Best of Both Worlds: Towards Adversarial Robustness with Transduction and Rejection

Nils Palumbo [*1], **Yang Guo** [*1], **Xi Wu** [2], **Jiefeng Chen** [1], **Yingyu Liang** [1], **Somesh Jha** [1]

[1] University of Wisconsin-Madison, [2] Google
npalumbo@wisc.edu, yguo@cs.wisc.edu, wu.andrew.xi@gmail.com,
jchen662@wisc.edu, yliang@cs.wisc.edu, jha@cs.wisc.edu

Abstract

Both transduction and rejection have emerged as key techniques to enable stronger defenses against adversarial perturbations, but existing work has not investigated the combination of transduction and rejection. Our theoretical analysis shows that combining the two can potentially lead to better guarantees than using transduction or rejection alone. Based on the analysis, we propose a defense algorithm that learns a transductive classifier with the rejection option and also propose a strong adaptive attack for evaluating our defense. The experimental results on MNIST and CIFAR-10 show that it has strong robustness, outperforming existing baselines, including those using only transduction or rejection.

1 Introduction

While machine learning has made significant progress, the brittleness of learning systems to adversarial inputs is a significant challenge in real-world deployments. Robust learning systems become crucial, especially for security-sensitive applications. However, even for medium magnitudes of adversarial perturbations to the inputs, the robustness of existing defense methods is not satisfactory.

Recent studies have suggested new alternatives for improving robustness. Two promising new settings are transduction and rejection. With rejection, the learning model is allowed to reject perturbed inputs. Existing theoretical analysis (e.g., [Tra21]) shows that there exist classifiers with the rejection option (a.k.a. selective classifiers) that can tolerate twice the magnitude of adversarial perturbations than traditional classifiers without rejection. With transduction,[2] the unlabeled test input data are available to the training algorithm and the model is only required to correctly classify these given test inputs. Existing theoretical analysis (e.g., [MHS21]) shows transduction can improve the sample complexity to get generalization guarantees from potentially exponential size to polynomial size, though may reduce the tolerance of the magnitude by half. See Appendix A for more discussion on related work.

This work investigates the potential benefit of combining both transduction and rejection. We provide a novel theoretical analysis showing the combination can enjoy the best of both worlds: it can guarantee generalization with a polynomial sample size without reducing the tolerance of the perturbation magnitude. Based on the analysis, we propose an algorithm for transductively learning selective classifiers and also design a strong adaptive attack for evaluating the method. We then perform experiments on synthetic and real data comparing our proposed method to existing baselines. The experimental results show that our method can achieve better robustness than existing baselines. In particular, it outperforms the methods using only transduction or only rejection, empirically confirming the benefit of combining the two techniques.

[*]Equal contribution.
[2]Robust learning with transduction is also called dynamic defense, test-time adaptive defense, etc.

ML Safety Workshop, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

	Robust Risk	Robust Risk (with Rejection)
Inductive	$\mathrm{R}_{\mathcal{U}}(h;\mathcal{D}) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\sup_{z\in\mathcal{U}(x)} \mathbb{1}\{h(z)\neq y\}\right]$	$\mathrm{R}_{\mathcal{U},\mathrm{rej}}(h;\mathcal{D}) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\sup_{z\in\mathcal{U}(x)} \mathbb{1}\{h(z)\notin\{y,\perp\}\vee h(x)\neq y\}\right]$
Transductive	$\mathrm{R}^{\mathrm{trans}}_{\mathcal{U}}(h;\mathcal{D}) = \mathop{\mathbb{E}}_{\substack{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}^n\\(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\sim\mathcal{D}^m}}\left[\sup_{\tilde{\boldsymbol{z}}\in\mathcal{U}(\tilde{\boldsymbol{x}})} \frac{1}{m}\sum_{i=1}^m \mathbb{1}\left\{h\left(\tilde{z}_i\right)\neq\tilde{y}_i\right\}\right]$	$\mathrm{R}^{\mathrm{trans}}_{\mathcal{U},\mathrm{rej}}(h;\mathcal{D}) = \mathop{\mathbb{E}}_{\substack{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}^n\\(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\sim\mathcal{D}^m}}\left[\sup_{\tilde{\boldsymbol{z}}\in\mathcal{U}(\tilde{\boldsymbol{x}})} \frac{1}{m}\sum_{i=1}^m \mathbb{1}\left\{\begin{pmatrix}(\mathbb{A}(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{z}})\left(\tilde{z}_i\right)\notin\{\tilde{y}_i\}\wedge\tilde{z}_i=\tilde{x}_i)\\ \vee\left(\begin{array}{l}\mathbb{A}(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{z}})\left(\tilde{z}_i\right)\notin\{\tilde{y}_i,\perp\}\\ \wedge\tilde{z}_i\neq\tilde{x}_i\end{array}\right)\end{pmatrix}\right\}\right]$

Table 1: Summary of the robust risk in all settings.

	Realizable		Agnostic Generalization Bound
	Condition	Generalization Bound	
Inductive [MHS19]	$\mathrm{OPT}_{\mathcal{U}} = 0$	$O\left(\frac{2^{\mathrm{VC}(\mathcal{H})}\log(n)+\log(1/\delta)}{n}\right)$	$\mathrm{OPT}_{\mathcal{U}} + O\left(\sqrt{\frac{2^{\mathrm{VC}(\mathcal{H})}+\log(1/\delta)}{n}}\right)$
Transduction [MHS21]	$\mathrm{OPT}_{\mathcal{U}^2} = 0$	$O\left(\frac{\mathrm{VC}(\mathcal{H})\log(n)+\log(1/\delta)}{n}\right)$	$2\mathrm{OPT}_{\mathcal{U}^2} + O\left(\sqrt{\frac{\mathrm{VC}(\mathcal{H})+\log(1/\delta)}{n}}\right)$
Rejection (Theorem 1)[4]	$\mathrm{OPT}_{\mathcal{U},\mathrm{rej}} = 0$	$O\left(\frac{2^{\mathrm{VC}(T(\mathcal{H}))}\log(n)+\log(1/\delta)}{n}\right)$	$\mathrm{OPT}_{\mathcal{U},\mathrm{rej}} + O\left(\sqrt{\frac{2^{\mathrm{VC}(T(\mathcal{H}))}+\log(1/\delta)}{n}}\right)$
Transduction + Rejection (Theorem 2)	$\mathrm{OPT}_{\mathcal{U}} = 0$	$O\left(\frac{\mathrm{VC}(\mathcal{H})\log(n)+\log(1/\delta)}{n}\right)$	$4\mathrm{OPT}_{\mathcal{U}} + O\left(\sqrt{\frac{\mathrm{VC}(\mathcal{H})+\log(1/\delta)}{n}}\right)$

Table 2: Summary of generalization bounds for the four settings. $\mathrm{VC}(T(\mathcal{H})) \leq (\mathrm{VC}(\mathcal{H}_r) + \mathrm{VC}(\mathcal{H}_c))\log(\mathrm{VC}(\mathcal{H}_r) + \mathrm{VC}(\mathcal{H}_c))$, where $\mathcal{H}_c, \mathcal{H}_r$, represents the hypothesis class of classifier and rejector for the selective classifier respectively.

