Learn2Weight: Parameter Adaptation against Similar-domain Adversarial Attacks

Siddhartha Datta
University of Oxford
siddhartha.datta@cs.ox.ac.uk

Abstract

Recent work in black-box adversarial attacks for NLP systems has attracted much attention. Prior black-box attacks assume that attackers can observe output labels from target models based on selected inputs. In this work, inspired by adversarial transferability, we propose a new type of black-box NLP adversarial attack that an attacker can choose a similar domain and transfer the adversarial examples to the target domain and cause poor performance in target model. Based on domain adaptation theory, we then propose a defensive strategy, called Learn2Weight, which trains to predict the weight adjustments for a target model in order to defend against an attack of similar-domain adversarial examples. Using Amazon multi-domain sentiment classification datasets, we empirically show that Learn2Weight is effective against the attack compared to standard black-box defense methods such as adversarial training and defensive distillation. This work contributes to the growing literature on machine learning safety.

1 Introduction

As machine learning models are applied to more and more real-world tasks, addressing machine learning safety is becoming an increasingly pressing issue. Deep learning algorithms have been shown to be vulnerable to adversarial examples (Szegedy et al., 2013; Goodfellow et al., 2014; Papernot et al., 2016a). In particular, prior black-box adversarial attacks assume that the adversary is not aware of the target model architecture, parameters or training data, but is capable of querying the target model with supplied inputs and obtaining the output predictions. The phenomenon that adversarial examples generated from one model may also be adversarial to another model is known as adversarial transferability (Szegedy et al., 2013).

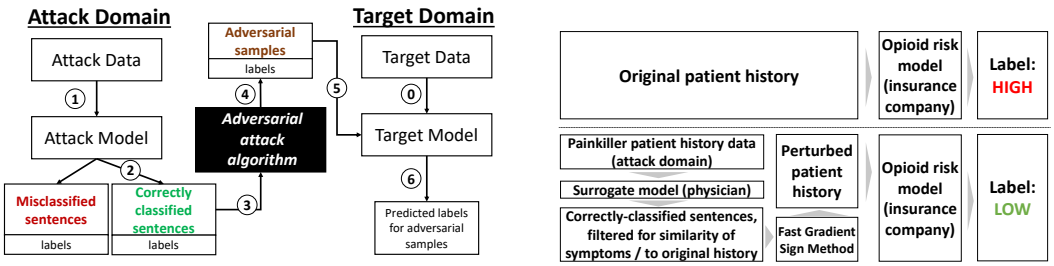
Motivated by adversarial transferability, we conjecture another black-box attack pipeline where the adversary does not even need to have access to the target model nor query labels from crafted inputs. Instead, as long as the adversary knows the task of the target, they can choose a similar domain to build a substitute model, and then attack the target model with adversarial examples that are generated from the attack domain.

The similar-domain adversarial attack may be more practical than prior blackbox attacks as label querying from the target model is not needed. This attack can be illustrated with the following example (Figure 1b) in medical insurance fraud (Finlayson et al., 2019). Insurance companies may use hypothetical opioid risk models to classify the likelihood (high/low) of a patient to abuse the opioids to be prescribed, based on the patient's medical history as text input. Physicians can run the original patient history through the attack pipeline to generate an adversarial patient history, where the original is more likely to be rejected ("High" risk) and the adversarial is more likely to be accepted ("Low" risk). Perturbations in patient history could be, for example, a slight perturbation from "alcohol abuse" to "alcohol dependence", and it may successfully fool the insurance company's model.

Based on domain adaption theory (Ben-David et al., 2010), we conjecture that domain-variant features cause the success of the similar-domain attack. The adversarial examples with domain-variant features are likely to reside in the low-density regions (far away from decision boundary) of the empirical distribution of the target training data which could fool the target model (Zhang et al., 2019b). Literature indicates that worsened generalizability is a tradeoff faced by existing defenses such as adversarial training (Raghunathan et al., 2019) and domain generalization techniques (Wang et al., 2019). In trying to increase robustness against adversarial inputs, a model faces a tradeoff of weakened accuracy towards clean inputs. Given that an adversarial training loss function is composed of a loss against

Proceedings of the 29th International Conference on Computational Linguistics, pages 4832–4843
October 12–17, 2022.

Figure 1: Diagrammatic representation of the attack



(a) Generalized architecture of similarity-based attacks.

(b) Flow of how an adversary physician can leverage similarity attack to fool opioid risk models.

clean inputs and loss against adversarial inputs, improper optimization where the latter is highly-optimized and the former weakly-optimized does not improve general performance in the real-world. To curb this issue, methods have been proposed (Schmidt et al., 2018; Zhang et al., 2019b; Lamb et al., 2019), such as factoring in under-represented data points in training set.

To defend against this similar-domain adversarial attack, we propose a meta learning approach, **Learn2Weight**, so that the target model's decision boundary can adapt to the examples from low-density regions. Experiments confirm the effectiveness of our approach against the similar-domain attack over other baseline defense methods. Moreover, our approach is able to improve robustness accuracy without losing the target model's standard generalization accuracy.

