Agnostic Multi-Robust Learning Using ERM

Saba Ahmadi
Toyota Technological Institute at Chicago

Avrim Blum
Toyota Technological Institute at Chicago

Omar Montasser
UC Berkeley

Kevin Stangl
Toyota Technological Institute at Chicago

Abstract

A fundamental problem in robust learning is asymmetry: a learner needs to correctly classify every one of exponentially-many perturbations that an adversary might make to a test-time natural example. In contrast, the attacker only needs to find one successful perturbation. Xiang et al. (2022) proposed an algorithm that in the context of patch attacks for image classification, reduces the effective number of perturbations from an exponential to a polynomial number of perturbations and learns using an ERM oracle. However, to achieve its guarantee, their algorithm requires the natural examples to be robustly realizable. This prompts the natural question; can we extend their approach to the non-robustly-realizable case where there is no classifier with zero robust error?

Our first contribution is to answer this question affirmatively by reducing this problem to a setting in which an algorithm proposed by Feige et al. (2015) can be applied, and in the process extend their guarantees. Next, we extend our results to a multi-group setting and introduce a novel agnostic multi-robust learning problem where the goal is to learn a predictor that achieves low robust loss on a (potentially) rich collection of subgroups.

1 INTRODUCTION

Robustness to adversarial examples is considered a major contemporary challenge in machine learning. Adversarial examples are carefully crafted perturbations or manipulations of natural examples that cause machine learning predictors to miss-classify at test-time (Goodfellow et al., 2014). One particularly challenging aspect of this problem is the asymmetry between the learner and the adversary. Specifically, a learner needs to produce a predictor that is *correct* on a randomly drawn natural example and *robust* to potentially *exponentially* many possible perturbations of it; while, the adversary needs to find just a single perturbation that fools the learner. In fact, because of this, adversarially robust learning has proven to require more sophisticated learning algorithms that go beyond standard Empirical Risk Minimization (ERM) in non-robust learning (Montasser et al., 2019).

In patch attacks on images, for instance, an adversary can select one of an exponential number of designs for a patch to be placed in the image in order to cause a classification error. To address this exponential asymmetry between the learner and the adversary, recently Xiang et al. (2022) introduced a clever algorithmic scheme, known as Patch-Cleanser, that provably reduces the exponential number of ways that an adversary can attack to a polynomial number of ways through the idea of masking images.

Specifically, Patch-Cleanser's *double-masking* approach is based on zero-ing out two different contiguous blocks of an input image, hopefully to remove the adversarial patch. For each one-masked image, if for all possible locations of the second mask, the prediction model outputs the same classification, it means that the first mask removed the adversarial patch, and the agreed-upon prediction is correct. Any disagreements in these predictions imply that the mask was not covered by the first patch. Crucially, the Patch-Cleanser algorithm requires a *two-mask correctness* guarantee from an underlying predictor F that is defined as follows: for a given input image x and label y, if for any pair of masks applied to x, predictor F outputs the correct prediction y, then F has a two-correctness guarantee on (x, y) (see Definition 2 in Xiang et al., 2022). In order to train a predictor with the two-mask correctness guarantee, Xiang et al. (2022) augment the training dataset with pairs of masks at random locations of training images, and perform *empirical risk minimization* (ERM) on the augmented dataset.

Our Contributions When no predictor is perfectly correct on *all* perturbations (e.g., all two-mask operations), which we refer to as the the *non-realizable* or *agnostic* setting, we exhibit an example where plain ERM on the augmented

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

dataset fails (See Example 1). At a high-level, the main issue is that plain ERM on the augmented data-set treats all mistakes equally and so this could lead to learning a predictor with very high robust loss, i.e. on *many* training examples. Our first contribution is to investigate whether the reduction proposed by Xiang et al. (2022) can be extended to the *non-realizable* setting. We answer this question affirmatively in Section 3, by building upon a prior work by Feige et al. (2015).

Next, in Section 4, we consider a multi-group setting and investigate the question of *agnostic multi-robust learning* using an ERM oracle. This question is inspired by the literature on *multi-calibration* and *multi-group learning* (Hébert-Johnson et al., 2017; Kim et al., 2019; Rothblum and Yona, 2021; Tosh and Hsu, 2021; Globus-Harris et al., 2022). Our objective is that given a hypothesis class $\mathcal{H}$ and a (potentially) rich collection of subgroups $\mathcal{G}$, learn a predictor h such that for each group $g \in \mathcal{G}$, h has low robust loss on $g \in \mathcal{G}$. However, we highlight that the prior work on multi-group learning does not extend to the setting of robust loss since they do not consider adversarial perturbations of natural examples. To our knowledge, our work is the first to consider the notion of multi-group learning for robust loss. That being said we emphasize that there is a trade-off here; our guarantees are for the *more challenging objective of robust loss*, but they are weaker than the ones given for PAC learning in the prior work. A detailed comparison is given in Section 1.1.

Our motivation for studying multi-robustness is two-fold. First, to prohibit the adversary from targeting a specific demographic group for adverse treatment. Additionally, it can increase the overall performance of the model by forcing the model to be robust on vulnerable examples. For instance, imagine a self-driving car system with a vision system recording a drive and we consider adversarial examples attacking individual frames of the video. Ideally, the system would have robust performance over every frame. However, average robust error of 1% could be very problematic if those errors instead of occurring uniformly then those errors concentrated on a specific adjacent set of frames. In this example, imagine that the protected groups are nearby frames so that we maintain smooth and reliable performance *locally and globally*.

To achieve multi-robustness, using plain ERM can fail by *concentrating* the overall robust loss on a *few* groups, instead of *spreading* the loss across *many* groups. However, building on our algorithm in Section 3 we propose Algorithm 2 that runs an additional layer of boosting with respect to groups to achieve multi-robustness guarantees across groups. We propose two types of multi-robustness guarantees, the first one is a randomized approach that guarantees the expected robust loss on each group is low (Theorem 12). Next, we add a de-randomization step to derive deterministic guarantees for the robust loss incurred on each group (Theorem 13).

1.1 Related Work

Patch Attacks Patch attacks (Brown et al., 2017; Karmon et al., 2018; Yang et al., 2020) are an important threat model in the general field of test-time evasion attacks (Goodfellow et al., 2014). Patch attacks realize adversarial test time evasion attacks to computer vision systems in the wild by printing and attaching a patch to an object. To mitigate this threat, there has been an active line of research for providing certifiable robustness guarantees against them (see e.g., McCoyd et al., 2020; Xiang et al., 2020; Xiang and Mittal, 2021; Metzen and Yatsura, 2021; Zhang et al., 2020; Chiang et al., 2020).

Adversarial Learning using ERM Recent work by Feige et al. (2015) gives a reduction algorithm for adversarial learning using an ERM oracle, but their guarantee is only for finite hypothesis classes. We observe in this work that we can apply their reduction algorithm to our problem, and along the way, we extend the guarantees of their algorithm. A more detailed comparison is provided in Section 3.1.

Multi-group Learning Interestingly, the notion of multi-robustness has connections with a thriving area of work in algorithmic fairness centered on the notion of multi-calibration (Hébert-Johnson et al., 2017; Kim et al., 2019; Rothblum and Yona, 2021; Tosh and Hsu, 2021; Globus-Harris et al., 2022; Gopalan et al., 2022). The promise of these multi-guarantees, given a rich set of groups, is to ensure uniformly acceptable performance on many groups simultaneously.

Specifically, Rothblum and Yona (2021) show how to learn a predictor such that the loss experienced by every group is not much larger than the best possible loss for this group within a given hypothesis class. However, we highlight that the prior work on multi-group learning does not extend to the setting of robust loss since their goal is not to minimize the robust loss by taking into consideration different perturbations of natural examples. In contrast, our approach can achieve multi-robustness guarantees by utilizing two layers of boosting to ensure 'emphasis' on both specific groups and the adversarial perturbations.

Tosh and Hsu (2021); Globus-Harris et al. (2022) study the problem of minimizing a general loss function over a collection of subgroups. Their approach can capture the robust loss, however, the main distinction between their algorithm and our approach is that unlike them, we do not use group membership during the test time. This is essential when groups correspond to protected features, and therefore in some scenarios, it would be undesirable to incorporate them in decision models. Additionally, if we interpret some of the groups in our setting as objects to be classified like

a stop-sign group or fire-hydrant group, then an approach that needs to detect group membership is too strong an assumption since the correct classification of those objects is our original goal.

However, we highlight that there is a trade-off here; To our knowledge, our work is the first one to achieve guarantees for the *more challenging objective of robust learning without having access to the group membership of examples* but at the cost of achieving a weaker upper bound on the robust loss incurred on each group compared to the previous work on multi-group PAC learning. A detailed comparison is given in Section 4.1.

2 SETUP AND NOTATION

Let $\mathcal{X}$ denote the instance space and $\mathcal{Y}$ denote the label space. Our main objective is to be robust against adversarial patches $\mathcal{A} : \mathcal{X} \to 2^{\mathcal{X}}$, where $\mathcal{A}(x)$ represents the (potentially infinite) set of adversarially patched images that an adversary might attack with at test-time on input x. Xiang et al. (2022) showed that even though the space of adversarial patches $\mathcal{A}(x)$ can be exponential or infinite, one can consider a "covering" function $\mathcal{U} : \mathcal{X} \to 2^{\mathcal{X}}$ of masking operations on images where $|\mathcal{U}(x)|$ shows the covering set on input image x and is polynomial in the image size. Thus, for the remainder of the paper, we focus on the task of learning a predictor robust to a perturbation set $\mathcal{U} : \mathcal{X} \to 2^{\mathcal{X}}$, where $\mathcal{U}(x)$is the set of allowed masking operations that can be performed on x. We assume that $\mathcal{U}(x)$ is finite where $|\mathcal{U}(x)| \leq k$.

We observe m iid samples $S \sim \mathcal{D}^m$ from an unknown distribution $\mathcal{D}$, and our goal is to learn a predictor $\hat{h}$ achieving small robust risk:

$$\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[\max_{z\in\mathcal{U}(x)} \mathbb{1}[\hat{h}(z) \neq y]]. \tag{1}$$

Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a hypothesis class, and denote by $\mathrm{vc}(\mathcal{H})$ its VC dimension. Let $\mathsf{ERM}_{\mathcal{H}}$ be an ERM oracle for $\mathcal{H}$ that returns a hypothesis $h \in \mathcal{H}$ that minimizes empirical loss. For any set arbitrary set W, denote by $\Delta(W)$ the set of distributions over W.