2 Summary of Results

Preliminary. Let $\mathcal{X}$ denote the input space, $\mathcal{Y}$ the label space, $\mathcal{D}$ the clean data distribution over $\mathcal{X}\times\mathcal{Y}$. Let $\mathcal{U}(x)$ denote the set of possible perturbations of an input x, e.g., the ℓ_p ball of radius ϵ: $\mathcal{U}(x) = \{z : \|z-x\|_p \leq \epsilon\}$. We restrict the class of $\mathcal{U}$ to those for which $\forall x \in \mathcal{X}\ x \in \mathcal{U}(x)$. In the traditional robust classification setting (also called the inductive setting), given a classifier $h : \mathcal{X} \mapsto \mathcal{Y}$, the adversary aims to find a perturbation for each clean data point (x, y) to incur the maximum error, i.e., the error is $\sup_{z\in\mathcal{U}(x)} \mathbb{1}\{h(z)\neq y\}$. In the new setting with rejection, the selective classifier can output rejection (denoted by $\perp$), i.e., $h : \mathcal{X} \mapsto \mathcal{Y}\cup\{\perp\}$. An error occurs only when h rejects a clean input or accepts and misclassifies, i.e., the error is $\sup_{z\in\mathcal{U}(x)} \mathbb{1}\{h(z)\notin\{y,\perp\}\vee h(x)\neq y\}$. In the new setting with transduction, the learning algorithm (the transductive learner) has access to the unlabeled test input data and the goal is to predict labels for these given test inputs (need not for other test inputs). Formally, there are n i.i.d. training sample $(\boldsymbol{x},\boldsymbol{y}) \sim \mathcal{D}^n$ and m i.i.d. test samples $(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}) \sim \mathcal{D}^m$, and the adversary can perturb $\tilde{\boldsymbol{x}}$ to $\tilde{\boldsymbol{z}} \in \mathcal{U}(\tilde{\boldsymbol{x}})$.[3] The learner $\mathbb{A}$ is given $(\boldsymbol{x},\boldsymbol{y})$ and $\tilde{\boldsymbol{z}}$ and it outputs m labels as predictions for $\tilde{\boldsymbol{z}}$ (denoted as $h(\tilde{\boldsymbol{z}}) = (h(\tilde{z}_i))_{i=1}^m$). That is, it is a function $\mathbb{A} : (\mathcal{X}\times\mathcal{Y})^n \times \mathcal{X}^m \mapsto (\mathcal{Y}\cup\{\perp\})^m$. An error occurs when the prediction is not correct on some test input, i.e., the error is $\sup_{\tilde{\boldsymbol{z}}\in\mathcal{U}(\tilde{\boldsymbol{x}})} \frac{1}{m}\sum_{i=1}^m \mathbb{1}\left\{h\left(\tilde{z}_i\right)\neq\tilde{y}_i\right\}$ where $h(\tilde{\boldsymbol{z}}) = \mathbb{A}(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{z}})$.

This work considers the new setting combining transduction and rejection. A transductive learner for selective classifiers is a function $\mathbb{A}' : (\mathcal{X}\times\mathcal{Y})^n \times \mathcal{X}^m \mapsto (\mathcal{Y}\cup\{\perp\})^m$. An error occurs when it rejects a clean test input or accepts and misclassifies, i.e., the error is $\sup_{\tilde{\boldsymbol{z}}\in\mathcal{U}(\tilde{\boldsymbol{x}})} \mathrm{err}^{\mathrm{rej}}_{\tilde{\boldsymbol{x}},\tilde{\boldsymbol{z}},\tilde{\boldsymbol{y}}}(h)$ where $\mathrm{err}^{\mathrm{rej}}_{\tilde{\boldsymbol{x}},\tilde{\boldsymbol{z}},\tilde{\boldsymbol{y}}}(h) = \frac{1}{m}\sum_{i=1}^m \mathbb{1}\left\{(h\left(\tilde{z}_i\right)\notin\{\tilde{y}_i\}\wedge\tilde{z}_i=\tilde{x}_i)\vee(h\left(\tilde{z}_i\right)\notin\{\tilde{y}_i,\perp\}\wedge\tilde{z}_i\neq\tilde{x}_i)\right\}$ and $h(\tilde{\boldsymbol{z}}) = \mathbb{A}'(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{z}})$. The definitions of robust risks in different settings are summarized in Table 1. And we define the optimal risk without rejection as $\mathrm{OPT}_{\mathcal{U}} := \inf_{h\in\mathcal{H}} \mathrm{R}_{\mathcal{U}}(h;\mathcal{D})$ and with rejection as $\mathrm{OPT}_{\mathcal{U},\mathrm{rej}} := \inf_{h\in\mathcal{H}} \mathrm{R}_{\mathcal{U},\mathrm{rej}}(h;\mathcal{D})$.

Main Results. We provide generalization bounds for the setting with rejection alone and that with both transduction and rejection. The bounds, together with known results for the other two settings, are summarized in Table 2 (formal statements/proofs in Section 3 and the appendix). We can see that in both realizable (i.e., there exist 0 robust risk models) and agnostic cases, combining transduction and rejection leads to the best bounds.

Section 4 presents our novel defense method utilizing transduction and rejection. Section 5 presents experimental results showing its strong empirical performance on CIFAR-10 and MNIST, e.g., on

[3] Here $\boldsymbol{x} = (x_i)_{i=1}^n$ and similarly with $\boldsymbol{y}, \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}$, etc. We overload the notation: $\mathcal{U}(\tilde{\boldsymbol{x}}) := \{\mathbf{u} \in \mathcal{X}^m : u_i \in \mathcal{U}(\tilde{x}_i)\}$.
[4] T in the bound refers to a transformation on the hypothesis class given by Equation 10 in the appendix.

CIFAR-10, we achieve 72.5% transductive robust accuracy with rejection, a significant improvement on the current state-of-the-art result of robust accuracy 66.56% [CAS+20] for the perturbation considered (l_∞ with budget $\epsilon = 8/255$).

3 Theoretical Analysis

Here we present the theorems for the realizable case in two settings: rejection only, and combining transduction with rejection. The proofs and the results for the agnostic case are in the Appendix E.

Rejection Only. We view the selective classifier $h \in \mathcal{H}$ as a composition of a classifier $h_c \in \mathcal{H}_c$ and a rejector $h_r \in \mathcal{H}_r$, where $h(x) = \begin{cases} h_c(x) & h_r(x) = \not\perp \\ \perp & h_r(x) = \perp \end{cases}$.

Theorem 1. *For any $n \in \mathbb{N}, \delta \in (0, 1/2)$, class $\mathcal{H} = \mathcal{H}_c \times \mathcal{H}_r$, perturbation set $\mathcal{U}$, and any $\epsilon, \delta \in (0, 1/2)$ and distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ satisfying $OPT_{\mathcal{U},rej} = 0$, there exists an algorithm $\mathbb{A} : (\mathcal{X} \times \mathcal{Y})^n \mapsto \mathcal{H}$ satisfying for $\epsilon = \frac{2^{\mathrm{VC}(T(\mathcal{H}))} \log(n) + \log(1/\delta)}{n}$,*

$$\Pr_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}^n} \left[\mathrm{R}_{\mathcal{U},rej}(\mathcal{A}(\boldsymbol{x},\boldsymbol{y}); \mathcal{D}) \leq \epsilon \right] \geq 1 - \delta.$$

The key strategy is to construct a compression scheme on the transformed hypothesis class $T(\mathcal{H})$. We obtain a similar guarantee to the inductive setting without rejection [MHS19] (see Table 2), but with dependence on $\mathrm{VC}(T(\mathcal{H}))$ than $\mathrm{VC}(\mathcal{H})$. So while [Tra21] shows rejection may double the perturbation magnitude tolerated, the sample complexity can be still exponentially large.