Our contribution can be summarized as follows [†]:

- We are among the first to demonstrate the similar-domain adversarial attack, leveraging domain adaptation to create adversarial perturbations that compromise NLP models. This attack pipeline relaxes the previous black-box attack assumption that the adversary has access to the target model and can query the model with crafted examples.
- We propose a defensive strategy for this attack based on domain adaptation theory and meta learning. Experiments show the effectiveness of our approach over existing defenses against the similar-domain adversarial attack.

[†] † indicates supplementary information can be found in the appendix (Appendix: Datta (2022)).

2 Related Work

Zhang et al. (2020) provides a survey of adversarial attacks in NLP. Existing research proposes different attack methods for generating adversarial text examples (Moosavi-Dezfooli et al., 2016; Ebrahimi et al., 2018; Wallace et al., 2019). The crafted adversarial text examples have been shown to fool state-of-the-art NLP systems, e.g. BERT (Jin et al., 2019). A large body of adversarial attack research focuses on black-box attack where the adversary builds a substitute model by querying the target model with supplied inputs and obtaining the output predictions. The key idea behind such black-box attack is that adversarial examples generated from one model may also be misclassified by another model, which is known as adversarial transferability (Szegedy et al., 2013; Cheng et al., 2019). While prior work examines the transferability between different models trained over the same dataset, or the transferability between the same or different models trained over disjoint subsets of a dataset, our work examines the adversarial transferability between different domains, which we call a similar-domain adversarial attack.

3 Similar-domain Adversarial Attack

3.1 Adversarial attack background

Adversarial attacks modify inputs to cause errors in machine learning inference (Szegedy et al., 2013). We use the basic gradient-based attack method *Fast Gradient Sign Method (FGSM)* (Goodfellow et al., 2014), with perturbation rate $\varepsilon = 0.4$. Other NLP adversarial generation algorithms could also be used, such as *Rand-FGSM* (Tramèr et al., 2017), *Basic Iterative Method* (Kurakin et al., 2016c,a; Xie et al., 2018), DeepFool (Moosavi-Dezfooli et al., 2016), HotFlip (Ebrahimi et al., 2018), uni-

Attack domain: baby, Target domain: books		
Original sentence (Actual label: Pos)	I purchased this toy for my son when he was 4 months old. At first, he seemed a little intimidated by the toys.	Pos (0.712)
Adversarial sentence	*I obtained this toys for my children when he was 4 weeks senior. At first, he hoped a modest harassed by the toy.*	Neg (0.364)
Original sentence (Actual label: Pos)	It felt like a big commitment for me to have to run the program 2 times a day, and near the end of my pregnancy I was annoyed with having anything strapped across my belly.	Pos (0.825)
Adversarial sentence	*It felt like a big committed for me to have to run the program 2 length a day, and near the end of my pregnancy I was annoyed with takes anything strapped across my belly.*	Neg (0.420)
Attack domain: dvd, Target domain: baby		
Original sentence (Actual label: Pos)	Fast times at ridgemont high is a clever, insightful, and wicked film! It is not just another teen movie.	Pos (0.614)
Adversarial sentence	*Sooner days at ridgemont high is a sane, thoughtful, and wicked flick! It is not just another adolescent flick.*	Neg (0.335)
Original sentence (Actual label: Pos)	This dvd gives a very good 60 minute workout. As others have pointed out the cardio is very dancy. The first time I did it, I felt a bit awkward with the steps.	Pos (0.647)
Adversarial sentence	*This dvd gives a awfully okay 60 minute exercise. As others have pointed out the cardio is very dancy. The first time I did it, I perceived a bit awkward with the steps.*	Neg (0.258)

Table 1: Comparison of attack domain sentences correctly classified when unperturbed by respective attack domain models and target domain models, then misclassified after perturbation by target models trained on **books** and **baby** domain. The perturbations are in blue, and prediction confidence in brackets.

versal adversarial trigger (Wallace et al., 2019), and TextFooler (Jin et al., 2019). To perform gradient-based perturbations upon discrete space data, we follow Papernot et al. (2016b) to generate adversarial text. Our proposed similar-domain adversarial attack is in-variant to adversarial algorithm, meaning that the adversarial algorithm used would not affect the attack performance.

Definition 1. NLP Adversarial Generation. We denote $\texttt{Adv}(\theta; \mathbf{x}; \varepsilon)$ as an NLP adversarial generation method. The goal of $\texttt{Adv}$ is to maximize the misclassification rate on perturbed inputs: $\mathbf{x}^{\text{adv}} = \texttt{Adv}(\theta; \mathbf{x})$ *s.t.* $y \neq f(\theta; \mathbf{x}^{\text{adv}})$.

