Knowledge Enhanced Machine Learning Pipeline against Diverse Adversarial Attacks

Nezihe Merve Gürel [1 *] **Xiangyu Qi** [2 *] **Luka Rimanic** [1] **Ce Zhang** [1] **Bo Li** [3]

Abstract

Despite the great successes achieved by deep neural networks (DNNs), recent studies show that they are vulnerable against adversarial examples, which aim to mislead DNNs by adding small adversarial perturbations. Several defenses have been proposed against such attacks, while many of them have been adaptively attacked. In this work, we aim to enhance the ML robustness from a different perspective by leveraging *domain knowledge*: We propose a Knowledge Enhanced Machine Learning Pipeline (KEMLP) to integrate domain knowledge (i.e., logic relationships among different predictions) into a probabilistic graphical model via first-order logic rules. In particular, we develop KEMLP by integrating a diverse set of weak auxiliary models based on their logical relationships to the main DNN model that performs the target task. Theoretically, we provide convergence results and prove that, under mild conditions, the prediction of KEMLP is more robust than that of the main DNN model. Empirically, we take road sign recognition as an example and leverage the relationships between road signs and their shapes and contents as domain knowledge. We show that compared with adversarial training and other baselines, KEMLP achieves higher robustness against physical attacks, $\mathcal{L}_p$ bounded attacks, unforeseen attacks, and natural corruptions under both whitebox and blackbox settings, while still maintaining high clean accuracy.

[*]Equal contribution [1]ETH Zurich, Zurich, Switzerland [2]Zhejiang University, China (work done during remote internship at UIUC) [3]University of Illinois at Urbana-Champaign, Illinois, USA. Correspondence to: Nezihe Merve Gürel <nezihe.guerel@inf.ethz.ch>, Xiangyu Qi <unispac@zju.edu.cn>, Ce Zhang <ce.zhang@inf.ethz.ch>, Bo Li <lbo@illinois.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

1. Introduction

Recent studies show that machine learning (ML) models are vulnerable to different types of adversarial examples, which are adversarially manipulated inputs aiming to mislead ML models to make arbitrarily incorrect predictions (Szegedy et al., 2013; Goodfellow et al., 2015; Bhattad et al., 2020; Eykholt et al., 2018). Different defense strategies have been proposed against such attacks, including adversarial training (Shafahi et al., 2019; Madry et al., 2017), input processing (Ross and Doshi-Velez, 2018), and approaches with certified robustness against $\mathcal{L}_p$ bounded attacks (Cohen et al., 2019; Yang et al., 2020a). However, these defenses have either been adaptively attacked again (Carlini and Wagner, 2017a; Athalye et al., 2018) or can only certify the robustness within a small ℓ_p perturbation radius. In addition, when models are trained to be robust against one type of attack, their robustness is typically not preserved against other attacks (Schott et al., 2018; Kang et al., 2019). Thus, despite the rapid recent progress on robust learning, it is still challenging to provide robust ML models against a diverse set of adversarial attacks in practice.

In this paper, we take a different perspective towards training robust ML models against diverse adversarial attacks by integrating *domain knowledge* during prediction, given the observation that human with knowledge is quite resilient against these attacks. We will first take stop sign recognition as a simple example to illustrate the potential role of knowledge in ML prediction. In this example, the **main task** is to predict whether a stop sign appears in the input image. Training a DNN model for this task is known to be vulnerable against a range of adversarial attacks (Eykholt et al., 2018; Xiao et al., 2018a). However, upon such a DNN model, if we could (1) build a detector for a different **auxiliary task**, e.g., detecting whether an octagon appears in the input by using other learning strategies such as traditional computer vision techniques, and (2) integrate the **domain knowledge** such that *"A stop sign should be of an octagon shape"*, it is possible that additional information could enable the ML system to detect or defend against attacks, which lead to conflicts between the DNN prediction and domain knowledge. For instance, if a speed limit sign with *rectangle* shape is misrecognized as a stop sign, the

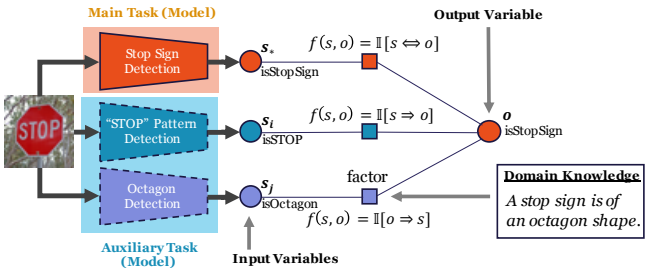

Figure 1. An overview of the KEMLP framework. KEMLP constructs a factor graph by modeling the output of ML models as random input variables, and the KEMLP prediction as a random output variable. It integrates domain knowledge via factors connecting different random variables.

ML system would identify this conflict and try to correct the prediction.

Inspired by this intuition, we aim to understand how to *enhance the robustness of ML models via domain knowledge integration*. Despite the natural intuition in the previous simple example, providing a technically rigorous treatment to this problem is far from trivial, yielding the following questions: How should we integrate domain knowledge in a principled way? When will integrating domain knowledge help with robustness and will there be a tradeoff between robustness and clean accuracy? Can integration of domain knowledge genuinely bring additional robustness benefits against practical attacks when compared with state-of-the-art defenses?

In this work, we propose KEMLP, a framework that facilitates the integration of *domain knowledge* in order to improve the robustness of ML models. Figure 1 illustrates the KEMLP framework. In KEMLP, the outputs of different ML models are modeled as random input variables, whereas the output of KEMLP is modeled as another variable. To integrate domain knowledge, KEMLP introduces corresponding factors connecting these random variables. For example, as illustrated in Figure 1, the knowledge rule "*A stop sign is of an octagon shape*" introduces a factor between the input variable (i.e., the output of the octagon detector) and the output variable (i.e., output of the stop sign detector) with a factor function that *the former implies the latter*. To make predictions, KEMLP runs statistical inference over the factor graph constructed by integrating all such domain knowledge expressed as first-order logic rules, and output the marginal probability of the output variable.

Based on KEMLP, our main goal is to understand two fundamental questions based on KEMLP: (1) *What type of knowledge is needed to improve the robustness of the joint inference results from KEMLP, and can we prove it?* (2) *Can we show that knowledge integration in the KEMLP framework can provide significant robustness gain over powerful state-of-the-art models?*

We conduct theoretical analysis to understand the first question, focusing on two specific types of knowledge rules: (1) *permissive knowledge* of the form "$B \implies A$", and (2) *preventive knowledge* of the form "$A \implies B$", where A represents the main task, B an auxiliary task and $\implies$ denotes logical implication. We focus on the *weighted robust accuracy*, which is a weighted average of accuracies on benign and adversarial examples, respectively, and we derive sufficient conditions under which KEMLP outperforms the main task model alone. Under mild conditions, we show that integrating multiple weak auxiliary models, both in their robustness and quality, together with the permissive and preventive rules, the weighted robust accuracy of KEMLP can be guaranteed to improve over the single main task model. To our best knowledge, this is the first analysis of proposed form, focusing on the intersection of knowledge integration, joint inference, and robustness.

We then conduct extensive empirical studies to understand the second question. We focus on the road sign classification task and consider the state-of-the-art adversarial training models based on both the $\mathcal{L}_p$ bounded perturbation and occlusion perturbations (Wu et al., 2019) as our baselines as well as the main task model. We show that by training weak auxiliary models for recognizing the shapes and contents of road signs, together with the corresponding knowledge rules as illustrated in Figure 1, KEMLP achieves significant improvements on their robustness compared with baseline main task models against a *diverse* set of adversarial attacks while maintaining similar or even higher clean accuracy, given its improvement on the tradeoff between clean accuracy and robustness. In particular, we consider existing physical attacks (Eykholt et al., 2018), $\mathcal{L}_p$ bounded attacks (Madry et al., 2017), unforeseen attacks (Kang et al., 2019), and common corruptions (Hendrycks and Dietterich, 2019), under both whitebox and blackbox settings. To our best knowledge, KEMLP is the first ML model robust to diverse attacks in practice with high clean accuracy. Our code is publicly available for reputability [1].

Technical Contributions. In this paper, we take the *first* step towards integrating *domain knowledge* with ML to improve its robustness against different attacks. We make contributions on both theoretical and empirical fronts.