Transduction + Rejection. Following [MHS21], define the set of robust hypotheses $\Delta^{\mathcal{U}}_{\mathcal{H}}(\boldsymbol{z}, \boldsymbol{y}, \tilde{\boldsymbol{z}})$ as:

$$\Delta^{\mathcal{U}}_{\mathcal{H}}(\boldsymbol{z}, \boldsymbol{y}, \tilde{\boldsymbol{z}}) = \left\{ \mathrm{R}_{\mathcal{U}^{-1}}(h; \boldsymbol{z}, \boldsymbol{y}) = 0 \wedge \mathrm{R}_{\mathcal{U}^{-1}}(h; \tilde{\boldsymbol{z}}) = 0 \right\} \tag{1}$$

where $\mathcal{H}$ is a binary hypothesis class and where $\mathrm{R}_{\mathcal{U}}(h; \boldsymbol{z}, \boldsymbol{y}) = \sup_{\tilde{\boldsymbol{x}} \in \mathcal{U}(\boldsymbol{z})} \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{h(\tilde{\mathrm{x}}_i) \neq y_i\}$ and $\mathrm{R}_{\mathcal{U}}(h; \boldsymbol{z}) = \mathrm{R}_{\mathcal{U}}(h; \boldsymbol{z}, h(\boldsymbol{z}))$. Following Tramèr [Tra21], we can define a transformation $F_{\mathcal{U}}$ that maps a classifier without rejection, c, to the selective classifier $c' = F_{\mathcal{U}}(c)$:

$$F_{\mathcal{U}}(c)(x) = \begin{cases} c(x) & \text{if } \forall x' \in \mathcal{U}^{-1}(x)\ c(x') = c(x) \\ \perp & \text{otherwise} \end{cases}. \tag{2}$$

We will use $F_{\mathcal{U}^{1/2}}$ for the following result. For example, for ℓ_p-norm perturbation with an adversarial budget ϵ, $\mathcal{U}$ is an ℓ_p ball of radius ϵ, and $\mathcal{U}^{1/2}$ is one of radius $\epsilon/2$.

Theorem 2. *For any $n \in \mathbb{N}$, $\delta > 0$, class $\mathcal{H}$, perturbation set $\mathcal{U}$ such that $\mathcal{U} = \mathcal{U}^{-1}$, and distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ satisfying $OPT_{\mathcal{U}} = 0$, for $\epsilon = \frac{\mathrm{VC}(\mathcal{H}) \log(2n) + \log(1/\delta)}{n}$:*

$$\Pr_{\substack{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}^n \\ (\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\sim\mathcal{D}^n}} \left[\forall\, \boldsymbol{z} \in \mathcal{U}(\boldsymbol{x}), \forall \tilde{\boldsymbol{z}} \in \mathcal{U}(\tilde{\boldsymbol{x}}), \forall \hat{h} \in F_{\mathcal{U}^{1/2}}(\Delta^{\mathcal{U}}_{\mathcal{H}}(\boldsymbol{z}, \boldsymbol{y}, \tilde{\boldsymbol{z}})) : \mathrm{err}^{rej}_{\tilde{\boldsymbol{x}},\tilde{\boldsymbol{z}},\tilde{\boldsymbol{y}}}(\hat{h}) \leq \epsilon \right] \geq 1 - \delta.$$

For $\mathcal{U}$ satisfying our conditions (including l_p balls), we obtain a stronger guarantee than is possible without rejection [MHS21]. Compared to the guarantee for transduction without rejection [MHS21] (see Table 2), our result requires weaker assumptions on the hypothesis class $\mathcal{H}$: we need $\mathrm{OPT}_{\mathcal{U}} = 0$ rather than $\mathrm{OPT}_{\mathcal{U}^2} = 0$, tolerating twice the adversarial budget. Compared to the result for rejection only, this bound has a linear sample complexity rather than exponential. Therefore, combining transduction and rejection has the advantage of both techniques.

4 Defense Methods with Transduction and Rejection

Theorem 2 suggests we should first obtain a classifier $h \in \Delta^{\mathcal{U}}_{\mathcal{H}}(\boldsymbol{z}, \boldsymbol{y}, \tilde{\boldsymbol{z}})$, and then apply transformation $F_{\mathcal{U}^{1/2}}$ in Equation 2 to get a selective classifier $\hat{h} = F_{\mathcal{U}^{1/2}}(h)$ to predict on the test inputs. We describe the resulting defense, which we refer to as TLDR (Transductive Learning via Defense with Rejection).

To get h, we perform adversarial training on both the training set and the test set, using a robust cross-entropy objective. As in TADV [CGW+21] we train with private randomness. As labels are not

present on the test set, we drop the base loss term on the test set and use robust loss only. Specifically, given the labeled training data T and the test inputs E, we optimize the following objective:

$$\min_h \frac{1}{n} \sum_{(x,y)\in T} \left[\mathcal{L}_{\text{CE}}(h^{softmax}(x), y) + \max_{x' \in \mathcal{U}(x)} \mathcal{L}_{\text{CE}}\left(h^{softmax}(x'), y\right) \right] + \frac{\lambda}{m} \sum_{x \in E} \left[\max_{x' \in \mathcal{U}(x)} \mathcal{L}_{\text{CE}}\left(h^{softmax}(x'), h(x)\right) \right] \tag{3}$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss and $h = g \circ h^{softmax}$. $h^{softmax}$ returns the softmax activation for the model h, and g is an argmax function.

Having trained h, we now discuss how to implement $F_{\mathcal{U}^{1/2}}$ in our experiments. We only need to determine for each test input x, if $F_{\mathcal{U}^{1/2}}(h)$ needs to reject x. Hence, we can use a standard inductive attack, e.g., PGD, to check if there exists x' such that $x \in \mathcal{U}^{1/2}(x')$ and $h(x') \neq h(x)$. Specifically, we find x' to maximize $\mathcal{L}_{\text{CE}}(h^{softmax}(x'), h(x))$ subject to $x \in \mathcal{U}^{1/2}(x')$, and reject x if $h(x') \neq h(x)$.

5 Experiments

Evaluation Methods. For inductive classifiers, we use the classic PGD on cross-entropy. For classifiers with rejection, we design a loss $\mathcal{L}_{\text{REJ}}$ that takes into account the rejection option. For inductive classifiers with rejection, we use PGD on $\mathcal{L}_{\text{REJ}}$. For transductive classifiers, we use GMSA on cross-entropy, which has been shown to be a strong adaptive attack on transduction [CGW+21]. Finally, for transductive classifiers with rejection, we use GMSA on $\mathcal{L}_{\text{REJ}}$.