3.2 Similar-domain Adversarial attack

We present the architecture of similar-domain adversarial attack in Figure 1a. The defender, the target of the attack, constructs a target model (parameters θ_i) trained on domain text data X_i ⓪. An attacker, only having a rough idea about the target's task but lacking direct access to the target data or target model parameters, collects attack data from a similar domain $X_j \sim \mathcal{X}$ and trains an attack model (parameters θ_j) ①. They run the attack model on the test data ② to obtain correctly-classified instances ③. They chooses an adversarial attack algorithm and generate a set of adversarial samples X_j^{adv} ④. They expose X_j^{adv} to the target model, hoping X_j^{adv} misleads the target model to produce an output of their choice ⑤. The attacker's objective is to maximize the misclassification per label and minimize the accuracy w.r.t. perturbed inputs (max Eqt 1), while the defender's objective is to maximize the accuracy w.r.t. perturbed inputs (min Eqt 1). This type of attack works best as an adversarial attack that compromises systems that base decision-making on one-instance.

$$\mathbb{E}_{\mathbf{x}_j, y_j \sim X_j, Y_j}[f(\theta_i; \texttt{Adv}(\theta_j; \mathbf{x}_j)) - y_j] \quad (1)$$

Definition 2. Similar-domain Adversarial Attack. Target model f, trained on target domain data X_i, is a deep neural network model with weights θ_i mapping text instances to labels: $Y_i = f(\theta_i; X_i)$. An adversary chooses source attack domain X_j, builds substitute model $f(\theta_j; X_j)$, and generates a set of adversarial examples X_j^{adv} from X_j using $\texttt{Adv}(\theta_j; X_j)$, such that during an attack $f(\theta_i; X_j^{\text{adv}}) = f(\theta_j; X_j^{\text{adv}})$.

4 Is the Attack Effective?

4.1 Setup

(Datasets) We sample domains from 25 domain datasets, each containing 1,000 positive and 1,000 negative reviews for an Amazon product category, sourced from the Amazon multi-domain sentiment classification benchmark (Blitzer et al., 2007).
(Models) We evaluated our setup on several architectures commonly-used for sentiment classification, including LSTM (Wang et al., 2018), GRU, BERT (Devlin et al., 2019), CNN (Kim, 2014), and Logistic Regression (Maas et al., 2011).

Target Domain	book			magazine			baby		
Original Accuracy	0.880			0.960			0.890		
Intra-attack Accuracy	0.525			0.570			0.632		
Attack Domain	magazine	baby	dvd	baby	dvd	book	dvd	book	magazine
Unperturbed Accuracy	0.745	0.726	0.646	0.673	0.663	0.739	0.652	0.624	0.665
After-attack Accuracy	0.395	0.398	0.421	0.343	0.366	0.381	0.386	0.365	0.401
SharedVocab	0.455	0.381	0.255	0.381	0.345	0.260	0.255	0.270	0.260
Transfer Loss	0.000	0.017	0.071	0.010	0.022	0.079	0.050	0.066	0.069

Table 2: *Domain shift & similarity:* Sorted in descending order of domain similarity, we observe a lower after-attack accuracy when domain similarity increases.

(Domain similarity) refers to the similarity between attacker's chosen domain and defender's domain. **SharedVocab** measures the overlap of unique words, in each of the datasets; a higher degree of overlapping vocabulary implies the two domains are more similar. We also use **Transfer Loss**, a standard metric for domain adaptation (Blitzer et al., 2007; Glorot et al., 2011), to measure domain similarity; lower loss indicates higher similarity. The test error from a target model trained on target domain X_i and evaluated on attack domain X_j returns transfer error $e(X_j, X_i)$. The baseline error $e(X_i, X_i)$ term is the test error obtained from target model trained on target domain (train) data X_i and tested on target domain (evaluation) data X_i. This computes the transfer loss, $tf(X_j, X_i) = e(X_j, X_i) - e(X_i, X_i)$.

(Accuracy) We first report the accuracy of the target models on the target domain test samples before the attack as the *original accuracy*. Then we measure the accuracy of the target models against adversarial samples crafted from the attack domain samples, denoted as the *after-attack accuracy*. *Intra-attack accuracy* denotes the after-attack accuracy where the attack domain is identical to the target domain. By comparing original and after-attack accuracy, we can evaluate the success of the attack. The greater the gap between the original and after-attack accuracy, the more successful the attack. *Unperturbed accuracy* measures the accuracy of the target model against the complete, unperturbed test set of the attack domain, to demonstrate that any drop in classification accuracy is not from domain shift alone but from adversarial transferability.

4.2 Results

The similar-domain adversarial attack results are presented in Table 2. We see a significant gap between original accuracy and after-attack accuracy, indicating that this attack can impose a valid threat to a target NLP system. After the similar-domain adversarial attack, the accuracy drops dramatically by a large margin. Take the book target domain as an example: when the attack domain is magazine, the after-attack accuracy drops to 0.398, and when the attack domain is baby, the accuracy is 0.421. Moreover, we observe a positive correlation between transfer loss and after-attack accuracy, and a negative correlation between shared vocab and after-attack accuracy.

5 Defending Against Similar-domain Adversarial Attack

In order to defend against a similarity based adversarial attack, it is critical to block adversarial transferability. Adversarial training is the most intuitive yet effective defense strategy for adversarial attack (Goodfellow et al., 2014; Madry et al., 2017). However, this may not be effective for two reasons. First, there is no formal guidance for generating similar-domain adversarial examples because the defender has no idea what the attack data domain is. Second, simply feeding the target model with adversarial examples may even hurt the generalization of the target model (Su et al., 2018; Raghunathan et al., 2019; Zhang et al., 2019a), which is also confirmed in our experiments.