- We propose KEMLP, which integrates a main task ML model with a set of weak auxiliary task models, together with different knowledge rules connecting them.
- Theoretically, we provide the robustness guarantees for KEMLP and prove that under mild conditions, the prediction of KEMLP is more robust than that of a single main task model.
- Empirically, we develop KEMLP based on different main

[1] https://github.com/AI-secure/Knowledge-Enhanced-Machine-Learning-Pipeline

task models, and evaluate them against a diverse set of attacks, including physical attacks, $\mathcal{L}_p$ bounded attacks, unforeseen attacks, and common corruptions. We show that the robustness of KEMLP outperforms all baselines by a wide margin, with comparable and often higher clean accuracy.

2. Related Work

In the following, we review several bodies of literature that are relevant to the objective of our paper.

Adversarial examples are carefully crafted inputs aiming to mislead well-trained ML models (Goodfellow et al., 2015; Szegedy et al., 2013). A variety of approaches to generate such adversarial examples have also been proposed based on different perturbation measurement metrics, including $\mathcal{L}_p$ bounded, unrestricted, and physical attacks (Wong et al., 2019; Bhattad et al., 2020; Xiao et al., 2018b;c; Eykholt et al., 2018).

Defense methods against such attacks have been proposed. Empirically, *adversarial training* (Madry et al., 2017) has shown to be effective, together with feature quantization (Xu et al., 2017) and reconstruction approaches (Samangouei et al., 2018). Certified robustness has also been studied by propagating the interval bound of a NN (Gowal et al., 2018), or randomized smoothing of a given model (Cohen et al., 2019). Several approaches have further improved it: by choosing different smoothing distributions for different L_p norms (Dvijotham et al., 2020; Zhang et al., 2020; Yang et al., 2020a), or training more robust smoothed classifiers via data augmentation (Cohen et al., 2019), unlabeled data (Carmon et al., 2019), adversarial training (Salman et al., 2019), and regularization (Li et al., 2019; Zhai et al., 2019). While most prior defenses focus on leveraging statistical properties of an ML model to improve its robustness, they can only be robust towards a specific type of attack, such as ℓ_p bounded attacks. This paper aims to explore how to utilize knowledge inference information to improve the robustness of a logically connected ML pipeline against a diverse set of attacks.

Joint inference has been studied to take multiple predictions made by different models, together with the relations among them, to make a final prediction (Xu et al., 2020; Deng et al., 2014; Poon and Domingos, 2007; McCallum, 2009; Chen et al., 2014; Chakrabarti et al., 2014; Biba et al., 2011). These approaches usually use different inference models, such as factor graphs (Wainwright and Jordan, 2008), Markov logic networks (Richardson and Domingos, 2006) and Bayesian networks (Neuberg, 2003), as a way to characterize their relationships. The programmatic weak supervision approaches (Ratner et al., 2016; 2017) also perform joint inference by employing labeling functions and using generative modeling techniques, which aims to create noisy training data. In this paper, we take a different perspective on this problem — we explore the potential of using joint inference with the objective of integrating domain knowledge and to eventually improving the ML robustness. As we will see, by integrating domain knowledge, it is possible to improve the learning robustness by a wide margin.

3. KEMLP: Knowledge Enhanced Machine Learning Pipeline

We first present the proposed framework KEMLP, which aims to improve the robustness of an ML model by integrating a diverse set of domain knowledge. In this section, we formally define the KEMLP framework.

We consider a classification problem under a supervised learning setting, defined on a feature space $\mathcal{X}$ and a finite label space $\mathcal{Y}$. We refer to $x \in \mathcal{X}$ as an input and $y \in \mathcal{Y}$ as the target variable. An input x can be a benign example or an adversarial example. To model this, we use $z \in \{0, 1\}$, a latent variable that is not exposed to KEMLP. That is, x is an adversarial example with $(x, y) \sim \mathcal{D}_a$ whenever $z = 1$, and $(x, y) \sim \mathcal{D}_b$ otherwise, where $\mathcal{D}_a$ and $\mathcal{D}_b$ represent the adversarial and benign data distributions. We let $\pi_{\mathcal{D}_a} = \mathbb{P}(z = 1)$ and $\pi_{\mathcal{D}_b} = \mathbb{P}(z = 0)$, implying $\pi_{\mathcal{D}_a} + \pi_{\mathcal{D}_b} = 1$. For convenience, we denote $\mathbb{P}_{\mathcal{D}_a}(x, y) = \mathbb{P}(x, y|z = 1)$ and $\mathbb{P}_{\mathcal{D}_b}(x, y) = \mathbb{P}(x, y|z = 0)$. In the following, to ease the exposition, we slightly abuse the notation and use probability densities for discrete distributions.

Given an input x whose corresponding z is unknown (benign or adversarial), KEMLP aims to predict the target variable y by employing a set of *models*. These predictive models are constructed, say, using ML or some other traditional rule-based methods (e.g., edge detector). For simplicity, we describe the KEMLP framework as a binary classification task, in which case $\mathcal{Y} = \{0, 1\}$, noting that the multi-class scenario is a simple extension of it. We introduce the KEMLP framework as follows.

Models Models are a collection of predictive ML models, each of which takes as input x and outputs some predictions. In KEMLP, we distinguish three different type of models.

- *Main task model*: We call the (untrusted) ML model whose robustness users want to enhance as the *main task model*, denoting its predictions by $s_* \in \mathcal{Y}$.
- *Permissive models*: Let $s_{\mathcal{I}} = \{s_i : i \in \mathcal{I}\}$ be a set of m permissive models, each of which corresponds to the prediction of one ML model. Conceptually, permissive models are usually designed for specific events which are *sufficient* for inferring $y = 1$: $s_i \implies y$.
- *Preventative models*: Similarly, we have n preventative

models: $s_{\mathcal{J}} = \{s_j : j \in \mathcal{J}\}$, each of which corresponds to the prediction of one ML model. Conceptually, preventative models capture the events that are *necessary* for the event $y = 1$: $y \implies s_j$.

Knowledge Integration Given a data example $(x, y) \sim \mathcal{D}_b$ or $(x, y) \sim \mathcal{D}_a$, y is unknown to KEMLP. We create a factor graph to embed the domain knowledge as follows. The outputs of each model over x become *input variables*: $s_*, s_{\mathcal{I}} = \{s_i : i \in \mathcal{I}\}, s_{\mathcal{J}} = \{s_j : j \in \mathcal{J}\}$. KEMLP also has an output variable $o \in \mathcal{Y}$, which corresponds to its prediction. Different models introduce different types of factors connecting these variables:

- Main model: KEMLP introduces a factor between the main model s_* and the output variable o with factor function $f_*(o, s_*) = \mathbb{1}\{o = s_*\}$;
- Permissive model: KEMLP introduces a factor between each permissive model s_i and the output variable o with factor function $f_i(o, s_i) = \mathbb{1}\{s_i \implies o\}$.
- Preventative model: KEMLP introduces a factor between each preventative model s_j and the output variable o with factor function $f_j(o, s_j) = \mathbb{1}\{o \implies s_j\}$.

Learning with KEMLP To make a prediction, KEMLP outputs the *probability* of the output variable o. KEMLP assigns a weight for each model and constructs the following statistical model:

$$\begin{aligned}\mathbb{P}[o|s_*, s_{\mathcal{I}}, s_{\mathcal{J}}, w_*, w_{\mathcal{I}}, w_{\mathcal{J}}, b_o] \;\propto\; & \\ \exp\{b_o + w_* f_*(o, s_*)\}\times & \\ \exp\Big\{\sum_{i\in\mathcal{I}} w_i f_i(o, s_i)\Big\} \times \exp\Big\{\sum_{j\in\mathcal{J}} w_j f_j(o, s_j)\Big\} &\end{aligned}$$

where w_*, w_i, w_j are the corresponding weights for models s_*, s_i, s_j, $w_{\mathcal{I}} = \{w_i : i \in \mathcal{I}\}, w_{\mathcal{J}} = \{w_j : j \in \mathcal{J}\}$ and b_o is some bias parameter that depends on o. For the simplicity of exposition, we use an equivalent notation by putting all the weights and outputs of factor functions into vectors using an ordering of models. More precisely, we define

$$\begin{aligned}\mathbf{w} &= [1; w_*; (w_i)_{i\in\mathcal{I}}; (w_j)_{j\in\mathcal{J}}], \\ \mathbf{f}_o(s_*, s_{\mathcal{I}}, s_{\mathcal{J}}) &= [b_o; f_*(o, s_*); (f_i(o, s_i))_{i\in\mathcal{I}}; (f_j(o, s_j))_{j\in\mathcal{J}}],\end{aligned}$$

for $o \in \mathcal{Y}$. All concatenated vectors from above are in $\mathbb{R}^{m+n+2}$. Given this, an equivalent form of KEMLP's statistical model is

$$\mathbb{P}[o|s_*, s_{\mathcal{I}}, s_{\mathcal{J}}, \mathbf{w}] = \frac{1}{Z_{\mathbf{w}}} \exp(\langle \mathbf{w}, \mathbf{f}_o(s_*, s_{\mathcal{I}}, s_{\mathcal{J}})\rangle) \quad (1)$$

where $Z_{\mathbf{w}}$ is the normalization constant over $o \in \mathcal{Y}$. With some abuse of notation, $\mathbf{w}$ is meant to govern all parameters including weights and biases whenever used with probabilities.

