ROBUSTNESS AGAINST RELATIONAL ADVERSARY

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time adversarial attacks have posed serious challenges to the robustness of machine-learning models, and in many settings the adversarial perturbation need not be bounded by small ℓ_p-norms. Motivated by the semantics-preserving attacks in vision and security domain, we investigate *relational adversaries*, a broad class of attackers who create adversarial examples that are in a reflexive-transitive closure of a logical relation. We analyze the conditions for robustness and propose *normalize-and-predict* – a learning framework with provable robustness guarantee. We compare our approach with adversarial training and derive an unified framework that provides benefits of both approaches. Guided by our theoretical findings, we apply our framework to image classification and malware detection. Results of both tasks show that attacks using relational adversaries frequently fool existing models, but our unified framework can significantly enhance their robustness.

1 INTRODUCTION

The robustness of machine learning (ML) systems has been challenged by test-time attacks using adversarial examples (Szegedy et al., 2013). These adversarial examples are intentionally manipulated inputs that preserve the essential characteristics of the original inputs, and thus are expected to have the same test outcome as the originals by human standard; yet they severely affect the performance of many ML models across different domains (Moosavi-Dezfooli et al., 2016; Eykholt et al., 2018; Qin et al., 2019). As models in high-stake domains such as system security are also undermined by attacks (Grosse et al., 2017; Rosenberg et al., 2018; Hu & Tan, 2018), robust ML in adversarial test environment becomes an imperative task for the ML community.

Existing work on test-time attacks predominately considers ℓ_p-norm bounded adversarial manipulation (Goodfellow et al., 2014; Carlini & Wagner, 2017). However, in many security-critical settings, the adversarial examples need not respect the ℓ_p-norm constraint as long as they preserve the malicious semantics. In malware detection, for example, a malware author can implement the same function using different APIs, or bind a malware within benign softwares like video games or office tools. The modified malware preserves the malicious functionality despite the drastically different syntactic features. Hence, focusing on adversarial examples of small ℓ_p-norm in this setting will fail to address a sizable attack surface that attackers can exploit to evade detectors.

In addition to security threats, another rising concern on ML models is the spurious correlations they could have learned in a biased data set. Ribeiro et al. (2016) show that a highly accurate wolf-vs-husky-dog classifier indeed bases its prediction on the presence/absence of snow in the background. A reliable model, in contrast, should be robust to changes of this nature. Although dubbed as semantic perturbation or manipulation (Mohapatra et al., 2020; Bhattad et al., 2019), these changes do not alter the core of the semantics of input data, thus, we still consider them to be semantics-preserving pertaining to the classification task. Since such semantics-preserving changes often resulted in large ℓ_p-norms, they are likely to render the existing ℓ_p-norm based defenses ineffective.

In this paper, we consider a general attack framework in which attackers create adversarial examples by transforming the original inputs via a set of rules in a semantics-preserving manner. Unlike the prior works (Rosenberg et al., 2018; Hu & Tan, 2018; Hosseini et al., 2017; Hosseini & Poovendran, 2018) which investigate specific adversarial settings, our paper extends the scope of attacks to general logical transformation: we unify the threat models into a powerful relational adversary, which can readily incorporate more complex input transformations.

From the defense perspective, recent work has started to look beyond ℓ_p-norm constraints, including adversarial training (Grosse et al., 2017; Rosenberg et al., 2019; Lei et al., 2019), verification-loss regularization (Huang et al., 2019) and invariance-induced regularization (Yang et al., 2019). Adversarial training in principle can achieve high robust accuracy when the adversarial example in the training loop maximizes the loss. However, finding such adversarial examples is in general NP-hard (Katz et al., 2017), and we show in Sec 4 that it is even PSPACE-hard for semantics-preserving attacks that are considered in this paper. Huang et al. (2019) and Yang et al. (2019) add regularizers that incorporate model robustness as part of the training objective. However, such regularization can not be strictly enforced in training, and neither can the model robustness. These limitations still cause vulnerability to semantics-preserving attacks.

Normalize-and-Predict Learning Framework This paper attempts to overcome the limitations of prior work by introducing a learning framework that guarantees robustness by design. In particular, we target a *relational* adversary, whose admissible manipulation is specified by a logical relation. A logical relation is a set of input pairs, each of which consists of a source and target of an atomic, semantics-preserving transformation. We consider a strong adversary who can apply an arbitrary number of transformations. Our paper makes the following contribution towards the theoretical understanding of robust ML against relational adversaries:

1. We formally describe admissible adversarial manipulation using logical relations, and characterize the necessary and sufficient conditions for robustness to relational adversaries.

2. We propose *normalize-and-predict* (hereinafter abbreviated as *N&P*), a learning framework that first converts each data input to a well-defined and unique normal form and then trains and classifies over the normalized inputs. We show that our framework has guaranteed robustness, and characterize conditions to different levels of robustness-accuracy trade-off.

3. We compare *N&P* to the popular adversarial training framework, which directly optimizes for accuracy under attacks. We show that *N&P* has the advantage in terms of explicit robustness guarantee and reduced training complexity, and in certain cases yields the same model accuracy as adversarial training. Motivated by the comparison, we propose a unified framework, which selectively normalizes over relations that tend to preserve the model accuracy and adversarially trains over the rest. Our unified approach gets the benefits from both frameworks.

We then apply our theoretical findings to malware detection and image classification. For the former, first, we formulate two types of common program transformation — (1) addition of redundant libraries and API calls, and (2) substitution of equivalent API calls — as logical relations. Next, we instantiate our learning framework to these relations, and propose two generic relational adversarial attacks to determine the robustness of a model. Finally, we perform experiments over *Sleipnir*, a real-world WIN32 malware data set. Regarding image classification, we reused an attack method proposed by the prior work (Hosseini & Poovendran, 2018) — shifting of the hue in the HSV color space — that can be deemed as a specific instantiation of our attack framework. We then compare the accuracy and robustness of ResNet-32 (He et al., 2016), a common image classification model, trained with the unified framework against the standard adversarial training on CIFAR-10 (Krizhevsky et al., 2009). The results we obtained in both tasks show that:

1. Attacks using addition and substitution suffice to evade existing ML malware detectors.

2. Our unified approach using input normalization and adversarial training achieves highest robust accuracy among all baselines in malware detection. The drop in accuracy on clean inputs is small and the computation cost is lower than pure adversarial training.

3. When trained with the unified learning framework, ResNet-32 achieves similar clean accuracy but significantly higher robust accuracy than adversarial training alone.

Finally, based on our theoretical and empirical results, we conclude that input normalization is vital to robust learning against relational adversaries. We believe techniques that can improve the quality of normalization are promising directions for future work.

2 RELATED WORK.

Test-time attacks using adversarial examples have been extensively studied in the past several years. Research has shown ML models are vulnerable to such attack in a variety of application domains (Moosavi-Dezfooli et al., 2016; Chen et al., 2017; Papernot et al., 2017; Eykholt et al., 2018; Ebrahimi et al., 2018; Qin et al., 2019; Yang et al., 2020) including system security where reliable defense is absolutely essential. For instance, Grosse et al. (2017) and Al-Dujaili et al. (2018) evade API/library usage based malware detectors by adding redundant API calls; Rosenberg et al. (2018), Hu & Tan (2018), and Rosenberg et al. (2019) successfully attack running-time behavior based detectors by adding redundant execution traces; Pierazzi et al. (2020) extend the attacks from feature-space to problem-space, propose a framework to describe real-world attacker's constraints and create realistic attack instances using automated software transplantation.

On the defense end, the work closest to ours in spirit is Yang et al. (2019), which adds invariance-induced regularizers to the training process. Their work however differs from ours in two major ways. First, their work considers a specific spatial transformation attack in image classification; our work considers a general adversary based on logic relations. Second, their regularizer may not enforce the model robustness on finite samples as they are primarily interested in enhancing the model accuracy. In contrast, our framework emphasizes robustness which is enforced by design. Grosse et al. (2017); Al-Dujaili et al. (2018); Rosenberg et al. (2019) improve robustness via adversarial training; we show such approach is hard to optimize. Incer et al. (2018); Kouzemtchenko (2018) enforce monotonicity over model outputs so that the addition of feature values always increase the maliciousness score. These approaches are limited to guarding against the addition attacks, thus lacks generality. Last, Xu et al. (2017) use feature squeezing, which quantizes the feature values in order to reduce the number of adversarial choices. However, their defense is for ℓ_p-norm adversaries and thus inapplicable for relational attacks.

Normalization is a technique to reduce the number of syntactically distinct instances. First introduced to network security in the early 2000s in the context of intrusion detection systems (Handley et al., 2001), it was later applied to malware detection (Christodorescu et al., 2007; Coogan et al., 2011; Bichsel et al., 2016; Salem & Banescu, 2016; Baumann et al., 2017). Our work addresses the open question whether normalization is useful for ML under relational adversary by investigating its impact on both model robustness and accuracy.

3 BACKGROUND

In this section, we first describe the learning task, then formalize the potential adversarial manipulation as logical relations, and eventually derive the notion of robustness to relational adversaries.