5.1 Parameter Adaptation

Meta learning techniques that modify parameters (Ha et al., 2016; Hu et al., 2018; Kuen et al., 2019) are concerned with adapting weights from one model into another, and generating/predicting the complete set of weights for a model given the input samples. In our context, distinctly different

weights are produced for target models trained on inputs of different domains, and feature transferability (Yosinski et al., 2014) in the input space can be expected to translate to weights transferability in the parameter space. Rather than completely regenerating classification weights, our model robustification defense, *Learn2Weight*, predicts the perturbation to existing weights $\theta^* = \theta_i + \widehat{\Delta\theta}$ for each new instance.

5.2 Learn2Weight (L2W)†

We conjecture that an effective defense strategy is to perturb the target model weights depending on the feature distribution of the input instance. In inference (Algorithm 1), L2W recalculates the target model weights depending on the input. During training (Algorithm 2), L2W trains on sentences from different domains and a weight differential for that domain (the weight adjustment required to tune the target model's weights to adapt to the input's domain). We obtain the weight differential $\Delta\theta$ by finding the difference between the weights θ_j trained on sentence:label pairs from a specific domain $X_j \sim \mathcal{X}$ and weights θ_i trained on sentence:label pairs from the target domain X_i. Other training models may be possible; here we trained a sequence-to-sequence network (Sutskever et al., 2014) on sentence:$\Delta\theta$ pairs.

5.3 Perturbation Sets Generation†

To generate synthetic domains of varying domain similarity $\mathbf{S} = \{X_j : Y_j\}_{j=1}^{T}$ so that defenders defend their model using only target domain data X_i, a defender iteratively generates perturbation sets that minimizes transfer loss while maximizing adversarial perturbations (Algorithm 3). A *perturbation set* is a set containing subsets of perturbed inputs (Alzantot et al., 2018; Wong et al., 2019). To construct one perturbation set (Eqt 2), we utilize an iterative minimax algorithm, where we iteratively apply a maximizing adversarial perturbation factor $\varepsilon \geq \varepsilon_{\text{min}}$, and accept the batch of perturbed inputs if it yields a minimizing input distance $\texttt{dist} \leq d_{\text{max}}$. We repeat this T times. We use transfer loss as the distance metric to optimize for domain similarity. We retain FGSM as the adversarial attack algorithm.

$$
\begin{aligned}
X^* &:= \min \texttt{dist}(X^*, X_i) \leq d_{\text{max}} \\
X^* &:= \min \arg\max_{\varepsilon \sim [\varepsilon_{\text{min}}, 1]} \texttt{dist}(\texttt{Adv}(\theta_i; X_i; \varepsilon), X_i) \\
X^* &:= \min \arg\max_{\varepsilon \sim [\varepsilon_{\text{min}}, 1]} [e(\texttt{Adv}(\theta_i; X_i; \varepsilon), X_i) - e(X_i, X_i)]
\end{aligned}
\tag{2}
$$

Algorithm 1: Learn2Weight (Inference)

inference $(X_j^{\text{adv}}, \hbar(\theta^{m\ell}), \ell(\theta_i))$

Input : test-time inputs X_j^{adv}; L2W $\hbar(\theta^{m\ell})$; base learner $\ell(\theta_i)$

Output : label $\hat{y}$

Compute parameter differential w.r.t. X_j^{adv}.

$\widehat{\Delta\theta} \leftarrow \hbar(\theta^{m\ell}; X_j^{\text{adv}})$

Update θ^{ℓ}.

$\hat{y} \leftarrow \ell(\theta_i + \widehat{\Delta\theta}; X_j^{\text{adv}})$

return $\hat{y}$

Algorithm 2: Learn2Weight (Training)

train $(\mathbf{S}, \mathbf{D}, \theta_i, \mathbf{E}^{\ell}, \mathbf{E}^{m\ell})$

Input : domains (perturbation sets) $\mathbf{S}$, target domain $\mathbf{D} = \{X_i : Y_i\}$, base learner parameters θ_i, epochs $\mathbf{E}^{\ell}$ & $\mathbf{E}^{m\ell}$

Output : L2W parameters $\theta^{m\ell}$

Initialize empty set Θ to store parameter differential.

$\Theta \leftarrow \emptyset$;

Compute $X_j \mapsto \Delta\theta$.

foreach $X_j : Y_j \in (\mathbf{D} \cup \mathbf{S})$ **do**

  for $e \leftarrow 0$ **to** $\mathbf{E}^{\ell}$ **do**

    $\theta_{j,e}^{\ell} := \theta_{j,e-1}^{\ell} - \sum_{\mathbf{x},y}^{X_j,Y_j} \frac{\partial \mathcal{L}(\mathbf{x},y)}{\partial \theta^{\ell}}$

  $\Delta\theta \leftarrow \theta_j^{\ell} - \theta_i$

  $\Theta \leftarrow \Delta\theta$;

Compute $\theta^{m\ell}$.

