Bayes classifier g for estimating Y from X (unconditionally) that is also Bayes when X and Y are both conditioned on some arbitrary class pair $a \in \mathcal{A}'$. For convenience, we define $\mathcal{B} = \{(s,t) \in \mathcal{Y} \times \mathcal{Y} | s = t\}$ as the set of all "identical" pairs. Then, the assumption is *guaranteed to hold* if the samples together with their (correct) labels following the joint distribution $P_{XY|\mathcal{B}}$ are *perfectly separable* by some classifier $f \in \mathcal{F}$. To see this, let's first consider the case where $\mathcal{A}' = \mathcal{B}$. We can easily construct the desired function $g = f \circ \delta$ from f, with δ being an identity mapping. Then, given that $\mathbb{E}_{P_{XY|\mathcal{B}}}[l(Y, f(X))] = 0$ for l being the 0-1 loss (which is due to that $X, Y \sim P_{XY|\mathcal{B}}$ is perfectly separable by f), $\mathbb{E}_{P_{XY|\mathcal{B}}}[l(Y, g(X))] = 0$ will also hold since $\delta(X) = X$ by our construction. Since the loss is defined to be non-negative, we will then have $\mathbb{E}_{P_{XY|a}}[l(Y, g(X))] = 0$ for $\forall a \in \mathcal{A}' = \mathcal{B}$. Next, we consider the case where $\mathcal{A}' \neq \mathcal{B}$. We first construct an injective mapping $\phi : \mathcal{A}' \to \mathcal{B}$, such that for any $a = (s,t) \in \mathcal{A}'$ and $a' = (s', t') = \phi(a) \in \mathcal{B}$ (with $s' = t'$ by the definition of $\mathcal{B}$), $a' = \phi(a)$ *if and only if* $s = s'$. The existence of such ϕ is guaranteed by that: (a) both $\mathcal{A}'$ and $\mathcal{B}$ satisfy condition (1) in Def. 3.1 (see the definition of $\mathcal{A}'$ in Sec. B.1), and (b) $|\mathcal{A}'| \leq |\mathcal{B}| = |\mathcal{Y}|$ (which allows each element in $\mathcal{A}'$ to have an image in $\mathcal{B}$). Thus, we can easily rearrange the output neurons of f based on the mapping ϕ. In particular, for any $a = (s,t) \in \mathcal{A}'$ and its associated $a' = (s', t') = \phi(a)$, we relabel class t' (where $t' = s' = s$) to class t. If two different class pairs $a_i = (s_i, t) \in \mathcal{A}'$ and $a_j = (s_j, t) \in \mathcal{A}'$ share the same target class t, the rearranged classifier will predict to class t if f predicts to any of s_i and s_j. Then, we will also obtain a desired

classifier g satisfying Assumption B.1 by affiliating an identity trigger embedding function δ to the classifier rearranged from f following the procedure above.

In the proof of Thm. 4.2, we will also need the following lemmas.

Lemma B.2. *(***Generalized Data Processing Inequality** *(Xu & Raginsky, 2022)) Suppose random variables X^δ and Y are conditionally independent given X. Then, for any loss function l, we have:*

$$R^{\mathcal{F}}(Y|X^\delta) \geq R^{\mathcal{G}}(Y|X).$$

Lemma B.3. *There always exists δ such that*

$$R^{\mathcal{F}}(Y|X^\delta) = R^{\mathcal{G}}(Y|X).$$

Moreover, for each $a \in \mathcal{A}'$, there also exists δ such that

$$R_a^{\mathcal{F}}(Y|X^\delta) = R_a^{\mathcal{G}}(Y|X).$$

Proof. For the unconditional case, we construct $\delta = f^{-1} \circ g^*$ with arbitrary $f \in \mathcal{F}$ and arbitrary $g^* \in \mathcal{G}^*$, such that $X^\delta = \delta(X) = f^{-1}(g^*(X))$. Then, we have

$$\begin{aligned} R^{\mathcal{G}}(Y|X) &= \mathbb{E}_{P_{XY}}[l(Y, g^*(X))] && \triangleright \text{ Eq. (11) and (12)} \\ &= \mathbb{E}_{P_{XY} \cdot P_{X^\delta|X}}[l(Y, f(X^\delta))] && \triangleright \text{ Construction of } \delta \\ &\geq R^{\mathcal{F}}(Y|X^\delta) && \triangleright \text{ Eq. (3) and (4)} \end{aligned}$$

According to Lemma B.2, since X^δ and Y are indeed conditionally independent given X, equality must hold in above for the constructed δ.

For the conditional case and for each $a \in \mathcal{A}'$, a similar proof can be applied with δ constructed by choosing g^* from $\mathcal{G}_a^*$. □

Lemma B.4. *If δ minimizes $R^{\mathcal{F}}(Y|X^\delta)$, then, for any $a \in \mathcal{A}'$: (1) $R_a^{\mathcal{F}}(Y|X^\delta) = R_a^{\mathcal{G}}(Y|X)$; (2) $\mathcal{F}^\delta \subset \mathcal{F}_a^\delta$.*

Proof. Considering an arbitrary $f^* \in \mathcal{F}^\delta$ and an arbitrary $g^* \in \mathcal{G}^*$ satisfying $g^* \in \mathcal{G}_a^*$ for $\forall a \in \mathcal{A}'$ (existence of such g^* is guaranteed by Assumption B.1), for the estimation of Y from both X and X^δ, we have the following relationship between the Bayes risks *with* and *without* conditioning:

$$\begin{aligned} R^{\mathcal{G}}(Y|X) &= \mathbb{E}_{P_{XY}}[l(Y, g^*(X))] && \triangleright \text{ Eq. (11) and (12)} \\ &= \sum_{a \in \mathcal{A}'} P_A(a) \mathbb{E}_{P_{XY|a}}[l(Y, g^*(X))] && \triangleright \text{ Conditioning} \\ &= \sum_{a \in \mathcal{A}'} P_A(a) R_a^{\mathcal{G}}(Y|X) && \triangleright \text{ Eq. (13) and (14)} \\ R^{\mathcal{F}}(Y|X^\delta) &= \mathbb{E}_{P_{X^\delta Y}}[l(Y, f^*(X^\delta))] && \triangleright \text{ Eq. (3) and (4)} \\ &= \sum_{a \in \mathcal{A}'} P_A(a) \mathbb{E}_{P_{X^\delta Y|a}}[l(Y, f^*(X^\delta))] && \triangleright \text{ Conditioning} \\ &\geq \sum_{a \in \mathcal{A}'} P_A(a) R_a^{\mathcal{F}}(Y|X^\delta) && \triangleright \text{ Eq. (15) and (16)} \end{aligned}$$