In Section 3, we focus on a single-group setting where the benchmark $\mathsf{OPT}_{\mathcal{H}}$ is defined as follows:

$$\mathsf{OPT}_{\mathcal{H}} \triangleq \min_{h\in\mathcal{H}} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \max_{z\in\mathcal{U}(x)} \mathbb{1}\left[h(z) \neq y\right]. \tag{2}$$

In Section 4, we consider a multi-group setting, where the instance space $\mathcal{X}$ is partitioned into a set of g groups $\mathcal{G} = \{G_1, \ldots, G_g\}$. These groups solely depend on the features x and not the labels. The goal is to learn a predictor that has low robust loss on all the groups simultaneously. In this setup, the benchmark $\mathsf{OPT}^{\mathcal{D}}_{\max}$ is as follows:

$$\mathsf{OPT}^{\mathcal{D}}_{\max} = \min_{h\in\mathcal{H}} \max_{j\in[g]} \mathop{\mathbb{E}}_{(x,y)\sim D} \left[\max_{z\in\mathcal{U}(x)} \mathbb{1}[h(z) \neq y] \middle| x \in G_j\right] \tag{3}$$

3 MINIMIZING ROBUST LOSS USING AN ERM ORACLE

First, we show an example where the approach of Xiang et al. (2022) of calling $\mathsf{ERM}_{\mathcal{H}}$ on the inflated dataset, i.e., original training points plus all possible perturbations resulting from the allowed masking operations, fails by obtaining a multiplicative gap of $k - 1$ in the robust loss between the optimal robust classifier and the classifer returned by $\mathsf{ERM}_{\mathcal{H}}$, where k is the size of the perturbation sets. This gap exists since ERM can exhibit a solution that incorrectly classifies at least one perturbation per natural example, while there is a robust classifier that concentrates error on one natural example, thus getting low robust loss.

Example 1. *Consider the following example in $\mathcal{R}$. There is a training set $\{z_1, \cdots, z_{2n}\}$ of original examples, where examples $\{z_1, \cdots, z_n\}$ are positively labeled and are located at $x = 1$. $\{z_{n+1}, \cdots, z_{2n}\}$ are negatively labeled and are at $x = -1$. Each example z_i has $k = n$ perturbations denoted by $\{z_{i,1}, \cdots, z_{i,k}\}$.*

For each of the negative examples $\{z_{n+1}, \cdots, z_{2n-1}\}$, all their perturbations are at $x = -0.75$. For the negative example z_{2n}, all its perturbations, i.e. $\{z_{2n,1}, \cdots, z_{2n,k}\}$, are at $x = 0$. For each positive example z_i where $i \in \{1, \cdots, n-1\}$, one of their perturbations $z_{i,1}$ is at $x = 0$ and the rest, i.e. $\{z_{i,2}, \cdots, z_{i,k}\}$, are at $x = 0.75$. For the positive example z_n, all its perturbations $z_{n,1}, \cdots, z_{n,k}$ are at $x = 0.75$.

The adversarial training procedure considered in the paper by Xiang et al. (2022) runs ERM on the augmented dataset (original examples and all their perturbations) to minimize the 0/1 loss. ERM finds a threshold classifier h_{ERM} with threshold $\tau = \varepsilon_1$ for any $0 < \varepsilon_1 < 0.75$ that classifies any points with $x < \tau$ as negative and points with $x \geq \tau$ as positive. As a result, for each positive example z_i for $i \in \{1, \cdots, n-1\}$, the perturbation $z_{i,1}$ is getting classified mistakenly which causes a robust loss on z_i. Therefore, h_{ERM} has a robust loss of $(n-1)/2n$ since $n-1$ of the positive examples are not robustly classified. However, there exists a threshold classifier h^* with threshold $\tau = \varepsilon_2$ for any $-0.75 < \varepsilon_2 < 0$ that only makes mistakes on perturbations of z_{2n} and thus has a robust loss of $1/2n$. However, its 0/1 loss is higher than h_{ERM} and therefore ERM does not pick it. Therefore, ERM can be suboptimal up to a multiplicative factor of $n-1$ for any arbitrary value of n. An illustration is given in Figure 1.

Next, we present our first contribution: we show in Theorem 1 that Algorithm 1 proposed by Feige et al. (2015)

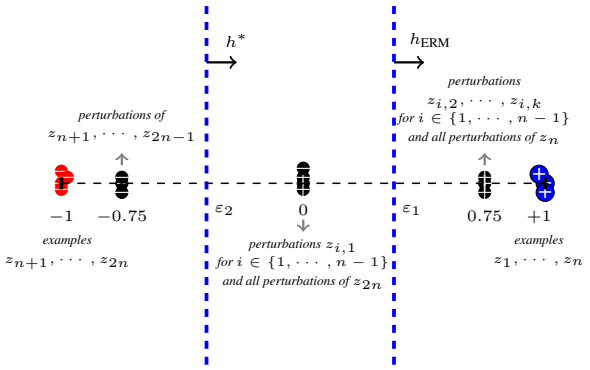

Figure 1: ERM failure mode in the robustly un-realizable case. Blue, red, and black points show respectively original examples with a positive label, original examples with a negative label, and perturbations of original examples.

learns a predictor that is simultaneously robust to a set of (polynomially many) masking operations, using an $\mathsf{ERM}_{\mathcal{H}}$ oracle. The algorithm is based on prior work, but the analysis and application are novel in this work. A detailed comparison with Feige et al. (2015) is given in Section 3.1. The main interesting feature of this algorithm is that it achieves stronger robustness guarantees in the non-realizable regime when $\mathsf{OPT}_{\mathcal{H}} \gg 0$, where the approach of Xiang et al. (2022) can fail as mentioned in Example 1.

Algorithm 1 Feige, Mansour, and Schapire (2015)

Input weight update parameter $\eta > 0$, number of rounds T, and training dataset $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ and corresponding weights $p_1, \cdots, p_m$

Set $w_1(z, (x, y)) = 1$, for each $(x, y) \in S, z \in \mathcal{U}(x)$.

Set $P^1(z, (x, y)) = \frac{w_1(z,(x,y))}{\sum_{z' \in \mathcal{U}(x)} w_1(z',(x,y))}$, for each $(x, y) \in S, z \in \mathcal{U}(x)$.

for *each* $t \in \{1, \cdots T\}$ **do**

Call ERM on the empirical weighted distribution:

$$h_t = \operatorname*{argmin}_{h \in \mathcal{H}} \sum_{(x,y) \in S} \sum_{z \in \mathcal{U}(x)} p_{(x,y)} P^t(z, (x, y)) \mathbb{1}\left[h_t(z) \neq y\right]$$

for *each* $(x, y) \in S$ *and* $z \in \mathcal{U}(x)$ **do**

$w_{t+1}(z, (x, y)) = (1 + \eta \mathbb{1}\left[h_t(z) \neq y\right]) \cdot w_t(z, (x, y))$

$P^{t+1}(z, (x, y)) = \frac{w_t(z,(x,y))}{\sum_{z' \in \mathcal{U}(x)} w_t(z',(x,y))}$

Output The majority-vote predictor $\mathrm{MAJ}(h_1, \ldots, h_T)$.

Theorem 1. *Set* $T(\varepsilon) = \frac{32 \ln k}{\varepsilon^2}$ *and* $m(\varepsilon, \delta) = O\left(\frac{\mathrm{vc}(\mathcal{H})(\ln k)^2}{\varepsilon^4} \ln\left(\frac{\ln k}{\varepsilon^2}\right) + \frac{\ln(1/\delta)}{\varepsilon^2}\right)$. *Then, for any distribution* $\mathcal{D}$ *over* $\mathcal{X} \times \mathcal{Y}$, *with probability at least* $1 - \delta$ *over* $S \sim \mathcal{D}^{m(\varepsilon,\delta)}$, *running Algorithm 1 where* $p_{(x,y)} = 1/m$ *for all* $(x, y) \in S$ *for* $T(\varepsilon)$ *rounds produces* $h_1, \ldots, h_{T(\varepsilon)}$ *satisfying:*

$$\mathop{\mathbb{E}}_{(x,y) \sim \mathcal{D}} \left[\max_{z \in \mathcal{U}(x)} \mathbb{1}\left[\mathrm{MAJ}(h_1, \ldots, h_{T(\varepsilon)})(z) \neq y\right]\right] \leq 2\mathsf{OPT}_{\mathcal{H}} + \varepsilon$$

where $\mathrm{MAJ}(h_1, \ldots, h_{T(\varepsilon)})$ *shows the majority-vote of predictors* $h_1, \ldots, h_{T(\varepsilon)}$.

Remark 1. *In the approach proposed by Xiang et al. (2022), the robust loss with respect to the (exponentially many) patches is upper bounded by the robust loss with respect to the (polynomially many) masking operations. Therefore, Theorem 1 implies that the robust loss against patches is at most* $2\mathsf{OPT}_{\mathcal{H}} + \varepsilon$.

3.1 Comparison with prior related work

As presented, Feige et al. (2015) only considered *finite* hypothesis classes $\mathcal{H}$ and provided generalization guarantees depending on $\log |\mathcal{H}|$. On the other hand, we consider here infinite classes $\mathcal{H}$ with bounded VC dimension and provide tighter robust generalization bounds (see Theorem 1). We would also like to highlight another difference. Given an output of $h_1, \ldots, h_T$ from Algorithm 1, the guarantee provided by Feige et al. (2015) is on average and does not exactly capture the notion of robust loss i.e. the loss on input x is $\sup_{z \in \mathcal{U}(x)} \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}[h_t(z) \neq y]$ (Lemma 3 states their result). We emphasize that this is different from the *robust loss* guarantee that we obtain in Theorem 1 for a single classifier, i.e. the loss on input x is captured as $\sup_{z \in \mathcal{U}(x)} \mathbb{1}[\mathrm{MAJ}(h_1, \ldots, h_T)(z) \neq y]$. In particular, unlike the guarantee provided by Feige et al. (2015) in which the adversary chooses $z \in \mathcal{U}(x)$ and then we can probabilistically choose a classifier to classify it, to implement the Patch-Cleanser reduction we need a single classifier that is *simultaneously* correct on *all* $z \in \mathcal{U}(x)$. Because of the difference in guarantees derived, we incur a multiplicative factor of 2 compared with their bound.

The robust learning guarantee (Attias et al., 2022, Theorem 2) assumes access to a *robust* ERM oracle, which minimizes the robust loss on the training dataset. On the other hand, at the expense of higher sample complexity, we provide a robust learning guarantee using only an ERM oracle which is a more common and simpler assumption in the challenging *non-realizable* setting. Prior work due to Montasser et al. (2020) considered using an ERM oracle for robust learning but only in the simpler realizable setting (when $\mathsf{OPT}_{\mathcal{H}} = 0$).

3.2 Proof of Theorem 1

Before proceeding with the proof of Theorem 1, we describe at a high-level the proof strategy. The main insight is to solve a finite zero-sum game. In particular, our goal is to find a mixed-strategy over the hypothesis class that is

approximately close to the value of the game:

$$\mathsf{OPT}_{S,\mathcal{H}} \triangleq \min_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m} \max_{z_i\in\mathcal{U}(x_i)} \mathbb{1}\left[h(z_i)\neq y_i\right].$$

We observe that Algorithm 1 due to Feige et al. (2015) solves a similar finite zero-sum game (see Lemma 3), and then we relate it to the value of the game we are interested in (see Lemma 2). Combined together, this only establishes that we can minimize the robust loss on the empirical dataset using an ERM oracle. We then appeal to uniform convergence guarantees for the robust loss in Lemma 4 to show that, with a large enough training data, our output predictor achieves robust risk that is close to the value of the game.