for $e \leftarrow 0$ **to** $\mathbf{E}^{m\ell}$ **do**

  $\theta_e^{m\ell} := \theta_{e-1}^{m\ell} - \sum_{X_j,\Delta\theta}^{(X_i \cup \mathbf{S}),\Theta} \frac{\partial \mathcal{L}(X_j,\Delta\theta)}{\partial \theta^{m\ell}}$

return $\theta^{m\ell}$

Algorithm 3: Perturbation Sets Generation

PerturbationSet $(\mathbf{D}, \theta_i; T, R; \texttt{dist}, d_{\text{max}}; \varepsilon, \gamma)$

Input : target domain $\mathbf{D} = \{X_i : Y_i\}$, parameters θ_i; number of perturbation sets $T = 10$, max iterations $R = 10$; distance metric $\texttt{dist} = tf(X_i, X_j)$, max distance $d_{\text{max}} = 0.1$; initial perturbation rate $\varepsilon = 0.9$, perturbation learning rate $\gamma = 0.05$;

Output : set $\mathbf{S}$ containing T perturbation sets

Initialize empty $\mathbf{S}$ to store perturbation sets S_t.

$\mathbf{S} \leftarrow \emptyset$;

while $t < T$ **do**

  Run next iteration r until S_t meets conditions.

  for $r \leftarrow 0$ **to** R **do**

    Apply adversarial perturbations to X.

    $S_{t,r} \leftarrow \texttt{Adv}(\theta_i; X_i; \varepsilon)$;

    Evaluate distance conditions.

    if $\texttt{dist}(S_{t,r}, X_i) \leq d_{\text{max}}$ **then**

      if $\sigma^2(\mathbf{S} \cup S_{t,r}) > \sigma^2(\mathbf{S})$ **then**

        $\mathbf{S} \leftarrow \{S_{t,r} : Y_i\}$;

        continue;

    else

      Adjust hyperparameters.

      $\varepsilon \leftarrow \varepsilon - \gamma$;

  $t \leftarrow t + 1$;

return $\mathbf{S}$

5.4 Explanation: Blocking Transferability

To facilitate our explanation, we adapt from domain adaptation literature (Ben-David et al., 2010; Liu et al., 2019; Zhang et al., 2019c):

$$e(X_j^{\text{adv}}, X_i) \leq e(X_i, X_i) + d_{\mathcal{H}\Delta\mathcal{H}}(X_j^{\text{adv}}, X_i) + \lambda \quad (3)$$

where $\mathcal{H}$ is the hypothesis space, h is a hypothesis function that returns labels $\{0, 1\}$, and $e(X_i, X_i)$ and $e(X_j^{\text{adv}}, X_i)$ are the generalization errors from passing target domain data X_i and adversarial data X_j^{adv} through a classifier trained on X_i. $d_{\mathcal{H}\Delta\mathcal{H}}(X_j^{\text{adv}}, X_i)$ is the $\mathcal{H}\Delta\mathcal{H}$-distance between X_i and X_j^{adv}, and measures the divergence between the feature distributions of X_j^{adv} and X_i. $e_{X_j^{\text{adv}}}(h, h^{'})$ and $e_{X_i}(h, h^{'})$ represent the probability that h disagrees with $h^{'}$ on the label of an input in the domain space X_j^{adv} and X_i respectively.

$$\begin{aligned} d_{\mathcal{H}\Delta\mathcal{H}}(X_j^{\text{adv}}, X_i) &= \sup_{h,h^{'}\in\mathcal{H}} |e_{X_j^{\text{adv}}}(h, h^{'}) - e_{X_i}(h, h^{'})| \\ d_{\mathcal{H}\Delta\mathcal{H}}(X_j^{\text{adv}}, X_i) &= \sup_{h,h^{'}\in\mathcal{H}} \left|\mathbb{E}_{x_j\sim X_j}[|(h(x_j) - h^{'}(x_j)|]\right| \\ &\quad - \left|\mathbb{E}_{x_i\sim X_i}[|(h(x_i) - h^{'}(x_i)|]\right| \end{aligned} \quad (4)$$

Divergence $d_{\mathcal{H}\Delta\mathcal{H}}$ measures the divergence between feature distributions X_j^{adv} and X_i. Higher $d_{\mathcal{H}\Delta\mathcal{H}}$ indicates less shared features between 2 domains. The greater the intersection between feature distributions, the greater the proportion of domain-variant features; one approach to domain adaptation is learning domain-invariant features representations (Zhao et al., 2019) to minimize $d_{\mathcal{H}\Delta\mathcal{H}}$.

Explaining similarity-domain attacks. As demonstrated by empirical results, $e(X_j^{\text{adv}}, X_i)$ increases in a similarity-based attack setting, and this would arise if $d_{\mathcal{H}\Delta\mathcal{H}}$ increases correspondingly. $d_{\mathcal{H}\Delta\mathcal{H}}$ computes inconsistent labels from inconsistent feature distributions, and attributes the success of the attack to domain-variant features.