We perform experiments on MNIST and CIFAR-10. All models are adversarially trained via a robust cross-entropy objective, discussed above, with the inductive models dropping the second transductive regularization term in Equation 3. On MNIST, we use a LeNet architecture, with a full adversarial budget of $\epsilon = 0.3$ in l_∞; on CIFAR-10, we use a ResNet-20 architecture, with a budget of $\epsilon = 8/255$ in l_∞. We use ADAM with a learning rate of 0.001 with 40 epochs. The PGD attacks use 200 steps for MNIST and 100 for CIFAR-10. We use 10 iterations of GMSA; we report the stronger of GMSA_{MIN} and GMSA_{AVG}. See Appendix C for the experimental details and Appendix B for ablation studies.

Baselines. We compare our method to standard adversarial training (AT) [GSS15] [MMS+19], with and without rejection, as well as existing transductive defenses: runtime masking and cleansing (RMC) [WYW20a], domain adversarial neural network (DANN) [AGL+15], and transductive adversarial training (TADV) [CGW+21].

Defense	Attacker	MNIST		CIFAR-10	
		p_{REJ}	Robust accuracy	p_{REJ}	Robust accuracy
AT	PGD ($\mathcal{L}_{\text{CE}}$)	–	0.920	–	0.587
AT (with rejection)	PGD ($\mathcal{L}_{\text{REJ}}$)	0.736	0.970	0.384	0.634
RMC	GMSA ($\mathcal{L}_{\text{CE}}$)	–	0.588	–	0.396
DANN	GMSA ($\mathcal{L}_{\text{CE}}$)	–	0.062	–	0.055
TADV	GMSA ($\mathcal{L}_{\text{CE}}$)	–	0.943	–	0.541
TLDR (ours) (no rejection)	GMSA ($\mathcal{L}_{\text{CE}}$)	–	0.900	–	0.516
TLDR (ours) (with rejection)	GMSA ($\mathcal{L}_{\text{REJ}}$)	**0.588**	**0.967**	**0.208**	**0.739**

Table 3: Results on MNIST and CIFAR-10. Robust accuracy is 1 - robust error; see section 2. p_{REJ} is the percentage of inputs rejected. The baseline results are from [CGW+21]. The strongest attack against each defense is shown; for DANN, the strongest attack against the strongest base model is shown.

Results. Table 4 shows that transduction and rejection both increase performance independently, while combining both techniques leads to the best results. In particular, our defense outperforms existing transductive defenses such as RMC and DANN. It also outperforms the strongest existing baseline of 66.56% robust accuracy on CIFAR-10 [CAS+20] (note that 66.56% is for the classic inductive setting without rejection and transduction). These results provide positive support for the benefit of combining transduction and rejection to improve adversarial robustness.

References

[AGL+15] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks, 2015.

[Ass83] Patrick Assouad. Densité et dimension. In *Annales de l'Institut Fourier*, volume 33, pages 233–282, 1983.

[BEHW89] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989.

[CAS+20] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

[CGW+21] Jiefeng Chen, Yang Guo, Xi Wu, Tianqi Li, Qicheng Lao, Yingyu Liang, and Somesh Jha. Towards adversarial robustness via transductive learning. *arXiv preprint arXiv:2106.08387*, 2021.

[CH20] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020.

[CRC+21] Jiefeng Chen, Jayaram Raghuram, Jihye Choi, Xi Wu, Yingyu Liang, and Somesh Jha. Revisiting adversarial robustness of classifiers with a reject option. In *The AAAI-22 Workshop on Adversarial Machine Learning and Beyond*, 2021.

[CW17] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[GKKM20] Shafi Goldwasser, Adam Tauman Kalai, Yael Kalai, and Omar Montasser. Beyond perturbations: Learning guarantees with arbitrary adversarial test examples. *Advances in Neural Information Processing Systems*, 33:15859–15870, 2020.

[Goo19] Ian Goodfellow. A research agenda: Dynamic models to defend against correlated attacks. *arXiv preprint arXiv:1903.06293*, 2019.

[GSS14] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[GSS15] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.

[HKS19] Steve Hanneke, Aryeh Kontorovich, and Menachem Sadigurschi. Sample compression for real-valued learners. In *Algorithmic Learning Theory*, pages 466–488. PMLR, 2019.

[KCF20] Masahiro Kato, Zhenghang Cui, and Yoshihiro Fukuhara. Atro: Adversarial training with a rejection option. *arXiv preprint arXiv:2010.12905*, 2020.

[LF19] Cassidy Laidlaw and Soheil Feizi. Playing it safe: Adversarial robustness with an abstain option. *arXiv preprint arXiv:1911.11253*, 2019.

[LW86] Nick Littlestone and Manfred Warmuth. Relating data compression and learnability. 1986.

[MDFF16] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[MHS19] Omar Montasser, Steve Hanneke, and Nathan Srebro. Vc classes are adversarially robustly learnable, but only improperly. In *Conference on Learning Theory*, pages 2512–2530. PMLR, 2019.

[MHS21] Omar Montasser, Steve Hanneke, and Nathan Srebro. Transductive robust learning guarantees. *arXiv preprint arXiv:2110.10602*, 2021.

[MMS+17] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[MMS+19] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019.

[MY16] Shay Moran and Amir Yehudayoff. Sample compression schemes for vc classes. *Journal of the ACM (JACM)*, 63(3):1–10, 2016.

[PZH+22] Tianyu Pang, Huishuai Zhang, Di He, Yinpeng Dong, Hang Su, Wei Chen, Jun Zhu, and Tie-Yan Liu. Two coupled rejection metrics can tell adversarial examples apart. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15223–15233, 2022.

[SDM+20] Angelo Sotgiu, Ambra Demontis, Marco Melis, Battista Biggio, Giorgio Fumera, Xiaoyi Feng, and Fabio Roli. Deep neural rejection against adversarial examples. *EURASIP Journal on Information Security*, 2020(1):1–10, 2020.

[SF12] Robert E Schapire and Yoav Freund. Boosting. adaptive computation and machine learning. *MIT Press, Cambridge, MA*, 1(1.2):9, 2012.

[SHS20] David Stutz, Matthias Hein, and Bernt Schiele. Confidence-calibrated adversarial training: Generalizing to unseen attacks. In *International Conference on Machine Learning*, pages 9155–9166. PMLR, 2020.

[SLK20] Fatemeh Sheikholeslami, Ali Lotfi, and J Zico Kolter. Provably robust classification of adversarial examples with detection. In *International Conference on Learning Representations*, 2020.

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[TCBM20] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *Advances in Neural Information Processing Systems*, 33:1633–1645, 2020.

[Tra21] Florian Tramer. Detecting adversarial examples is (nearly) as hard as classifying them. *arXiv preprint arXiv:2107.11630*, 2021.

[WJS+21] Dequan Wang, An Ju, Evan Shelhamer, David Wagner, and Trevor Darrell. Fighting gradients with gradients: Dynamic defenses against adversarial attacks, 2021.

[WYW20a] Yi-Hsuan Wu, Chia-Hung Yuan, and Shan-Hung Wu. Adversarial robustness via runtime masking and cleansing. In Hal DaumÃ© III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10399–10409. PMLR, 13–18 Jul 2020.

[WYW20b] Yi-Hsuan Wu, Chia-Hung Yuan, and Shan-Hung Wu. Adversarial robustness via runtime masking and cleansing. In *International Conference on Machine Learning*, pages 10399–10409. PMLR, 2020.

[ZYJ+19] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019.