Combining the above, we have:

$$\begin{aligned} R^{\mathcal{F}}(Y|X^\delta) - R^{\mathcal{G}}(Y|X) &\geq \sum_{a \in \mathcal{A}'} P_A(a)(R_a^{\mathcal{F}}(Y|X^\delta) - R_a^{\mathcal{G}}(Y|X)) && (17) \\ &\geq 0 && \triangleright \text{ Lemma B.2} \end{aligned}$$

Since that δ minimizes $R^{\mathcal{F}}(Y|X^\delta)$ is given, by Lemma B.2 and Lemma B.3, we have $R^{\mathcal{F}}(Y|X^\delta) - R^{\mathcal{G}}(Y|X) = 0$. Thus, the inequalities above both become equality. Since $P_A(a) > 0$ for $\forall a \in \mathcal{A}'$ (see the settings of $\mathcal{A}'$ in Sec. B.1), item (1) of the lemma, i.e. $R^{\mathcal{F}}_a(Y|X^\delta) = R^{\mathcal{G}}_a(Y|X)$ for $\forall a \in \mathcal{A}'$, is proved.

Next, we prove item (2) of the lemma by contradiction. Assume that there exist $a \in \mathcal{A}'$ and $f' \in \mathcal{F}^\delta$ such that $f' \notin \mathcal{F}^\delta_a$. Then,

$$\begin{aligned}
R^{\mathcal{F}}(Y|X^\delta) &= \mathbb{E}_{P_{X^\delta Y}}[l(Y, f'(X^\delta))] && \triangleright \text{ Eq. (3) and (4)} \\
&= \sum_{a' \in \mathcal{A}' \setminus a} P_A(a') \mathbb{E}_{P_{X^\delta Y|a'}}[l(Y, f'(X^\delta))] + P_A(a) \mathbb{E}_{P_{X^\delta Y|a}}[l(Y, f'(X^\delta))] && \triangleright \text{ Conditioning} \\
&> \sum_{a' \in \mathcal{A}' \setminus a} P_A(a') R^{\mathcal{F}}_{a'}(Y|X^\delta) + P_A(a) R^{\mathcal{F}}_a(Y|X^\delta) && \triangleright \text{ Eq. (16) and } f' \notin \mathcal{F}^\delta_a \\
&= \sum_{a' \in \mathcal{A}'} P_A(a') R^{\mathcal{F}}_{a'}(Y|X^\delta)
\end{aligned}$$

Thus, the inequality (17) becomes strict and moreover, $R^{\mathcal{F}}(Y|X^\delta) - R^{\mathcal{G}}(Y|X) > 0$. Here, we have reached a contradiction since $R^{\mathcal{F}}(Y|X^\delta) - R^{\mathcal{G}}(Y|X) = 0$ must hold when δ minimizes $R^{\mathcal{F}}(Y|X^\delta)$ as discussed above. □

Theorem B.5. *(***Restatement of Thm. 4.2***) For any class pair $a \in \mathcal{A}'$, consider a trigger embedding function δ that minimizes $R^{\mathcal{F}}_a(Y|X^\delta)$. Then, δ minimizes:*

$$\min_{f \in \mathcal{F}^\delta_a} \sum_{a' \in \mathcal{A}' \setminus a} P_{A|A \neq a}(a') \mathbb{E}_{P_{X^\delta Y|a'}}[l(Y, f(X^\delta))]$$

if and only if δ also minimizes $R^{\mathcal{F}}(Y|X^\delta)$.

Proof. For any $\delta \in \Delta$ and $a \in \mathcal{A}'$, we have the following lower bound for the minimum:

$$\begin{aligned}
&\min_{f \in \mathcal{F}^\delta_a} \sum_{a' \in \mathcal{A}' \setminus a} P_{A|A \neq a}(a') \mathbb{E}_{P_{X^\delta Y|a'}}[l(Y, f(X^\delta))] \\
\geq &\sum_{a' \in \mathcal{A}' \setminus a} P_{A|A \neq a}(a') R^{\mathcal{F}}_{a'}(Y|X^\delta) && \triangleright \text{ Eq. (16)} \quad (*) \\
\geq &\sum_{a' \in \mathcal{A}' \setminus a} P_{A|A \neq a}(a') R^{\mathcal{G}}_{a'}(Y|X) && \triangleright \text{ Lemma B.2} \quad (**)
\end{aligned}$$

Proof of sufficiency We show that if δ minimizes $R^{\mathcal{F}}(Y|X^\delta)$, the lower bound above will be reached, i.e. equality holds for both $(*)$ and $(**)$. First, by item (2) of Lemma B.4, there exist $f^* \in \mathcal{F}^\delta_a$ satisfying $f^* \in \mathcal{F}^\delta_{a'}$ for $\forall a' \in \mathcal{A}' \setminus a$. Thus, based on Eq. (15), equality holds for $(*)$. Next, by item (1) of Lemma B.4, equality holds for $(**)$.