Weight Learning During the training phase of KEMLP, we choose parameters $\mathbf{w}$ by performing standard maximum likelihood estimation over a training dataset. Given a particular input instance $x^{(n)}$, respective model predictions $s_*^{(n)}, s_{\mathcal{I}}^{(n)}, s_{\mathcal{J}}^{(n)}$, and the ground truth label $y^{(n)}$, we minimize the negative log-likelihood function in view of

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \Big\{ -\sum_n \log\Big(\mathbb{P}[o^{(n)} = y^{(n)}|s_*^{(n)}, s_{\mathcal{I}}^{(n)}, s_{\mathcal{J}}^{(n)}, \mathbf{w}]\Big)\Big\}.$$

Inference During the inference phase of KEMLP, given an input example $\hat{x}$, we predict $\hat{y}$ that has the largest probability given the respective model predictions $\hat{s}_*, \hat{s}_{\mathcal{I}}, \hat{s}_{\mathcal{J}}$, namely, $\hat{y} = \arg\max_{\tilde{y}\in\mathcal{Y}} \mathbb{P}[o = \tilde{y}|\hat{s}_*, \hat{s}_{\mathcal{I}}, \hat{s}_{\mathcal{J}}, \hat{\mathbf{w}}]$.

4. Theoretical Analysis

How does knowledge integration impact the robustness of KEMLP? In this section, we provide theoretical analysis about the impact of domain knowledge integration on the robustness of KEMLP. We hope to (1) depict the regime under which knowledge integration can help with robustness; (2) explain how a collection of "weak" (in terms of prediction accuracy) but "robust" auxiliary models, on tasks different from the main one, can be used to boost overall robustness. Here we state the main results, whereas we refer interested readers to Appendix A where we provide all relevant details.

Weighted Robust Accuracy Previous theoretical analysis on ML robustness (Javanmard et al., 2020; Xu et al., 2009; Raghunathan et al., 2020) have identified two natural dimensions of model quality: *clean accuracy* and *robust accuracy*, which are the accuracy of a given ML model on inputs x drawn from either the benign distribution $\mathcal{D}_b$ or adversarial distribution $\mathcal{D}_a$. In this paper, to balance their tradeoff, we use their weighted average as our main metric of interest. That is, given a classifier $h : \mathcal{X} \to \mathcal{Y}$ we define its *Weighted Robust Accuracy* as

$$\mathcal{A}_h = \pi_{D_a}\mathbb{P}_{\mathcal{D}_a}[h(x) = y] + \pi_{D_b}\mathbb{P}_{\mathcal{D}_b}[h(x) = y].$$

We use $\mathcal{A}^{\text{KEMLP}}$ and $\mathcal{A}^{\text{main}}$ to denote the weighted robust accuracies of KEMLP and main task model, respectively.

4.1. $\mathcal{A}^{\text{KEMLP}}$: Weighted Robust Accuracy of KEMLP

The goal of our analysis is to identify the regime under which $\mathcal{A}^{\text{KEMLP}} > \mathcal{A}^{\text{main}}$ is guaranteed. The main analysis to achieve this hinges on deriving the weighted robust accuracy $\mathcal{A}^{\text{KEMLP}}$ for KEMLP. We first describe the modeling assumptions of our analysis, and then describe two key characteristics of models, culminating in a lower bound of $\mathcal{A}^{\text{KEMLP}}$.

Modeling Assumptions We assume that for a fixed z, that is, for a fixed $\mathcal{D} \in \{\mathcal{D}_b, \mathcal{D}_a\}$, the models make independent

errors given the target variable. Thus, for all $\mathcal{D} \in \{\mathcal{D}_b, \mathcal{D}_a\}$, the class conditional distribution can be decomposed as

$$\mathbb{P}_{\mathcal{D}}[s_*, s_{\mathcal{I}}, s_{\mathcal{J}}|y] = \mathbb{P}_{\mathcal{D}}[s_*|y] \prod_{i\in\mathcal{I}} \mathbb{P}_{\mathcal{D}}[s_i|y] \prod_{j\in\mathcal{J}} \mathbb{P}_{\mathcal{D}}[s_j|y].$$

We also assume for simplicity that the main task model makes symmetric errors given the class of target variable, that is, $\mathbb{P}_{\mathcal{D}}[s_* \neq y|y]$ is fixed with respect to y for all $\mathcal{D} \in \{\mathcal{D}_b, \mathcal{D}_a\}$.

Characterizing Models: Truth Rate (α) and False Rate (ϵ) Each auxiliary model $k \in \mathcal{I} \cup \mathcal{J}$ is characterized by two values, their truth rate (α) and false rate (ϵ) over benign and adversarial distributions. These values measure the *consistency* of the model with the ground truth:

Permissive Models:

$$\alpha_{i,\mathcal{D}} := \mathbb{P}_{\mathcal{D}}[s_i = y|y = 1], \quad \epsilon_{i,\mathcal{D}} := \mathbb{P}_{\mathcal{D}}[s_i \neq y|y = 0]$$

Preventative Models:

$$\alpha_{j,\mathcal{D}} := \mathbb{P}_{\mathcal{D}}[s_j = y|y = 0], \quad \epsilon_{j,\mathcal{D}} := \mathbb{P}_{\mathcal{D}}[s_j \neq y|y = 1]$$

Note that, given the asymmetric nature of these auxiliary models, we do *not* necessarily have $\epsilon_{k,\mathcal{D}} = 1 - \alpha_{k,\mathcal{D}}$. In addition, for a high quality permissive model ($k \in \mathcal{I}$), or a high quality preventative model ($k \in \mathcal{J}$) for which the logic rules mostly hold, we expect $\alpha_{k,\mathcal{D}}$ to be large and $\epsilon_{k,\mathcal{D}}$ to be small.

We define the truth rate of main model over data examples drawn from $\mathcal{D} \in \{\mathcal{D}_b, \mathcal{D}_a\}$ as $\alpha_{*,\mathcal{D}} := \mathbb{P}_{\mathcal{D}}(s_* = y)$, and its false rate as $\epsilon_{*,\mathcal{D}} := \mathbb{P}_{\mathcal{D}}(s_* \neq y) = 1 - \alpha_{*,\mathcal{D}}$.

These characteristics are of integral importance to weighted robust accuracy of KEMLP. To combine all the models together, we define upper and lower bounds to truth rates and false rates. For the main model, we have ${}_\wedge\alpha_* := \min_{\mathcal{D}} \alpha_{*,\mathcal{D}}$ and ${}_\vee\alpha_* := \max_{\mathcal{D}} \alpha_{*,\mathcal{D}}$. For the auxiliary models, on the other hand, for each model index $k \in \mathcal{I} \cup \mathcal{J}$, we have

$${}_\wedge\alpha_k := \min_{\mathcal{D}} \alpha_{k,\mathcal{D}}, \quad {}_\wedge\epsilon_k := \min_{\mathcal{D}} \epsilon_{k,\mathcal{D}}$$
$${}_\vee\alpha_k := \max_{\mathcal{D}} \alpha_{k,\mathcal{D}}, \quad {}_\vee\epsilon_k := \max_{\mathcal{D}} \epsilon_{k,\mathcal{D}}.$$

Intuitively, the difference between ${}_\wedge\alpha$ and ${}_\vee\alpha$ (resp. ${}_\wedge\epsilon$ and ${}_\vee\epsilon$) indicates the "robustness" of each individual model. If a model performs very similarly when it is given a benign and an adversarial example, we have that ${}_\wedge\alpha$ should be similar to ${}_\vee\alpha$ (resp. ${}_\wedge\epsilon$ to ${}_\vee\epsilon$).