Supplementary Material

A Related Work

In recent years, there have been extensive studies on adversarial robustness in the classical inductive learning setting, where the model is fixed during the evaluation phase [CW17, GSS14, MDFF16]. Most popular and effective methods are adversarial training, such as PGD [MMS+17], TRADES [ZYJ+19]. These methods are effective against adversaries on small dataset like MNIST, but still ineffective on complex dataset like CIFAR-10 or ImageNet [CAS+20]. Researchers have proposed defense mechanisms beyond adversarial training but most defenses broke down under strong adaptive attacks [CH20, TCBM20].

To break this robust bottleneck in the classical inductive setting, people have proposed alternative settings with relaxed yet realistic assumptions, particularly by allowing rejection and transduction. In robust learning with rejection option, we allow rejection of adversarial examples instead of correctly classifying all of them [Tra21]. People have considered different variants of adversarial training with rejection option [LF19, PZH+22, CRC+21, KCF20, SDM+20], also different generalizations such as [SHS20](unseen attacks), [SLK20](certified robustness).

The other approach is to define alternative notion of adversarial robustness via the transductive learning, i.e. "dynamically" ensuring robustness on the particular given test samples than on all test samples. Previously, researchers consider the similar setting, but under the view of "dynamic defense" [Goo19, WJS+21, WYW20b]. [GKKM20] is the first paper to formalize transductive learning for robust learning. It proposes Rejectron, a selective transductive classifier, to handle general adversaries on test data, and presents novel theoretical guarantees. In comparison, our paper focuses on small-perturbation adversary as a more realistic robust learning setup. [CGW+21] formally defines the notion of transductive robustness as a maximin problem and presents a principled adaptive attack, GMSA. [MHS21] discusses robust transductive learning against small perturbation from a learning theory perspective and obtains corresponding sample complexity.

B Ablation Studies

As an effective adaptive attack for Tramèr-transform based detectors does not exist to our knowledge, we present an ablation of the adversarial loss.

Due to the computational cost, ablations are performed on synthetic data unless specified. We generate the data with 100 Gaussians (one per class) equally spaced in l_∞ with a separation of 3 units between means. The adversarial budget is 2 units, and we ensure that the data is sparse by generating 10 samples per class. The models are 10 layer feedforward networks with skip connections.

In each table of results, the row used in our later experiments is shown in bold.

B.1 $\mathcal{L}_{\text{REJ}}$

Loss	Rejection Rate	Robust Rejection Accuracy
Cross-Entropy	0.852	0.153
$\mathcal{L}_{\text{REJ}}$	**0.526**	**0.134**

While using cross-entropy directly is ideal for finding label-flipping perturbations, this approach does not take the rejection layer into account. The results shown above bear this out – cross-entropy finds more examples in the rejected region.

A successful attack needs to find samples which are distant from the decision boundary to avoid rejection: hence we directly penalize the nearness of the perturbed point to the decision boundary, motivating the decision-boundary loss as the the attack loss $\mathcal{L}'$ used by the inner PGD step.

B.2 Attack Algorithm

Attack	Defense	Rejection Rate	Robust Rejection Accuracy
GMSA	Transduction + Rejection	**0.531**	0.177
PGD	Transduction + Rejection	0.792	0.349
PGD	Rejection	0.680	**0.145**

All attacks are rejection-aware and optimize $\mathcal{L}_{\text{REJ}}$. Here, we can see that GMSA significantly outperforms even a rejection-aware transfer attack. While PGD is very strong against a fixed inductive model, it performs poorly compared to GMSA against a transductive defender.

B.3 Warm Start

Warm start (epochs)	Rejection Rate	Robust Rejection Accuracy
0	0.813	0.153
500	**0.531**	**0.177**
1000	0.830	0.171

The synthetic models are trained for 1000 epochs total; we see the best performance when the model has transductive regularization but is allowed to learn an initial baseline model before transductive regularization is used in training. Doing so reduces the risk of the regularization term harming performance.

B.4 Attack Radius

The theory suggests that incorporating rejection can allow a transductive learner to tolerate perturbations twice as large; we investigate how transduction and rejection affects the robustness as ϵ grows (models are adversarially trained with the corresponding ϵ and the detectors use a rejection radius of $\epsilon/2$). The results are shown for the natural choice of adversary, as in the experiment section (e.g. GMSA with $\mathcal{L}_{\text{REJ}}$ for the transduction+rejection). For detectors, the rejection rate scaling is shown.

We see that the combination of rejection and transduction does indeed maintain high accuracy for larger ϵ; at $\epsilon = 0.6$, it has 96.2% of the robust accuracy that transduction alone had for $\epsilon = 0.3$. This aligns with the theory, given the increased constant factors of $\text{OPT}_{\mathcal{U}^2}$ in Corollary 1 compared to the results for classifiers in [MHS21].

Note also the behavior of the inductive classifier: accuracy improves past $\epsilon = 0.6$. To see why, note that a model adversarially trained for $\epsilon \geq 1$ will return near-uniform predictions for all classes (resulting in a robust accuracy of approximately 10%, as seen), making finding adversarial examples slightly more difficult than for smaller ϵ where this does not occur. The decline in rejection rate for very large ϵ is a similar phenomenon.

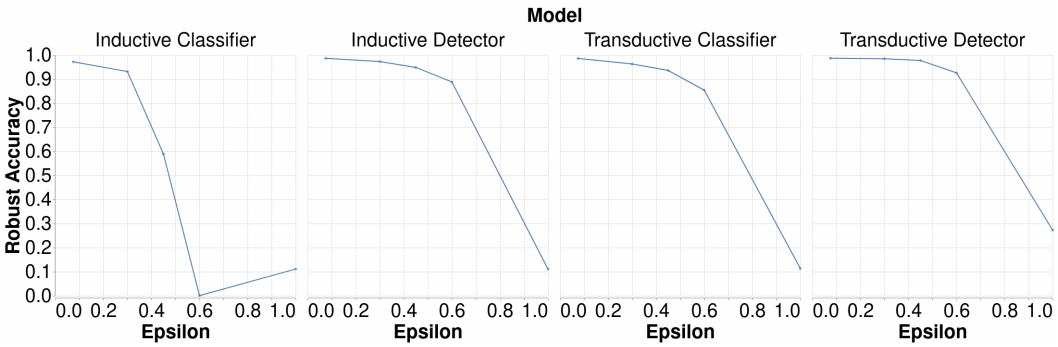

Figure 1: Robustness Scaling with ϵ: MNIST

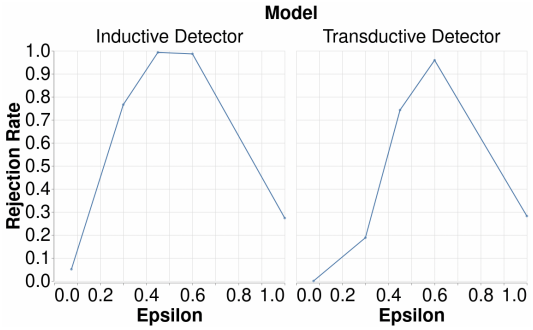

Figure 2: Rejection Rate Scaling with ϵ: MNIST

B.5 Rejection Radius

The natural approach given the theory is to reject samples where the model f is not robust with radius $\epsilon/2$; however, this may not be the strongest approach in practice. Below, we see that $\epsilon/5$ performs best on MNIST. If we consider a natural extension of Corollary 1 to variable rejection radius, the best choice may depend on the growth of $\text{OPT}_{\mathcal{U}_\epsilon^2}$ with ϵ, and hence dataset-specific tuning may be required for the best possible results. As expected, rejection rates rise steadily with the rejection radius, particularly adversarial rejection rates. In all cases, far more perturbed samples are rejected than clean samples.