Learning Task. We consider a data distribution $\mathcal{D}$ over a input space $\mathcal{X}$ and categorical label space $\mathcal{Y}$. We use bold face letters, e.g. $\mathbf{x}$, for input vectors and y for the label. Given a hypothesis class $\mathcal{H}$, the learner wants to learn a classifier $f : \mathcal{X} \to \mathcal{Y}$ in $\mathcal{H}$ that minimizes the risk over the data distribution. In non-adversarial settings, the learner solves $\min_{f\in\mathcal{H}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\, \ell(f, \mathbf{x}, y)$, where ℓ is a loss function. For classification, $\ell(f, \mathbf{x}, y) = \mathbb{1}(f(\mathbf{x}) \neq y)$.

Logical Relation. A relation $\mathcal{R}$ is a set of input pairs, where each pair $(\mathbf{x}, \mathbf{z})$ specifies a transformation of input $\mathbf{x}$ to output $\mathbf{z}$. We write $\mathbf{x} \to_{\mathcal{R}} \mathbf{z}$ iff $(\mathbf{x}, \mathbf{z}) \in \mathcal{R}$. We write $\mathbf{x} \to^*_{\mathcal{R}} \mathbf{z}$ iff $\mathbf{x} = \mathbf{z}$ or there exists $\mathbf{z}_0, \mathbf{z}_1, \cdots, \mathbf{z}_k$ $(k > 0)$ such that $\mathbf{x} = \mathbf{z}_0$, $\mathbf{z}_i \to_{\mathcal{R}} \mathbf{z}_{i+1}$ $(0 \leq i < k)$ and $\mathbf{z}_k = \mathbf{z}$. In other words, $\to^*_{\mathcal{R}}$ is the reflexive-transitive closure of $\to_{\mathcal{R}}$. We describe an example relation as follows:

Example 1 (Hue Shifting). *Let $\mathbf{x}_h$, $\mathbf{x}_s$, $\mathbf{x}_v$ denote the hue, saturation and value components of an image $\mathbf{x}$. In a hue shifting relation $\mathcal{R}$, $\mathbf{x} \to_{\mathcal{R}} \mathbf{z}$ iff $\mathbf{z}_h = (\mathbf{x}_h + \delta) \,\%\, 1$ where δ is a scalar, $\mathbf{z}_s = \mathbf{x}_s$, $\mathbf{z}_v = \mathbf{x}_v$. Since $\mathbf{x}_h$ changes in a circle, i.e., hue of 1 is equal to hue of 0. Hence, we compute the modulo of the hue component with 1 to map $\mathbf{z}_h$ within [0,1] (Appendix B gives the background of HSV).*

In this paper, we also consider unions of relations. Notice that a finite union $\mathcal{R}$ of m relations $\mathcal{R}_1, \cdots, \mathcal{R}_m$ is also a relation, and $\mathbf{x} \to_{\mathcal{R}} \mathbf{z}$ *iff* $\mathbf{x} \to_{\mathcal{R}_i} \mathbf{z}$ for any $i \in \{1, \cdots, m\}$.

Table 1: Comparison of training objective and test output for standard risk minimization learning scheme, *N&P* and adversarial training; f^* is the minimizer of the training objective.

	No Defense	Normalize-and-Predict	Adversarial Training
Train	$\min_{f} \sum_{(\mathbf{x},y)\in D} \ell(f,\mathbf{x},y)$	$\min_{f} \sum_{(\mathbf{x},y)\in D} \ell(f,\mathcal{N}(\mathbf{x}),y)$	$\min_{f} \max_{A(\cdot)} \sum_{(\mathbf{x},y)\in D} \ell(f,A(\mathbf{x}),y)$
Test	$f^*(x)$	$f^*(\mathcal{N}(x))$	$f^*(x)$

Threat Model. A test-time adversary replaces a clean test input $\mathbf{x}$ with an adversarially manipulated input $A(\mathbf{x})$, where $A(\cdot)$ represents the attack algorithm. We consider an adversary who wants to maximize the classification error rate: $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\, \mathbb{1}(f(A(\mathbf{x})) \neq y)$.

We assume *white-box* attacks[1], i.e. the adversary has total access to f, including its structures, model parameters and any defense mechanism in place. To maintain the malicious semantics, the adversarial input $A(\mathbf{x})$ should belong to a feasible set $\mathcal{T}(\mathbf{x})$. In this paper, we focus on $\mathcal{T}(\mathbf{x})$ that is described by relation. We consider a logical relation $\mathcal{R}$ that is known to both the learner and the adversary, and we define a relational adversary as the following.

Definition 1 (relational adversary)**.** *An adversary is said to be $\mathcal{R}$-relational if $\mathcal{T}(\mathbf{x}) = \{\mathbf{z} \,|\, \mathbf{x} \to^*_{\mathcal{R}} \mathbf{z}\}$, i.e. each element in $\mathcal{R}$ represents an admissible transformation, and the adversary can apply arbitrary number of transformation specified by $\mathcal{R}$.*

We can then define the robustness of a classifier f by how often its prediction is consistent under attack, and robust accuracy as the fraction of predictions that are both robust and accurate.

Definition 2 (Robustness and robust accuracy)**.** *Let $Q(\mathcal{R}, f, \mathbf{x})$ be the following statement: $\forall\, \mathbf{z}((\mathbf{x} \to^*_{\mathcal{R}} \mathbf{z}) \Rightarrow f(\mathbf{x}) = f(\mathbf{z}))$. Then, a classifier f is robust at $\mathbf{x}$ if $Q(\mathcal{R}, f, \mathbf{x})$ is true, and the robustness of f to an $\mathcal{R}$-relational adversary is: $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{X}}}\, \mathbb{1}_{Q(\mathcal{R},f,\mathbf{x})}$, where $\mathbb{1}_{(\cdot)}$ indicates the truth value of a statement and $\mathcal{D}_{\mathcal{X}}$ is the marginal distribution over inputs. The robust accuracy of f w.r.t. an $\mathcal{R}$-relational adversary is then: $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\, \mathbb{1}_{Q(\mathcal{R},f,\mathbf{x})\wedge f(\mathbf{x})=y}$.*

Notice that the robust accuracy of a classifier is no more than the robustness in value because of the extra requirement of $f(\mathbf{x}) = y$. Meanwhile, a classifier with the highest robustness accuracy may not always have the highest robustness and vice versa: an intuitive example is that a constant classifier is always robust but not necessarily robustly accurate. In Sec 4, we will discuss both objectives and characterize the trade-off between them.

4 *N&P* – A Provably Robust Learning Framework

In this section, we introduce *N&P*, a learning framework which learns and predicts over normalized training and test inputs. We first identify the necessary and sufficient condition for robustness, and propose a normalization procedure that makes *N&P* provably robust to $\mathcal{R}$-relational adversaries. Finally, we analyze the performance of *N&P*: since *N&P* guarantees robustness, the analysis will focus on robustness-accuracy trade-off and provide an in-depth understanding to causes of such trade-off.

4.1 An Overview of the *N&P* Framework

In *N&P*, the learner first specifies a normalizer $\mathcal{N} : \mathcal{X} \to \mathcal{X}$. We call $\mathcal{N}(\mathbf{x})$ the 'normal form' of input $\mathbf{x}$. The learner then both trains the classifier and predicts the test label over the normal forms instead of the original inputs. Let D denote the training set. In the empirical risk minimization learning scheme, for example, the learner will now solve the following problem

$$\min_{f\in\mathcal{H}} \sum_{(\mathbf{x},y)\in D} \ell(f, \mathcal{N}(\mathbf{x}), y), \tag{1}$$

and use the minimizer f^* as the classifier. During test-time, the model will predict $f^*(\mathcal{N}(\mathbf{x}))$. Table 1 compares the *N&P* learning pipeline to normal risk minimization and adversarial training.

[1] We consider a strong white-box attacker to avoid interference from security by obscurity, which is shown fragile in various other adversarial settings (Carlini & Wagner, 2017).

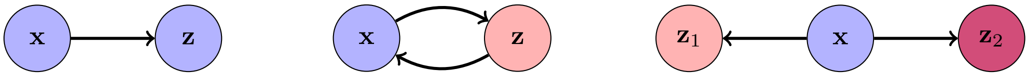

Figure. 1: Relations with different robustness-accuracy trade-off. Different node colors indicate different most likely labels. Appendix A.7 gives a detailed explanation on why semantics-preserving transformation can still change the labels of data. **Left:** *N&P* preserves natural accuracy; **Middle:** *N&P* preserves robust accuracy; **Right:** *N&P* causes suboptimal robust accuracy: suppose $\mu(\mathbf{x}) = 0.02, \mu(\mathbf{z}_1) = \mu(\mathbf{z}_2) = 0.49$, and η is deterministic. *N&P* predict the same label and thus has accuracy at most 0.49, while the highest robust accuracy is 0.98 by predicting the true label for $\mathbf{z}_1$ and $\mathbf{z}_2$.

4.2 Finding the Normalizer

The normalizer $\mathcal{N}$ is crucial for achieving robustness: intuitively, if $\mathbf{x}$ and its adversarial example $\mathbf{x}_{adv}$ share the same normal form, then the prediction will be robust. Meanwhile, a constant $\mathcal{N}$ is robust, but has no utility as $f(\mathcal{N}(\cdot))$ is also constant. Therefore, we seek an $\mathcal{N}$ that perform only the necessary normalization for robustness and has minimal impact on accuracy.