The truth and false rates of models directly influence the factor weights which govern the influence of models in the main task. In Appendix A.2 we prove that the optimal weight of an auxiliary model is bounded by $w_k \geq \log {}_\wedge\alpha_k(1 - {}_\vee\epsilon_k)/(1 - {}_\wedge\alpha_k){}_\vee\epsilon_k$, for all $k \in \mathcal{I} \cup \mathcal{J}$. That is, the lowest truth rate and highest false rate of an auxiliary model (resp. ${}_\wedge\alpha_k$ and ${}_\vee\epsilon_k$) are indicative of its influence in the main task. By taking partial derivatives, this lower bound can be shown to be increasing in ${}_\wedge\alpha_k$ and decreasing in ${}_\vee\epsilon_k$. That is, as the lowest truth rate of a model gets higher, KEMLP increases its influence in the weighted majority voting accordingly – in the above nonlinear fashion. The lowest truth rate is often determined by the *robust accuracy*. As a result, the more "robust" an auxiliary model is, the larger the influence on KEMLP, which naturally contributes to its robustness.

Weighted Robust Accuracy of KEMLP We now provide a lower bound on the weighted robust accuracy of KEMLP, which can be written as

$$\mathcal{A}^{\text{KEMLP}} = \mathbb{E}_{\mathcal{D}\sim\{\mathcal{D}_a,\mathcal{D}_b\}} \mathbb{E}_{y\sim\mathcal{Y}} \big[\mathbb{P}_{\mathcal{D}}[o = y|y, \mathbf{w}]\big]. \quad (2)$$

We first provide one key technical lemma followed by the general theorem.

We see that the key component in $\mathcal{A}^{\text{KEMLP}}$ is $\mathbb{P}_{\mathcal{D}}[o = y|y, \mathbf{w}]$, the conditional probability that a KEMLP pipeline outputs the correct prediction. Using knowledge aggregation rules f_*, f_i and f_j, as well as (1), for each $\mathcal{D} \in \{\mathcal{D}_b, \mathcal{D}_a\}$ we have

$$\begin{aligned}\mathbb{P}_{\mathcal{D}}[o = y|y, \mathbf{w}] &= \mathbb{P}_{\mathcal{D}}\Big[\mathbb{P}[o = y|s_*, s_{\mathcal{I}}, s_{\mathcal{J}}, \mathbf{w}] > 1/2\Big|y\Big] \\ &= \mathbb{P}_{\mathcal{D}}\big[\langle \mathbf{w}, \mathbf{f}_y(s_*, s_{\mathcal{I}}, s_{\mathcal{J}}) - \mathbf{f}_{1-y}(s_*, s_{\mathcal{I}}, s_{\mathcal{J}})\rangle > 0|y\big].\end{aligned}$$

To bound the above value, we need to characterize the concentration behavior of the random variable

$$\Delta_{\mathbf{w}}(y, s_*, s_{\mathcal{I}}, s_{\mathcal{J}}) := \langle \mathbf{w}, \mathbf{f}_y(s_*, s_{\mathcal{I}}, s_{\mathcal{J}}) - \mathbf{f}_{1-y}(s_*, s_{\mathcal{I}}, s_{\mathcal{J}})\rangle.$$

That is, we need to bound its left tail below zero. For this purpose, we reason about its expectation, leading to the following lemma.

Lemma 1. *Let $\Delta_{\mathbf{w}}$ be a random variable defined above. Suppose that KEMLP uses optimal parameters $\mathbf{w}$ such that $\mathbb{P}[y|s_*, s_{\mathcal{I}}, s_{\mathcal{J}}] = \mathbb{P}[o|s_*, s_{\mathcal{I}}, s_{\mathcal{J}}, \mathbf{w}]$. Let also r_y denote the log-ratio of class imbalance $\log \frac{\mathbb{P}[y=1]}{\mathbb{P}[y=0]}$. For a fixed $y \in \mathcal{Y}$ and $\mathcal{D} \in \{\mathcal{D}_b, \mathcal{D}_a\}$, one has*

$$\begin{aligned}&\mathbb{E}_{s_*, s_{\mathcal{I}}, s_{\mathcal{J}}}[\Delta_{\mathbf{w}}(y, s_*, s_{\mathcal{I}}, s_{\mathcal{J}})|y] \\ &\geq \mu_{d_{*,\mathcal{D}}} + y\mu_{d_{\mathcal{I},\mathcal{D}}} + (1 - y)\mu_{d_{\mathcal{J},\mathcal{D}}} + (2y - 1)r_y := \mu_{y,\mathcal{D}},\end{aligned}$$

where

$$\mu_{d_{*,\mathcal{D}}} = \alpha_{*,\mathcal{D}} \log \frac{{}_\wedge\alpha_*}{1 - {}_\wedge\alpha_*} + (1 - \alpha_{*,\mathcal{D}}) \log \frac{1 - {}_\vee\alpha_*}{{}_\vee\alpha_*},$$

$$\begin{aligned}\mu_{d_{\mathcal{I},\mathcal{D}}} = &\sum_{i\in\mathcal{I}} \alpha_{i,\mathcal{D}} \log \frac{{}_\wedge\alpha_i}{{}_\vee\epsilon_i} + (1 - \alpha_{i,\mathcal{D}}) \log \frac{1 - {}_\vee\alpha_i}{1 - {}_\wedge\epsilon_i} \\ &- \sum_{j\in\mathcal{J}} \epsilon_{j,\mathcal{D}} \log \frac{{}_\vee\alpha_j}{{}_\wedge\epsilon_j} - (1 - \epsilon_{j,\mathcal{D}}) \log \frac{1 - {}_\wedge\alpha_j}{1 - {}_\vee\epsilon_j},\end{aligned}$$

and

$$\mu_{d_{\mathcal{J},\mathcal{D}}} = \sum_{j\in\mathcal{J}} \alpha_{j,\mathcal{D}} \log \frac{{}_{\wedge}\alpha_j}{{}_{\vee}\epsilon_j} + (1-\alpha_{j,\mathcal{D}}) \log \frac{1-{}_{\vee}\alpha_j}{1-{}_{\wedge}\epsilon_j} - \sum_{i\in\mathcal{I}} \epsilon_{i,\mathcal{D}} \log \frac{{}_{\vee}\alpha_i}{{}_{\wedge}\epsilon_i} - (1-\epsilon_{i,\mathcal{D}}) \log \frac{1-{}_{\wedge}\alpha_i}{1-{}_{\vee}\epsilon_i}.$$

Proof Sketch. This lemma can be derived by first decomposing $\Delta_{\mathbf{w}}$ into parts that are relevant for $s_*, s_{\mathcal{I}}, s_{\mathcal{J}}$, namely there exist $d_{*,\mathcal{D}}, d_{\mathcal{I},\mathcal{D}}, d_{\mathcal{J},\mathcal{D}}$ such that

$$\Delta_{\mathbf{w}}(y, s_*, s_{\mathcal{I}}, s_{\mathcal{J}}) = d_{*,\mathcal{D}} + y d_{\mathcal{I},\mathcal{D}} + (1-y) d_{\mathcal{J},\mathcal{D}} + (2y-1) r_y.$$

Then we prove that $\mu_{*,\mathcal{D}} \leq \mathbb{E}[d_{*,\mathcal{D}}]$ for the main model, and $\mu_{d_{\mathcal{K},\mathcal{D}}} \leq \mathbb{E}[d_{\mathcal{K},\mathcal{D}}]$ for $\mathcal{K} \in \{\mathcal{I}, \mathcal{J}\}$, the permissive and preventative models. The full proof is presented in Appendix A.3.

Discussion The above lemma illustrates the relationship between the models and $\mathcal{A}^{\text{KEMLP}}$. Intuitively, the larger $\mu_{y,\mathcal{D}}$ is, the further away the expectation of $\Delta_{\mathbf{w}}(y, s_*, s_{\mathcal{I}}, s_{\mathcal{J}})$ is from 0, and thus, the larger the probability that $\Delta_{\mathbf{w}}(y, s_*, s_{\mathcal{I}}, s_{\mathcal{J}}) > 0$. We see that $\mu_{y,\mathcal{D}}$ consists of three terms: $\mu_{d_{*,\mathcal{D}}}$, $\mu_{d_{\mathcal{I},\mathcal{D}}}$, $\mu_{d_{\mathcal{J},\mathcal{D}}}$, measuring the contributions from the main model for all y, permissive models and preventative models for $y=1$ and $y=0$, respectively. More specifically, $\mu_{y,\mathcal{D}}$ is increasing in terms of a weighted sum of α_i, and decreasing in terms of a weighted sum of ϵ_j. When $s_i \implies y$ holds (permissive models), it implies a large α_i for $y=1$, whereas when $y \implies s_j$ holds (preventative model) it implies a small ϵ_j for $y=1$. Thus, this lemma connects the property of auxiliary models to the weighted robust accuracy of KEMLP.

4.2. Convergence of $\mathcal{A}^{\text{KEMLP}}$

Now we are ready to present our convergence result.