Proof of necessity We prove by contradiction. Suppose δ *does not* minimize $R^{\mathcal{F}}(Y|X^\delta)$, by Lemma B.2 and Lemma B.3, we will have:

$$R^{\mathcal{F}}(Y|X^\delta) - R^{\mathcal{G}}(Y|X) > 0$$

Then, based on inequality (17), *at least one* of the following must hold:

$$\begin{aligned}
&\text{(A)} \quad R^{\mathcal{F}}(Y|X^\delta) > \sum_{a' \in \mathcal{A}'} P_A(a') R^{\mathcal{F}}_{a'}(Y|X^\delta) \\
\text{or} \quad &\text{(B)} \quad R^{\mathcal{F}}_{a'}(Y|X^\delta) - R^{\mathcal{G}}_{a'}(Y|X) > 0 \quad \text{for some } a' \in \mathcal{A}'
\end{aligned}$$

If (B) holds, we will further have $R^{\mathcal{F}}_{a'}(Y|X^\delta) - R^{\mathcal{G}}_{a'}(Y|X) > 0$ for some $a' \in \mathcal{A}' \setminus a$. This is because for the given a, $R^{\mathcal{F}}_a(Y|X^\delta) - R^{\mathcal{G}}_a(Y|X) = 0$ due to both that δ minimizes $R^{\mathcal{F}}_a(Y|X^\delta)$ and the existence of such minimum (based on Lemma B.3). Then, equality *cannot* be achieved for $(**)$ and we have reached a contradiction.

But if (B) does not hold, (A) must hold. Again, for $(*)$ being equal, there must exist $f^* \in \mathcal{F}_a^\delta$ satisfying $f^* \in \mathcal{F}_{a'}^\delta$ for $\forall a' \in \mathcal{A}' \setminus a$. In other words, there exists (at least one) $f^* \in \cup_{a' \in \mathcal{A}'} \mathcal{F}_{a'}^\delta \neq \emptyset$. Thus, we have:

$$
\begin{aligned}
R^{\mathcal{F}}(Y|X^\delta) &\leq \mathbb{E}_{P_{X^\delta Y}}[l(Y, f^*(X^\delta))] && \triangleright \text{ Eq. (4)} \\
&= \sum_{a' \in \mathcal{A}'} P_A(a') \mathbb{E}_{P_{X^\delta Y|a'}}[l(Y, f^*(X^\delta))] && \triangleright \text{ Conditioning} \\
&= \sum_{a' \in \mathcal{A}'} P_A(a') R_{a'}^{\mathcal{F}}(Y|X^\delta) && \triangleright \text{ Eq. (16)}
\end{aligned}
$$

This is a clear contradiction with (A). □

C. Supplementary of the Main Experiments on Backdoor Model Detection

C.1. Details for the Datasets

CIFAR-10 is a benchmark dataset with 32×32 color images from 10 classes for different categories of objects (Krizhevsky, 2012). The training set contains 50,000 images and the test set contains 10,000 images, both evenly distributed in the 10 classes.

GTSRB is an image dataset for German traffic signs from 43 classes (Stallkamp et al., 2012). The training set and the test set contain 39,209 and 12,630 images respectively. The image sizes vary in a relatively large range. Thus, we resize all the images to 32×32 in our experiments for convenience.

Imagenette consists of 224×224 color images from ten selected classes of the ImageNet dataset (Deng et al., 2009) that are easily classified. The training set and the test set contain 9,469 and 3,925 images respectively.

C.2. Details for the Backdoor Triggers

In our experiments in Sec. 5, we considered a global, perturbation-based trigger with a big 'X' shape (dubbed "Pert"), and a local patch trigger (dubbed "Patch"). The Pert trigger is generated by positively perturbing each pixel on both diagonals of the image by the same perturbation size for all three color channels. For CIFAR-10, GTSRB, and Imagenette, we set the perturbation size to 5/255, 15/255, and 15/255, respectively. For the Patch trigger, we replace a small area of the image (for all three channels) with an image patch with the same shape and size. For CIFAR-10, GTSRB, and Imagenette, we use 3×3, 2×2, and 8×8 square patches respectively. For each attack, the location for the patch replacement and the color for each pixel in the patch are both randomly selected. Examples of both triggers and the image embedded with each trigger (compared with the original, trigger-free image) are shown in Fig. 5.

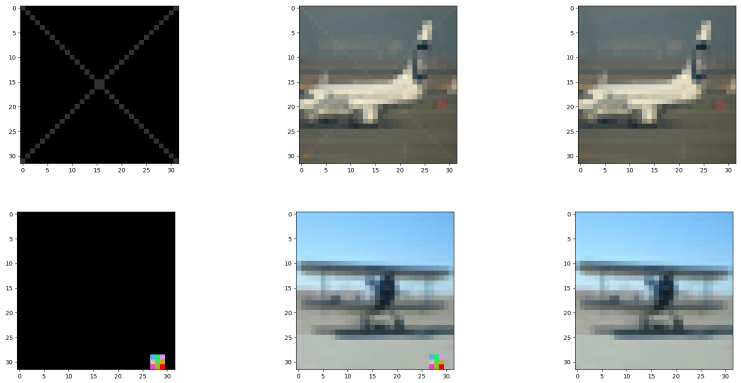

Figure 5. Top: example of the Pert trigger (amplified to 50/255 perturbation size for better visualization), an image (from CIFAR-10) embedded with the Pert trigger (with perturbation size 5/255), and the original clean image without the trigger. Bottom: example of the Patch trigger, an image (also from CIFAR-10) embedded with the Patch trigger, and the original clean image without the trigger.

C.3. Training Configurations and Attack Effectiveness

For all three datasets, the training is performed on the training set specified in Apdx. C.1. For CIFAR-10 and Imagenette, the training images are augmented by random horizontal flipping. For GTSRB, the training images are augmented by random rotation of ± 5 degrees. We use ResNet-18 (He et al., 2016) as the model architecture for CIFAR-10 and Imagenette. For GTSRB, we use the model with the top performance on the leaderboard (Leaderboard, 2018). For CIFAR-10, GTSRB, and Imagenette, training is performed using the Adam optimizer (D. P. Kingma, 2015) for 200, 100, and 80 epochs, respectively, with a learning rate of 10^{-3} and a mini-batch size of 64. When there is no attack, this training configuration achieves around 93%, 98%, and 88% accuracy (ACC) for the three datasets, respectively. The same set of configurations is also used for training the classifiers under the attacks we created. The effectiveness of an attack is jointly measured by the ASR and ACC of the model. The ASR for an X2X attack is the misclassification rate from the backdoor source classes to their designated target class when the samples from these source classes are embedded with the backdoor trigger. In Tab. 8, for each combination of the dataset, trigger, and attack setting, we show the average and the minimum ASR, together with the

Table 8. The average (avg) and the minimum (min) ASR and ACC for each combination of the trigger and the attack setting on CIFAR-10, GTSRB, and Imagenette, and the average and the minimum ACC for the benign classifiers on each dataset for reference. The ASRs and the ACCs are all in percentage. All attacks we created are successful with ASR $\geq 78\%$.