We first construct the **relational graph** $G_\mathcal{R} = \{V, E\}$ of $\mathcal{R}$: the vertex set V contains all elements in $\mathcal{X}$; the edge set E contains an edge $(\mathbf{x}, \mathbf{z})$ *iff* $(\mathbf{x}, \mathbf{z}) \in \mathcal{R}$. Then, a directed path exists from $\mathbf{x}$ to $\mathbf{z}$ *iff* $\mathbf{x} \rightarrow^*_\mathcal{R} \mathbf{z}$. We derive the following necessary and sufficient condition for robustness under *N&P* in Observation 1, and thus obtain a normalizer $\mathcal{N}$ in Proposition 1 that guarantees robustness.

Observation 1 (Condition for Robustness)**.** *Let $C_1, \cdots, C_k$ denote the weakly connected components (WCC) in $G_\mathcal{R}$. A classifier f is robust for all $\mathbf{x} \in C_i$* iff *$f(\mathbf{x})$ returns the same label for all $\mathbf{x} \in C_i$.*

Proposition 1 (Choice of Normalizer)**.** *Let $\mathcal{N}$ be a function that maps an input $\mathbf{x} \in C_i$ to any deterministic element in C_i. Then $f(\mathcal{N}(\cdot))$ is robust to $\mathcal{R}$-relational adversaries.*[2]

4.3 Robustness-Accuracy Trade-off

Optimal Accuracy under N&P. Let $\mu(\mathbf{x})$ denote the probability mass of $\mathbf{x}$. The label of an input $\mathbf{x}$ may also be probabilistic in nature, therefore we use $\eta(\mathbf{x}, l) = \Pr(y = l | \mathbf{x})$ to denote the probability that $\mathbf{x}$ has label l. [3] Then the optimal robust accuracy using *N&P*, denoted by $Acc^*_\mathcal{R}$, is $\sum_{C_i} \max_{l \in \mathcal{Y}} \sum_{\mathbf{x} \in C_i} \mu(\mathbf{x}) \eta(\mathbf{x}, l)$, which happens when $f(\mathcal{N}(\mathbf{x})) = \arg\max_{l \in \mathcal{Y}} \sum_{\mathbf{x} \in C_i} \mu(\mathbf{x}) \eta(\mathbf{x}, l)$ for $\mathbf{x} \in C_i$. Intuitively, f shall assign the most likely label of random samples in C_i to all $\mathbf{x} \in C_i$.

Price of Robustness. In *N&P*, the optimal robust accuracy depends on $\mathcal{R}$. We then observe the following fundamental robustness-accuracy trade-off: as the relation becomes more complicated, we may lose accuracy for enforcing invariant model predictions, and such loss is the price of robustness.

Observation 2 (Robustness-accuracy trade-off)**.** *Let $\mathcal{R}'$ and $\mathcal{R}$ be two relations s.t. $\mathcal{R}' = \mathcal{R} \bigcup \{(\mathbf{x}, \mathbf{z})\}$, i.e. $\mathcal{R}'$ allows an extra transformation from $\mathbf{x}$ to $\mathbf{z}$ than $\mathcal{R}$. Let $C_{\mathbf{x},\mathcal{R}}$ denote the WCC in $G_\mathcal{R}$ that contains $\mathbf{x}$, and l_C be the most likely label of inputs in a WCC C. Then $Acc^*_{\mathcal{R}'} - Acc^*_\mathcal{R} \leq 0$ for all $\mathcal{R}, \mathcal{R}'$ pairs, and the equality only holds when $l_{C_{\mathbf{x},\mathcal{R}}} = l_{C_{\mathbf{z},\mathcal{R}}}$.*

The intuition is that the extra edge on the relation graph may join two connected components which are otherwise separate. As a result, a model under *N&P* will predict the same label for the two components, thus the accuracy on one component will drop if two components have different labels.

We further characterize three different levels of trade-offs (Figure 1). First, if two inputs $\mathbf{x}, \mathbf{z}$ have the same most likely label on $\mathcal{D}$, then the optimal accuracy under *N&P* is the same as before normalization, in other words, robustness is obtained *for free*. Second, if both $(\mathbf{x}, \mathbf{z})$ and $(\mathbf{z}, \mathbf{x})$ are in $\mathcal{R}$ but $\mathbf{x}, \mathbf{z}$ have different most likely labels, then the model with the highest natural accuracy, which predicts the most likely label of $\mathbf{x}$ and $\mathbf{z}$ respectively, do not have any robustness. In contrast, *N&P* achieves the optimal robust accuracy by predicting a *single* label — the most likely label of samples in $\{\mathbf{x}, \mathbf{z}\}$ — for both $\mathbf{x}$ and $\mathbf{z}$. Third, if $\mathbf{x}$ can only be one-way transformed to two inputs $\mathbf{z}_1, \mathbf{z}_2$ with different

[2] Appendix C.1 shows a decidable algorithm of realizing such an $\mathcal{N}$ given $G_\mathcal{R}$.

[3] For example, a ransomware and a zip tool may have the same static feature vector $\mathbf{x}$. The label of a randomly drawn $\mathbf{x}$ is probabilistic, and the probability depends on the frequency that each software appears.

most likely labels, then *N&P* may have suboptimal robust accuracy. An absolutely robust classifier need to predict the same label for $\mathbf{x}, \mathbf{z}_1$ and $\mathbf{z}_2$, while the classifier with the highest robust accuracy should predict the mostly likely labels for $\mathbf{z}_1$ and $\mathbf{z}_2$ if $\mathbf{z}_1, \mathbf{z}_2$ appear more frequently than $\mathbf{x}$.

5 Comparing and Unifying *N&P* with Adversarial Training

N&P differs from the adversarial training — the most widely acknowledged defense mechanism against test-time adversary — in its objective and procedure. While each approach has its own limitation against relational adversaries, we show that they can complement each other and be unified into one framework that enjoys the benefits from both worlds.

Comparative Advantages. The performance of adversarial training depends on the quality of the adversarial examples. However, we show in Proposition 2 that the inner maximization problem is in general computationally infeasible for relational adversaries.

Proposition 2 (Hardness of Inner Maximization)**.** *The inner optimization problem of adversarial training is PSPACE-hard for relational adversaries.*

Intuitively, the search space of a relational adversary can grow combinatorially with the number of transformations, and the proposition follows the classical results of reachability analysis in model checking (Kozen, 1977). The *N&P* framework, in contrast, solves a typical minimization problem, and thus reduces the computation complexity if an efficient normalizer exists. Meanwhile, we show in Appendix A.4 that robust accuracy can be achieved with a simpler model class on normalized inputs than on original inputs; reduced model complexity may also improve the sample efficiency of the underlying learning algorithm. On the other hand, *N&P* may incur excessive loss in accuracy to enforce robustness, for example, the last scenario in Figure 1, in which case, adversarial training will be a better choice for overall utility.

A Unified Framework. Motivated by the above observations, we propose a unified framework: for a relation $\mathcal{R}$, we strategically select a subset $\mathcal{R}' \subset \mathcal{R}$ to normalize inputs, and adversarially train on the normalized inputs. Let $\mathcal{N}_{R'}$ denote the normalizer for $\mathcal{R}'$. Formally, the learner solves

$$\min_{f \in \mathcal{H}} \max_{A(\cdot)} \sum_{(\mathbf{x},y) \in D} \ell\left(f, A\left(\mathcal{N}_{\mathcal{R}'}(\mathbf{x})\right), y\right), \tag{2}$$

during training to obtain a minimizer f^*, and predicts $f^*(\mathcal{N}_{\mathcal{R}'}(\mathbf{x}))$ at test-time. The classifier f^* will be robust to $\mathcal{R}'$-relational adversary, and have potentially higher robust accuracy than using *N&P* alone. In particular, if $\mathcal{R}'$ is *reversible* by Definition 3, then our unified framework preserves the optimal robust accuracy as shown in Theorem 1.

Definition 3. *A relation $\mathcal{R}'$ is reversible iff $\mathbf{x} \rightarrow_{\mathcal{R}'^*} \mathbf{z}$ implies $\mathbf{z} \rightarrow_{\mathcal{R}'^*} \mathbf{x}$ and vice versa.*

Theorem 1 (Preservation of robust accuracy)**.** *Let f^* be the classifier that minimizes the objective of our unified framework over data distribution $\mathcal{D}$, and let f^*_{adv} minimize the objective of adversarial training over $\mathcal{D}$. Then, in principle, $f^*(\mathcal{N}_{\mathcal{R}'}(\cdot))$ and f^*_{adv} have the same optimal robust accuracy if $\mathcal{R}'$ is reversible.*

The proof can be found in Appendix A.5. In essence, Theorem 1 is a generalization of the second scenario in Figure 1, in particular, we extend the same principle applied to $(\mathbf{x}, \mathbf{z})$ to all possible pairs of inputs in the relational graph induced by $\mathcal{R}'$. Note that reversible relation is also common: if $\mathbf{z}$ is $\mathbf{x}$'s adversarial example, then $\mathbf{x}$ is also likely to be an adversarial choice of $\mathbf{z}$. Observation 2 and Theorem 1 provide a general guideline for selecting $\mathcal{R}'$: choose the reversible subset of $\mathcal{R}$ first, and then consider transformations that cause little drop in optimal robust accuracy.

Regarding the efficiency of normalization, we show in Appendix A.6 that the strongest adversarial example satisfies the requirment of Proposition 1, and thus can be used as the normal form. Therefore, in theory, *N&P* is at least as efficient as the optimal adversarial training. In practice, the normalizer we use in our empirical evaluation are all more efficient than adversarial training.