Theorem 1 (Convergence of $\mathcal{A}^{\text{KEMLP}}$)**.** *For $y \in \mathcal{Y}$ and $\mathcal{D} \in \{\mathcal{D}_b, \mathcal{D}_a\}$, let $\mu_{y,\mathcal{D}}$ be defined as in Lemma 1. Suppose that the modeling assumption holds, and suppose that $\mu_{d_{\mathcal{K},\mathcal{D}}} > 0$, for all $\mathcal{K} \in \{\mathcal{I}, \mathcal{J}\}$ and $\mathcal{D} \in \{\mathcal{D}_b, \mathcal{D}_a\}$. Then*

$$\mathcal{A}^{KEMLP} \geq 1 - \mathbb{E}_{\mu_{y,\mathcal{D}}}[\exp\left(-2\mu_{y,\mathcal{D}}^2/v^2\right)], \tag{3}$$

where v^2 is the variance upper bound to $\mathbb{P}[o=y|y,\mathbf{w}]$ with

$$v^2 = 4\Big(\log \frac{{}_{\vee}\alpha_*}{1-{}_{\wedge}\alpha_*}\Big)^2 + \sum_{k\in\mathcal{I}\cup\mathcal{J}} \Big(\log \frac{{}_{\vee}\alpha_k(1-{}_{\wedge}\epsilon_k)}{{}_{\wedge}\epsilon_k(1-{}_{\vee}\alpha_k)}\Big)^2.$$

Proof Sketch. We begin by subtracting the term $\mu_{y,\mathcal{D}}$ from $\mathbb{P}_{\mathcal{D}}(o=y|y,\mathbf{w})$, and then decomposing the result into individual summands, where each summand is induced by a single model. We then treat each summand as a bounded increment whose sum is a submartingale. Followed by an application of generalized bounded difference inequality (van de Geer, 2002), we arrive at the proof, whose full details can be found in Appendix A.4.

Discussion In the following, we attempt to understand the scaling of the weighted robust accuracy of KEMLP in terms of models' characteristics.

Impact of truth rates and false rates: We note that $\mu_{d_{\mathcal{K},\mathcal{D}}}$ for $\mathcal{K} \in \{\mathcal{I}, \mathcal{J}\}$, which is an additive component of $\mu_{y,\mathcal{D}}$, poses importance to understand the factors contributing to the performance of KEMLP. Generally, larger $\mu_{d_{\mathcal{K},\mathcal{D}}}$ (hence $\mu_{y,\mathcal{D}}$) would increase the right tail probability of $\Delta_{\mathbf{w}}(y, s_*, s_{\mathcal{I}}, s_{\mathcal{J}})$ leading to a larger weighted accuracy for KEMLP. Although exceptions exist in cases where the variance increases disproportionally, here in our discussion we first focus on parameters that increase $\mu_{d_{\mathcal{K},\mathcal{D}}}$. Towards that, we simplify our exposition and let each auxiliary model have the same truth and false rate over both benign and adversarial examples, and within each type, where the exact parameters are given by $\alpha_k := \alpha_{k,\mathcal{D}} = {}_{\wedge}\alpha_{k,\mathcal{D}} = {}_{\vee}\alpha_{k,\mathcal{D}}$ and $\epsilon_k := \epsilon_{k,\mathcal{D}} = {}_{\wedge}\epsilon_{k,\mathcal{D}} = {}_{\vee}\epsilon_{k,\mathcal{D}}$, for $k \in \mathcal{I}\cup\mathcal{J}$. In this simplified setting where the expected performance improvement by the auxiliary models is given by $\mu_{d_{\mathcal{K},\mathcal{D}}}$ for $\mathcal{K} \in \{\mathcal{I}, \mathcal{J}\}$ and fixed with respect to $\mathcal{D}$, one can observe through partial derivatives that $\mu_{d_{\mathcal{K},\mathcal{D}}}$ is increasing over α_k and decreasing over ϵ_k. This explains why the two types of knowledge rules would help: high-quality permissive models would have high truth rate and low false rate (α_i and ϵ_i), as well as the preventative models (α_j and ϵ_j), yet with different coverages for $y \in \mathcal{Y}$.

Auxiliary models in KEMLP - the more the merrier? Next, we investigate the effect of the number of auxiliary models. To simplify, let $|\mathcal{I}| = |\mathcal{J}|$, and let $\hat{\mu}_{y,\mathcal{D}}$ be a random variable with $\hat{\mu}_{y,\mathcal{D}} = \mu_{y,\mathcal{D}}/(n+1)$, and $\hat{v}^2 = v^2/(n+1)$. The exponent thus becomes $-\mu_{y,\mathcal{D}}^2/v^2 = -(n+1)\hat{\mu}_{y,\mathcal{D}}^2/\hat{v}^2$. One can show that $\hat{\mu}_{y,\mathcal{D}}^2/\hat{v}^2 \geq c$ for some positive constant c, implying that $\mathcal{A}^{\text{KEMLP}} \geq 1 - \exp(-2(n+1)c)$. That is, increasing the number of models generally improves the weighted robust accuracy of KEMLP. To demonstrate this, we now focus on understanding the scaling of weighted robust accuracy on a simplified setting. We assume that the auxiliary models are *homogeneous* for each type: permissive or preventative. For example, α_k is fixed with respect to $k \in \mathcal{I}\cup\mathcal{J}$, hence we drop the subscripts, i.e., $\alpha_{k,\mathcal{D}} = \alpha$ and $\epsilon_{k,\mathcal{D}} = \epsilon$. We assume that the same number of auxiliary models are used, namely $|\mathcal{I}| = |\mathcal{J}| = n$, and that the classes are balanced with $\mathbb{P}_{\mathcal{D}}(y=1) = \mathbb{P}_{\mathcal{D}}(y=0)$, for all $\mathcal{D} \in \{\mathcal{D}_b, \mathcal{D}_a\}$. Finally, we let $\alpha_{*,\mathcal{D}_b} = 1$ and $\alpha_{*,\mathcal{D}_a} = 0$, and $\alpha - \epsilon > 0$. Then, the following holds.

Corollary 1 (Homogenous models)**.** *The weighted robust accuracy of KEMLP in the homogeneous setting satisfies*

$$\mathcal{A}^{KEMLP} \geq 1 - \exp\big(-2n(\alpha-\epsilon)^2\big).$$

In particular, one has $\lim_{n\to\infty} \mathcal{A}^{KEMLP} = 1$.

For this particular case, the predicted class for the target

variable y is based upon an (unweighted) majority voting decision. The above result suggests that for a setting where the auxiliary models are homogeneous with different coverage, the performance of KEMLP to predict the output variable y robustly is determined by: (a) the difference between the probability of predicting the output variable correctly and that of making an erroneous prediction, that is, $\alpha - \epsilon$, and (b) the number of auxiliary models. Consequently, $\mathcal{A}^{\text{KEMLP}}$ converges to 1 exponentially fast in the number of auxiliary models as long as $\alpha - \epsilon > 0$, which is naturally satisfied by the principle KEMLP employs while constructing the logical relations between the output variable and different knowledge.

4.3. Comparing $\mathcal{A}^{\text{KEMLP}}$ and $\mathcal{A}^{\text{main}}$

Theorem 1 guarantees that the addition of models allows the weighted robust accuracy of KEMLP to converge to 1 exponentially fast. We now introduce a sufficient condition under which $\mathcal{A}^{\text{KEMLP}}$ is strictly better than $\mathcal{A}^{\text{main}}$.

Theorem 2 (Sufficient condition for $\mathcal{A}^{\text{KEMLP}} > \mathcal{A}^{\text{main}}$)**.** *Let the number of permissive and preventative models be the same and denoted by n such that $n := |\mathcal{I}| = |\mathcal{J}|$. Note that the weighted accuracy of the main model in terms of its truth rate is simply $\alpha_* := \sum_{\mathcal{D}\in\{\mathcal{D}_b,\mathcal{D}_a\}} \pi_{\mathcal{D}}\alpha_{*,\mathcal{D}}$. Moreover, let $\mathcal{K}, \mathcal{K}' \in \{\mathcal{I}, \mathcal{J}\}$ with $\mathcal{K} \neq \mathcal{K}'$ and for any $\mathcal{D} \in \{\mathcal{D}_b, \mathcal{D}_a\}$, let*

$$\gamma_{\mathcal{D}} := \frac{1}{n+1} \min_{\mathcal{K}} \left\{ \alpha_{*,\mathcal{D}} - 1/2 + \sum_{k\in\mathcal{K}} \alpha_{k,\mathcal{D}} - \sum_{k'\in\mathcal{K}'} \epsilon_{k',\mathcal{D}} \right\}.$$

If $\gamma_{\mathcal{D}} > \sqrt{\frac{4}{n+1} \log \frac{1}{1-\alpha_*}}$ for all $\mathcal{D} \in \{\mathcal{D}_b, \mathcal{D}_a\}$, then $\mathcal{A}^{\text{KEMLP}} > \mathcal{A}^{\text{main}}$.