(a) CIFAR-10

Setting	Attack	avg ASR	min ASR	avg ACC	min ACC
Benign	-	-	-	93.54	93.24
A2O	Patch	99.73	98.08	93.08	92.59
	Pert	97.87	95.87	93.06	92.76
2to2	Patch	97.90	95.80	92.76	91.70
	Pert	98.06	95.60	91.85	91.62
5to5	Patch	93.49	90.20	93.14	92.66
	Pert	93.10	87.12	91.91	91.48
8to8	Patch	91.71	90.12	93.15	92.62
	Pert	90.06	87.85	91.96	91.34
A2Ar	Patch	91.44	89.88	93.32	92.92
	Pert	87.27	86.62	93.35	93.08

(b) GTSRB

Setting	Attack	avg ASR	min ASR	avg ACC	min ACC
Benign	-	-	-	98.46	98.27
A2O	Patch	99.99	99.93	98.12	97.69
	Pert	98.45	98.01	97.63	97.36
20to20	Patch	96.82	93.86	98.16	97.88
	Pert	95.49	93.92	98.05	97.60
30to30	Patch	95.85	91.75	98.13	97.77
	Pert	93.67	90.35	98.14	97.69
40to40	Patch	94.10	88.37	98.02	97.66
	Pert	93.58	92.11	98.10	97.78
A2Ar	Patch	94.44	93.37	98.14	97.89
	Pert	93.55	92.13	98.26	98.04

(c) ImageNette

Setting	Attack	avg ASR	min ASR	avg ACC	min ACC
Benign	-	-	-	88.81	87.95
A2O	Patch	99.51	99.14	88.36	87.57
	Pert	99.70	99.53	88.52	88.20
3to3	Patch	90.19	83.06	87.74	85.43
	Pert	92.66	89.71	88.36	87.85
5to5	Patch	81.04	78.26	87.46	85.89
	Pert	88.71	86.17	88.58	87.90
8to8	Patch	-	-	-	-
	Pert	83.47	79.05	87.91	86.42
A2Ar	Patch	-	-	-	-
	Pert	82.31	81.20	88.17	87.44

average and the minimum ACC for the ten classifiers we trained. As a reference, the average and the minimum ACC for the ten benign classifiers for each dataset are also shown in Tab. 8.

C.4. Review of the Model Detection Methods Compared in Our Experiments

Neural Cleanse (NC) is a typical reverse-engineering-based model detection method (Wang et al., 2019). It assumes an A2O attack and reverse-engineers a patch trigger with a size as small as possible for each putative target class using the algorithm described in Sec. C.5.2. The premise behind NC is that the backdoor trigger will likely have a small size for human imperceptibility (which is generally true in practice), while the minimum size of a common patch that induces a large fraction of images to be misclassified to a non-backdoor target class will likely be large. With a trigger reverse-engineered for each class, NC adopts an unsupervised, MAD-based anomaly detector to infer if, for any class, the size of the reverse-engineered trigger is abnormally small based on a derived anomaly score. The classifier is deemed to be attacked if the anomaly score is larger than a prescribed threshold (which indicates the existence of a reverse-engineered trigger with abnormally small size). In our experiments, we use 20 clean images per class for detection and setting the threshold of the anomaly score to 2 (Wang et al., 2019). Note that this threshold, though claimed to be associated with a 95% detection confidence level,

implicitly assumes that the estimation of the MAD uses only a single null statistic, while the actual anomaly detection procedure of NC uses all the trigger size statistics for the estimation of MAD. Moreover, threshold 2 is associated with the assumption that an anomaly may exist on both tails of the null distribution, i.e. both overly small and overly large trigger sizes are considered outliers, though a true detection should only be triggered by abnormally small trigger sizes (i.e. the small outliers). Differently, our UMD determines a (single-tailed) confidence threshold based on the actual number of null statistics used for the estimation of MAD (see Sec. 4.2.2), which is more robust than NC to the changes of the domain size. Note that based on Eq. (9), the same threshold 2 used by NC will be obtained if we set $N = 1$ (for a single null statistic) and $\beta = 0.025$ (for a single-tailed 0.025 significance level). Despite the issue with the detection threshold, NC is not able to detect most X2X attacks except A2O attacks[3] by design. Moreover, NC is not implemented with class pair detection since once an attack is detected, all the class pairs with the target class being the detected target class will be treated as backdoor class pairs (by the definition of A2O attacks).

ABS is also a reverse-engineering-based detector that assumes an A2O setting for potential attacks (Liu et al., 2019). But before reverse-engineering the trigger, ABS first identifies a subset of neurons (e.g.) from the penultimate layer with the largest "stimulation" to particular neurons in the output layer. That is, for any of these identified neurons, a large activation will subsequently lead to a large value for some neurons in the output layer. Thus, ABS performs trigger reverse-engineering with a constraint to only boost the activation of these selected neurons. The premise behind the design is that backdoor triggers will likely cause a large activation for some neurons in the intermediate layers. Then, for each putative target class, the reverse-engineered trigger is embedded into a set of clean images and a REASR score is obtained as the misclassification fraction to the target class for these trigger-embedded images. In the inference step, a larger REASR indicates that the classifier is more likely to be attacked. Note that REASR is actually the "transferability" of the reverse-engineered trigger from one group of samples to another with respect to the same target class. It is different from our TR statistic designed for each ordered pair of class pairs and does not endow ABS with the capability to detect general X2X attacks except A2O attacks. In our experiments, we follow the descriptions in the original ABS paper by using one image per class and selecting 10 neurons from the penultimate layer of each classifier for detection. For each putative target class, 30% of the images are used for trigger reverse-engineering, and the remaining 70% images are used to compute the REASR score. Since ABS does not propose a practical method to select a threshold for the REASR score in an unsupervised fashion, in our experiments, *based on the resulting REASR scores*, we choose the threshold for ABS that keeps an approximately 95% false detection rate across all three datasets (for a fair comparison with other methods adopting the same confidence level) while maximizing the overall true positive rate.