6 Experiment

We now evaluate the effectiveness of our unified framework against relational attacks. In particular, we seek answers to the following questions:

Table 2: Malware Detection: False Negative Rate (FNR) and False Positive Rate (FPR) on *Sleipnir*.

	Unified (Ours)		Adv-Trained		Al-Dujaili et al. (2018)		Natural	
	FNR(%)	FPR(%)	FNR(%)	FPR(%)	FNR(%)	FPR(%)	FNR(%)	FPR(%)
Natural	5.0±0.4	11.9±1.2	5.8±0.9	12.1±1.2	6.4±0.5	10.7±0.3	6.2±0.6	10.0±0.6
Adversarial	5.5±0.5	11.9±1.2	27.9±8.2	12.1±1.2	89.9±7.8	10.7±0.3	100±0.0	10.0±0.6

1. Do relational attacks pose real threats to existing ML models?
2. How effective is our unified framework in enhancing robustness, and do the results corroborate with the theory?

We investigate these aspects over two real world tasks — malware detection and image classification. For each task, we identify relations that do not alter the essential semantics of the inputs. Our result shows that the models obtained from our unified framework has the highest robust accuracy compared to adversarially trained models and unprotected models.

6.1 Malware Detection

We evaluate a malware detection task on *Sleipnir*, a data set containing Windows binary API usage features of 34,995 malware and 19,696 benign software, extracted from their Portable Executable (PE) files using LIEF (Thomas, 2017). The detection is exclusively based on the API usage of a malware. There are 22,761 unique API calls in the data set, so each PE file is represented by a binary indicator vector $\mathbf{x} \in \{0,1\}^m$, where $m = 22,761$. Note that this is the same encoding scheme adopted by Al-Dujaili et al. (2018). We sample 19,000 benign PEs and 19,000 malicious PEs to construct the training (60%), validation (20%), and test (20%) sets.

Existing ℓ_p norm based attacks are not applicable for relational adversaries. Meanwhile, exhaustive search over adversarial choices may be computationally prohibitive. Therefore, we propose two heuristic attack algorithms – GreedyByGroup and GreedyByGrad – to validate models' robust accuracy. Both algorithms are greedy and iterative in nature. Detailed algorithm descriptions are in Appendix C.2.

GreedyByGroup takes a test input vector $\mathbf{x}$ and a maximum number of iterations K. In each iteration, it partitions $\mathcal{R}$ into subsets of relations $\mathcal{R}_1, \cdots, \mathcal{R}_m$, and finds the instance within the transitive closure of each $\mathcal{R}_i$ that maximizes the loss. These instances from all $\mathcal{R}_i$s are combined to create the new version of $\mathbf{x}^{adv}$. Notice the attack reduces to exact search if $\mathcal{R}$ is not partitioned.

GreedyByGrad takes a test input vector $\mathbf{x}$, a maximum number m of transformation to apply in each iteration, and a maximum number of iteration K. In each iteration, it makes a first-order approximation of the change in test loss caused by each transformation, and then applies the transformations with top m approximated increases in test loss to create the new version of $\mathbf{x}^{adv}$.

Relation and Attacks. The goal of an adversary is to evade a malware detector. A common strategy that (Al-Dujaili et al., 2018) also adopts is adding redundant API calls. This strategy can be described by an additive relation: $(\mathbf{x}, \mathbf{z}) \in \mathcal{R}$ *iff* $\mathbf{z}$ is obtained by flipping some $\mathbf{x}$'s feature values from 0 to 1. We also consider a new attacking strategy, which substitutes API calls with functionally equivalent counterparts. This strategy can be described by an equivalence relation: $(\mathbf{x}, \mathbf{z}) \in \mathcal{R}$ *iff* $\mathbf{z}$ is obtained by changing some of $\mathbf{x}$'s feature values from 1 to 0 in conjunction with some of $\mathbf{x}$'s other feature values changed from 0 to 1. With expert knowledge, we extract nearly 2,000 equivalent API groups described in Appendix C.3. We use three attack algorithms — GreedyByGrad, GreedyByGroup and the `rfgsm_k` additive attack presented by Al-Dujaili et al. (2018) — and consider the attack to be successful if any algorithm fools the detector.

Model and Baselines. We compare four ML detectors. The **Unified** detector is realized using our unified framework in Sec 5: we normalize over the equivalence relation based on the functionally equivalent API groups, and then adversarially trains over the additive relation. The **Adv-Trained** detector is adversarially trained with the best adversarial example generated using GreedyByGrad and the `rfgsm_k` additive attack (Al-Dujaili et al., 2018) as GreedyByGroup is too computationally expensive to be included in the training loop. We also include the model proposed by Al-Dujaili et al. (2018), which is adversarially trained against only the `rfgsm_k` additive attack, and a **Natural**

Table 3: Image Classification: Classification accuracy on CIFAR10, same relation in training and testing. The first column specifies the attack parameters used in test-time. The parameters are in the form of (ℓ_∞-norm, PGD step size, PGD steps, number of hue-shifts). The models are adversarially trained using $(4/255, 2/255, 3, 20)$.

	Unified (Ours)	Adv-Trained (Combined)	Adv-Trained (PGD only)
Natural	73.4±1.0	73.7±1.2	78.6±0.7
$(4/255, 2/255, 3, 20)$	54.9±1.2	50.0±0.8	28.5±0.6
$(4/255, 2/255, 3, 200)$	54.9±1.2	49.1±0.6	28.3±0.6
$(4/255, 2/255, 15, 20)$	54.3±1.1	48.6±0.9	27.9±0.6
$(4/255, 2/255, 200, 20)$	54.2±1.2	48.2±0.9	27.8±0.6
$(6/255, 2/255, 15, 20)$	42.4±1.4	34.5±1.1	17.4±0.8
$(6/255, 2/255, 200, 200)$	42.3±1.5	33.8±1.0	17.2±0.8

Table 4: Image Classification: Classification accuracy on CIFAR10, relation in training is a subset of relation in testing. The attacker uses a 15-step PGD attack with ℓ_∞-norm $4/255$ and step size $2/255$, and randomly samples 500 combinations of hue, brightness and constrast adjustment factors.

	Unified (Ours)	Adv-Trained (Combined)	Adv-Trained (PGD only)
Natural	73.4±1.0	73.7±1.2	78.6±0.7
Adversarial	47.0±1.1	41.7±1.1	19.7±0.8

model with no defense. We use the same network architecture as Al-Dujaili et al. (2018), a fully-connected neural net with three hidden layers, each with 300 ReLU nodes, to set up a fair comparison. We train each baseline to minimize the negative log-likelihood loss for 20 epochs, and pick the model with the lowest validation loss. We run five different data splits.

Results. As Table 2 shows, relational attacks are overwhelmingly effective to detectors that are oblivious to potential transformations. Adversarial examples almost always (>99% FNR) evade the naturally trained model, and also evade the detector in Al-Dujaili et al. (2018) most of the time (>89% FNR) as it does not consider API substitution. On the defense end, **Unified** achieves the highest robust accuracy: the evasion rate (FNR) only increases by 0.5% on average. **Adv-Trained** comes second but the evasion rate is still 22.1% higher. The evasion is mostly caused by GREEDYBYGROUP, the attack that is too computationally expensive to be included in the training loop. This result corroborates with the theoretical advantage of *N&P*: its robustness guarantee is independent of training algorithms. Last, all detectors using robust learning techniques have higher FPR compared to **Natural**, which is expected because of the inevitable robustness-accuracy trade-off. However, the difference is much smaller compared to the cost due to attacks, and thus the trade-off is worthwhile.

6.2 IMAGE CLASSIFICATION

We evaluate the effectiveness of our unified framework on CIFAR10 containing 50,000 training and 10,000 test images of size 32x32 pixels. We randomly sample 5,000 images for validation and train on the remaining 45,000 images.

Relation and Attacks. We consider a relation induced by hue shifting specified in Example 1. Due to the shape bias property (Landau et al., 1988), humans can still correctly classify most images after the adjustment of color hue. Therefore, we consider this relation to be semantics-preserving. The attacker uses a combination of ℓ_∞ and relational attacks: it first shifts the color hue of the image, and then generates ℓ_∞ adversarial example using PGD attack. For each image, the attacker tries different hue adjustments, which evenly split the hue space. In addition, we consider an attacker that can also adjust the brightness and contrast of the image by a factor in $[0.8, 1.2]$. It tries 500 random combination of hue, brightness and contrast adjustments followed by PGD attack.

Model and Baselines. The **Unified** classifier is obtained with our unified framework in Sec 5: we adjust hue of the input such that the pixel at the top-left corner has hue value 1, and then adversarially train against the PGD attack. [4] We also consider two adversarial training baselines: the first uses the combined attack (PGD and hue adjustment) in training, while the second only uses the PGD attack. We train a ResNet32 network for 100 epochs in all configurations, and pick the model with the lowest validation loss. We also run five different data splits.

Results. Table 3 shows the results against attackers using hue-shift and ℓ_∞ perturbation. Although adversarial training against only the PGD attack has higher clean input accuracy, the combined attack heavily reduces its test accuracy, indicating again the effectiveness of simple relational attack to unprotected models. **Unified** achieves the highest robust accuracy against the combined attack – $\geq 4.8\%$ higher compared to adversarial training with the combined attack over all attack parameters. This result shows the advantage of normalization over reversible relations, as projected by our analysis in Sec 5. In addition, Table 4 shows the results against an attacker using more transformations than the ones normalized in training. Our unified approach still achieves the highest accuracy with a substantial margin over the baselines. Although the attacker may use more transformations, normalization can still reduce the search space of adversarial examples and increase robustness.