Proof Sketch. We first approximate $\Delta_{\mathbf{w}}(y, s_*, s_{\mathcal{I}}, s_{\mathcal{J}})$ with a Poisson Binomial random variable and apply the relevant Chernoff bound. Imposing a strict bound between the Chernoff result and the true and false rates of main model concludes the proof. We note that this bound is slightly simplified, and our full proof in the Appendix A.5 is tighter.

Discussion We start by noting that $\gamma_{\mathcal{D}}$ is a combined truth rate of all models normalized over the number of models. That is, for a fixed distribution $\mathcal{D}$, $\alpha_{*,\mathcal{D}} - 1/2$ indicates the truth rate of main task model over a random classifier and $\sum_{k\in\mathcal{K}} \alpha_{k,\mathcal{D}} - \sum_{k'\in\mathcal{K}'} \epsilon_{k',\mathcal{D}}$ refers to the improvement by the auxiliary models on top of the main task model. More specifically, in cases where the true class of output variable is positive with $y = 1$, $\sum_{i\in\mathcal{I}} \alpha_{i,\mathcal{D}} - \sum_{j\in\mathcal{J}} \epsilon_{j,\mathcal{D}}$ account for the total (and unnormalized) success of permissive models in identifying $y = 1$ interfered by the failure of preventative model in identifying $y = 1$ (resp. For $y = 0$, $\mathcal{K} = \mathcal{J}$). Hence, $\gamma_{\mathcal{D}}$ is the "worst-case" combined truth rate of all models, where the worst-case refers to minimization over all possible labels of target variable.

Theorem 2 therefore forms a relationship between the improvement of KEMLP over the main task model and the combined truth rate of models, and theoretically justifies our intuition – larger truth rates and lower false rates of individual auxiliary models result in larger combined truth rate $\gamma_{\mathcal{D}}$, hence making the sufficient condition more likely to hold. Additionally, employing a large number of auxiliary models is found to be beneficial for better KEMLP performance, as we conclude in Corollary 1 as well. Our finding here also confirms that in the extreme scenarios where the main task model has a perfect clean and robust truth rate ($\alpha_* = 1$), it is *not* possible to improve upon the main task model. Conversely, when $\alpha_* = 0$, any improvement by KEMLP would result in absolute improvement over the main model.

5. Experimental Evaluation

In this section, we evaluate KEMLP based on the traffic sign recognition task against different adversarial attacks and corruptions, including the physical attacks (Eykholt et al., 2018), $\mathcal{L}_\infty$ bounded attacks, unforeseen attacks (Kang et al., 2019), and common corruptions (Hendrycks and Dietterich, 2019). We show that under both whitebox and blackbox settings against a *diverse* set of attacks, 1) KEMLP achieves significantly higher robustness than baselines, 2) KEMLP maintains similar clean accuracy with a strong main task model whose clean accuracy is originally high (e.g., vanillar CNN), 3) KEMLP even achieves higher clean accuracy than a relatively weak main task model whose clean accuracy is originally low as a tradeoff for its robustness (e.g., adversarially trained models).

5.1. Experimental Setup

Dataset Following existing work (Eykholt et al., 2018; Wu et al., 2019) that evaluate ML robustness on traffic sign data, we adopt LISA (Mogelmose et al., 2012) and GTSRB (Stallkamp et al., 2012) for training and evaluation. All data are processed by standard crop-and-resize to 32×32 as described in (Sermanet and LeCun, 2011). In this paper, we conduct the evaluation on two dataset settings: *1) Setting-A:* a subset of GTSRB, which contains 12 types of German traffic signs. In total, there are 14880 samples in the training set, 972 samples in the validation set, and 3888 samples in the test set; *2) Setting-B:* a modified version of Setting-A, where the German stop signs are replaced with the U.S. stop signs from LISA, following (Eykholt et al., 2018).

Models We adopt the GTSRB-CNN architecture (Eykholt et al., 2018) as the main task model. KEMLP is constructed based on the main task model together with a set of auxiliary task models (e.g., color, shape, and content detectors). To

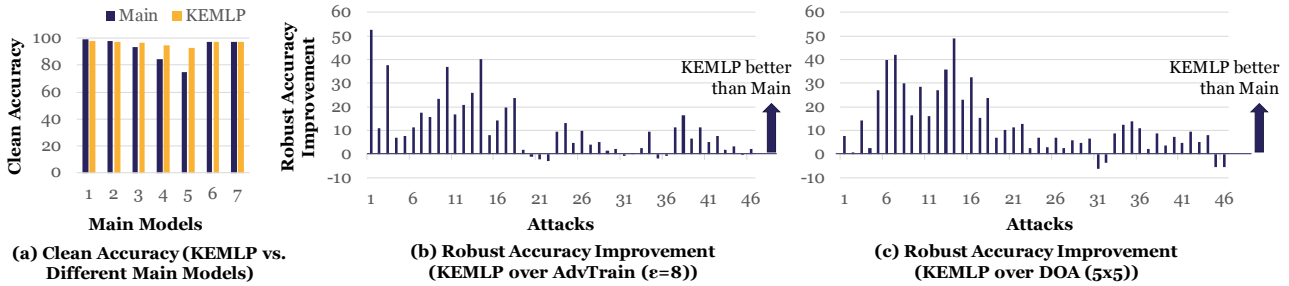

Figure 2. (a) Clean accuracy and (b) (c) robust accuracy improvement of KEMLP ($\beta = 0.5$) over baselines against different attacks under both whitebox and blackbox settings. The represented attack list and results of other baselines are in Appendix B.2.

Table 1. Model performance (%) under physical attacks ($\beta = 0.4$). Performance gain and loss of KEMLP over baselines are highlighted.

	Main			KEMLP		
	Clean Acc	Robust Acc	W-Robust Acc	Clean Acc	Robust Acc	W-Robust Acc
GTSRB-CNN	100	5	52.5	100(±0)	87.5(+82.5)	93.75(+41.25)
AdvTrain ($\epsilon = 4$)	100	12.5	56.25	100(±0)	90(+77.5)	95(+38.75)
AdvTrain ($\epsilon = 8$)	97.5	37.5	67.5	100(+2.5)	90(+52.5)	95(+27.5)
AdvTrain ($\epsilon = 16$)	87.5	50	68.75	100(+12.5)	90(+40)	95(+26.25)
AdvTrain ($\epsilon = 32$)	62.5	32.5	47.5	100(+37.5)	90(+57.5)	95(+47.5)
DOA (5x5)	95	90	92.5	100(+5)	100(+10)	100(+7.5)
DOA (7x7)	57.5	32.5	45	100(+42.5)	100(+67.5)	100(+55)

train the weights of factors in KEMLP, we use β to denote the prior belief on balance between benign and adversarial distributions. More details on implementation are provided in Appendix B.3.

Baselines To demonstrate the superiority of KEMLP, we compare it with two state-of-the-art baselines: **adversarial training** (Madry et al., 2017) and **DOA** (Wu et al., 2019), which are strong defenses against $\mathcal{L}_p$ bounded attacks and physically attacks respectively. Detailed setup for baselines is given in Appendix B.1.

Evaluated Attacks and Corruptions We consider four types of attacks for thorough evaluation: *1) physical attacks* on stop signs (Eykholt et al., 2018); *2) $\mathcal{L}_\infty$ bounded attacks* (Madry et al., 2017) with $\epsilon \in \{4, 8, 16, 32\}$; *3) Unforeseen attacks*, which produce a diverse set of unforeseen test distributions (e.g. Elastic, JPEG, Fog) distinct from $\mathcal{L}_p$ bounded perturbation (Kang et al., 2019); *4) common corruptions* (Hendrycks and Dietterich, 2019). We present examples of these adversarial instances in Appendix B.4. For each attack, we consider both the *whitebox attack* against the main task model and *blackbox attack* by distilling either the main task model or the whole KEMLP pipeline. More details can be found in Appendix B.2.

5.2. Evaluation Results

Here we compare the clean accuracy, robust accuracy, and weighted robustness (W-Robust Accuracy) for baselines and KEMLP under different attacks and settings.