PT-RED detects imperceptible, perturbation-based triggers by performing trigger reverse-engineering for each class pair. However, its inference step, which is based on probabilistic modeling with a threshold that controls the false detection rate, relies on the assumption of a single backdoor target class. Thus, it is capable of detecting X-to-one attacks with an inference of the source classes. But still, PT-RED cannot handle X2X attacks with more than one backdoor target class. In our experiments, we use 10 images per class for PT-RED and set the desired false detection rate to 5% (i.e. 95% confidence) based on the original paper.

MNTD trains a binary[4] meta-classifier on features extracted from a large number of shadow classifiers with and without attack. Given a classifier to be inspected, features extracted from the classifier following the same procedure as for the shadow classifiers are fed into the meta-classifier to produce a score – if a score is larger than a prescribed threshold, the classifier is deemed to be attacked, otherwise, it is not attacked. However, for the unsupervised model detection problem, a proper threshold is hard to choose. Thus, in our experiments, we choose a threshold based on the resulting scores to fix a 5% false detection rate (i.e. 95% confidence) while maximizing the true detection rate for a fair comparison with other methods. Moreover, since MNTD cannot cover the enormous space of attack settings for the X2X attack (see Fig. 2) when training the shadow classifiers, its effectiveness largely depends on the generalization capability of the attack settings (as well as the model architecture, the trigger, and so on) from the shadow classifiers to the actual classifier being attacked. Thus, based on the design, it is questionable for MNTD to detect X2X attacks with arbitrary settings. In our experiments, we train a meta-classifier for each dataset using the code provided by the authors of META. In particular, the shadow classifiers with the attack are trained in the A2O setting. From our empirical results in Tab. 2, MNTD does not perform well even against A2O attacks, showing a poor generalization of the model architecture from the shadow models to the actual models to be

[3] A variant of NC with class-pair-wise trigger reverse-engineering was suggested by Wang et al. (2019) for detecting X2O attacks but without adequate evaluation on complicated datasets beyond MNIST (Lecun et al., 1998).

[4] A one-class variant of MNTD is also proposed as a baseline by Xu et al. (2021). However, the performance of this variant is not comparable to MNTD with the binary meta-classifier, thus is not evaluated in our experiments.

inspected. Finally, MNTD only performs model detection without the inference of backdoor class pairs.

K-Arm focuses on solving the trigger reverse-engineering problem for each putative target class without knowing the actual number of source classes. Again, it well-addresses the X-to-one attacks, but cannot detect X2X attacks with more than one target class. Moreover, K-Arm uses the reverse-engineered trigger size for detection inference, which requires supervision for picking a threshold. In our experiments, we pick a threshold for K-Arm for each dataset and for each trigger type to control the false detection rate to 5% while maximizing the true positive rate. For the reverse-engineering step, we use 40 images per class. Finally, like all the other methods reviewed above, K-Arm is not implemented with backdoor class pair inference.

C.5. Trigger Reverse-Engineering Algorithms

In this paper, we have considered three trigger reverse-engineering algorithms. In our experiments in Sec. 5, we equip UMD with the algorithms used by PT-RED (Xiang et al., 2020) and NC (Wang et al., 2019) to address the perturbation trigger and the patch trigger respectively. In our experiments in Sec. 5.4, we show that UMD can even incorporate with the intermediate-layer trigger reverse-engineering technique (Xiang et al., 2020) to address the stronger sample-specific backdoor attack. Here, we introduce these algorithms in detail.

C.5.1. Reverse-Engineering Perturbation Triggers

Perturbation triggers take the form $\delta(X) = [X + v]_{\mathrm{c}}$ for the embedding function where v is a perturbation with a small $||v||_2$ for human imperceptibility and $[\cdot]_{\mathrm{c}}$ is a clipping function. Thus, reverse-engineering a perturbation trigger solves problem (1) with $d(X, \delta(X)) = ||X - \delta(X)||_2 \approx ||v||_2$, i.e.:

$$\begin{aligned} \underset{v}{\text{minimize}} \quad & ||v||_2 \\ \text{subject to} \quad & v \in \arg\min_{v'} \mathbb{E}_{P_{XY|a}}[l(Y, f([X + v']_{\mathrm{c}}))] \end{aligned}$$

Empirically, for class pair $a = (s, t)$ and l being the 0-1 loss, the above problem can be reformulated as (Xiang et al., 2020):

$$\begin{aligned} \underset{v}{\text{minimize}} \quad & ||v||_2 \\ \text{subject to} \quad & \frac{1}{|\mathcal{D}_s|} \sum_{x \in \mathcal{D}_s} \mathbb{1}[f([x + v]_{\mathrm{c}}) = t] \geq \pi \end{aligned} \tag{18}$$

where $\mathcal{D}_s$ is the subset of samples in $\mathcal{D}_{\mathrm{c}}$ from class s, $\mathbb{1}[\cdot]$ is the indicator function (for counting the number of misclassifications from class s to class t), and π is a targeted misclassification fraction (which approximates one minus the Bayes error rate in practice). Typically, π is set large for a relatively large "pair ASR" assumed for a successful attack. But an overly large π may not be achievable for a backdoor class pair even with the actual trigger used by the attacker. In practice, π can be set large but not clearly larger than the ACC of the classifier to be inspected (which can be evaluated on the small dataset possessed by the defender). The reasons are the following. For X2X attacks with an A2Ar setting, the ASR of a successful attack will not likely exceed the ACC. For X2X attacks with other settings, the ASR of a successful attack may be larger than the ACC (even close to 100%). Since there is no prior knowledge about the attack setting, having a large π without exceeding the ACC much will enlarge the probability for the trigger reverse-engineered for backdoor class pairs being close to the actual trigger used by the attacker. Thus, in our experiments, we set $\pi = 0.9$ for all datasets, which is sufficiently large without clearly exceeding the ACC of the classifiers to be inspected.