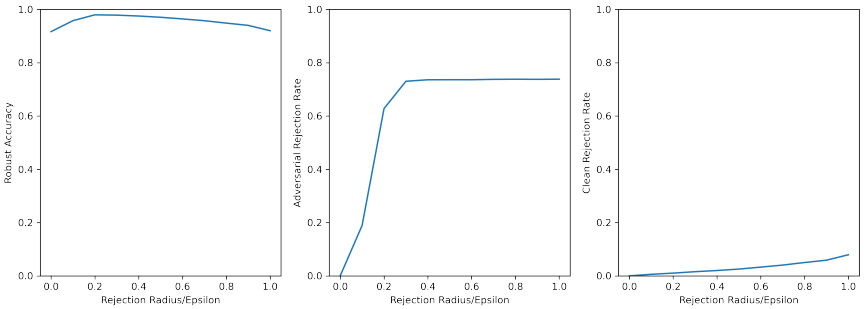

Figure 3: Effects of Rejection Radius on Robustness and Rejection Rates: MNIST

B.6 GMSA Method

We compare the results of GMSA_{AVG}, which optimizes the average loss of past iterations, and GMSA_{MIN}, which optimizes the worst-case loss. See [CGW+21]. We can see that while the two perform about the same on the transductive detectors (GMSA_{MIN} performs slightly better), GMSA_{AVG} is much stronger for the classifiers.

Defense	Attacker	MNIST		CIFAR-10	
		p_{REJ}	Robust accuracy	p_{REJ}	Robust accuracy
TLDR (no rejection)	GMSA_{AVG} ($\mathcal{L}_{\text{CE}}$)	–	0.900	–	0.516
TLDR (with rejection)	GMSA_{AVG} ($\mathcal{L}_{\text{REJ}}$)	0.796	0.968	0.195	0.744
TLDR (no rejection)	GMSA_{MIN} ($\mathcal{L}_{\text{CE}}$)	–	0.914	–	0.601
TLDR (with rejection)	GMSA_{MIN} ($\mathcal{L}_{\text{REJ}}$)	0.588	0.967	0.208	0.739

Table 4: Results for GMSA_{AVG} and GMSA_{MIN} targeting TLDR

C Adaptive Attacks for Defenses with Transduction and Rejection

As a strong attacker for transductively learned detectors does not exist to our knowledge, we present one here. We base the attack on GMSA [CGW+21]; our contribution is a novel loss function which, when maximized by GMSA, allows the attacker to avoid select perturbed points which are less likely to be rejected. Note furthermore that this attack, without GMSA, is effective against selective classifiers based on the transformation F (and via Tramèr's equivalency, selective classifiers in general). Our attack on a fixed model is described in 1.

Algorithm 1 Inductive Rejection-Aware Attack

Require: A model h, and a clean labelled test point (x, y), an adversarial budget of ϵ, and a radius used for rejection $\epsilon_{\text{defense}}$.

1: Search for a perturbation $\tilde{x}$ of x for which h predicts $\hat{y} \neq y$ robustly.

$$\tilde{x} = \arg\max_{\|\tilde{x}-x\|\leq\epsilon} \left[\mathcal{L}_{\text{CE}}(h^{softmax}(\tilde{x}), y) + \lambda \left\| \tilde{x} - \arg\max_{\|x'-\tilde{x}\|\leq\epsilon_{\text{defense}}} \mathcal{L}_{\text{DB,h}}(x') \right\| \right]$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss, $h^{softmax}$ returns the softmax activations of h and where $\mathcal{L}_{\text{DB,h}}(x') = \text{rank}_2 h^{softmax}(x') - \max h^{softmax}(x')$

2: **return** $\tilde{x}$

Attacking the detector $F(h)$ forces an adversary to solve a more difficult problem: that of finding adversarial examples which fool h but which are not rejected (i.e. where h is robustly fooled).

If $\mathcal{U}$ is defined by a maximum perturbation of ϵ in some metric, then this can be done by solving

$$\arg\max_{\tilde{x}\in\mathcal{U}(x)} \mathcal{L}_{\text{CE}}(h^{softmax}(\tilde{x}), y)$$

such that $\tilde{x}$ is at least ϵ from the decison boundary.

While the constraint

$$\min_{h(x')\neq h(\tilde{x})} \|\tilde{x} - x'\| \geq \epsilon$$

is most natural, it is equivalent to the condition that the point within ϵ of $\tilde{x}$ closest to the decision boundary is ϵ from $\tilde{x}$.

Now, we let $\mathcal{L}_{\text{DB,h}}(x')$ be the *surrogate loss* function on the closeness to the decision boundary. The loss increases for x' nears the decision boundary of h, and the condition is equivalent to

$$\left\| \tilde{x} - \arg\max_{\|x'-\tilde{x}\|\leq\epsilon} \mathcal{L}_{\text{DB,h}}(x') \right\| = \epsilon$$

We discuss our choice of $\mathcal{L}_{\text{DB,h}}$ in subsection C.1. b

We can then solve the optimization by solving the Lagrangian

$$\arg\max_{\tilde{x}\in\mathcal{U}(x)} \mathcal{L}_{\text{CE}}(h^{\text{softmax}}(\tilde{x}), y) + \lambda \left\| \tilde{x} - \arg\max_{\|x'-\tilde{x}\|\leq\epsilon} \mathcal{L}_{\text{DB,h}}(x') \right\|$$

Thus, we define the *rejection loss* to be

$$\mathcal{L}_{\text{REJ}}(\tilde{x}, y) := \mathcal{L}_{\text{CE}}(h^{\text{softmax}}(\tilde{x}), y) + \lambda \left\| \tilde{x} - \arg\max_{\|x'-\tilde{x}\|\leq\epsilon} \mathcal{L}_{\text{DB,h}}(x') \right\|. \tag{4}$$

By maximizing $\mathcal{L}_{\text{REJ}}$ via PGD can find examples $\tilde{x}$ which evade our defense.

C.1 $\mathcal{L}_{\text{DB,h}}$

It remains to find some $\mathcal{L}_{\text{DB,h}}$ which is maximized at the decision boundary.

As all models we considered use a softmax layer for their final predictions, the decision boundary is defined by those points for which the top-two logits are equal. This suggest a natural loss function which satisfies the requirements:

$$\mathcal{L}_{\text{DB,h}}(x') = \text{rank}_2 h^{softmax}(x') - \max h^{softmax}(x')$$

where $h^{softmax}$ returns the softmax activations of the model h.

C.2 Handling transductive rejection loss

To handle the clean branch of the robust risk, we apply a simple post-processing step after solving for $\tilde{\mathrm{x}}$: if the model does not robustly fail at $\tilde{\mathrm{x}}$, we replace $\tilde{\mathrm{x}}$ with x, allowing the model to be penalized for incorrect predictions or rejections at the clean points. To improve stability for the transductive attacks, we furthermore restrict the set of $\tilde{\mathrm{x}}$ replaced by clean points to those where $F(h)(x) \neq y$ (i.e. those where the attack $\tilde{\mathrm{x}} = x$ would succeed).