Clean accuracy of KEMLP First, we present the clean accuracy of KEMLP and baselines in Figure 2 (a) and Tables 1–4. As demonstrated, the clean accuracy of KEMLP is generally high (over 90%), by either maintaining the high clean accuracy of strong main task models (e.g., vanilla DNN) or improving upon the weak main task models with relatively low clean accuracy (e.g., adversarially trained models). It is clear that KEMLP can relax the tradeoff between benign and robust accuracy and maintain the high performance for both via knowledge integration.

Robustness against diverse attacks We then present the robustness of KEMLP based on different main task models against the physical attacks, which is very challenging to defend currently (Table 1), ℓ_p bounded attacks (Table 2), unseen attacks (Table 3), and common corruptions (Table 4) under whitebox attack setting. The corresponding results for blackbox setting can be found in Appendix B.5. From the tables, we observe that KEMLP achieves significant *robustness gain* over baselines. Note that although adversarial training improves the robustness against $\mathcal{L}_\infty$ attacks and DOA helps to defend against physical attacks, they are not robust to other types of attacks or corruptions. In contrast, KEMLP presents general robustness against a range of attacks and corruptions without further adaptation.

Performance stability of KEMLP We conduct additional ablation studies on β, representing the prior belief on the benign and adversarial distribution balance. We set $\beta = 0.5$ for KEMLP indicating a balanced random guess

Table 2. Accuracy (%) under whitebox $\mathcal{L}_\infty$ attacks ($\beta = 0.8$)

Models		$\epsilon = 0$	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 16$	$\epsilon = 32$
GTSRB-CNN	Main	**99.38**	67.31	43.13	13.50	3.63
	KEMLP	98.28(−1.10)	85.39(+18.08)	71.76(+28.63)	48.89(+35.39)	26.13(+22.50)
AdvTrain ($\epsilon = 4$)	Main	97.94	87.94	68.85	38.66	8.77
	KEMLP	97.89(−0.05)	**92.80**(+4.86)	79.58(+10.73)	57.48(+18.82)	28.58(+19.81)
AdvTrain ($\epsilon = 8$)	Main	93.72	84.21	71.76	43.16	13.01
	KEMLP	96.79(+3.07)	92.08(+7.87)	81.58(+9.82)	59.18(+16.02)	30.61(+17.60)
AdvTrain ($\epsilon = 16$)	Main	84.54	78.58	71.89	55.99	19.55
	KEMLP	94.68(+10.14)	91.64(+13.06)	**85.55**(+13.66)	67.98(+11.99)	32.61(+13.06)
AdvTrain ($\epsilon = 32$)	Main	74.74	70.24	65.61	56.22	29.04
	KEMLP	91.46(+16.72)	88.58(+18.34)	83.23(+17.62)	**72.02**(+15.80)	**41.90**(+12.86)
DOA (5x5)	Main	97.43	57.46	28.76	5.81	0.85
	KEMLP	97.45(+0.02)	83.85(+26.39)	67.98(+39.22)	45.27(+39.46)	24.28(+23.43)
DOA (7x7)	Main	97.27	38.50	9.75	2.83	0.67
	KEMLP	97.22(−0.05)	80.89(+42.39)	63.40(+53.65)	49.20(+46.37)	31.04(+30.37)

Table 3. Accuracy (%) under whitebox unforeseen attacks ($\beta = 0.8$)

		Clean	Fog-256	Fog-512	Snow-0.25	Snow-0.75	Jpeg-0.125	Jpeg-0.25	Gabor-20	Gabor-40	Elastic-1.5	Elastic-2.0
GTSRB-CNN	Main	**99.38**	59.65	34.18	56.58	24.54	55.74	27.01	57.25	32.41	44.78	24.31
	KEMLP	98.28(−1.10)	76.95(+17.30)	62.83(+28.65)	78.94(+22.36)	53.22(+28.68)	79.63(+23.89)	63.40(+36.39)	80.17(+22.92)	65.20(+32.79)	69.34(+24.56)	52.37(+28.06)
AdvTrain ($\epsilon = 4$)	Main	97.94	55.53	29.50	66.31	32.61	56.58	28.11	73.30	46.76	57.25	30.09
	KEMLP	97.89(−0.05)	76.08(+20.55)	61.96(+32.46)	80.45(+14.14)	57.84(+25.23)	84.23(+27.65)	68.57(+40.46)	81.48(+8.18)	65.77(+19.01)	71.19(+13.94)	50.33(+20.24)
AdvTrain ($\epsilon = 8$)	Main	93.72	50.03	23.56	63.71	34.93	57.56	26.16	76.72	53.76	48.25	24.46
	KEMLP	96.79(+3.07)	76.59(+26.56)	63.97(+40.41)	81.40(+17.69)	57.07(+22.14)	85.11(+27.55)	68.70(+42.54)	85.29(+8.57)	68.90(+15.14)	68.78(+20.53)	49.31(+24.85)
AdvTrain ($\epsilon = 16$)	Main	84.54	47.92	19.75	66.46	37.60	66.56	34.23	78.01	64.33	55.48	32.28
	KEMLP	94.68(+10.14)	77.13(+29.21)	64.38(+44.63)	**81.64**(+15.18)	58.20(+20.60)	**86.99**(+20.43)	70.40(+36.17)	**87.42**(+9.41)	72.61(+8.28)	67.31(+11.83)	50.28(+18.00)
AdvTrain ($\epsilon = 32$)	Main	74.74	48.71	22.84	61.78	38.91	63.58	43.49	70.37	65.20	54.58	39.45
	KEMLP	91.46(+16.72)	**79.22**(+30.51)	**66.33**(+43.49)	81.20(+19.42)	**64.53**(+25.62)	86.70(+23.12)	**73.38**(+29.89)	87.04(+16.67)	**74.92**(+9.72)	66.38(+11.80)	54.76(+15.31)
DOA (5x5)	Main	97.43	58.00	32.69	61.19	28.34	41.13	11.29	55.43	29.55	58.02	32.74
	KEMLP	97.45(+0.02)	76.85(+18.85)	63.07(+30.38)	78.78(+17.59)	56.76(+28.42)	78.60(+37.47)	61.78(+50.49)	80.25(+24.82)	63.89(+34.34)	**72.69**(+14.67)	**57.51**(+24.77)
DOA (7x7)	Main	97.27	59.88	38.01	62.47	30.17	23.46	3.65	54.58	27.29	56.33	30.97
	KEMLP	97.22(−0.05)	78.09(+18.21)	62.76(+24.75)	79.68(+17.21)	58.26(+28.09)	74.25(+50.79)	61.39(+57.74)	79.06(+24.48)	62.29(+35.00)	71.27(+14.94)	55.09(+24.12)

Table 4. Accuracy (%) under common corruptions ($\beta = 0.2$)

		Clean	Fog	Contrast	Brightness
GTSRB-CNN	Main	**99.38**	76.23	57.61	85.52
	KEMLP	98.28(−1.10)	**78.14**(+1.91)	72.43(+14.82)	**89.58**(+4.06)
AdvTrain ($\epsilon = 4$)	Main	97.94	63.81	42.31	78.47
	KEMLP	97.89(−0.05)	70.29(+6.48)	67.46(+25.16)	86.70(+8.23)
AdvTrain ($\epsilon = 8$)	Main	93.72	59.05	31.97	78.47
	KEMLP	96.79(+3.07)	67.41(+8.36)	66.69(+34.72)	85.91(+7.44)
AdvTrain ($\epsilon = 16$)	Main	84.54	56.58	34.31	78.01
	KEMLP	94.68(+10.14)	66.80(+10.22)	68.39(+34.08)	86.14(+8.13)
AdvTrain ($\epsilon = 32$)	Main	74.74	50.87	30.45	71.30
	KEMLP	91.46(+16.72)	64.94(+14.07)	68.31(+37.86)	83.20(+11.90)
DOA (5x5)	Main	97.43	73.95	62.24	83.92
	KEMLP	97.45(+0.02)	76.08(+2.13)	**74.38**(+12.14)	87.60(+3.68)
DOA (7x7)	Main	97.27	73.41	57.54	83.56
	KEMLP	97.22(−0.05)	76.00(+2.59)	72.40(+14.86)	87.78(+4.22)

for the distribution tradeoff. We show the clean accuracy and robustness of KEMLP and baselines under diverse 46 attacks in Figure 2. We can see that KEMLP consistently and significantly outperforms the baselines, which indicates the performance stability of KEMLP regarding different distribution ratio β. More results can be found in Appendix B.5, with additional discussions in Appendix B.6.