Lemma 2. *For any dataset $S = \{(x_1,y_1),\dots,(x_m,y_m)\} \in (\mathcal{X}\times\mathcal{Y})^m$ with corresponding weights $p_1,\cdots,p_m = 1/m$,*

$$\begin{aligned}
&\mathsf{OPT}_{S,\mathcal{H}} = \min_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m} \max_{z_i\in\mathcal{U}(x_i)} \mathbb{1}\left[h(z_i)\neq y_i\right] \\
&\geq \min_{Q\in\Delta(\mathcal{H})} \max_{\substack{P_1\in\Delta(\mathcal{U}(x_1)),\\ \dots,\\ P_m\in\Delta(\mathcal{U}(x_m))}} \frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{z_i\sim P_i}\mathop{\mathbb{E}}_{h\sim Q} \mathbb{1}\left[h(z_i)\neq y_i\right]
\end{aligned}$$

Lemma 3 (Feige, Mansour, and Schapire (2015))**.** *For any data set $S = \{(x_1,y_1),\dots,(x_m,y_m)\} \in (\mathcal{X}\times\mathcal{Y})^m$ with corresponding weights $p_1,\cdots,p_m = 1/m$, running Algorithm 1 for T rounds produces a mixed-strategy $\hat{Q} = \frac{1}{T}\sum_{t=1}^{T} h_t \in \Delta(\mathcal{H})$ satisfying:*

$$\begin{aligned}
&\max_{\substack{P_1\in\Delta(\mathcal{U}(x_1)),\\ \dots,\\ P_m\in\Delta(\mathcal{U}(x_m))}} \frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{z_i\sim P_i} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}\left[h_t(z_i)\neq y_i\right] \\
&\leq \min_{Q\in\Delta(\mathcal{H})} \max_{\substack{P_1\in\Delta(\mathcal{U}(x_1)),\\ \dots,\\ P_m\in\Delta(\mathcal{U}(x_m))}} \frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{z_i\sim P_i}\mathop{\mathbb{E}}_{h\sim Q} \mathbb{1}\left[h(z_i)\neq y_i\right] + \\
&2\sqrt{\frac{\ln k}{T}}
\end{aligned}$$

Lemma 4 (VC Dimension for the Robust Loss (Attias et al., 2022))**.** *For any class $\mathcal{H}$ and any $\mathcal{U}$ such that $\sup_{x\in\mathcal{X}} |\mathcal{U}(x)| \leq k$, denote the robust loss class of $\mathcal{H}$ with respect to $\mathcal{U}$ by*

$$\mathcal{L}^{\mathcal{U}}_{\mathcal{H}} = \{(x,y)\mapsto \max_{z\in\mathcal{U}(x)} \mathbb{1}\left[h(z)\neq y\right] : h\in\mathcal{H}\}.$$

Then, it holds that $\mathrm{vc}(\mathcal{L}^{\mathcal{U}}_{\mathcal{H}}) \leq O(\mathrm{vc}(\mathcal{H})\log(k))$.

We are now ready to proceed with the proof of Theorem 1.

Proof of Theorem 1. Let $S\sim\mathcal{D}^m$ be an iid sample from $\mathcal{D}$, where the size of the sample m will be determined later. By invoking Lemma 3 and Lemma 2, we observe that running Algorithm 1 on S with corresponding weights $p_1,\cdots,p_m = 1/m$ for T rounds, produces $h_1,\dots,h_T$ satisfying

$$\max_{\substack{P_1\in\Delta(\mathcal{U}(x_1)),\\ \dots,\\ P_m\in\Delta(\mathcal{U}(x_m))}} \frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{z_i\sim P_i} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}\left[h_t(z_i)\neq y_i\right] \leq \mathsf{OPT}_{S,\mathcal{H}} + \frac{\varepsilon}{4}$$

Next, the average robust loss for the majority-vote predictor $\mathrm{MAJ}(h_1,\dots,h_T)$ can be bounded from above as follows:

$$\begin{aligned}
&\frac{1}{m}\sum_{i=1}^{m} \max_{z_i\in\mathcal{U}(x_i)} \mathbb{1}\left[\mathrm{MAJ}(h_1,\dots,h_T)(z_i)\neq y_i\right] \\
&\leq \frac{1}{m}\sum_{i=1}^{m} \max_{z_i\in\mathcal{U}(x_i)} 2\mathop{\mathbb{E}}_{t\sim[T]} \mathbb{1}\left[h_t(z_i)\neq y_i\right] \\
&= 2\frac{1}{m}\sum_{i=1}^{m} \max_{z_i\in\mathcal{U}(x_i)} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}\left[h_t(z_i)\neq y_i\right] \\
&\leq 2 \max_{\substack{P_1\in\Delta(\mathcal{U}(x_1)),\\ \dots,\\ P_m\in\Delta(\mathcal{U}(x_m))}} \frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{z_i\sim P_i} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}\left[h_t(z_i)\neq y_i\right] \\
&\leq 2\mathsf{OPT}_{S,\mathcal{H}} + \frac{\varepsilon}{2}.
\end{aligned}$$

In the second line above, the factor 2 shows up since for any arbitrary example (z,y), if at least half the predictors make a mistake then the majority-vote is wrong, and otherwise it is correct. The factor 2 is used as a correction so that RHS is bigger than LHS, where the edge case is exactly when half the predictors make a mistake.

Next, we invoke Lemma 4 to obtain a uniform convergence guarantee on the robust loss. In particular, we apply Lemma 4 on the *convex-hull* of $\mathcal{H}$: $\mathcal{H}^T = \{\mathrm{MAJ}(h_1,\dots,h_T) : h_1,\dots,h_T\in\mathcal{H}\}$. By a classic result due to Blumer et al. (1989), it holds that $\mathrm{vc}(\mathcal{H}^T) = O(\mathrm{vc}(\mathcal{H})T\ln T)$. Combining this with Lemma 4 and plugging-in the value of $T = \frac{32\ln k}{\varepsilon^2}$, we get that the VC dimension of the robust loss class of $\mathcal{H}^T$ is bounded from above by

$$\mathrm{vc}(\mathcal{L}^{\mathcal{U}}_{\mathcal{H}^T}) \leq O\left(\frac{\mathrm{vc}(\mathcal{H})(\ln k)^2}{\varepsilon^2}\ln\left(\frac{\ln k}{\varepsilon^2}\right)\right).$$

Finally, using Vapnik's "General Learning" uniform convergence (Vapnik, 1982), with probability at least $1-\delta$ over $S\sim\mathcal{D}^m$ where $m = O\left(\frac{\mathrm{vc}(\mathcal{H})(\ln k)^2}{\varepsilon^4}\ln\left(\frac{\ln k}{\varepsilon^2}\right) + \frac{\ln(1/\delta)}{\varepsilon^2}\right)$, it holds that

$$\begin{aligned}
&\forall f\in\mathcal{H}^T : \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\max_{z\in\mathcal{U}(x)} \mathbb{1}\left[f(z)\neq y\right]\right] \\
&\leq \frac{1}{m}\sum_{i=1}^{m} \max_{z_i\in\mathcal{U}(x_i)} \mathbb{1}\left[f(z_i)\neq y_i\right] + \frac{\varepsilon}{4}
\end{aligned}$$

This also applies to the particular output $\mathrm{MAJ}(h_1,\dots,h_T)$

of Algorithm 1, and thus

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\max_{z\in\mathcal{U}(x)} \mathbb{1}\left[\mathrm{MAJ}(h_1,\dots,h_{T(\varepsilon)})(z)\neq y\right]\right]\\
&\leq \frac{1}{m}\sum_{i=1}^{m}\max_{z_i\in\mathcal{U}(x_i)} \mathbb{1}\left[\mathrm{MAJ}(h_1,\dots,h_T)(z_i)\neq y_i\right]+\frac{\varepsilon}{4}\\
&\leq 2\mathsf{OPT}_{S,\mathcal{H}}+\frac{\varepsilon}{2}+\frac{\varepsilon}{4}.
\end{aligned}$$

Finally, by applying a standard Chernoff-Hoeffding concentration inequality, we get that $\mathsf{OPT}_{S,\mathcal{H}} \leq \mathsf{OPT}_{\mathcal{H}}+\frac{\varepsilon}{8}$. Combining this with the above inequality concludes the proof. □

4 MULTI-ROBUSTNESS GUARANTEES ON A SET OF GROUPS

In this section, we propose a boosting algorithm that learns a predictor with a low robust loss on a collection of subgroups simultaneously. First, we consider the case of disjoint groups and present our training-time algorithm for this case in Section 4.2. Section 4.4 provides generalization guarantees. In Section 4.3, we show a reduction from overlapping groups to disjoint groups. In the following, first we formalize the notions of robust loss on a specific group and multi-robustness.

When the training dataset S is partitioned into g groups $\mathcal{G}=\{G_1,\dots,G_g\}$, the empirical robust loss of a predictor h on group G_j is defined as follows:

$$\ell_j^{\text{rob}}(h)=\frac{1}{|G_j|}\sum_{(x,y)\in G_j}\max_{z\in\mathcal{U}(x)}\mathbb{1}[h(z)\neq y] \tag{4}$$

The learning benchmark that we compete with on a dataset S for the robust loss on each group is $\mathsf{OPT}^S_{\max}$ that is defined as follows:

$$\mathsf{OPT}^S_{\max}=\min_{h\in\mathcal{H}}\max_{j\in[g]}\frac{1}{|G_j|}\sum_{(x,y)\in G_j}\max_{z\in\mathcal{U}(x)}\mathbb{1}[h(z)\neq y] \tag{5}$$

Definition 1 (Multi-Robustness)**.** *A hypothesis h is multi-robust on a dataset S if it achieves the following guarantee:*

$$\max_{j\in[g]}\frac{1}{|G_j|}\sum_{(x,y)\in G_j}\max_{z\in\mathcal{U}(x)}\mathbb{1}[h(z)\neq y]\leq \mathsf{OPT}^S_{\max}+\varepsilon$$

Definition 2 (β-Multi-Robustness)**.** *A hypothesis h is β-multi-robust on a dataset S if it achieves the following guarantee:*

$$\max_{j\in[g]}\frac{1}{|G_j|}\sum_{(x,y)\in G_j}\max_{z\in\mathcal{U}(x)}\mathbb{1}[h(z)\neq y]\leq \beta(\mathsf{OPT}^S_{\max}+\varepsilon)$$

Definition 3 (Multi-Robustness on Average)**.** *A set of hypotheses $\mathcal{H}'=\{h_1,\dots,h_T\}$ is multi-robust on a dataset S on average if the the following property holds:*

$$\frac{1}{T}\max_{j\in[g]}\sum_{t=1}^{T}\ell_j^{rob}(h_t)\leq \mathsf{OPT}^S_{\max}+\varepsilon$$

Remark 2. *Definition 1 is a stronger notion of multi-robustness compared to Definition 3.*

Summary of Results. Section 4.2 investigates the case of disjoint groups and proposes a two-layer boosting algorithm (Algorithm 2) that achieves multi-robustness on the training dataset S. First, we show that $\mathcal{H}'=\{h_1,\dots,h_T\}$ returned by Algorithm 2 is multi-robust on average (Theorem 8). Theorem 10 exhibits that the majority-vote classifier over $\mathcal{H}'$, i.e. $\mathrm{MAJ}(h_1,\dots,h_T)$, obtains β-multi-robustness for $\beta=2$. We remark that although Theorem 8 achieves a tighter upper bound on the multi-robustness guarantee, Theorem 10 gives a guarantee for the stronger notion of multi-robustness. In Section 4.3, we show a reduction from overlapping groups to disjoint groups. Section 4.4 provides generalization guarantees for both notions of average multi-robustness and β-multi-robustness.