C.3 Transductive Attack Details

We present two rejection-aware transductive attacks: a stronger but more computationally intensive rejection-aware GMSA (Algorithm 2) and a weaker but faster rejection-aware transfer attack which takes the transductive robust rejection risk into account (Algorithm 3).

Algorithm 2 Rejection-Aware GMSA

Require: A clean training set T, a clean test set E, a transductive learning algorithm for classifiers $\mathbb{A}$, an adversarial budget of ϵ, *mode* either MIN or AVG, a radius used for rejection $\epsilon_{\text{defense}}$, and a maximum number of iterations $N \geq 1$.

Search for a perturbation of the test set which fools the model space induced by $(T, \mathcal{U}(E))$.
$E' = E$
$\hat{E} = E$
$err_{\max} = -\inf$
for i=0,...,N-1 **do**
  Train a transductive model on the perturbed data.
  $h^{(i)} = \mathbb{A}(T, \pi_x(E'))$

$$err = \frac{1}{|E'|} \sum_{i=1}^{|E'|} \mathbb{1}\left\{\left(F(h^{(i)})(\tilde{x}_i) \notin \{\tilde{y}_i\} \wedge \tilde{x}_i = x_i\right) \vee \left(F(h^{(i)})(\tilde{x}_i) \notin \{\tilde{y}_i, \perp\} \wedge \tilde{x}_i \neq x_i\right)\right\}$$

{The $\tilde{x}_i$ and the x_i are the i$^{\text{th}}$ datapoints of E' and E, repectively; y_i is the true label.}
  if $err_{\max} < err$ **then**
    $\hat{E} = E'$
  end if
  for $j = 1, \ldots, |E|$ **do**
    if *mode* = MIN **then**

$$\tilde{\mathrm{x}}_j = \arg \max_{\|\tilde{\mathrm{x}} - x_j\| \leq \epsilon} \min_{1 \leq k \leq i} \mathcal{L}_{\mathrm{REJ}_{h^{(k)}}}(\tilde{\mathrm{x}}, y_j)$$

    else

$$\tilde{\mathrm{x}}_j = \arg \max_{\|\tilde{\mathrm{x}} - x_j\| \leq \epsilon} \frac{1}{i} \sum_{k=1}^{i} \mathcal{L}_{\mathrm{REJ}_{h^{(k)}}}(\tilde{\mathrm{x}}, y_j)$$

    end if
    {Select whether to perturb by comparing success rates on the clean and perturbed samples.}

$$err_{\text{clean}} = \frac{1}{i} \sum_{0 \leq k \leq i} \mathbb{1}\left[F\left(h^{(k)}\right)(x_j) \neq y_j\right]$$

$$err_{\text{perturbed}} = \frac{1}{i} \sum_{0 \leq k \leq i} \mathbb{1}\left[F\left(h^{(k)}\right)(\tilde{\mathrm{x}}_j) \notin \{y_j, \perp\}\right]$$

    {Do not perturb if the perturbation reduces robust rejection accuracy less on average than leaving the points unchanged.}
    if $err_{\text{perturbed}} < err_{\text{clean}}$ **then**
      $\tilde{\mathrm{x}}_j = x_j$
    end if
    $E'_j = \tilde{\mathrm{x}}_j, y_i$
  end for
end for
return $\hat{E}$

Algorithm 3 TRANSDUCTIVE REJECTION-AWARE TRANSFER

Require: A model h, a clean labelled test point (x, y), an adversarial budget of ϵ, and a radius used for rejection $\epsilon_{\text{defense}}$.

Search for a perturbation $\tilde{x}$ of x for which h predicts $\hat{y} \neq y$ robustly.

1:
$$\tilde{x} = \arg \max_{\|\tilde{x} - x\| \leq \epsilon} \left[\mathcal{L}_{\text{CE}}(h^{\text{softmax}}(\tilde{x}), y) + \lambda \left\| \tilde{x} - \arg \max_{\|x' - \tilde{x}\| \leq \epsilon_{\text{defense}}} \mathcal{L}_{\text{DB,h}}(x') \right\| \right]$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss, h^{softmax} returns the softmax activations of h and where $\mathcal{L}_{\text{DB,h}}(x) = \text{rank}_2 h^{\text{softmax}}(x) - \max h^{\text{softmax}}(x)$.

If the attack did not succeed vs h (h does not robustly predict $\hat{y} \neq y$), check whether to leave x unperturbed.

2:
$$x' = \arg \max_{\|x' - \tilde{x}\| \leq \epsilon_{\text{defense}}} \mathcal{L}_{\text{CE}}(f(x'), h(\tilde{x}))$$

3: **if** $h(x') \neq h(\tilde{x}) \vee h(\tilde{x}) = y$ **then**
4: Leave x unperturbed if $F(h)$ rejects it, or if $h(x) \neq y$.
5:
$$x'' = \arg \max_{\|x'' - x\| \leq \epsilon_{\text{defense}}} \mathcal{L}_{\text{CE}}(f(x''), h(x))$$

6: **if** $h(x) \neq y \vee h(x'') \neq h(x)$ **then**
7: $\tilde{x} = x$
8: **end if**
9: **end if**
10: **return** $\tilde{x}$

D Appendix: Implementation Details

D.1 Defense

In our implementation, we begin to incorporate the transductive term in our objective (see Equation 3) after initially training the model with the inductive loss term only; this allows learning a better baseline before we begin to enforce robustness about the test points. In our experiments, we use the transductive loss in the final half of the training epochs, and put 85% of the weight on the inductive term afterwards.

D.2 Adaptive Attack

Solving for the perturbation $\tilde{x}$ by iteratively optimizing $\mathcal{L}_{\text{REJ}}$ poses several difficulties.

First, the rejection-avoidance term $\left\| \tilde{x} - \arg\max_{\|x' - \tilde{x}\| \leq \epsilon} \mathcal{L}_{\text{DB,h}}(x') \right\|$ is not differentiable with respect to $\tilde{x}$. While it is possible to approximate the derivative with the derivative of a proxy (e.g. differentiating though some fixed number of PGD steps, necessitating second-order optimization), this is extremely expensive and does not improve results in our experiments (see below).

Intuitively, we might see that this would be the case: if the decision boundary is smooth, we might expect the maximizers in $\mathcal{U}(x + \Delta)$ and $\mathcal{U}(x)$ to be the same for small Δ unless x' is near the border of $\mathcal{U}(x)$ given that $\mathcal{U}(x + \Delta) \approx \mathcal{U}(x)$. In this case, approximating x' as constant with respect to x is reasonable.

In addition, note that if $h(x) = y$, the adversary must find a $\tilde{x}$ where $h(\tilde{x}) \neq y$ which is not rejected: if maximizing $\mathcal{L}_{\text{REJ}}$ with PGD, the rejection-avoidance term penalizes moving $\tilde{x}$ towards the decision boundary. As this is necessary to find a valid attack (when $h(\tilde{x}) = y$ at initialization), we adjust λ adaptively during optimization by setting it to zero or negating it when $h(\tilde{x}) = y$.