To solve (18) in practice, we minimize the following differentiable surrogate objective function using stochastic gradient descent (Xiang et al., 2020):

$$J_{st}^{\mathrm{pert}}(v) = -\frac{1}{|\mathcal{D}_s|} \sum_{x \in \mathcal{D}_s} p(t|[x + v]_c), \tag{19}$$

with learning rate 10^{-4} and initial $v = 0$. $p(t|x)$ denotes the classifier's posterior for class t for arbitrary input $x \in \mathcal{X}$. The minimization of Eq. (19) terminates when π misclassification is achieved on $\mathcal{D}_s$.

C.5.2. Reverse-Engineering Patch Triggers

Patch triggers take the form $\delta(X) = (1 - m) \odot X + m \odot u$, where u is a small image patch, m is a binary mask, and $\odot$ represents element-wise multiplication. For human imperceptibility, the patch size, which is solely determined by m,

is usually small. Thus, the distance metric in problem (1) can be specified by $d(X, \delta(X)) = ||X - \delta(X)||_0 \approx ||m||_0$. Accordingly, for each class pair $a = (s, t)$, we solve:

$$\begin{aligned} \underset{\{m,u\}}{\text{minimize}} \quad & ||m||_0 \\ \text{subject to} \quad & \{u, m\} \in \underset{\{u',m'\}}{\arg\min}\, \mathbb{E}_{P_{XY|a}}[l(Y, f((1-m') \odot X + m' \odot u'))] \end{aligned}$$

Similarly, the problem above can be reformulated as the following:

$$\begin{aligned} \underset{\{m,u\}}{\text{minimize}} \quad & ||m||_0 \\ \text{subject to} \quad & \frac{1}{|\mathcal{D}_s|} \sum_{x \in \mathcal{D}_s} \mathbb{1}[f((1-m) \odot x + m \odot u)) = t] \geq \pi \end{aligned} \tag{20}$$

Again, we set $\pi = 0.9$ for all datasets considered in our experiments. Then, problem (20) can be solved by minimizing the surrogate objective function proposed by NC (Wang et al., 2019):

$$J_{st}^{\text{patch}}(u, m) = -\frac{1}{|\mathcal{D}_s|} \sum_{x \in \mathcal{D}_s} \log p(t|(1-m) \odot X + m \odot u)) + \lambda ||m||_1, \tag{21}$$

where λ is the Lagrange multiplier and the patch size is measured using the ℓ_1 norm (instead of the ℓ_0 norm in problem (20)) for differentiability. As suggested by Wang et al. (2019), the mask m and the patch u are both initialized to be image-wide and with initial values around 0.5 (for pixel values in [0, 1]) when minimizing Eq. (21). The multiplier λ is adjusted based on whether the π misclassification fraction from class s to class t (i.e. the constraint of problem (20) is achieved). More details about such adjustment and the learning rate can be found in the original implementation provided by Wang et al. (2019). To avoid poor local optimum when minimizing Eq. (21), we solve problem (20) for multiple times (e.g. 5 trials for CIFAR-10 and Imagenette and 3 trials for GTSRB), each with a randomly initialized m and u. The solution with the minimum $||m||_1$ over all trials is deemed to be the reverse-engineered trigger.

C.5.3. Reverse-Engineering Sample-Specific Triggers

In fact, sample-specific triggers still use a *common* δ, which may be as sophisticated as an autoencoder, for trigger embedding. The term "sample-specific" actually refers to that $\delta(x_i) - x_i$ and $\delta(x_j) - x_j$ are different for different samples x_i and x_j. Unfortunately, accurate estimation of δ (e.g. estimating all the parameters of δ if it is an autoencoder) for a sample-specific trigger is still an open problem. But using the method proposed by Xiang et al. (2020), we can estimate a simple additive perturbation in the intermediate layer of the classifier to approximate δ. More specifically, suppose $f = f_2 \circ f_1$ where $f_1 : \mathcal{X} \to \mathcal{Z}$ maps an input to the intermediate feature space $\mathcal{Z}$ and $f_2 : \mathcal{Z} \to \mathcal{Y}$ maps an intermediate feature to the output space $\mathcal{Y}$. For each class pair $a = (s, t)$, we solve:

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & ||w||_2 \\ \text{subject to} \quad & \frac{1}{|\mathcal{D}_s|} \sum_{x \in \mathcal{D}_s} \mathbb{1}[f_2(f_1(x) + w) = t] \geq \pi \end{aligned}$$

by minimizing:

$$J_{st}^{\text{inter}}(w) = -\frac{1}{|\mathcal{D}_s|} \sum_{x \in \mathcal{D}_s} p'(t|f_1(x) + w),$$

using the same settings as for perturbation reverse-engineering in the input layer. Here, $p'(t|\cdot)$ denotes the posterior of class t for intermediate features. In the experiments in Sec. 5.4, our UMD uses this technique to reverse-engineer the sample-specific trigger embedded by WaNet at the output layer of the first "block" of ResNet-18 (with four "blocks" in total) (He et al., 2016) and achieves excellent detection performance in both model inference and pair inference.

C.6. Additional Results: Detection Performance of UMD with Different Choice of Confidence Level

In Sec. 5 of the main paper, we showed the detection performance of UMD for a 95% confidence level for a fair comparison with the SOTA baselines. Here, in Tab. 9, we show the model detection performance (via MIA) of UMD for a range of

confidence levels from 0.6 to 0.999. Clearly, more aggressive confidence thresholds (with a confidence level $< 0.95\%$) slightly increase the true positive rate (i.e. an increment in MIA for classifiers being attacked) at the cost of a slight increment in the false positive rate (i.e. a decrement in MIA for benign classifiers). On the other hand, more conservative thresholds (with a confidence level $> 0.95\%$) slightly reduce the false positive rate, but the true positive rate is not affected much. The results show that UMD prefers a more conservative confidence level since the attacks are typically associated with a large anomaly score if the putative backdoor class pairs are correctly selected.