4.1 Comparison to Prior Work on Multi-group Learning

Rothblum and Yona (2021) study agnostic multi-group *PAC learning* and their algorithm returns a hypothesis h such that for each group G_j in a collection of groups $\mathcal{G}$:

$$\mathbb{E}\left[\ell(h(x),y)|x\in G_j\right]\leq \min_{h_{G_j}\in\mathcal{H}}\mathbb{E}\left[\ell(h_{G_j}(x),y)|x\in G_j\right]$$

That is, the hypothesis h must compete against a hypothesis $h_{G_j}\in\mathcal{H}$ trained specifically to minimize the error over the group $G_j\in\mathcal{G}$, for every group in the collection. *However, their results do not extend to the case of robust loss.* In contrast, in our notion of multi-robustness loss that holds for the *more challenging objective of robust learning*, our benchmark is weaker (Definition 1). We leave it as an open question to study whether our upper bounds for the robust loss over a collection of groups can be strengthened.

4.2 Boosting algorithm achieving multi-robustness guarantees:

In this section, we present Algorithm 2 that obtains multi-robustness guarantees on a set of *disjoint* groups. The algorithm follows the idea proposed by Freund and Schapire (1996) that obtains boosting by playing a repeated game. Initially a sample set $S=\{(x_1,y_1),\dots,(x_m,y_m)\}$ partitioned into a set of disjoint groups $\mathcal{G}=\{G_1,\dots,G_g\}$ is received as input. P_j^t shows the normalized weight of group G_j in step t. Initially, for each group G_j, $P_j^t=1/g$. In

each round t, the weight of each group gets split between its examples equally: $p_i = P_j^t/|G_j|$ where $(x_i, y_i) \in G_j$. Subsequently, an oracle call is made to Algorithm 1 with sample weights $p_1, \ldots, p_m$. Lemma 6 shows at each iteration t, Algorithm 1 returns a hypothesis h_t such that its average robust loss across the groups is at most $\mathsf{OPT}^S_{\max} + \varepsilon$. In the next iteration $t+1$, for each group G_j, the weights of examples in G_j get decreased by a multiplicative factor of $1 - \delta m_j^{\text{rob}}(h_t)$ where $m_j^{\text{rob}}(h_t) = 1 - \ell_j^{\text{rob}}(h_t)$ and $\delta = \sqrt{\ln g / T}$. Theorem 8 exhibits that after $T = \mathcal{O}(\ln g/\varepsilon^2)$ rounds, Algorithm 2 outputs a set of hypotheses $\mathcal{H}' = \{h_1, \ldots, h_T\}$ such that for each group G_j the average multi-robustness guarantee is obtained, i.e., $\frac{1}{T}\sum_{t=1}^T \ell_j^{\text{rob}}(h_t) \le \mathsf{OPT}^S_{\max} + \varepsilon$. Theorem 10 provides that $\text{MAJ}(h_1, \ldots, h_t)$ achieves β-multi-robustness guarantee for $\beta = 2$.

Algorithm 2 Boosting Algorithm Achieving Multi-Robustness

Input training dataset $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ partitioned into a set of groups $\{G_1, \cdots, G_g\}$
Initially, $\forall 1 \le j \le g : P_j^t = 1/g$
for $t = 1, \ldots, T$ **do**
  $p_i = P_j^t/|G_j|$ where $(x_i, y_i) \in G_j$
  Call Algorithm 1 on S with weights $(p_1, \ldots, p_m)$ for $T' = \frac{36 \ln k}{\varepsilon^2}$ rounds.
  Update P_j^t, for all $j \in [g]$:

$$P_j^{t+1} = \frac{P_j^t \cdot \left(1 - \delta m_j^{\text{rob}}(h_t)\right)}{Z_t}$$

  where $m_j^{\text{rob}}(h_t) = 1 - \ell_j^{\text{rob}}(h_t)$, Z_t is a normalization factor, and $\delta = \sqrt{\frac{\ln g}{T}}$.
Output $\mathcal{H}' = \{h_1, \cdots, h_T\}$

Remark 3. *We remark that the output of Algorithm 2 is a set of majority-vote classifiers over $\mathcal{H}$:*

$$\mathcal{H}' = \Big\{\text{MAJ}(h_{1,1}, \ldots, h_{1,T'}), \ldots, \text{MAJ}(h_{T,1}, \ldots, h_{T,T'}) : \forall i \in [T], \forall j \in [T'], h_{i,j} \in \mathcal{H}\Big\}$$

Before proving the multi-robustness guarantees, we show that Lemma 6 holds. In order to prove that Lemma 6 holds, first we show in Lemma 5 that an extension of Lemma 3 holds when $p_1, \cdots, p_m$ are arbitrary weights such that $\sum_{i=1}^m p_i = 1$. Next, we restate the guarantee of the Multiplicative Weights algorithm that is a generalization of *Weighted Majority* algorithm (Littlestone and Warmuth, 1994) and is equivalent to *Hedge* developed by Freund and Schapire (1997).

Lemma 5 (Extension to general weights)**.** *For any dataset $S = \{(x_1, y_1), \ldots, (x_m, y_m)\} \in (\mathcal{X} \times \mathcal{Y})^m$ and any corresponding weights $p_1, \ldots, p_m > 0$ such that $\sum_{i=1}^m p_i = 1$, running Algorithm 1 for T rounds produces a mixed-strategy $\hat{Q} = \frac{1}{T}\sum_{t=1}^T h_t \in \Delta(\mathcal{H})$ satisfying:*

$$\max_{\substack{P_1 \in \Delta(\mathcal{U}(x_1)), \\ \ldots, \\ P_m \in \Delta(\mathcal{U}(x_m))}} \sum_{i=1}^m p_i \cdot \mathop{\mathbb{E}}_{z_i \sim P_i} \frac{1}{T}\sum_{t=1}^T \mathbb{1}\left[h_t(z_i) \ne y_i\right]$$

$$\le \min_{Q \in \Delta(\mathcal{H})} \max_{\substack{P_1 \in \Delta(\mathcal{U}(x_1)), \\ \ldots, \\ P_m \in \Delta(\mathcal{U}(x_m))}} \sum_{i=1}^m p_i \cdot \mathop{\mathbb{E}}_{z_i \sim P_i} \mathop{\mathbb{E}}_{h \sim Q} \mathbb{1}\left[h(z_i) \ne y_i\right] + 2\sqrt{\frac{\ln k}{T}}$$

Lemma 6. *In each round t of Algorithm 2, by making an oracle-call to Algorithm 1 after $T' = \frac{4 \ln k}{\varepsilon^2}$ rounds, a hypothesis h_t is outputted such that $\mathbb{E}_{j \sim P^t}[\ell_j^{rob}(h_t)] = \sum_{j \in [g]} P_j^t \ell_j^{rob}(h_t) \le \mathsf{OPT}^S_{\max} + \varepsilon$.*

Theorem 7 (Mutiplicative Weights Algorithm (Kale, 2007))**.** *For any sequence of costs of experts $\mathbf{m}_1, \cdots, \mathbf{m}_T$ revealed by nature where all the costs are in $[0, 1]$, the sequence of mixed strategies $\mathbf{p}_1, \cdots, \mathbf{p}_T$ produced by the Multiplicative Weights algorithm satisfies:*

$$\sum_{t=1}^T \mathbf{m}_t \cdot \mathbf{p}_t \le (1+\delta) \min_{\mathbf{p}} \sum_{t=1}^T \mathbf{m}_t \cdot \mathbf{p} + \frac{\ln n}{\delta}$$

where n is the number of experts.

Theorem 8. *When $T = \mathcal{O}(\frac{\ln g}{\varepsilon^2})$, Algorithm 2 computes a set of hypotheses $\mathcal{H}' = \{h_1, \cdots, h_T\}$, such that for each group G_j, $\frac{1}{T}\sum_{t=1}^T \ell_j^{rob}(h_t) \le \mathsf{OPT}^S_{\max} + \varepsilon$.*

Proof. In each iteration t, we define average loss and reward terms as follows:

$$L(P^t, h_t) = \mathop{\mathbb{E}}_{j \sim P_t}\left[\ell_j^{\text{rob}}(h_t)\right] = \sum_{j \in [g]} P_j^t \ell_j^{\text{rob}}(h_t),$$

$$M(P^t, h_t) = \mathop{\mathbb{E}}_{j \sim P_t}\left[m_j^{\text{rob}}(h_t)\right]$$

Substituting $\ell_j^{\text{rob}}(h_t) = 1 - m_j^{\text{rob}}(h_t)$ provides:

$$M(P^t, h_t) = \sum_{j \in [g]} P_j^t (1 - \ell_j^{\text{rob}}(h_t)) = 1 - \sum_{j \in [g]} P_j^t \ell_j^{\text{rob}}(h_t) = 1 - L(P^t, h_t)$$

Now by setting $T = \frac{9 \ln g}{\varepsilon^2}$ which implies that $\delta = \sqrt{\frac{\ln g}{T}} = \frac{\varepsilon}{3}$, and by using the guarantee of Theorem 7, the following

bound is obtained.

$$\frac{1}{T}\sum_{t=1}^{T} M(P^t, h_t) \leq \frac{(1+\delta)}{T}\min_{j\in[g]}\sum_{t=1}^{T} M(j, h_t) + \frac{\ln g}{\delta T}$$
$$\rightarrow \frac{1}{T}\sum_{t=1}^{T} M(P^t, h_t) \leq \frac{1}{T}\min_{j\in[g]}\sum_{t=1}^{T} M(j, h_t) + \delta + \frac{\ln g}{\delta T}$$
$$\rightarrow \frac{1}{T}\sum_{t=1}^{T} M(P^t, h_t) \leq \frac{1}{T}\min_{j\in[g]}\sum_{t=1}^{T} M(j, h_t) + \frac{2\varepsilon}{3}$$

where $M(j, h_t)$ is the reward term when the whole probability mass is concentrated on group G_j. Therefore for each group $j \in [g]$:

$$\frac{1}{T}\sum_{t=1}^{T} M(j, h_t) \geq \frac{1}{T}\sum_{t=1}^{T} M(P^t, h_t) - \frac{2\varepsilon}{3} \tag{6}$$