7 Conclusion and Future Work

In this work, we set the first step towards robust learning against relational adversaries: we theoretically characterize the conditions for robustness and the sources of robustness-accuracy trade-off, and propose a provably robust learning framework. Our empirical evaluation shows that a combination of input normalization and adversarial training can significantly enhance model robustness. For future work, we see automatic detection of semantics-preserving transformation as a promising addition to our current expert knowledge approach, and plan to extend the normalization approach to deal with other kinds of attacks beyond relational adversaries.

[4] Given an input image in RGB format, we first convert the image to HSV format, and then add a scalar to the hue of all pixels. The scalar is determined by 1 - (the hue of the pixel on the first row and first column). The hue values are then projected back to the [0,1] interval by taking the remainder over 1. Finally, we convert the image back to RGB for classification.

REFERENCES

Abdullah Al-Dujaili, Alex Huang, Erik Hemberg, and Una-May O'Reilly. Adversarial deep learning for robust detection of binary encoded malware. In *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 76–82. IEEE, 2018.

Richard Baumann, Mykolai Protsenko, and Tilo Müller. Anti-proguard: Towards automated deobfuscation of android apps. In *Proceedings of the 4th Workshop on Security in Highly Connected IT Systems*, SHCIS '17, pp. 7–12, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5271-0. doi: 10.1145/3099012.3099020. URL `http://doi.acm.org/10.1145/3099012.3099020`.

Anand Bhattad, Min Jin Chong, Kaizhao Liang, Bo Li, and David A Forsyth. Unrestricted adversarial examples via semantic manipulation. *arXiv preprint arXiv:1904.06347*, 2019.

Benjamin Bichsel, Veselin Raychev, Petar Tsankov, and Martin Vechev. Statistical deobfuscation of android applications. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pp. 343–355, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4139-4. doi: 10.1145/2976749.2978422. URL `http://doi.acm.org/10.1145/2976749.2978422`.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57. IEEE, 2017.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 15–26. ACM, 2017.

Mihai Christodorescu, Somesh Jha, Johannes Kinder, Stefan Katzenbeisser, and Helmut Veith. Software transformations to improve malware detection. *Journal in Computer Virology*, 3:253–265, 10 2007. doi: 10.1007/s11416-007-0059-8.

Kevin Coogan, Gen Lu, and Saumya Debray. Deobfuscation of virtualization-obfuscated software: A semantics-based approach. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*, CCS '11, pp. 275–284, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0948-6. doi: 10.1145/2046707.2046739. URL `http://doi.acm.org/10.1145/2046707.2046739`.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 31–36, 2018.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1625–1634, 2018.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Kathrin Grosse, Nicolas Papernot, Praveen Manoharan, Michael Backes, and Patrick McDaniel. Adversarial examples for malware detection. In *European Symposium on Research in Computer Security*, pp. 62–79. Springer, 2017.

Mark Handley, Vern Paxson, and Christian Kreibich. Network intrusion detection: Evasion, traffic normalization, and end-to-end protocol semantics. In *Proceedings of the 10th Conference on USENIX Security Symposium - Volume 10*, SSYM'01, Berkeley, CA, USA, 2001. USENIX Association. URL `http://dl.acm.org/citation.cfm?id=1251327.1251336`.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Hossein Hosseini and Radha Poovendran. Semantic adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1614–1619, 2018.

Hossein Hosseini, Baicen Xiao, Mayoore Jaiswal, and Radha Poovendran. On the limitation of convolutional neural networks in recognizing negative images. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 352–358. IEEE, 2017.

Weiwei Hu and Ying Tan. Black-box attacks against rnn based malware detection algorithms. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. Achieving verified robustness to symbol substitutions via interval bound propagation. pp. 4074–4084, 2019.

Inigo Incer, Michael Theodorides, Sadia Afroz, and David Wagner. Adversarially robust malware detection using monotonic classification. In *the Fourth ACM International Workshop on Security and Privacy Analytics (IWSPA)*, Tempe, AZ, USA, Mar. 2018.

G. Katz, C. Barrett, D.L. Dill, K. Julian, and M.J. Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, 2017.

A. Koschan, M. Abidi, and M.A. Abidi. *Digital Color Image Processing*. Wiley, 2008. ISBN 9780470147085.

Alex Kouzemtchenko. Defending malware classification networks against adversarial perturbations with non-negative weight restrictions. *arXiv preprint arXiv:1806.09035*, 2018.

Dexter Kozen. Lower bounds for natural proof systems. In *FOCS*, 1977.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

Barbara Landau, Linda B Smith, and Susan S Jones. The importance of shape in early lexical learning. *Cognitive development*, 3(3):299–321, 1988.

Qi Lei, Lingfei Wu, Pin-Yu Chen, Alexandros G Dimakis, Inderjit S Dhillon, and Michael Witbrock. Discrete adversarial attacks and submodular optimization with applications to text classification. *Systems and Machine Learning (SysML)*, 2019.

Jeet Mohapatra, Tsui-Wei Weng, Pin-Yu Chen, Sijia Liu, and Luca Daniel. Towards verifying robustness of neural networks against a family of semantic perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 244–252, 2020.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582, 2016.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, 2017.

Fabio Pierazzi, Feargus Pendlebury, Jacopo Cortellazzi, and Lorenzo Cavallaro. Intriguing properties of adversarial ml attacks in the problem space. In *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 1308–1325. IEEE Computer Society, 2020. doi: 10.1109/SP40000.2020.00073. URL `https://doi.ieeecomputersociety.org/10.1109/SP40000.2020.00073`.

Yao Qin, Nicholas Carlini, Garrison Cottrell, Ian Goodfellow, and Colin Raffel. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In *International Conference on Machine Learning*, pp. 5231–5240, 2019.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144. ACM, 2016.

Ishai Rosenberg, Asaf Shabtai, Lior Rokach, and Yuval Elovici. Generic black-box end-to-end attack against state of the art api call based malware classifiers. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pp. 490–510. Springer, 2018.

Ishai Rosenberg, Asaf Shabtai, Yuval Elovici, and Lior Rokach. Defense methods against adversarial examples for recurrent neural networks. *arXiv preprint arXiv:1901.09963*, 2019.

Aleieldin Salem and Sebastian Banescu. Metadata recovery from obfuscated programs using machine learning. In *Proceedings of the 6th Workshop on Software Security, Protection, and Reverse Engineering*, SSPREW '16, pp. 1:1–1:11, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4841-6. doi: 10.1145/3015135.3015136. URL http://doi.acm.org/10.1145/3015135.3015136.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Romain Thomas. Lief - library to instrument executable formats. https://lief.quarkslab.com/, April 2017.

Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

Fanny Yang, Zuowen Wang, and Christina Heinze-Deml. Invariance-inducing regularization using worst-case transformations suffices to boost accuracy and spatial robustness. In *Advances in Neural Information Processing Systems*, pp. 14757–14768, 2019.

Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael I Jordan. Greedy attack and gumbel attack: Generating adversarial examples for discrete data. *Journal of Machine Learning Research*, 21(43):1–36, 2020.

A Proofs and Explanation for Theoretical Results

In this section, we present the omitted proofs for theorems and observations due to page limit of the main body.

A.1 Proof for Observation 1

The if direction holds because any $\mathbf{x} \in C_i$ can only be transformed into $\mathbf{z} \in C_i$ by the maximal property of weakly connected component. The contrapositive of the only if direction holds because if $f(\mathbf{x}) \neq f(\mathbf{z})$ for a pair $\mathbf{x}, \mathbf{z} \in C_i$, then there must exist two adjacent nodes $\mathbf{z}_1, \mathbf{z}_2$ on the path between $\mathbf{x}$ and $\mathbf{z}$ such that one can be transformed to another yet $f(\mathbf{z}_1) \neq f(\mathbf{z}_2)$.

A.2 Proof for Observation 2

We first write the full version for Observation 2 in the following claim.

Claim 1. *Let $\mathcal{R}'$ and $\mathcal{R}$ be two relations such that $\mathcal{R}' = \mathcal{R} \bigcup \{(\mathbf{x}, \mathbf{z})\}$, i.e. $\mathcal{R}'$ allows an extra transformation from $\mathbf{x}$ to $\mathbf{z}$ than $\mathcal{R}$. Let $G_{\mathcal{R}}, G_{\mathcal{R}'}$ denote their relation graphs and $C_{\mathbf{x},\mathcal{R}}$ be the weakly connected component (WCC) in $G_{\mathcal{R}}$ that contains $\mathbf{x}$. In addition, let μ, η be the same as defined in Sec 4.3, and $l_C = \arg\max_{l \in \mathcal{Y}} \sum_{\mathbf{x} \in C} \mu(\mathbf{x})\eta(\mathbf{x}, l)$, i.e. the most likely label of inputs in C. Then the change of best attainable robust accuracy from $\mathcal{R}$ to $\mathcal{R}'$, denoted by $\Delta_{\mathcal{R},\mathcal{R}'}$, is*

$$\Delta_{\mathcal{R},\mathcal{R}'} = Acc^*_{\mathcal{R}'} - Acc^*_{\mathcal{R}} \tag{3}$$

$$= \sum_{\mathbf{x}' \in C_{\mathbf{x},\mathcal{R}'}} \mu(\mathbf{x}')\eta(\mathbf{x}, l_{C_{\mathbf{x},\mathcal{R}'}}) - \left(\sum_{\mathbf{x}' \in C_{\mathbf{x},\mathcal{R}}} \mu(\mathbf{x}')\eta(\mathbf{x}', l_{C_{\mathbf{x},\mathcal{R}}}) + \sum_{\mathbf{x}' \in C_{\mathbf{z},\mathcal{R}} \setminus C_{\mathbf{x},\mathcal{R}}} \mu(\mathbf{x}')\eta(\mathbf{x}', l_{C_{\mathbf{z},\mathcal{R}}}) \right). \tag{4}$$

The change $\Delta_{\mathcal{R},\mathcal{R}'} \leq 0$ for all $\mathcal{R}, \mathcal{R}'$ pairs, and the equality only holds when $l_{C_{\mathbf{x},\mathcal{R}}} = l_{C_{\mathbf{z},\mathcal{R}}}$.