FGSM and variants adjust the input data to maximize the loss based on the backpropagated gradients of a model trained on X_j. As our pipeline used correctly-labelled sentences before adversarially perturbing them, we can infer that perturbations applied to X_j were not class-dependent (i.e. the success of the attack is not based on the removal of class-specific features), but class-independent features. It is already difficult for a model trained on X_j to classify when there is insufficient class-dependent features (hence a high $tf(X_j^{\text{adv}}, X_i)$); in a cross-domain setting, it must be even more difficult for a model trained on X_i to classify given a shortage of domain-invariant, class-dependent features.

$$\begin{aligned} d_{\mathcal{H}\Delta\mathcal{H}} &\geq e(X_j^{\text{adv}}, X_i) - e(X_i, X_i) - \lambda \\ d_{\mathcal{H}\Delta\mathcal{H}} &\geq tf(X_j^{\text{adv}}, X_i) - \lambda \end{aligned} \quad (5)$$

Explaining Learn2Weight. L2W minimizes divergence by training on $\{d_{\mathcal{H}\Delta\mathcal{H}}(X_j, X_i) : \Delta\theta\}$ pairs, such that $\Delta\theta = L2W(d_{\mathcal{H}\Delta\mathcal{H}}(X_j, X_i))$, where $d_{\mathcal{H}\Delta\mathcal{H}}(X_j, X_i)$ is reconstructed from the difference between X_j and X_i. The target model possesses a decision boundary (Liu et al., 2019) to classify inputs based on whether they cross the boundary or not; adversarial inputs have a tendency of being near the boundary and fooling it. Meta learning applies perturbations to the decision boundary such that the boundary covers certain adversarial inputs otherwise misclassified, and in this way blocks transferability. The advantage of training on multiple domains $\{X_j\}_{j=1}^{T}$ is that the after-L2W divergence between X_j^{adv} and X_i is smaller because L2W's weight perturbations render the decision boundary more precise in classifying inputs.

Explaining perturbation sets. We attributed why adversarial sentences X_j^{adv} are computed to be domain-dissimilar despite originating from X_j due to insufficient domain-invariant, class-dependent features resulting in low $e(X_j^{\text{adv}}, X_i)$, i.e. low $tf(X_j^{\text{adv}}, X_i)$. To replicate this phenomenon in natural domains, we iteratively perturb X_i to increase the proportion of class-independent features. This approximates the real-world similarity-based attack scenario where class-dependent features may be limited for inference. By generating the synthetic data, we are feeding L2W attack data with variations in $d_{\mathcal{H}\Delta\mathcal{H}}$ and class-independent feature distributions. This prepares L2W to robustify weights θ_i when such feature distributions are met.

Target Domain	magazine			baby		
Attack Domain	baby	dvd	book	dvd	book	magazine
After-attack Accuracy	0.381	0.366	0.343	0.365	0.386	0.401
After-defense Accuracy						
Adversarial Training	0.639	0.559	0.657	0.558	0.577	0.661
Defensive Distillation	0.549	0.561	0.597	0.588	0.629	0.577
Perturbation Sets Adversarial Training	0.608	0.637	0.620	0.604	0.620	0.587
Learn2Weight	**0.796**	**0.842**	**0.843**	**0.774**	**0.751**	**0.737**

Table 3: *After-defense Accuracy:* Learn2Weight outperforms the baseline and ablation methods.

Target Domain	Attack Domain	After-Attack Accuracy					After-Defense Accuracy				
		BERT	LSTM	GRU	CNN	LogReg	BERT	LSTM	GRU	CNN	LogReg
book	dvd	0.342	0.413	0.477	0.335	0.440	0.786	0.847	0.804	0.816	0.782
	kitchenware	0.350	0.372	0.325	0.353	0.425	0.765	0.826	0.795	0.742	0.767
	electronics	0.400	0.389	0.416	0.315	0.460	0.792	0.812	0.784	0.770	0.725
dvd	book	0.326	0.434	0.479	0.383	0.490	0.816	0.795	0.824	0.804	0.794
	kitchenware	0.355	0.370	0.379	0.359	0.490	0.728	0.796	0.755	0.735	0.695
	electronics	0.387	0.377	0.332	0.348	0.455	0.825	0.836	0.812	0.834	0.796
electronics	book	0.425	0.394	0.473	0.358	0.474	0.775	0.821	0.795	0.782	0.712
	dvd	0.342	0.395	0.452	0.368	0.493	0.784	0.845	0.855	0.842	0.792
	kitchenware	0.390	0.384	0.464	0.329	0.432	0.730	0.824	0.753	0.724	0.678

Table 4: *Models:* L2W retains high after-defense accuracy at varying attack model architectures.

6 Experiments†

6.1 Baselines

Defensive distillation (Papernot et al., 2016c, 2017): The high-level implementation of *defensive distillation* is to first train an initial model against target domain inputs and labels, and retrieve the raw class probability scores. The predicted probability values would be used as the new labels for the same target sentences, and we would train a new model based on this new label-sentence pair.

Adversarial training (Goodfellow et al., 2014; Madry et al., 2017): It is shown that injecting adversarial examples throughout training increases the robustness of target neural network models. In this baseline, target model is trained with both original training data and adversarial examples generated from original training data. However, since the adversarial examples are still generated from the target domain, it is unlikely that the method can defend against a similar-domain adversarial attack, which is the result of domain-variant features.