Lemma 6 provides that in each iteration t, $L(P^t, h_t) \leq \mathsf{OPT}^S_{\max} + \varepsilon/3$ given that Algorithm 1 is executed for $T' = \frac{36\ln k}{\varepsilon^2}$ rounds. Thus, at each iteration t, $M(P^t, h_t) \geq 1 - (\mathsf{OPT}^S_{\max} + \varepsilon/3)$. Therefore, $\frac{1}{T}\sum_{t=1}^{T} M(P^t, h_t) \geq 1 - (\mathsf{OPT}^S_{\max} + \varepsilon/3)$; combining with Equation 6 implies that:

$$\frac{1}{T}\sum_{t=1}^{T} M(j, h_t) \geq \frac{1}{T}\sum_{t=1}^{T} M(P^t, h_t) - \frac{2\varepsilon}{3}$$
$$\geq 1 - (\mathsf{OPT}^S_{\max} + \frac{\varepsilon}{3}) - \frac{2\varepsilon}{3} = 1 - (\mathsf{OPT}^S_{\max} + \varepsilon)$$

Plugging in the definition of $L(P^t, h_t)$ implies that:

$$\frac{1}{T}\sum_{t=1}^{T} L(j, h_t) \leq \mathsf{OPT}^S_{\max} + \varepsilon$$

Which concludes the proof. □

Corollary 9. *Theorem 8 implies that if for each example a predictor is picked uniformly at random from $\mathcal{H}'$ to predict its label, then for each group $G_j \in \mathcal{G}$, the expected robust loss is at most $\mathsf{OPT}^S_{\max} + \varepsilon$.*

Theorem 10. *When $T = \mathcal{O}(\frac{\ln g}{\varepsilon^2})$, Algorithm 2 computes a set of hypotheses $\mathcal{H}' = \{h_1, \ldots, h_T\}$ such that for each group G_j, $\ell^{rob}_j(\mathrm{MAJ}(h_1, \cdots, h_T)) \leq 2(\mathsf{OPT}^S_{\max} + \varepsilon)$.*

Proof. By Theorem 8, after $T = \mathcal{O}(\frac{\ln g}{\varepsilon^2})$ rounds, for each group G_j, $\frac{1}{T}\sum_{t=1}^{T} \ell^{\mathrm{rob}}_j(h_t) \leq \mathsf{OPT}^S_{\max} + \varepsilon$. Therefore, the total number of robustness mistakes on G_j across all the classifiers $h_1, \cdots, h_T$ is at most $T(\mathsf{OPT}^S_{\max} + \varepsilon)|G_j|$ which is equal to $T/2 \cdot 2(\mathsf{OPT}^S_{\max} + \varepsilon)|G_j|$.

Therefore, the fraction of examples in G_j that at least $T/2$ of the classifiers in $h_1, \cdots h_T$ make a robustness mistake on is at most $2(\mathsf{OPT}^S_{\max} + \varepsilon)$. Hence, the fraction of examples in G_j that are not robustly classified by the majority-vote classifier is at most $2(\mathsf{OPT}^S_{\max} + \varepsilon)$. □

4.3 Reduction from overlapping groups to disjoint groups

When the groups are overlapping, we reduce it to the case of disjoint groups. The reduction is as follows: for an input instance $\mathcal{I}(\mathcal{G} = \{G_1, \ldots, G_g\}, S)$ of overlapping groups, create a new instance $\mathcal{I}'(\mathcal{G}' = \{G'_1, \ldots, G'_g\}, S')$ as follows. Initially, for all $G'_j \in \mathcal{G}'$, G'_j is an empty set. For each example $(x_i, y_i) \in S$ that belongs to a set of groups $\mathcal{G}_i = \{G_{i,1}, \cdots, G_{i,|\mathcal{G}_i|}\} \subseteq \mathcal{G}$ in $\mathcal{I}$, create identical copies of (x_i, y_i) and assign each copy including the original example to exactly one of the groups in $\mathcal{G}'_i = \{G'_{i,1}, \cdots, G'_{i,|\mathcal{G}'_i|}\}$. Now we have an instance $\mathcal{I}'$ with disjoint groups. By executing Algorithm 2 on $\mathcal{I}'$, it returns a predictor h that achieves a β-multi-robustness guarantee. First, we argue that if h is used on $\mathcal{I}$, it achieves a multi-robustness guarantee of $\beta \cdot (\mathsf{OPT}^{\mathcal{I}'}_{\max} + \varepsilon)$. This is the case since either h makes a robustness mistake on all copies of an example or does not make any robustness mistakes on any of them. Next, we show that $\mathsf{OPT}^{\mathcal{I}'}_{\max} \leq \mathsf{OPT}^{\mathcal{I}}_{\max}$. Consider a predictor $h^* \in \mathcal{H}$ that achieves multi-robustness of $\mathsf{OPT}^{\mathcal{I}}_{\max}$ on $\mathcal{I}$. If h^* is used on $\mathcal{I}'$, for each example $(x, y) \in S$ that h^* has zero robust loss on, it does not make any mistakes on any of its copies in $\mathcal{I}'$. Additionally, if h^* makes a robustness mistake on (x, y), then it makes a robustness mistake on all its copies in $\mathcal{I}'$. Thus, h^* achieves a multi-robustness guarantee of $\mathsf{OPT}^{\mathcal{I}}_{\max}$ on $\mathcal{I}'$. Therefore, $\mathsf{OPT}^{\mathcal{I}'}_{\max} \leq \mathsf{OPT}^{\mathcal{I}}_{\max}$, and a $\beta(\mathsf{OPT}^{\mathcal{I}'}_{\max} + \varepsilon)$ multi-robustness guarantee on $\mathcal{I}$ implies $\beta(\mathsf{OPT}^{\mathcal{I}}_{\max} + \varepsilon)$ multi-robustness. A similar argument holds for the average multi-robustness guarantee.

Remark 4. *When $|\mathcal{G}|$ is large, this reduction becomes computationally inefficient, since in the worst case, the number of samples gets increased by a multiplicative factor of $|\mathcal{G}|$. However, this reduction is equivalent to keeping only one copy of each sample $(x_i, y_i) \in S$ and when executing Algorithm 2, in each iteration t, assigning it a weight of $p_i = \sum_{j\in[g]:(x_i,y_i)\in G_j} P^t_j/|G_j|$.*

4.4 Generalization Guarantees

In this section, we derive generalization guarantees for multi-robustness. First, Lemma 11 shows how to bound the VC-Dimension of the intersection of robust loss and groups. We can then invoke this Lemma to get uniform convergence guarantees that will allow us to get concentration for the conditional robust loss across groups (see Definition 1).

Lemma 11 (VC Dimension of Intersection of Robust Loss and Groups)**.** *For any class $\mathcal{H}$, any perturbation set $\mathcal{U}$, and any group class $\mathcal{G}$, denote the intersection function class by*

$$\mathcal{F}^{\mathcal{U}}_{\mathcal{H},\mathcal{G}} \triangleq \{(x, y) \mapsto \max_{z\in\mathcal{U}(x)} \mathbb{1}\left[h(z) \neq y\right] \wedge \mathbb{1}[x \in G_j] :$$
$$h \in \mathcal{H}, G_j \in \mathcal{G}\}$$

Then, it holds that $\mathrm{vc}(\mathcal{F}^{\mathcal{U}}_{\mathcal{H},\mathcal{G}}) \leq \tilde{O}\left(\mathrm{vc}(\mathcal{L}^{\mathcal{U}}_{\mathcal{H}}) + \mathrm{vc}(\mathcal{G})\right)$.

Theorem 12 (Generalization guarantees for average multi-robustness)**.** *With* $T = \mathcal{O}(\ln g/\varepsilon^2)$ *and* $m = \tilde{O}\left(\frac{\mathrm{vc}(\mathcal{H})\ln^2(k)}{\varepsilon^4} + \frac{\mathrm{vc}(\mathcal{G})+\ln(1/\delta)}{\varepsilon^2}\right)$*, Algorithm 2 computes a set of hypotheses* $\mathcal{H}' = \{h_1, \ldots, h_T\}$*, such that* $\forall G_j \in \mathcal{G}$*,*

$$\frac{1}{T}\sum_{t=1}^{T} \Pr_{(x,y)\in\mathcal{D}}\left[\exists z \in \mathcal{U}(x) : h_t(z) \neq y \mid x \in G_j\right]$$
$$\leq \left(1 + \frac{\varepsilon}{\Pr_{\mathcal{D}}(x \in G_j)}\right)\left(\mathsf{OPT}^{S}_{\max} + \varepsilon\right) + \frac{\varepsilon}{\Pr_{\mathcal{D}}(x \in G_j)}$$

Theorem 13 (Generalization guarantees for β-multi-robustness)**.** *With* $T = \mathcal{O}(\ln g/\varepsilon^2)$*,* $m = \tilde{O}\left(\frac{\mathrm{vc}(\mathcal{H})\ln(g)\ln^2(k)}{\varepsilon^6} + \frac{\mathrm{vc}(\mathcal{G})+\ln(1/\delta)}{\varepsilon^2}\right)$*, and* $\beta = 2$*, Algorithm 2 computes a set of hypotheses* $\mathcal{H}' = \{h_1, \ldots, h_T\}$*, such that* $\forall G_j \in \mathcal{G}$*,*

$$\Pr_{(x,y)\in\mathcal{D}}\left[\exists z \in \mathcal{U}(x) : \mathrm{MAJ}(h_1, \ldots, h_T)(z) \neq y \mid x \in G_j\right]$$
$$\leq \left(1 + \frac{\varepsilon}{\Pr_{\mathcal{D}}(x \in G_j)}\right)\left(\beta(\mathsf{OPT}^{S}_{\max} + \varepsilon)\right) + \frac{\varepsilon}{\Pr_{\mathcal{D}}(x \in G_j)}$$

Remark 5. *In Section A.9, we show how to achieve generalization guarantees in terms of* $\mathsf{OPT}^{\mathcal{D}}_{\max}$ *instead of* $\mathsf{OPT}^{S}_{\max}$*.*

5 CONCLUSION

We exhibited an example showing how using ERM on an augmented dataset to learn a robust classifier can fail when the examples are robustly un-realizable. Next, we provided a "boosting-style" algorithm that uses ERM and obtains strong robust learning guarantees in the non-realizable regime. This work provides theoretical evidence that our existing methods of learning accurate classifiers i.e. ERM, can be modified effectively to learn robust classifiers even in the agnostic robust regime. Next, we introduced a new multi-robustness objective to obtain robustness guarantees simultaneously across a collection of subgroups. We showed this objective can be achieved by adding a second layer of boosting to the first algorithm.

Adversarial examples exist for many types of classifiers but are especially salient with modern neural-based vision methods. However, due to the large capacity of these networks, it is not clear that they would benefit from boosting. Therefore, the fact that our algorithms rely on boosting should not be interpreted as a firm recommendation to use boosting with neural networks, but instead as a theoretical proof-of-concept that plain ERM can be used to learn robust models, given the right algorithmic scheme, especially if such a scheme can reduce the effective number of perturbations available to the adversary.

Acknowledgements

This work was supported in part by the National Science Foundation under grants CCF-2212968 and ECCS-2216899, by the Simons Foundation under the Simons Collaboration on the Theory of Algorithmic Fairness, and by the Defense Advanced Research Projects Agency under cooperative agreement HR00112020003. The views expressed in this work do not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred. Approved for public release; distribution is unlimited. This work was done when OM was a PhD student at the Toyota Technological Institute at Chicago.