6. Discussions and Future Work

In this paper, we propose KEMLP, which integrates *domain knowledge* with a set of weak auxiliary models to enhance the ML robustness against a diverse set of adversarial attacks and corruptions. While our framework can be extended to other applications, for any knowledge system, one naturally needs domain experts to design the knowledge rules specific to that application. Here we aim to introduce this framework as a prototype, provide a rigorous analysis of it, and demonstrate the benefit of such construction on an application. Nevertheless, there is probably no universal strategy on how to aggregate knowledge for any arbitrary application, and instead, application-specific constructions are needed. We do believe that, once the principled framework of knowledge fusion is ready, application-specific developments of knowledge rules will naturally follow, similar to what happened previously for knowledge-enriched joint inference.

Acknowledgements

CZ and the DS3Lab gratefully acknowledge the support from the Swiss National Science Foundation (Project Number 200021_184628), Innosuisse/SNF BRIDGE Discovery (Project Number 40B2-0_187132), European Union Horizon 2020 Research and Innovation Programme (DAPHNE, 957407), Botnar Research Centre for Child Health, Swiss Data Science Center, Alibaba, Cisco, eBay, Google Focused Research Awards, Oracle Labs, Swisscom, Zurich Insurance, Chinese Scholarship Council, and the Department of Computer Science at ETH Zurich. BL and the SLLab would like to acknowledge the support from NSF grant No.1910100, NSF CNS 20-46726 CAR, and Amazon Research Award.

References

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Interna-*

tional Conference on Machine Learning, pages 274–283. PMLR, 2018.

Kazuoki Azuma. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal, Second Series*, 19(3):357–367, 1967.

Anand Bhattad, Min Jin Chong, Kaizhao Liang, Bo Li, and David A Forsyth. Unrestricted adversarial examples via semantic manipulation. 2020. URL https://openreview.net/forum?id=Sye_OgHFwH.

Marenglen Biba, Stefano Ferilli, and Floriana Esposito. Protein fold recognition using markov logic networks. In *Mathematical Approaches to Polymer Sequence Analysis and Related Problems*, pages 69–85. Springer, 2011.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

Qi-Zhi Cai Cai, Chang Liu, and Dawn Song. Curriculum adversarial training. pages 3740–3747, 7 2018. doi: 10.24963/ijcai.2018/520. URL https://doi.org/10.24963/ijcai.2018/520.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14, 2017a.

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

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems*, pages 11190–11201, 2019.

Deepayan Chakrabarti, Stanislav Funiak, Jonathan Chang, and Sofus Macskassy. Joint inference of multiple label types in large networks. In *International Conference on Machine Learning*, pages 874–882. PMLR, 2014.

Liwei Chen, Yansong Feng, Jinghui Mo, Songfang Huang, and Dongyan Zhao. Joint inference for knowledge base population. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1912–1923, 2014.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019.

Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. Large-scale object classification using label relation graphs. In *European conference on computer vision*, pages 48–64. Springer, 2014.

Krishnamurthy Dj Dvijotham, Jamie Hayes, Borja Balle, Zico Kolter, Chongli Qin, Andras Gyorgy, Kai Xiao, Sven Gowal, and Pushmeet Kohli. A framework for robustness certification of smoothed classifiers using f-divergences. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJlKrkSFPH.

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*, pages 1625–1634, 2018.

Ian J Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *arXiv preprint arXiv:1312.6082*, 2013.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136, 2019.

Adel Javanmard, Mahdi Soltanolkotabi, and Hamed Hassani. Precise tradeoffs in adversarial training for linear regression. In *Conference on Learning Theory*, pages 2034–2078. PMLR, 2020.

Daniel Kang, Yi Sun, Dan Hendrycks, Tom Brown, and Jacob Steinhardt. Testing robustness against unforeseen adversaries. *arXiv preprint arXiv:1908.08016*, 2019.

Sanjay Kariyappa and Moinuddin K Qureshi. Improving adversarial robustness of ensembles with diversity training. *arXiv preprint arXiv:1901.09981*, 2019.

Nasser Kehtarnavaz, Norman C Griswold, and DS Kang. Stop-sign recognition based on color/shape processing. *Machine Vision and Applications*, 6(4):206–208, 1993.

Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. In *Advances in Neural Information Processing Systems*, pages 9464–9474, 2019.

Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 369–385, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Andrew McCallum. Joint inference for natural language processing. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 1–1, 2009.

Jun Miura, Tsuyoshi Kanda, and Yoshiaki Shirai. An active vision system for real-time traffic sign recognition. In *ITSC2000. 2000 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 00TH8493)*, pages 52–57. IEEE, 2000.

Andreas Mogelmose, Mohan Manubhai Trivedi, and Thomas B Moeslund. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1484–1497, 2012.

Jeet Mohapatra, Ching-Yun Ko, Sijia Liu, Pin-Yu Chen, Luca Daniel, et al. Rethinking randomized smoothing for adversarial robustness. *arXiv preprint arXiv:2003.01249*, 2020.

Leland Gerson Neuberg. Causality: Models, reasoning, and inference, 2003.

Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. *arXiv preprint arXiv:1901.08846*, 2019.

Hoifung Poon and Pedro Domingos. Joint inference in information extraction. In *AAAI*, volume 7, pages 913–918, 2007.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy. *arXiv preprint arXiv:2002.10716*, 2020.

Alexander Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. *Advances in neural information processing systems*, 29:3567, 2016.

Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 11, page 269. NIH Public Access, 2017.

Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

Andrew Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. " grabcut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3): 309–314, 2004.

Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems*, pages 11292–11303, 2019.

Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.

Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. Towards the first adversarially robust neural network model on mnist. In *International Conference on Learning Representations*, 2018.

Pierre Sermanet and Yann LeCun. Traffic sign recognition with multi-scale convolutional networks. In *The 2011 International Joint Conference on Neural Networks*, pages 2809–2813. IEEE, 2011.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *Advances in Neural Information Processing Systems*, pages 3358–3369, 2019.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332, 2012.

Thilo Strauss, Markus Hanselmann, Andrej Junginger, and Holger Ulmer. Ensemble methods as a defense to adversarial perturbations against deep neural networks. *arXiv preprint arXiv:1709.03423*, 2017.

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.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SyxAb30cY7.

Jonathan Uesato, Brendan O'donoghue, Pushmeet Kohli, and Aaron Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*, pages 5025–5034. PMLR, 2018.

Sara A van de Geer. On Hoeffding's inequality for dependent random variables. In *Empirical process techniques for dependent data*, pages 161–169. Springer, 2002.

Martin J Wainwright and Michael Irwin Jordan. *Graphical models, exponential families, and variational inference*. Now Publishers Inc, 2008.

Eric Wong, Frank R Schmidt, and J Zico Kolter. Wasserstein adversarial examples via projected sinkhorn iterations. *arXiv preprint arXiv:1902.07906*, 2019.

Tong Wu, Liang Tong, and Yevgeniy Vorobeychik. Defending against physically realizable attacks on image classification. *arXiv preprint arXiv:1909.09552*, 2019.

Chaowei Xiao, Ruizhi Deng, Bo Li, Fisher Yu, Mingyan Liu, and Dawn Song. Characterizing adversarial examples based on spatial consistency information for semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 217–234, 2018a.

Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *IJCAI*, 2018b.

Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. In *International Conference on Learning Representations*, 2018c. URL https://openreview.net/forum?id=HyydRMZC-.

Huan Xu, Constantine Caramanis, and Shie Mannor. Robustness and regularization of support vector machines. *Journal of machine learning research*, 10(7), 2009.

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

Zhe Xu, Ivan Gavran, Yousef Ahmad, Rupak Majumdar, Daniel Neider, Ufuk Topcu, and Bo Wu. Joint inference of reward machines and policies for reinforcement learning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, pages 590–598, 2020.

Greg Yang, Tony Duan, J Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. In *International Conference on Machine Learning*, pages 10693–10705. PMLR, 2020a.

Huanrui Yang, Jingyang Zhang, Hongliang Dong, Nathan Inkawhich, Andrew Gardner, Andrew Touchet, Wesley Wilkes, Heath Berry, and Hai Li. Dverge: Diversifying vulnerabilities for enhanced robust generation of ensembles. *arXiv preprint arXiv:2009.14720*, 2020b.

Runtian Zhai, Chen Dan, Di He, Huan Zhang, Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh, and Liwei Wang. Macer: Attack-free and scalable robust training via maximizing certified radius. In *International Conference on Learning Representations*, 2019.

Dinghuai Zhang, Mao Ye, Chengyue Gong, Zhanxing Zhu, and Qiang Liu. Black-box certification with randomized smoothing: A functional optimization based framework. *arXiv preprint arXiv:2002.09169*, 2020.