Proof. First, we observe that $C_{\mathbf{x},\mathcal{R}'} = C_{\mathbf{x},\mathcal{R}} \bigcup C_{\mathbf{z},\mathcal{R}}$. Notice that $(\mathbf{x}, \mathbf{z})$ will not change the graph structure outside $C_{\mathbf{x},\mathcal{R}} \bigcup C_{\mathbf{z},\mathcal{R}}$: the maximal property of WCC guarantees that neither $\mathbf{x}$ nor $\mathbf{z}$ connect to nodes outside their own WCC. If $C_{\mathbf{x},\mathcal{R}}$ and $C_{\mathbf{z},\mathcal{R}}$ are two disjoint WCCs, then the path $\mathbf{x} \rightarrow_{\mathcal{R}} \mathbf{z}$ will join them to form $C_{\mathbf{z},\mathcal{R}'}$. Otherwise, $\mathbf{x}, \mathbf{z}$ are already in the same WCC, and thus $C_{\mathbf{x},\mathcal{R}'} = C_{\mathbf{x},\mathcal{R}} = C_{\mathbf{z},\mathcal{R}}$.

Since the graph structure outside $C_{\mathbf{x},\mathcal{R}'}$ does not change, it suffices to only look at change of best robust accuracy within $C_{\mathbf{x},\mathcal{R}'}$. The term $\sum\limits_{\mathbf{x}' \in C_{\mathbf{x},\mathcal{R}'}} \mu(\mathbf{x}')\eta(\mathbf{x}, l_{C_{\mathbf{x},\mathcal{R}'}})$ is the best robust accuracy in $C_{\mathbf{x},\mathcal{R}'}$. When $C_{\mathbf{x},\mathcal{R}} \neq C_{\mathbf{z},\mathcal{R}}$, the term $\sum\limits_{\mathbf{x}' \in C_{\mathbf{x},\mathcal{R}}} \mu(\mathbf{x}')\eta(\mathbf{x}', l_{C_{\mathbf{x},\mathcal{R}}})$ and $\sum\limits_{\mathbf{x}' \in C_{\mathbf{z},\mathcal{R}} \setminus C_{\mathbf{x},\mathcal{R}}} \mu(\mathbf{x}')\eta(\mathbf{x}', l_{C_{\mathbf{z},\mathcal{R}}})$ are the best robust accuracy in $C_{\mathbf{x},\mathcal{R}}$ and $C_{\mathbf{z},\mathcal{R}}$, respectively. When $C_{\mathbf{x},\mathcal{R}} = C_{\mathbf{z},\mathcal{R}}$, the latter term becomes zero and the former is the best robust accuracy in $C_{\mathbf{x},\mathcal{R}}$. In both cases, the equation in Claim 1 holds by definition.

Next, we show $\Delta_{\mathcal{R},\mathcal{R}'} \leq 0$. First, consider $l_{C_{\mathbf{x},\mathcal{R}}} \neq l_{C_{\mathbf{z},\mathcal{R}}}$. No matter what $l_{C_{\mathbf{x},\mathcal{R}'}}$ is, it will be different from at least one of $l_{C_{\mathbf{x},\mathcal{R}}}, l_{C_{\mathbf{z},\mathcal{R}}}$. Suppose, $l_{C_{\mathbf{x},\mathcal{R}'}} \neq l_{C_{\mathbf{x},\mathcal{R}}}$, then the robust accuracy for inputs in $C_{\mathbf{x},\mathcal{R}}$ will drop. Similarly, the accuracy in $C_{\mathbf{z},\mathcal{R}}$ will drop if $l_{C_{\mathbf{x},\mathcal{R}'}} \neq l_{C_{\mathbf{z},\mathcal{R}}}$. Therefore, $\Delta_{\mathcal{R},\mathcal{R}'} < 0$. Second, when $l_{C_{\mathbf{x},\mathcal{R}}} = l_{C_{\mathbf{z},\mathcal{R}}}$, then we will have $l_{C_{\mathbf{x},\mathcal{R}'}} = l_{C_{\mathbf{x},\mathcal{R}}} = l_{C_{\mathbf{z},\mathcal{R}}}$ by definition, and the expression for $\Delta_{\mathcal{R},\mathcal{R}'}$ will evaluate to 0. □

We also note that if the majority label $l_{C,\mathcal{R}}$ is not unique for $C_{\mathbf{x},\mathcal{R}}$ and/or $C_{\mathbf{z},\mathcal{R}}$, then we consider $l_{C_{\mathbf{x},\mathcal{R}}} = l_{C_{\mathbf{z},\mathcal{R}}}$ if any majority label for $C_{\mathbf{x},\mathcal{R}}$ matches any one for $C_{\mathbf{z},\mathcal{R}}$.

A.3 Proof for Proposition 2

We first write the full statement of Proposition 2 in the following theorem.

Theorem 2. *Let $\mathcal{R} \subseteq \{0,1\}^d \times \{0,1\}^d$ be a relation. Given a function f, an input $\mathbf{x} \in \{0,1\}^d$ and a feasible set $\mathcal{T}(\mathbf{x}) = \{\mathbf{z} : \mathbf{x} \rightarrow^*_{\mathcal{R}} \mathbf{z}\}$, solving the following maximization problem:*

$$\max_{\mathbf{z} \in \mathcal{T}(\mathbf{x})} l(f, \mathbf{z}, y)$$

is PSPACE-hard when $l(f, \mathbf{x}, y)$ is the 0-1 classification loss.

Proof. Let $\alpha : \{0,1\}^d \rightarrow \{0,1\}$ be a predicate. Define a loss function $l(f, \mathbf{z}, y)$ as follows: $l(f, \mathbf{z}, y) = \alpha(\mathbf{z})$ (the loss function is essentially the value of the predicate). Note that $\max_{\mathbf{z} \in \mathcal{T}(\mathbf{x})} l(f, \mathbf{z}, y)$ is equal to 1 iff there exists a $\mathbf{z} \in \mathcal{T}(\mathbf{x})$ such that $\alpha(\mathbf{z}) = 1$. This is a well known problem in model checking called *reachability analysis*, which is well known to be PSPACE-complete (the reduction is from the problem of checking emptiness for a set of DFAs, which is known to be PSPACE-complete Kozen (1977)). □

Recall that the maximation problem $\max_{\mathbf{z} \in B_p(\mathbf{x}, \epsilon)} l(f, \mathbf{z}, y)$ used in adversarial training for the image modality was proven to be NP-hard Katz et al. (2017). Hence it seems that the robust optimization problem in our context is in a higher complexity class than in the image domain.

A.4 Model Capacity Requirement

In this section, we illustrate how the *N&P* framework can potentially help reduce the model complexity for learning a robustly accurate classifier. We start with the following proposition.

Proposition 3 (Model Capacity Requirement)**.** *For some hypothesis class $\mathcal{H}$ and relation $\mathcal{R}$, there exists $f \in \mathcal{H}$ such that $f(\mathcal{N}_{\mathcal{R}}(\cdot))$ is robustly accurate, but no $f \in \mathcal{H}$ can be robustly accurate on the original inputs. In other words, robustly accurate classifier can only be obtained after normalization.*

We first define an equivalence relation induced by equivalent coordinates over binary inputs, and then write the formal statement of the observation in the following claim.

Definition 4 (Equivalence relation induced by equivalent coordinates)**.** *Let $\mathbf{x} = (\mathbf{x}_1, \cdots, \mathbf{x}_d)$ be a binary input vector on $\{0,1\}^d$, where each $\mathbf{x}_i, i \in \{1, \cdots, d\}$ is a coordinate. Let $I = \{1, \cdots, d\}$ be the set of coordinate indices for inputs in $\mathcal{X}$ and $U = \{i_1, \cdots, i_m\} \subseteq I$. In an equivalence relation $\mathcal{R}$ induced by U, $\mathbf{x} \rightarrow_{\mathcal{R}} \mathbf{z}$* iff *1) $\mathbf{x}_i = \mathbf{z}_i$ for all $i \in I \backslash U$, and 2) $\bigvee_{i \in U} \mathbf{x}_i = \bigvee_{i \in U} \mathbf{z}_i$. Notice that $\mathbf{x} \rightarrow_{\mathcal{R}} \mathbf{z}$ iff $\mathbf{z} \rightarrow_{\mathcal{R}} \mathbf{x}$.*

The notation $\bigvee_{i \in U} \mathbf{x}_i$ means taking a *logic or* operation over all $\mathbf{x}_i$s for $i \in U$. Intuitively, having an equivalence relation induced by coordinates with indices in U means the presence of any combination of such coordinates is equivalent to any other combination. Usage of interchangeable APIs in malware implementation is an example of equivalence relation: the attacker can choose any combination from a set of equivalent APIs to implement the same functionality.