Table 9. MIA of our UMD for confidence levels (i.e. $1-\beta$) 0.6, 0.8, 0.9, 0.95, 0.99, and 0.999. Large confidence thresholds are helpful to reduce the false positive rate without much degradation in the true positive rate.

(a) CIFAR-10

Setting	benign	A2O	2to2	5to5	8to8	A2Ar
$1-\beta=0.6$	0.70	0.95	0.90	0.95	0.85	1.0
$1-\beta=0.8$	0.80	0.95	0.90	0.95	0.85	1.0
$1-\beta=0.9$	0.80	0.90	0.90	0.95	0.85	0.95
$1-\beta=0.95$	0.90	0.90	0.90	0.95	0.85	0.95
$1-\beta=0.99$	0.90	0.90	0.85	0.95	0.85	0.95
$1-\beta=0.999$	0.90	0.90	0.85	0.95	0.85	0.90

(b) GTSRB

Setting	benign	A2O	20to20	30to30	40to40	A2Ar
$1-\beta=0.6$	0.80	0.95	0.80	0.90	0.95	1.0
$1-\beta=0.8$	0.80	0.95	0.80	0.90	0.95	1.0
$1-\beta=0.9$	0.80	0.95	0.80	0.90	0.90	1.0
$1-\beta=0.95$	0.90	0.95	0.80	0.90	0.90	1.0
$1-\beta=0.99$	0.90	0.95	0.80	0.85	0.90	0.95
$1-\beta=0.999$	0.90	0.95	0.80	0.85	0.90	0.95

(c) ImageNette

Setting	benign	A2O	3to3	5to5	8to8	A2Ar
$1-\beta=0.6$	0.80	0.95	0.80	0.80	0.90	1.0
$1-\beta=0.8$	0.80	0.95	0.80	0.80	0.90	1.0
$1-\beta=0.9$	0.80	0.95	0.80	0.80	0.80	1.0
$1-\beta=0.95$	0.80	0.90	0.75	0.80	0.80	1.0
$1-\beta=0.99$	0.80	0.90	0.75	0.75	0.80	0.9
$1-\beta=0.999$	1.0	0.90	0.60	0.75	0.70	0.8

D. Supplementary of the Ablation Study

D.1. Baseline Variants of UMD

In this section, we provide details for the two baseline variants of UMD. The first baseline variant, UMD†, directly uses the perturbation or patch size of the reverse-engineered trigger for each class pair for anomaly detection, without using our TR statistic. Since UMD† does not select a subset of putative backdoor class pairs, all the trigger statistics are used for the estimation of MAD, i.e.:

$$\sigma^\dagger = \mathrm{med}_{a \in \mathcal{Y} \times \mathcal{Y} \setminus \mathcal{B}}(|z_a^{-1} - \mathrm{med}_{a' \in \mathcal{Y} \times \mathcal{Y} \setminus \mathcal{B}} z_{a'}^{-1}|)$$

Note that the set $\mathcal{B}$ contains all class pairs with the same source class and target class – trigger reverse-engineering is not performed for these class pairs. Then, an anomaly score is computed for each statistic by:

$$r^\dagger(z) = (z^{-1} - \mathrm{med}_{a \in \mathcal{Y} \times \mathcal{Y} \setminus \mathcal{B}} z_a^{-1})/(1.4826 \cdot \sigma^\dagger)$$

If for any class pair a, the anomaly score $r^\dagger(z_a)$ is larger than the confidence threshold determined by Eq. (9), we say that the classifier is attacked. And the K class pairs with the largest anomaly score are detected as the backdoor class pairs. K is the number of classes in the domain, which is also the largest number of class pairs that an X2X backdoor attack may involve.

The second baseline variant, UMD‡, uses TR, but in a naive way. For each class pair a *detected by* UMD†, UMD‡ performs a "double check" to see if the maximum "mutual-transferability" of a with all the other class pairs, i.e. $\max_{a' \neq a}(T_{aa'} + T_{a'a})$, is in the top K of all class pairs. If this is true, class pair a is admitted as a backdoor class pair; otherwise, a is deemed a non-backdoor class pair. Again, if there is at least one backdoor class pair being detected, the classifier is deemed to be attacked.

D.2. Additional Results for UMD Compared with the Two Baseline Variants

As shown in Sec. 5.3 of the main paper, the two baseline variants achieve an overly large false positive rate in model inference, though the confidence threshold is set for a 5% false positive rate. Especially, the variant UMD† tends to predict any given classifier to be "attacked". In Tab. 10, we show MIA for the two baseline variants and also our UMD for a range of confidence levels in $[0.95, 0.99, 0.999]$. Based on the results, even for extremely conservative confidence thresholds, the two baseline variants still have significantly high false positive rates (i.e. low MIA on benign classifiers). Indeed, an even larger confidence threshold may result in a meaningful false positive rate for the two variants while keeping a high true positive rate (i.e. a good separability between statistics for classifiers with and without attack), but such a threshold will be unknown to the defender *a priori* – the defender will likely set a reasonable confidence level (e.g. near 95%).

Table 10. MIA for UMD†, UMD‡, and our UMD for confidence levels 0.95, 0.99, 0.999.