References

Idan Attias, Aryeh Kontorovich, and Yishay Mansour. Improved generalization bounds for adversarially robust learning. *Journal of Machine Learning Research*, 23 (175):1–31, 2022.

A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery*, 36 (4):929–965, 1989.

Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

Ping-yeh Chiang, Renkun Ni, Ahmed Abdelkader, Chen Zhu, Christoph Studer, and Tom Goldstein. Certified defenses for adversarial patches. *CoRR*, abs/2003.06693, 2020. URL https://arxiv.org/abs/2003.06693.

Uriel Feige, Yishay Mansour, and Robert E. Schapire. Learning and inference in the presence of corrupted inputs. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, volume 40 of *JMLR Workshop and Conference Proceedings*, pages 637–657. JMLR.org, 2015. URL http://proceedings.mlr.press/v40/Feige15.html.

Yoav Freund and Robert E Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the ninth annual conference on Computational learning theory*, pages 325–332, 1996.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55 (1):119–139, 1997.

Ira Globus-Harris, Michael Kearns, and Aaron Roth. Beyond the frontier: Fairness without accuracy loss. *CoRR*, abs/2201.10408, 2022. URL https://arxiv.org/abs/2201.10408.

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

Parikshit Gopalan, Lunjia Hu, Michael P Kim, Omer Reingold, and Udi Wieder. Loss minimization through the

lens of outcome indistinguishability. *arXiv preprint arXiv:2210.08649*, 2022.

Úrsula Hébert-Johnson, Michael P. Kim, Omer Reingold, and Guy N. Rothblum. Calibration for the (computationally-identifiable) masses. *CoRR*, abs/1711.08513, 2017. URL http://arxiv.org/abs/1711.08513.

Satyen Kale. *Efficient algorithms using the multiplicative weights update method*. Princeton University, 2007.

Danny Karmon, Daniel Zoran, and Yoav Goldberg. Lavan: Localized and visible adversarial noise. In *International Conference on Machine Learning*, pages 2507–2515. PMLR, 2018.

Michael J. Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2569–2577. PMLR, 2018. URL http://proceedings.mlr.press/v80/kearns18a.html.

Michael P. Kim, Amirata Ghorbani, and James Zou. Multiaccuracy: Black-box post-processing for fairness in classification. AIES '19, page 247–254, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450363242. doi: 10.1145/3306618.3314287. URL https://doi.org/10.1145/3306618.3314287.

Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108 (2):212–261, 1994.

Michael McCoyd, Won Park, Steven Chen, Neil Shah, Ryan Roggenkemper, Minjune Hwang, Jason Xinyu Liu, and David A. Wagner. Minority reports defense: Defending against adversarial patches. *CoRR*, abs/2004.13799, 2020. URL https://arxiv.org/abs/2004.13799.

Jan Hendrik Metzen and Maksym Yatsura. Efficient certified defenses against patch attacks on image classifiers. *CoRR*, abs/2102.04154, 2021. URL https://arxiv.org/abs/2102.04154.

Omar Montasser, Steve Hanneke, and Nathan Srebro. VC classes are adversarially robustly learnable, but only improperly. In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 2512–2530, Phoenix, USA, 25–28 Jun 2019. PMLR.

Omar Montasser, Steve Hanneke, and Nati Srebro. Reducing adversarially robust learning to non-robust PAC learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/a822554e5403b1d370db84cfbc530503-Abstract.html.

Guy N. Rothblum and Gal Yona. Multi-group agnostic PAC learnability. *CoRR*, abs/2105.09989, 2021. URL https://arxiv.org/abs/2105.09989.

Christopher Tosh and Daniel Hsu. Simple and near-optimal algorithms for hidden stratification and multi-group learning. *CoRR*, abs/2112.12181, 2021. URL https://arxiv.org/abs/2112.12181.

V. Vapnik. *Estimation of Dependencies Based on Empirical Data*. Springer-Verlag, New York, 1982.

Chong Xiang and Prateek Mittal. Patchguard++: Efficient provable attack detection against adversarial patches. *CoRR*, abs/2104.12609, 2021. URL https://arxiv.org/abs/2104.12609.

Chong Xiang, Arjun Nitin Bhagoji, Vikash Sehwag, and Prateek Mittal. Patchguard: Provable defense against adversarial patches using masks on small receptive fields. *CoRR*, abs/2005.10884, 2020. URL https://arxiv.org/abs/2005.10884.

Chong Xiang, Saeed Mahloujifar, and Prateek Mittal. Patchcleanser: Certifiably robust defense against adversarial patches for any image classifier. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 2065–2082, 2022.

Chenglin Yang, Adam Kortylewski, Cihang Xie, Yinzhi Cao, and Alan Yuille. Patchattack: A black-box texture-based attack with reinforcement learning. In *European Conference on Computer Vision*, pages 681–698. Springer, 2020.

Zhanyuan Zhang, Benson Yuan, Michael McCoyd, and David Wagner. Clipped bagnet: Defending against sticker attacks with clipped bag-of-features. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 55–61. IEEE, 2020.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Not Applicable]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A Supplementary Materials

A.1 Proof of Lemma 2

Proof. By definition of $\mathsf{OPT}_{S,\mathcal{H}}$, it follows that

$$\begin{aligned}
\mathsf{OPT}_{S,\mathcal{H}} &= \min_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m} \max_{z_i\in\mathcal{U}(x_i)} \mathbb{1}\left[h(z_i)\neq y_i\right] \\
&\geq \min_{h\in\mathcal{H}} \max_{z_1\in\mathcal{U}(x_1),\ldots,z_m\in\mathcal{U}(x_m)} \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}\left[h(z_i)\neq y_i\right] \\
&\geq \min_{Q\in\Delta(H)} \max_{z_1\in\mathcal{U}(x_1),\ldots,z_m\in\mathcal{U}(x_m)} \frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{h\sim Q} \mathbb{1}\left[h(z_i)\neq y_i\right] \\
&\geq \min_{Q\in\Delta(\mathcal{H})} \max_{\substack{P_1\in\Delta(\mathcal{U}(x_1)),\\ \ldots,\\ P_m\in\Delta(\mathcal{U}(x_m))}} \frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{z_i\sim P_i}\mathop{\mathbb{E}}_{h\sim Q} \mathbb{1}\left[h(z_i)\neq y_i\right].
\end{aligned}$$

□

A.2 Proof of Lemma 3

Proof. By the minimax theorem and (Feige, Mansour, and Schapire, 2015, Equation 3 and 9 in proof of Theorem 1), we have that

$$\begin{aligned}
&\max_{\substack{P_1\in\Delta(\mathcal{U}(x_1)),\\ \ldots,\\ P_m\in\Delta(\mathcal{U}(x_m))}} \sum_{i=1}^{m} \mathop{\mathbb{E}}_{z_i\sim P_i} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}\left[h_t(z_i)\neq y_i\right] \leq \\
&\min_{Q\in\Delta(\mathcal{H})} \max_{\substack{P_1\in\Delta(\mathcal{U}(x_1)),\\ \ldots,\\ P_m\in\Delta(\mathcal{U}(x_m))}} \mathop{\mathbb{E}}_{z_i\sim P_i}\mathop{\mathbb{E}}_{h\sim Q} \mathbb{1}\left[h(z_i)\neq y_i\right] \\
&+ 2\frac{\sqrt{\mathcal{L}^* m\ln k}}{T},
\end{aligned}$$

where $\mathcal{L}^* = \sum_{i=1}^{m} \max_{z\in\mathcal{U}(x_i)} \sum_{t=1}^{T} \mathbb{1}\left[h_t(z)\neq y\right]$. By observing that $\mathcal{L}^* \leq mT$ and dividing both sides of the inequality above by m, we arrive at the inequality stated in the lemma. □

A.3 Proof of Lemma 5

Proof. We generalize the argument in Feige, Mansour, and Schapire (2015) to accommodate the weights on the samples $p_1,\ldots,p_m$. Specifically, let

$$L_T^{ON} = \sum_{t=1}^{T}\sum_{i=1}^{m}\sum_{z\in\mathcal{U}(x_i)} p_i P^t(z,(x_i,y_i))\mathbb{1}\left[h_t(z)\neq y_i\right]$$

be the loss of Algorithm 1 after T rounds, and let

$$L^* = \max_{P}\sum_{t=1}^{T}\sum_{i=1}^{m}\sum_{z\in\mathcal{U}(x_i)} p_i P(z,(x_i,y_i))\mathbb{1}\left[h_t(z)\neq y_i\right]$$

be the benchmark loss. We show that $L^*(1-\eta) - \frac{\ln k}{\eta} \leq L_T^{ON}$.

To this end, define $W_i^t = \left(\sum_{z \in \mathcal{U}(x_i)} w_t(z,(x_i,y_i))\right)^{p_i}$ and $W^t = \prod_{i=1}^m W_i^t$. Let

$$F_i^t = p_i \cdot \frac{\sum_{z \in \mathcal{U}(x)} w_t(z,(x,y)) \mathbb{1}\left[h_t(z) \neq y\right]}{\sum_{z \in \mathcal{U}(x)} w_t(z,(x,y))}$$
$$= p_i \sum_{z \in \mathcal{U}(x_i)} P^t(z,(x_i,y_i)) \mathbb{1}\left[h_t(z) \neq y\right]$$

be the loss of Algorithm 1 on example (x_i, y_i) at round t. Observe that by the Step 7 in Algorithm 1, it holds that $W_i^T \geq (1+\eta)^{p_i \max_{z \in \mathcal{U}(x_i)} \sum_{t=1}^T [h_t(z) \neq y]}$, and therefore $W^T \geq (1+\eta)^{L^*}$.

Observe also

$$W_i^{t+1} =$$
$$\left(\sum_{z:[h_t(z) \neq y]=0} w_t(z,(x,y)) + \sum_{z:[h_t(z) \neq y]=1} (1+\eta) w_t(z,(x,y)) \right)^{p_i}$$
$$= W_i^t \left(1 + \eta \frac{F_i^t}{p_i}\right)^{p_i}$$

This implies that

$$W^T = \prod_{i=1}^m W_i^T = \prod_{i=1}^m \left[k \prod_{t=1}^T \left(1 + \eta \frac{F_i^t}{p_i}\right) \right]^{p_i} =$$
$$k^{\sum_{i=1}^m p_i} \prod_{i=1}^m \prod_{t=1}^T \left(1 + \eta \frac{F_i^t}{p_i}\right)^{p_i}$$

Combining the above we have,

$$(1+\eta)^{L^*} \leq k \prod_{i=1}^m \prod_{t=1}^T \left(1 + \eta \frac{F_i^t}{p_i}\right)^{p_i}.$$

We then apply a logarithmic transformation on both sides

$$L^* \ln(1+\eta) \leq \ln k + \sum_{i=1}^m \sum_{t=1}^T p_i \ln\left(1 + \eta \frac{F_i^t}{p_i}\right).$$

Since $a - a^2 \leq \ln(1+a) \leq a$ for $a \geq 0$, we have

$$L^*(\eta - \eta^2) \leq \ln k + \sum_{i=1}^m \sum_{t=1}^T \eta F_i^t = \ln k + \eta L_T^{ON}.$$

By dividing by η and rearranging terms we get $L^*(1-\eta) - \frac{\ln k}{\eta} \leq L_T^{ON}$.