In Definition 4, we use U to represent the set of *indices* of the equivalent coordinates. In the following theorems and proofs, we overload U to also represent the set of equivalent coordinates directly, and the $\bigvee$ operation will be taken over all coordinates in U.

Claim 2. *Consider $\mathcal{X} = \{0,1\}^5$ and $\mathcal{Y} = \{0,1\}$. Let the coordinates of an input $\mathbf{x} \in \mathcal{X}$ be $\{\mathbf{x}_1, \mathbf{x}'_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$. Suppose we have an equivalence relation induced by $U = \{\mathbf{x}_1, \mathbf{x}'_1\}$. Meanwhile, the true label y of an input $\mathbf{x}$ is* 1 iff *any of the following clauses is true: 1) $(\mathbf{x}_2 = 1) \wedge (\mathbf{x}_3 = 1)$, 2) $(\mathbf{x}_1 \vee \mathbf{x}'_1 = 1) \wedge (\mathbf{x}_2 = 1)$, 3) $(\mathbf{x}_1 \vee \mathbf{x}'_1 = 1) \wedge (\mathbf{x}_3 = 1) \wedge (\mathbf{x}_4 = 1)$. Then*

1. *no linear model can classify the inputs with perfect robust accuracy, but*
2. *a robust and accurate linear model exists under normalize-and-predict.*

Proof. Let $\mathcal{H} = \{f_{\mathbf{w},b} : sgn(\langle \mathbf{w}, \mathbf{x} \rangle + b)\}$. Let $\mathbf{w}_1, \mathbf{w}'_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{w}_4$ denote the coordinates in $\mathbf{w}$ that corresponds to $\mathbf{x}_1, \mathbf{x}'_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$.

We know $y = 1$ if $\mathbf{x}_1 = 1, \mathbf{x}_2 = 1$ and the other coordinates are zero because the second clause in the labeling rule is satisfied. Therefore, in order to classify this input instance correctly, we must have $\mathbf{w}_2 + \mathbf{w}_1 + b > 0$. Since $\mathbf{x}_1$ and $\mathbf{x}'_1$ are equivalent, we should also have $\mathbf{w}_2 + \mathbf{w}'_1 + b > 0$.

Similarly, we know $y = 0$ if $\mathbf{x}_2 = 1, \mathbf{x}_4 = 1$ and the other coordinates are zero because none of the clauses are satisfied. Therefore, we must have $\mathbf{w}_2 + \mathbf{w}_4 + b < 0$.

In order to classify all possible $\mathbf{x}$ correctly, the classifier $f_{\mathbf{w},b}$ must satisfy

$$\mathbf{w}_2 + \mathbf{w}_4 + b < 0 \tag{5}$$
$$\mathbf{w}_2 + \mathbf{w}_1 + b > 0 \tag{6}$$
$$\mathbf{w}_2 + \mathbf{w}_1' + b > 0 \tag{7}$$
$$\mathbf{w}_3 + \mathbf{w}_1 + \mathbf{w}_1' + b < 0 \tag{8}$$
$$\mathbf{w}_3 + \mathbf{w}_4 + \mathbf{w}_1 + b > 0 \tag{9}$$

First, by Formula 5, 6 and 7, we have $\mathbf{w}_1 > \mathbf{w}_4$ and $\mathbf{w}_1' > \mathbf{w}_4$. However, by Formula 8 and 9, we have $\mathbf{w}_1 + \mathbf{w}_1' < \mathbf{w}_4 + \mathbf{w}_1$, which implies $\mathbf{w}_1' < \mathbf{w}_4$. Contradition. Therefore, no linear classifier can satisfy all the equations.

On the other hand, if we perform normalization by letting $\mathbf{x}_1 = \mathbf{x}_1 \vee \mathbf{x}_1'$ and removing $\mathbf{x}_1'$, then a classifier $f_{\mathbf{w},b}$ – with $\mathbf{w}_1 = 0.4, \mathbf{w}_2 = 0.7, \mathbf{w}_3 = 0.5, \mathbf{w}_4 = 0.2, b = -1$ – can perfectly classify $\mathbf{x}$. □

A.5 Proof for Theorem 1

We first write down the full formal statement.

Theorem 3 (Preservation of robust accuracy)**.** *Consider $\mathcal{H}$ to be the set of all labeling functions on $\{0,1\}^d$. Let f^* be the classifier that minimizes the objective of our unified framework over data distribution $\mathcal{D}$, i.e. the optimal solution to*

$$\min_{f\in\mathcal{H}} \max_{A(\cdot)} \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \ell\left(f, A\left(\mathcal{N}_{\mathcal{R}'}(\mathbf{x})\right), y\right),$$

where ℓ is the 0-1 classification. Meanwhile, let f^*_{adv} be the classifier that minimizes the objective of adversarial training over $\mathcal{D}$, i.e. the optimal solution to

$$\min_{f\in\mathcal{H}} \max_{A(\cdot)} \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \ell(f, A(\mathbf{x}), y).$$

Then in principle, $f^*(\mathcal{N}_{\mathcal{R}'}(\cdot))$ and f^*_{adv} have the same optimal robust accuracy if $\mathcal{R}'$ is reversible.

Proof. We know by definition that f^*_{adv} has the highest possible robust accuracy. Therefore, it suffices to show that there exists a classifier $f(\mathcal{N}_{\mathcal{R}'}(\cdot))$ under our unified framework that has at least as good robust accuracy as f^*_{adv}.

We consider an f under our unified framework as follows. Let C denote a connected component (which is also strongly connected for reversible relation) C in $G_{\mathcal{R}'}$ and $\mathcal{Y}_C$ be the set of all labels that f^*_{adv} assign to inputs in C, i.e. $\mathcal{Y}_C = \{y \in \mathcal{Y} \,|\, \exists\, \mathbf{x} \in C \; s.t. \; f^*_{adv}(\mathbf{x}) = y\}$. Let $C_\mathbf{x}$ denote the connected component which contains $\mathbf{x}$. We consider a classifier f such that

$$f(\mathbf{x}) = \arg\max_{l\in\mathcal{Y}_{C_\mathbf{x}}} \mathop{\mathbb{E}}_{(\mathbf{z},y)\sim\mathcal{D}} \mathbb{1}(\mathbf{z} \in C_\mathbf{x} \wedge y = l), \tag{10}$$

i.e. f predicts the same label for all inputs in a connected component, and chooses the label that maximizes the clean accuracy in the connected component. Notice that $f(\mathbf{x}) = f(\mathcal{N}_{\mathcal{R}'}(\mathbf{x}))$ by construction, so we will use $f(\mathbf{x})$ instead of $f(\mathcal{N}_{\mathcal{R}'}(\mathbf{x})$ for simplicity.

Recall that we say a classifier f is robustly accurate over an input-label pair $(\mathbf{x}, y)$ under $\mathcal{R}$ if and only if $f(\mathbf{z}) = y$ for all $\mathbf{z} : \mathbf{x} \to^*_{\mathcal{R}} \mathbf{z}$. Now to show that f with the prediction rule in Equation 10 has the same robust accuracy as f^*_{adv}, we prove the following claim.

Claim 3. *The classifier f is robustly accurate over $(\mathbf{x}, y)$ under $\mathcal{R}$ if f^*_{adv} is robustly accurate $(\mathbf{x}, y)$ under $\mathcal{R}$.*

We prove the contrapositive of the claim. Suppose f is not robustly accurate over $(\mathbf{x}, y)$. Let $\mathbf{z}$ denote the input such that $\mathbf{x} \to^*_{\mathcal{R}} \mathbf{z}$ and $f(\mathbf{z}) \neq y$.

First, by the definition of $f(\mathbf{z})$, we know that there must be a $\mathbf{z}' \in C_{\mathbf{z}}$ such that $f^*_{adv}(\mathbf{z}') = f(\mathbf{z}) \neq y$ because f only predicts the label that has been used by f^*_{adv} over some input in the same connected component. Since $C_{\mathbf{z}}$ is strongly connected by our initial assumption, we must have $\mathbf{z} \rightarrow^*_{\mathcal{R}'} \mathbf{z}'$.

Now, because $\mathbf{x} \rightarrow^*_{\mathcal{R}} \mathbf{z}$ and $\mathbf{z} \rightarrow^*_{\mathcal{R}'} \mathbf{z}'$, we have $\mathbf{x} \rightarrow^*_{\mathcal{R}} \mathbf{z}'$. Since $\mathbf{x} \rightarrow^*_{\mathcal{R}} \mathbf{z}'$ but $f^*_{adv}(\mathbf{z}') \neq y$, we can conclude that f^*_{adv} is not robustly accurate at $\mathbf{x}$, and thus prove the claim.