Perturbation sets adversarial training: This ablation baseline tests for incremental performance to a baseline defense using domain-variant inputs. We adapt adversarial training to be trained on perturbation sets (synthetic domains) generated with Algorithm 3 with respect to target domain X_i.

6.2 Learn2Weight Performance

Defense performance. We present the results of different defense baselines in Table 3. First, we can see that L2W achieves the highest after-defense accuracy against the adversarial attack. Take the *magazine* as target domain for example: if the adversary chooses to use *book* data as the attack domain, it would reduce the target model accuracy to 0.343. However, L2W can improve the performance to 0.843, which is a significant and substantial improvement against the attack. This improvement also exist across different target/attack domain pairs. Second, we see that all defense methods can improve the accuracy to some extent which indicates the importance and effectiveness of having robust training for machine learning models.

Attack model architectures. So far, all the results are conducted using the same LSTM as the target/attack model. Here, we keep the target model unchanged, but vary the architecture of the attack model for the generation of adversarial examples. LSTM (GRU) is configured with 64 cells, tokens embedded with respect to GloVe, `sigmoid` (`tanh`) activation function, randomly-initialized and trained with Adam optimizer and 80% (60%) dropout, based on Wang et al. (2018). CNN is configured with accepting tokens embedded with respect to GloVe (Pennington et al., 2014), 3 convo-

lutional layers with kernel widths of 3, 4, and 5, all with 100 output channels, and randomly-initialized, based on Kim (2014). We configure Logistic Regression based on Maas et al. (2011). Based on Devlin et al. (2019), we initialize a pretrained BERT with its own embeddings. Models are trained until reaching state-of-the-art validation accuracy (early-stopping pauses training at loss 0.5).

We present the results of different attack model architectures in Table 4. First, the similar-domain adversarial attack is model-agnostic and it does not require the target and attack model to have identical architectures. We can see that all four attack model architectures are able to reduce the target model accuracy. Second, the results suggest that L2W is also model-agnostic as it can substantially improve the after-defense accuracy regardless which attack model is used.

7 Conclusion

In this newly-proposed, empirically-effective similar-domain adversarial attack, an adversary can choose a similar domain to the target task, build a substitute model and produce adversarial examples to fool the target model. We also propose a defense strategy, Learn2Weight, that learns to adapt the target model's weight using crafted adversarial examples. Compared with other adversarial defense strategies, Learn2Weight can improve the target model robustness against the similar-domain attack. Our method demonstrates properties of a good adversarial defense, such as adopting a defense architecture that adapts to situations/inputs rather than compromising standard error versus robustness error, to leverage class-independent properties in domain-variant text, and factoring in domain similarity in adversarial robustness.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Vaughan. 2010. A theory of learning from different domains. *Machine Learning*, 79:151–175.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, pages 440–447.

Karsten M. Borgwardt, Arthur Gretton, Malte J. Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J. Smola. 2006. Integrating structured biological data by Kernel Maximum Mean Discrepancy. *Bioinformatics*, 22(14):e49–e57.

Shuyu Cheng, Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. 2019. Improving black-box adversarial attacks with a transfer-based prior. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 10934–10944. Curran Associates, Inc.

Xia Cui, Frans Coenen, and Danushka Bollegala. 2017. Tsp: Learning task-specific pivots for unsupervised domain adaptation. In *Machine Learning and Knowledge Discovery in Databases*, pages 754–771, Cham. Springer International Publishing.

Siddhartha Datta. 2022. Learn2weight: Parameter adaptation against similar-domain adversarial attacks.

Siddhartha Datta and Nigel Shadbolt. 2022a. Backdoors stuck at the frontdoor: Multi-agent backdoor attacks that backfire. In *International Conference on Learning Representations Workshop: Gamification and Multiagent Solutions*.

Siddhartha Datta and Nigel Shadbolt. 2022b. Interpolating compressed parameter subspaces.

Siddhartha Datta and Nigel Shadbolt. 2022c. Low-loss subspace compression for clean gains against multi-agent backdoor attacks. *arXiv preprint arXiv:2203.03692*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification. In *ACL*, pages 31–36.

Samuel G. Finlayson, John D. Bowers, Joichi Ito, Jonathan L. Zittrain, Andrew L. Beam, and Isaac S. Kohane. 2019. Adversarial attacks on medical machine learning. *Science*, 363(6433):1287–1289.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks.

Tomer Galanti and Lior Wolf. 2020. On the modularity of hypernetworks.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. 2019. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*, pages 513–520.

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

David Ha, Andrew Dai, and Quoc V Le. 2016. Hypernetworks. *arXiv preprint arXiv:1609.09106*.

Marton Havasi, Rodolphe Jenatton, Stanislav Fort, Jeremiah Zhe Liu, Jasper Snoek, Balaji Lakshminarayanan, Andrew M. Dai, and Dustin Tran. 2021. Training independent subnetworks for robust prediction.

Dan Hendrycks and Kevin Gimpel. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks.

Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross Girshick. 2018. Learning to segment every thing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4233–4241.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4129–4142, Hong Kong, China. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR.