By setting $\eta = \sqrt{\frac{\ln k}{L^*}}$ and observing that $L^* \leq T$, the remainder of the analysis follows similar to Feige, Mansour, and Schapire (2015, Equation 3-10 in proof of Theorem 1). □

A.4 Proof of Lemma 4

Proof. By finiteness of $\mathcal{U}$, observe that for any dataset $S \in (\mathcal{X} \times \mathcal{Y})^m$, each robust loss vector in the set of robust loss behaviors:

$$\Pi_{\mathcal{L}_{\mathcal{H}}^{\mathcal{U}}}(S) = \{(f(x_1,y_1), \ldots, f(x_m,y_m)) : f \in \mathcal{L}_{\mathcal{H}}^{\mathcal{U}}\}$$

maps to a 0-1 loss vector on the *inflated set* $S_{\mathcal{U}} = \{(z_1^1, y_1), \ldots, (z_1^k, y_1), \ldots, (z_m^1, y_m), \ldots, (z_m^k, y_m)\}$,

$$\Pi_{\mathcal{H}}(S_{\mathcal{U}}) = \{(h(z_1^1), \ldots, h(z_1^k), \ldots, h(z_m^1), \ldots, h(z_m^k)) : h \in \mathcal{H}\}$$

Therefore, it follows that $\left|\Pi_{\mathcal{L}_{\mathcal{H}}^{\mathcal{U}}}(S)\right| \leq |\Pi_{\mathcal{H}}(S_{\mathcal{U}})|$. Then, by applying the Sauer-Shelah lemma, it follows that $|\Pi_{\mathcal{H}}(S_{\mathcal{U}})| \leq O((mk)^{\mathrm{vc}(\mathcal{H})})$. Then, by solving for m such that $O((mk)^{\mathrm{vc}(\mathcal{H})}) \leq 2^m$, we get that $\mathrm{vc}(\mathcal{L}_{\mathcal{H}}^{\mathcal{U}}) \leq O(\mathrm{vc}(\mathcal{H}) \log(k))$. □

A.5 Proof of Lemma 6

Proof.

$$\mathop{\mathbb{E}}_{j\in[g]}[\ell_j^{rob}(h_t)] = \sum_j P_j^t(1/|G_j|) \sum_{(x,y)\in G_j} \max_{z\in\mathcal{U}(x)} \mathbb{1}\left[h_t(z)\neq y\right] \tag{7}$$

$$= \sum_{i=1}^{m} p_i \cdot \max_{z\in\mathcal{U}(x)} \mathbb{1}\left[h_t(z)\neq y\right] \tag{8}$$

$$\leq \max_{\substack{P_1'\in\Delta(\mathcal{U}(x_1)),\\ \dots,\\ P_m'\in\Delta(\mathcal{U}(x_m))}} \sum_{i=1}^{m} p_i \cdot \mathop{\mathbb{E}}_{z_i\sim P_i'} \frac{1}{T}\sum_{\tau=1}^{T} \mathbb{1}\left[h_\tau^{\text{FMS}}(z_i)\neq y_i\right] \tag{9}$$

$$\leq \min_{Q\in\Delta(\mathcal{H})} \max_{\substack{P_1'\in\Delta(\mathcal{U}(x_1)),\\ \dots,\\ P_m'\in\Delta(\mathcal{U}(x_m))}} \sum_{i=1}^{m} p_i \mathop{\mathbb{E}}_{z_i\sim P_i'} \mathop{\mathbb{E}}_{h\sim Q} \mathbb{1}\left[h(z_i)\neq y_i\right] + 2\sqrt{\frac{\ln k}{T}} \tag{10}$$

$$\leq \min_{h\in\mathcal{H}} \max_{\substack{P_1'\in\Delta(\mathcal{U}(x_1)),\\ \dots,\\ P_m'\in\Delta(\mathcal{U}(x_m))}} \sum_{i=1}^{m} p_i \cdot \mathop{\mathbb{E}}_{z_i\sim P_i'} \mathbb{1}\left[h(z_i)\neq y_i\right] + 2\sqrt{\frac{\ln k}{T}} \tag{11}$$

$$= \min_{h\in\mathcal{H}} \max_{\substack{z_1\in\mathcal{U}(x_1),\\ \dots,\\ z_m\in\mathcal{U}(x_m)}} \sum_{i=1}^{m} p_i \cdot \mathbb{1}\left[h(z_i)\neq y_i\right] + 2\sqrt{\frac{\ln k}{T}} \tag{12}$$

$$\leq \min_{h\in\mathcal{H}} \max_{j\in[g]} (1/|G_j|) \sum_{(x,y)\in G_j} \max_{z\in\mathcal{U}(x)} \mathbb{1}\left[h(z)\neq y\right] + 2\sqrt{\frac{\ln k}{T}} \tag{13}$$

$$= OPT_{\max} + 2\sqrt{\frac{\ln k}{T}} \tag{14}$$

Equation 7 holds by plugging in the definition of $\ell_j^{\text{rob}}(h_t)$(Equation 4). Equation 8 holds for a distribution $p_1,\dots,p_m$ on the samples. In Equation (9), h_t is replaced with the hypothesis selected by Algorithm 1 in each round t. Equation (10) holds by Lemma 5. Equation (12) holds since it suffices for the max-player to pick a pure strategy. Equation (13) holds since the whole probability mass is put as a uniform distribution on the worst-off group. Note that when defining $p_1,\cdots,p_m$, all individuals that belong to the same group have equal weights. □

A.6 Proof of Corollary 9

Proof. Expected robust loss on each group $G_j \in \mathcal{G}$ is:

$$\frac{1}{|G_j|} \sum_{(x,y)\in G_j} \max_{z\in\mathcal{U}(x)} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}[h_t(z)\neq y] \tag{15}$$

$$= \frac{1}{|G_j|} \sum_{(x,y)\in G_j} \max_{z\in\mathcal{U}(x)} \mathop{\mathbb{E}}_{h_t\sim U(\mathcal{H}')} \mathbb{1}[h_t(z)\neq y] \tag{16}$$

$$\leq \frac{1}{|G_j|} \sum_{(x,y)\in G_j} \mathop{\mathbb{E}}_{h_t\sim U(\mathcal{H}')} \max_{z\in\mathcal{U}(x)} \mathbb{1}[h_t(z)\neq y] \tag{17}$$

$$= \frac{1}{T}\sum_{t=1}^{T} \frac{1}{|G_j|} \sum_{(x,y)\in G_j} \max_{z\in\mathcal{U}(x)} \mathbb{1}[h_t(z)\neq y] \tag{18}$$

$$= \frac{1}{T}\sum_{t=1}^{T} \ell_j^{\text{rob}}(h_t) \leq \mathsf{OPT}_{\max}^{S} + \varepsilon \tag{19}$$

Where Equation 17 holds by Jensen's inequality and Equation 19 holds by Theorem 8. □

A.7 Proof of Lemma 11

Proof. The proof is inspired by the proof of (claim B.1 in Kearns et al., 2018) which proved a similar result for the standard 0-1 loss, and here we extend the result to the robust loss using essentially the same proof.

Let $S \subseteq \mathcal{X} \times \mathcal{Y}$ be a dataset of size m that is shattered by $\mathcal{F}^{\mathcal{U}}_{\mathcal{H},\mathcal{G}}$. Then, observe that, by definition of $\mathcal{F}^{\mathcal{U}}_{\mathcal{H},\mathcal{G}}$, the number of possible behaviors $\left|\Pi_{\mathcal{F}^{\mathcal{U}}_{\mathcal{H},\mathcal{G}}}(S)\right|$ is at most $\left|\Pi_{\mathcal{L}^{\mathcal{U}}_{\mathcal{H}}}(S)\right| \cdot |\Pi_{\mathcal{G}}(S)|$. By Sauer-Shelah Lemma, $\left|\Pi_{\mathcal{L}^{\mathcal{U}}_{\mathcal{H}}}(S)\right| \leq O(m^{\mathrm{vc}(\mathcal{L}^{\mathcal{U}}_{\mathcal{H}})})$ and $|\Pi_{\mathcal{G}}(S)| \leq O(m^{\mathrm{vc}(\mathcal{G})})$. Thus, $\left|\Pi_{\mathcal{F}^{\mathcal{U}}_{\mathcal{H},\mathcal{G}}}(S)\right| = 2^m \leq O(m^{\mathrm{vc}(\mathcal{L}^{\mathcal{U}}_{\mathcal{H}})+\mathrm{vc}(\mathcal{G})})$, and solving for m yields that $m = \tilde{O}(\mathrm{vc}(\mathcal{L}^{\mathcal{U}}_{\mathcal{H}}) + \mathrm{vc}(\mathcal{G}))$. Hence, $\mathrm{vc}(\mathcal{F}^{\mathcal{U}}_{\mathcal{H},\mathcal{G}}) \leq \tilde{O}\left(\mathrm{vc}(\mathcal{L}^{\mathcal{U}}_{\mathcal{H}}) + \mathrm{vc}(\mathcal{G})\right)$. □

A.8 Proof of Theorem 12

Proof. The output of Algorithm 2 is $\mathcal{H}' = \{h_1, \ldots, h_T\}$ where each of the predictors $h_1, \ldots, h_T$ is a majority-vote predictor over $\mathcal{H}$. Due to Blumer et al. (1989), the VC-dimension of the output space is $\mathrm{vc}(\mathcal{H}^{T'}) = \left(\mathrm{vc}(\mathcal{H})T' \ln T'\right)$ where T' is the number of rounds of Algorithm 1 in each oracle call.

Set $m = \tilde{O}\left(\frac{\mathrm{vc}(\mathcal{H}^{T'})\ln(k)+\mathrm{vc}(\mathcal{G})+\ln(1/\delta)}{\varepsilon^2}\right)$. By setting $T' = \mathcal{O}(\frac{\ln k}{\varepsilon^2})$ and by invoking Lemma 4 and Lemma 11 on the hypothesis class $\mathcal{H}$ and group class $\mathcal{G}$, we get the following uniform convergence guarantee. With probability at least $1-\delta$ over $S \sim \mathcal{D}^m$,

$$\left(\forall h \in \mathcal{H}^{T'}\right)\left(\forall G_j \in \mathcal{G}\right): \\ \left| \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\mathbb{1}[x \in G_j] \wedge \max_{z\in\mathcal{U}(x)} \mathbb{1}[h(z) \neq y]\right] - \frac{1}{m}\sum_{(x,y)\in S} \mathbb{1}[x \in G_j] \wedge \max_{z\in\mathcal{U}(x)} \mathbb{1}[h(z) \neq y]\right| \leq \varepsilon$$

We can rewrite the above guarantee in a conditional form which will be useful for us shortly in the proof. Namely, $\forall h \in \mathcal{H}^{T'}, \forall G_j \in \mathcal{G}$:

$$\Pr_{(x,y)\sim\mathcal{D}}[\exists z \in \mathcal{U}(x) : h(z) \neq y | x \in G_j] \leq \frac{\Pr_S(x \in G_j)}{\Pr_{\mathcal{D}}(x \in G_j)} \frac{1}{|G_j|} \sum_{(x,y)\in S \wedge x \in G_j} \max_{z\in\mathcal{U}(x)} \mathbb{1}[h(z) \neq y] + \frac{\varepsilon}{\Pr_D(x \in G_j)} \quad (20)$$

where $|G_j| = \sum_{(x,y)\in S} \mathbb{1}[x \in G_j]$.