Claim 3 suggests that f has at least the same robust accuracy as f^*_{adv}. Since $f(\mathbf{x}) = f(\mathcal{N}_{\mathcal{R}'}(\mathbf{x}))$ by our construction of f, we know $f(\mathcal{N}_{\mathcal{R}'}(\cdot))$ has at least the same robust accuracy as f^*_{adv} too. Moreover, $f^*(\mathcal{N}_{\mathcal{R}'}(\cdot))$, defined as the classifier with the highest robust accuracy after normalizaiton, should have at least the same robust accuracy as $f(\mathcal{N}_{\mathcal{R}'}(\cdot))$. Therefore, $f^*(\mathcal{N}_{\mathcal{R}'}(\cdot))$ has at least the same robust accuracy as f^*_{adv}. On the other hand, we know f^*_{adv} has the highest possible robust accuracy by definition. Therefore, we can conclude that $f^*(\mathcal{N}_{\mathcal{R}'}(\cdot))$ and f^*_{adv} have the same robust accuracy. □

A.6 Computation Efficiency of Normalizer for Reversible Relation

We show in Proposition 4 that normalizing over reversible relation is at most as hard as finding the strongest adversarial example, which is required for optimal adversarial training.

Proposition 4. *Let $(\mathbf{x}, y)$ denote a sample point where $\mathbf{x}$ is the input vector and y is the ground truth label. Let $\mathcal{R}'$ denote a reversible relation, and $\mathcal{T}(\mathbf{x}) = \{\mathbf{z} \,|\, \mathbf{x} \rightarrow^*_{\mathcal{R}'} \mathbf{z}\}$ denote the feasible set of adversarial examples. Let $\mathbf{z}^* = \arg\max_{\mathbf{z} \in \mathcal{T}(\mathbf{x})} \ell(f, \mathbf{z}, y)$, i.e. the most powerful adversarial example of $\mathbf{x}$ for model f, then the normalizer $\mathcal{N}(\mathbf{x}) = \mathbf{z}^*$ satisfies the robustness condition in Proposition 1.* [5]

Let C_i be any connected component in $G_{\mathcal{R}'}$. Since $\mathcal{R}'$ is reversible, C_i shall be strongly connected. The strongest adversarial example to an input $\mathbf{x} \in C_i$ shall be the same for all nodes in C_i. Meanwhile, the strongest adversarial example is deterministic given a consistent tie-breaker between adversarial choices with the same test loss. Therefore, $\mathcal{N}(\mathbf{x}) = \mathbf{z}^*$ satisfies the robustness condition.

A.7 Semantics-Preserving Transformation and Most Likely Label

In Figure 1, one may notice that two inputs $\mathbf{x}$ and $\mathbf{z}$ may have different most likely labels even though $\mathbf{z}$ is obtained from $\mathbf{x}$ by a semantics-preserving transformation. This phenomenon may look bizarre, but is indeed possible in real-world settings. The main reason is that the input vector may contain information that (1) is irrelevant to the essential semantics for classification task, and yet (2) may enhance classification accuracy on clean inputs.

Taking malware detection as an example. A group of zip tools may have the same static feature vector $\mathbf{x}$ as some ransomware. Authors of zip tools may agree to use a secret syntactic pattern with no semantic implication in an attempt to distinguish from ransomwares circulated on the web. Let $\mathbf{z}$ be the static feature vector of zip tools after the pattern is added. Now given the natural distribution and the absence of relational adversaries, $\mathbf{z}$ will certainly be classified as benign zip tools, while $\mathbf{x}$ is predominantly ransomware.

However, in the long run, an adaptive malware author will eventually know this secret pattern, and also add it to his ransomware. Similarly, the secret pattern may also be gradually abandoned by authors of zip tools as time goes by making the zip tools identical to the ransomware. In short, $(\mathbf{x}, \mathbf{z})$ is a semantics-preserving transformation whose source $\mathbf{x}$ and target $\mathbf{z}$ do not have the same most likely label on the clean distribution.

[5] A mild assumption is that the argmax solver has a consistent tie-breaker in case multiple optima exist.

B HUE, SATURATION AND VALUE

This section is heavily based on Hosseini & Poovendran (2018).

HSV (Hue, Saturation and Value) is an alternative to RGB (red, green and blue) color space and is known to more closely represent the way human vision perceives color (Koschan et al., 2008). The hue channel corresponds to the color's position on the color wheel. As hue increases from 0 to 1, the color transitions from red to orange, yellow, green, cyan, blue, magenta, and finally back to red. Saturation measures the colorfulness, i.e., setting saturation to 0 yields a grayscale image and increasing it to 1 generates the most colorful image with same colors. Value shows the brightness, which is maximum value of red, green and blue components. Figure 2 shows the effect of changing the hue component on a image.

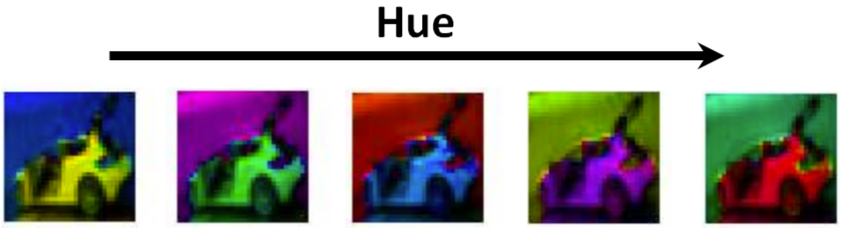

Figure. 2: Illustration of shifting hue component in HSV space on a sample image in CIFAR10.

C Algorithms and Experiment Implementation Details

In this section, we present the omitted algorithm descriptions and experiment implementation details.

C.1 Generic Normalizer

We propose a generic decidable algorithm of finding the normal form. Let C_i be any weakly connected component in $G_{\mathcal{R}}$ induced by $\mathcal{R}$. We first construct a new graph from C_i by condensing all nodes in the same cycle into a single node for all cycles in C_i. Suppose $S = \{\mathbf{x}_1, \cdots, \mathbf{x}_m\}$ is the set of nodes in a cycle. We only keep $\mathbf{x}_1$ in the graph, and replace edge $(\mathbf{x}_i, \mathbf{z})$ with $(\mathbf{x}_1, \mathbf{z})$ and edge $(\mathbf{z}, \mathbf{x}_i)$ with $(\mathbf{z}, \mathbf{x}_1)$ for all $\mathbf{x}_i \in S$ and all $\mathbf{z}$ in the graph. We repeat this procedure until no cycle exists in the remaining graph, and we call the final graph C_i'. Since C_i' is acyclic by construction, we can fix a topological order over its nodes. A generic normalizer $\mathcal{N}(\mathbf{x})$ can then return the element in C_i with the largest topological order.

We also want to note that in practice, the normalizer can be much more efficient than this generic one. Normalizing over equivalence relations, for example, only requires picking one representitive element from an equivalence group, which often does not require a graph travesal over $G_{\mathcal{R}'}$.

C.2 Generic Relational Attack Algorithms

In Sec 6, we introduce two generic relational attack algorithms – GreedyByGroup and GreedyByGrad. The algorithm boxes below shows the exact description of both algorithms.

Algorithm 1 GreedyByGroup $(\mathbf{x}, y, K)$

```
$\mathbf{x}^{adv} = \mathbf{x}, k = 0$
Partition $\mathcal{R}$ into $m$ groups $\{\mathcal{R}_1, \cdots, \mathcal{R}_m\}$.
while $k < K$ do
    $k = k + 1$
    for $\mathcal{R}_i \in \{\mathcal{R}_1, \cdots, \mathcal{R}_m\}$ do
        $\mathbf{x}_i = \arg \max_{\mathbf{z}:\mathbf{x}^{adv} \rightarrow^*_{R_i} \mathbf{z}} \ell(f, \mathbf{z}, y)$.
    end
    Combine $\mathbf{x}_i$s to obtain the new $\mathbf{x}^{adv}$
end
return $\mathbf{x}^{adv}$
```

Algorithm 2 GreedyByGrad$(\mathbf{x}, y, m, K)$

```
$\mathbf{x}^{adv} = \mathbf{x}, k = 0$
while $k < K$ do
    $k = k + 1$
    $g = \nabla_{\mathbf{x}} \ell(f, \mathbf{x}^{adv}, y)$
    for $(\mathbf{x}^{adv}, \mathbf{z}) \in \mathcal{R}$ do
        $c_{(\mathbf{x}^{adv}, \mathbf{z})} = \sum_{i=1}^{d} g_i(\mathbf{z}_i - \mathbf{x}_i^{adv})$
    end
    Apply the transformations with top $m$ largest
    positive $c_{(\mathbf{x}^{adv}, \mathbf{z})}$ to obtain the new $\mathbf{x}^{adv}$.
end
return $\mathbf{x}^{adv}$
```

C.3 Extract API Substitution Rules

In Sec 6, we consider malware authors who can substitute API calls with equivalent API calls to evade ML-based malware detector. We now explain how we extract the equivalent APIs. We identify four types of patterns for extracting equivalent APIs:

- API with the same name but located in different Dynamically Linkable Libraries (DLLs). For example, `memcpy`, a standard C library function, is shipped in libraries with different names, including `crtdll.dll`, `msvcr90.dll`, and `msvcr110.dll`.
- API with and without the `Ex` suffix. The `Ex` suffix represents an extension to the same API without the suffix.
- API with and without the `A` or `W` suffixes. The `A` suffix represents the single character version. The `W` suffix represents the wide character version.
- API with/without `_s` suffix. The `_s` suffix represents the secure version of an API.

Using these four patterns, we extracted about 2,000 equivalent API groups. About 500 of the groups have more than 2 APIs and the maximal group has 23 APIs.