Jason Kuen, Federico Perazzi, Zhe Lin, Jianming Zhang, and Yap-Peng Tan. 2019. Scaling object detection by transferring classification weights. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6044–6053.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016a. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016b. Adversarial examples in the physical world.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016c. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 957–966, Lille, France. PMLR.

Alex Lamb, Vikas Verma, Juho Kannala, and Yoshua Bengio. 2019. Interpolated adversarial training: Achieving robust neural networks without sacrificing too much accuracy. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, AISec'19, page 95–103, New York, NY, USA. Association for Computing Machinery.

Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. 2018. Defense against adversarial attacks using high-level representation guided denoiser. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 1778–1787. Computer Vision Foundation / IEEE Computer Society.

Etai Littwin, Tomer Galanti, Lior Wolf, and Greg Yang. 2020. On infinite-width hypernetworks. *Advances in Neural Information Processing Systems*, 2020-December. Publisher Copyright: © 2020 Neural information processing systems foundation. All rights reserved.; null ; Conference date: 06-12-2020 Through 12-12-2020.

Guanxiong Liu, Issa Khalil, Abdallah Khreishah, and NhatHai Phan. 2021. A synergetic attack against neural network classifiers combining backdoor and adversarial examples.

Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. 2019. Transferable adversarial training: A general approach to adapting deep classifiers. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4013–4022, Long Beach, California, USA. PMLR.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *ACL*, pages 142–150.

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

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, pages 2574–2582.

Muzammal Naseer, Salman H. Khan, Harris Khan, Fahad Shahbaz Khan, and Fatih Porikli. 2019. Cross-domain transferability of adversarial perturbations.

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. 2020. What is being transferred in transfer learning? In *Advances in Neural Information Processing Systems*, volume 33, pages 512–523. Curran Associates, Inc.

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms.

Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016a. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, page 506–519, New York, NY, USA. Association for Computing Machinery.

Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. 2016b. Crafting adversarial input sequences for recurrent neural networks. In *MILCOM 2016 - 2016 IEEE Military Communications Conference*, page 49–54. IEEE Press.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016c. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2021. Mind the style of text! adversarial and backdoor attacks based on text style transfer.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C Duchi, and Percy Liang. 2019. Adversarial training can hurt generalization. *arXiv preprint arXiv:1906.06032*.

Alexandre Ramé, Rémy Sun, and Matthieu Cord. 2021. Mixmo: Mixing multiple inputs for multiple outputs via deep subnetworks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 823–833.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. 2019. Learning to learn without forgetting by maximizing transfer and minimizing interference.

Evgenia Rusak, Lukas Schott, Roland S. Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. 2020. A simple way to make neural networks robust against diverse image corruptions. In *Computer Vision – ECCV 2020*, pages 53–69, Cham. Springer International Publishing.

Shiori Sagawa*, Pang Wei Koh*, Tatsunori B. Hashimoto, and Percy Liang. 2020. Distributionally robust neural networks. In *International Conference on Learning Representations*.

Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. 2020. Breeds: Benchmarks for subpopulation shift.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. 2018. Adversarially robust generalization requires more data. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 5019–5031, Red Hook, NY, USA. Curran Associates Inc.

Aman Sinha, Hongseok Namkoong, and John Duchi. 2018. Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. 2018. Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models. In *Proceedings of ECCV*, pages 631–648.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

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

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2017. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*.

Johannes von Oswald, Christian Henning, João Sacramento, and Benjamin F. Grewe. 2020. Continual learning with hypernetworks.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. In *EMNLP*, pages 2153–2162.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric Xing. 2019. Learning robust global representations by penalizing local predictive power. *Neural Information Processing Systems*, pages 10506–10518.

Jenq-Haur Wang, Ting-Wei Liu, Xiong Luo, and Long Wang. 2018. An LSTM approach to short text sentiment classification with word embeddings. In *ROCLING*, pages 214–223.

Zixin Wen and Yuanzhi Li. 2021. Toward understanding the feature learning process of self-supervised contrastive learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11112–11122. PMLR.

Eric Wong, Frank R. Schmidt, and J. Zico Kolter. 2019. Wasserstein adversarial examples via projected sinkhorn iterations. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6808–6817. PMLR.

Mitchell Wortsman, Maxwell Horton, Carlos Guestrin, Ali Farhadi, and Mohammad Rastegari. 2021. Learning neural network subspaces.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time.

Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan Yuille. 2018. Improving transferability of adversarial examples with input diversity.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3320–3328, Cambridge, MA, USA. MIT Press.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. 2019a. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of ICML*, pages 7472–7482.

Huan Zhang, Hongge Chen, Zhao Song, Duane Boning, Inderjit S. Dhillon, and Cho-Jui Hsieh. 2019b. The limitations of adversarial training and the blind-spot attack.

Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41.

Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. 2019c. Bridging theory and algorithm for domain adaptation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7404–7413, Long Beach, California, USA. PMLR.

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. 2019. On learning invariant representations for domain adaptation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7523–7532, Long Beach, California, USA. PMLR.

Yftah Ziser and Roi Reichart. 2019. Task refinement learning for improved accuracy and stability of unsupervised domain adaptation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5895–5906, Florence, Italy. Association for Computational Linguistics.