Theorem 8 shows that running Algorithm 2 produces hypotheses $h_1, \ldots, h_T$ such that, $\forall G_j \in \mathcal{G}$:

$$\frac{1}{T}\sum_{t=1}^{T} \frac{1}{|G_j|} \sum_{(x,y)\in S \wedge x \in G_j} \max_{z\in\mathcal{U}(x)} \mathbb{1}[h_t(z) \neq y] \leq \mathsf{OPT}^S_{\max} + \varepsilon \quad (21)$$

Equation 20 implies that $\forall G_j \in \mathcal{G}$,

$$\frac{1}{T}\sum_{t=1}^{T} \Pr_{(x,y)\sim\mathcal{D}}[\exists z \in \mathcal{U}(x) : h_t(z) \neq y | x \in G_j] \leq \frac{1}{T}\sum_{t=1}^{T} \frac{\Pr_S(x \in G_j)}{\Pr_{\mathcal{D}}(x \in G_j)} \frac{1}{|G_j|} \sum_{(x,y)\in S \wedge x \in G_j} \max_{z\in\mathcal{U}(x)} \mathbb{1}[h_t(z) \neq y] \quad (22)$$

$$+ \frac{\varepsilon}{\Pr_{\mathcal{D}}(x \in G_j)}, \quad (23)$$

Combining Equation 21 and Equation 23 implies:

$$\frac{1}{T}\sum_{t=1}^{T} \Pr_{(x,y)\in\mathcal{D}}[\exists z \in \mathcal{U}(x) : h_t(z) \neq y | x \in G_j] \leq \frac{\Pr_S(x \in G_j)}{\Pr_{\mathcal{D}}(x \in G_j)} \left(\mathsf{OPT}^S_{\max} + \varepsilon\right) + \frac{\varepsilon}{\Pr_{\mathcal{D}}(x \in G_j)} \quad (24)$$

Now, given additional samples $\tilde{m} = O\left(\frac{\mathrm{vc}(\mathcal{G})+\log(2/\delta)}{\varepsilon^2}\right)$, in addition to the above, we can guarantee that:

$$\forall G_j \in \mathcal{G} : \frac{\Pr_S(x \in G_j)}{\Pr_{\mathcal{D}}(x \in G_j)} \leq \frac{\Pr_{\mathcal{D}}(x \in G_j) + \varepsilon}{\Pr_{\mathcal{D}}(x \in G_j)} = 1 + \frac{\varepsilon}{\Pr_{\mathcal{D}}(x \in G_j)}. \tag{25}$$

Combining Equation 24 and Equation 25 implies that:

$$\frac{1}{T}\sum_{t=1}^{T} \Pr_{(x,y)\sim\mathcal{D}}[\exists z \in \mathcal{U}(x) : h_t(z) \neq y | x \in G_j] \leq \left(1 + \frac{\varepsilon}{\Pr_{\mathcal{D}}(x \in G_j)}\right)\left(\mathsf{OPT}^S_{\max} + \varepsilon\right) + \frac{\varepsilon}{\Pr_{\mathcal{D}}(x \in G_j)}$$

which completes the proof. We can also obtain a bound in terms of $\mathsf{OPT}^{\mathcal{D}}_{\max}$ instead of $\mathsf{OPT}^S_{\max}$ using a similar approach used in Section A.9.

□

A.9 Proof of Theorem 13

Proof. The output of Algorithm 2 is $\mathcal{H}' = \{h_1, \ldots, h_T\}$. Taking majority-vote over the predictors in $\mathcal{H}'$ is equivalent to taking the majority-vote of majority-vote predictors over $\mathcal{H}$. Therefore, due to Blumer et al. (1989), the VC-dimension of the output space is $\mathrm{vc}(\mathcal{H}^{T'})^T = \Big(\mathrm{vc}(\mathcal{H})T' \ln T'\Big)T \ln T$, where T' is the number of rounds of Algorithm 1 in each oracle call and T is the number of rounds of Algorithm 2.

Let the sample size $m = \tilde{O}\left(\frac{\mathrm{vc}(\mathcal{H}^{T'})^T \log(k)+\mathrm{vc}(\mathcal{G})+\log(1/\delta)}{\varepsilon^2}\right)$. By setting $T = \mathcal{O}(\ln g/\varepsilon^2)$ and $T' = \mathcal{O}(\frac{\ln k}{\varepsilon^2})$ and by invoking Lemma 4 and Lemma 11 on the hypothesis class $\mathcal{H}$ and group class $\mathcal{G}$, we get the following uniform convergence guarantee. With probability at least $1-\delta$ over the sample set $S \sim \mathcal{D}^m$, $\forall h \in (\mathcal{H}^{T'})^T$ and $\forall G_j \in \mathcal{G}$:

$$\left| \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\mathbb{1}[x \in G_j] \wedge \max_{z\in\mathcal{U}(x)} \mathbb{1}[h(z) \neq y]\right] - \frac{1}{m}\sum_{(x,y)\in S} \mathbb{1}[x \in G_j] \wedge \max_{z\in\mathcal{U}(x)} \mathbb{1}[h(z) \neq y] \right| \leq \varepsilon \tag{26}$$

We can rewrite the above guarantee in a conditional form which will be useful for us shortly in the proof. Namely, $\forall h \in (\mathcal{H}^{T'})^T$ and $\forall G_j \in \mathcal{G}$:

$$\Pr_{(x,y)\sim\mathcal{D}}[\exists z \in \mathcal{U}(x) : h(z) \neq y | x \in G_j] \leq \frac{\Pr_S(x \in G_j)}{\Pr_{\mathcal{D}}(x \in G_j)} \frac{1}{|G_j|} \sum_{(x,y)\in S \wedge x \in G_j} \max_{z\in\mathcal{U}(x)} \mathbb{1}[h(z) \neq y] + \frac{\varepsilon}{\Pr_{\mathcal{D}}(x \in G_j)} \tag{27}$$

where $|G_j| = \sum_{(x,y)\in S} \mathbb{1}[x \in G_j]$. Theorem 10 provides that $h^{\mathrm{maj}} = \mathrm{MAJ}(h_1, \ldots, h_T)$ satisfies that $\forall G_j \in \mathcal{G}$:

$$\frac{1}{|G_j|} \sum_{(x,y)\in S \wedge x \in G_j} \max_{z\in\mathcal{U}(x)} \mathbb{1}[h^{\mathrm{maj}}(z) \neq y] \leq \beta(\mathsf{OPT}^S_{\max} + \varepsilon) \tag{28}$$

Combining Equation 27 and Equation 28 implies that $\forall G_j \in \mathcal{G}$:

$$\Pr_{(x,y)\sim\mathcal{D}}\left[\exists z \in \mathcal{U}(x) : h^{\mathrm{maj}}(z) \neq y | x \in G_j\right] \leq \frac{\Pr_S(x \in G_j)}{\Pr_{\mathcal{D}}(x \in G_j)}\left(\beta(\mathsf{OPT}^S_{\max} + \varepsilon)\right) + \frac{\varepsilon}{\Pr_{\mathcal{D}}(x \in G_j)} \tag{29}$$

Now, given additional samples $\tilde{m} = O\left(\frac{\mathrm{vc}(\mathcal{G})+\log(2/\delta)}{\varepsilon^2}\right)$, guarantees that:

$$\forall G_j \in \mathcal{G} : \frac{\Pr_S(x \in G_j)}{\Pr_{\mathcal{D}}(x \in G_j)} \leq \frac{\Pr_{\mathcal{D}}(x \in G_j) + \varepsilon}{\Pr_{\mathcal{D}}(x \in G_j)} = 1 + \frac{\varepsilon}{\Pr_{\mathcal{D}}(x \in G_j)} \tag{30}$$

Combining Equation 29 and Equation 30 gives a refined bound on the average conditional robust loss that holds uniformly across groups. Namely, $\forall G_j \in \mathcal{G}$,

$$\Pr_{(x,y)\sim\mathcal{D}}\left[\exists z \in \mathcal{U}(x) : h^{\text{maj}}(z) \neq y | x \in G_j\right] \leq \left(1 + \frac{\varepsilon}{\Pr_{\mathcal{D}}(x \in G_j)}\right)\left(\beta(\mathsf{OPT}^{S}_{\max} + \varepsilon)\right) + \frac{\varepsilon}{\Pr_{\mathcal{D}}(x \in G_j)}$$

We can also obtain a guarantee in terms of $\mathsf{OPT}^{\mathcal{D}}_{\max}$ instead of $\mathsf{OPT}^{S}_{\max}$, as follows. Let $h^* \in \mathcal{H}$ be a predictor which attains $\mathsf{OPT}^{\mathcal{D}}_{\max}$ defined as

$$\mathsf{OPT}^{\mathcal{D}}_{\max} = \min_{h\in\mathcal{H}} \max_{G_j\in\mathcal{G}} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\max_{z\in\mathcal{U}(x)} \mathbb{1}[h(z) \neq y] \middle| x \in G_j\right].$$

Dividing both sides of Equation 26 by $\Pr_S(x \in G_j)$ provides that $\forall G_j \in \mathcal{G}, \forall h \in \mathcal{H}$:

$$\left|\frac{\Pr_{\mathcal{D}}(x \in G_j)}{\Pr_S(x \in G_j)} \Pr_{(x,y)\in\mathcal{D}}[\exists z \in \mathcal{U}(x) : h(z) \neq y | x \in G_j] - \Pr_{(x,y)\in S}[\exists z \in \mathcal{U}(x) : h(z) \neq y | x \in G_j]\right| \leq \frac{\varepsilon}{\Pr_S(x \in G_j)}$$

and thus it implies that

$$\Pr_{(x,y)\in S}[\exists z \in \mathcal{U}(x) : h(z) \neq y | x \in G_j] \leq \left(1 + \frac{\varepsilon}{\Pr_S(x \in G_j)}\right) \Pr_{(x,y)\sim\mathcal{D}}[\exists z \in \mathcal{U}(x) : h(z) \neq y | x \in G_j] + \frac{\varepsilon}{\Pr_S(x \in G_j)}$$

Supposing that $\forall G_j \in \mathcal{G}, \Pr_S(x \in G_j) \geq \gamma$. By taking a max over groups $G_j \in \mathcal{G}$, we get

$$\mathsf{OPT}^{S}_{\max} \leq (1 + \frac{\varepsilon}{\gamma})\mathsf{OPT}^{\mathcal{D}}_{\max} + \frac{\varepsilon}{\gamma}.$$

□