$1-\beta = 0.95$	2to2	5to5	8to8	A2Ar	Benign
UMD†	1.0	1.0	1.0	1.0	0
UMD‡	1.0	1.0	1.0	1.0	0.4
UMD	1.0	1.0	0.9	0.9	0.9

$1-\beta = 0.99$	2to2	5to5	8to8	A2Ar	Benign
UMD†	1.0	1.0	1.0	1.0	0
UMD‡	1.0	1.0	1.0	1.0	0.6
UMD	1.0	1.0	0.9	0.9	0.9

$1-\beta = 0.999$	2to2	5to5	8to8	A2Ar	Benign
UMD†	1.0	1.0	1.0	1.0	0
UMD‡	1.0	1.0	1.0	1.0	0.7
UMD	0.9	1.0	0.9	0.9	0.9

E. Others

E.1. Collateral Damage for O2O Attacks

Since our UMD will always select at least two different class pairs for inference, we are interested in its detection capability against attacks with only one backdoor class pair, i.e. an O2O attack. In Tab. 6, we show that UMD achieves relatively good performance against O2O attacks. This is because, for O2O attacks, non-backdoor class pairs may suffer from collateral damage, such that samples from the source class will be misclassified to the target class when the backdoor trigger used by the attacker is embedded. Thus, in addition to the true backdoor class pairs, there will exist effective "backdoor" class pairs that are not involved in the attack deliberately. These class pairs typically share the same target class as the true backdoor class pair, since a relation between the backdoor trigger and the adversarial target class has been established when the classifier is trained on the poisoned training set. In Fig. E.1, for all ten O2O attacks, we show the histogram of the pair-based ASR (i.e. the fraction of samples from the source class being misclassified to the target class when the backdoor trigger is embedded) for all non-backdoor class pairs sharing the same target class as the true backdoor class pair. There are several "non-backdoor" class pairs that have a pair-based ASR even larger than 80%.

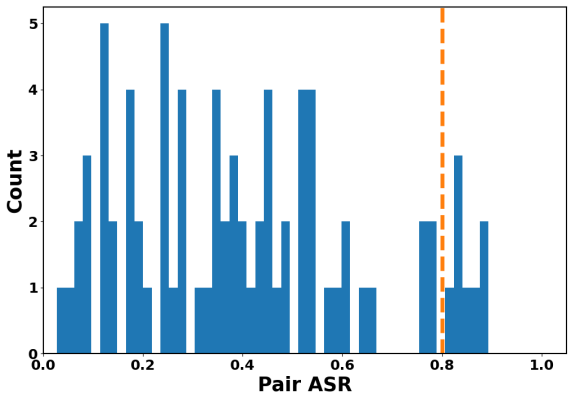

Figure 6. Collateral damage of O2O attacks. Some non-backdoor class pairs achieve as high as 80% ASR due to the attack.

E.2. Performance of UMD against X2X Attacks with Different Numbers of Images for Poisoning

In this section, we show the performance of UMD against X2X attacks with different numbers of poisoning images. We train three groups of classifiers with the A2Ar setting and the perturbation trigger on CIFAR-10, but with 1500, 6000, and 10000 poisoning images respectively. As shown in Fig. E.2, the average ASR of the attack grows with the number of poisoning samples, though still not exceeding the clean test accuracy, which is around 93%. As shown in Tab. 11, UMD achieves generally stable detection performance for all these choices of the number of poisoning images. For attacks with 1500 poisoning images, we observe a drop in the average PDR (though still with a perfect MIA). This is because, with only 1500 images for poisoning, not every backdoor class pair achieves a sufficiently large ASR – these class pairs are less distinguishable from non-backdoor class pairs than backdoor class pairs with high ASR, and thus are more difficult to detect.

Table 11. MIA and average PDR for UMD against A2Ar attacks on CIFAR-10 with 1500, 3000, 6000, and 10000 poisoned images.

No. images	1,500		3,000		6,000		10,000	
	MIA	PDR	MIA	PDR	MIA	PDR	MIA	PDR
UMD	1.0	0.70	0.90	0.92	1.0	0.84	0.90	0.92

E.3. Intuition Behind the Objective Function in Problem (6)

In Fig. E.3, we show a real TR-map for an A2A attack on CIFAR-10. Our clustering problem aims to find a "core" with the maximum "brightness" and an associated "periphary" with the maximum "darkness" (Borgatti & Everett, 2000).

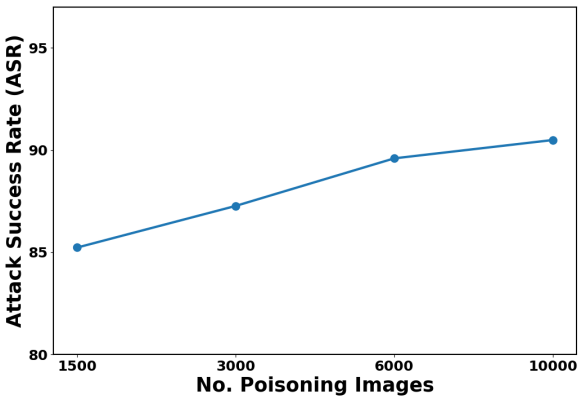

Figure 7. ASR for A2Ar attacks with different numbers of poisoning images on CIFAR-10.

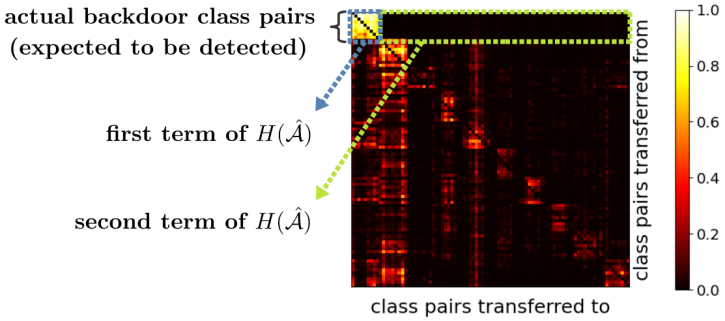

Figure 8. A real TR-map for an A2A attack on CIFAR-10. The orders of the 90 class pairs are the same for both axes.

E.4. Computational Cost of UMD

Empirically, each model inference on CIFAR-10, GTSRB, and Imagenette requires around 0.3h, 2.5h, and 4.3h, respectively, as measured on a single RTX 2080 Ti card. As an off-line detection procedure, this time cost is acceptable compared with the training time on each dataset. The main computational cost is induced by the need to determine for each of the $K(K-1)$ class pairs whether it is involved in a backdoor attack – trigger reverse-engineering is performed for each class pair. Since there is no constraint on the trigger reverse-engineering algorithm used by UMD, the efficiency of UMD can potentially be improved, e.g., by adopting the warm-up strategy by Shen et al. (2021) or the weighted-sum strategy by Xiang et al. (2021) to accelerate the trigger reverse-engineering process.