E Proof Details

Before introducing the proof for the generalization results, we first need to make some additional definitions. We define the *empirical robust risk* as

$$\hat{R}_{\mathcal{U}}(h; S) = \sum_{(x,y) \in S} \left[\sup_{z \in \mathcal{U}(x)} \mathbb{1}\{h(z) \neq y\} \right]$$

And we can define the *empirical robust risk under rejection* accordingly:

$$\hat{\mathrm{R}}_{\mathcal{U},\mathrm{rej}}(h;S) = \sum_{(x,y)\in S}\left[\sup_{z\in\mathcal{U}(x)} \mathbb{1}\{h(x)\neq y \vee h(z)\notin\{y,\perp\}\}\right]$$

And we can define the corresponding robust empirical risk minimization procedure (under rejection) as follows:

$$\mathrm{RERM}_{\mathcal{H}}(S) := \operatorname*{argmin}_{h\in\mathcal{H}} \hat{\mathrm{R}}_{\mathcal{U}}(h;S)$$

$$\mathrm{RERM}_{\mathcal{H},\mathrm{rej}}(S) := \operatorname*{argmin}_{h\in\mathcal{H}} \hat{\mathrm{R}}_{\mathcal{U},\mathrm{rej}}(h;S)$$

E.1 Inductive Realizable Case

Definition 1 (Realizable Robust PAC Learnability under Rejection)**.** *For $\mathcal{Y} = \{0,1\}$, $\forall\epsilon,\delta \in (0,1), \mathcal{H} = \mathcal{H}_c \times \mathcal{H}_r$, the sample complexity of realizable robust (ϵ,δ) - PAC learning of $\mathcal{H}$ with respect adversary $\mathcal{U}$ under rejection, denoted as $\mathcal{M}_{RE}(\epsilon,\delta;\mathcal{H},\mathcal{U})$, is defined as the smallest $m \in \mathbb{N}\cup\{0\}$ for which there exists a learning rule $\mathcal{A} : (\mathcal{X}\times\mathcal{Y})^m \longmapsto (\mathcal{Y}\cup\{\perp\})^{\mathcal{X}}$ s.t. for every data distribution $\mathcal{D}$ over $(\mathcal{X}\times\mathcal{Y})^m$ where there exists a predictor with rejection option $h^* \in \mathcal{H}$ with 0 risk, $\mathrm{R}_{\mathcal{U},rej}(h^*;\mathcal{D}) = 0$ with probability at least $1-\delta$ over $S \sim \mathcal{D}^m$,*

$$\mathrm{R}_{\mathcal{U},rej}(\mathcal{A}(S);\mathcal{D}) \leq \epsilon$$

If no such m exists, $\mathcal{M}_{RE}(\epsilon,\delta;\mathcal{H},\mathcal{U}) = \infty$. We say that $\mathcal{H}$ is robustly PAC learnable under rejection in the realizable setting with respect to adversary $\mathcal{U}$ if $\forall\epsilon,\delta\in(0,1)$, $\mathcal{M}_{RE}(\epsilon,\delta;\mathcal{H},\mathcal{U})$ is finite.

Theorem 3 (Sample Complexity for Realizable Robust PAC Learning under Rejection)**.** *In the realizable setting, for any $\mathcal{H} = \mathcal{H}_c\times\mathcal{H}_r$ and $\mathcal{U}$, and any $\epsilon,\delta\in(0,1/2)$,*

$$\mathcal{M}_{RE}(\epsilon,\delta;\mathcal{H},\mathcal{U}) = 2^{O((d_r+d_c)\log(d_r+d_c))}\frac{1}{\varepsilon}\log\left(\frac{1}{\varepsilon}\right) + O\left(\frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right)\right) \tag{5}$$

where $d_r = \mathrm{VC}(\mathcal{H}_r), d_c = \mathrm{VC}(\mathcal{H}_c)$.

The idea of the proof is to adapt the classical sample compression argument [LW86] with improvements based on [MHS19, HKS19, MY16]. The generalization result in the inductive case (Theorem 1) directly comes from Equation 30.

Proof. First, we define the concept of *sample compression scheme* and *sample compression algorithm*.

Definition 2 (Sample Compression Scheme)**.** *Given $\forall m\in\mathbb{N}$ samples, $S\sim\mathcal{D}^m$, a sample compression scheme of size k is defined by the following pair of functions:*

1. *Compression function $\kappa : (\mathcal{X}\times\mathcal{Y})^m \mapsto (\mathcal{X}\times\mathcal{Y})^{\leq k}$.*
2. *Reconstruction function: $\rho : (\mathcal{X}\times\mathcal{Y})^{\leq k} \mapsto \mathcal{H}$.*

An algorithm $\mathcal{A}$ is a sample compression algorithm if $\exists\kappa,\rho$ s.t. $\mathcal{A}(S) = (\kappa\circ\rho)(S)$.

Fix $\epsilon,\delta\in(0,1)$, $m > 2(d_r+d_c)\log(d_r+d_c)$. Let the compression parameter, $n = O((d_r+d_c)\log(d_r+d_c))$. Let $\mathcal{D}$ be any distribution, then by realizability of the learner, $\inf_{h\in\mathcal{H}}\mathrm{R}_{\mathcal{U},\mathrm{rej}}(h;\mathcal{D}) = 0$. Thus, $\forall S$ sampled from $\mathcal{D}$, we have $\hat{\mathrm{R}}_{\mathcal{U},\mathrm{rej}}(\mathrm{RERM}_{\mathcal{H},\mathrm{rej}}(S);S) = 0$.

Compression First, we define a compression function κ as through the following inflation and discretization procedure. Given the training data $S := \{(x_i,y_i)\}_{i\in[m]}$, we define the following index mapping:

$$I(x) = \min\{i\in[m] : x\in\mathcal{U}(x_i)\}, \quad \forall x\in\bigcup_{i\in[m]}\mathcal{U}(x_i). \tag{6}$$

In another word, this index function outputs the first indexed training sample to include x in its neighborhood.

Then, we consider the set of RERM mapping learned by a size n subset of the training data:

$$\hat{\mathcal{H}} = \{\text{RERM}_{\mathcal{H},\text{rej}}(L) : L \subseteq S, |L| = n\}. \tag{7}$$

Note that

$$|\hat{\mathcal{H}}| \leq |\{L : L \subseteq S, |L| = n\}| = \binom{m}{n} \leq \left(\frac{em}{n}\right)^n. \tag{8}$$

Then, we inflate the data in the following way:

$$S_{\mathcal{U}} = \bigcup_{i\in[m]} \{(x_{I(x)}, x, y_{I(x)}) : x \in \mathcal{U}(x_i)\}. \tag{9}$$

Note that $x_{I(x)}$ can be different from x_i.

Let's define the following transformation T:

$$T(h)(x, x', y) := \mathbb{1}\{h(x) \neq y \vee h(x') \notin \{y, \perp\}\},\ h \in \mathcal{H}. \tag{10}$$

And we can obtain the transformed hypothesis class $T(\mathcal{H}) := \{T(h)|h \in \mathcal{H}\}$.

Now, we proceed to define the *dual space* $\mathcal{G}$ of $T(\mathcal{H})$ as the following set of functions.

$$\mathcal{G} := \{g_{(x,x',y)}|g_{(x,x',y)}(t) = t(x, x', y),\ t \in T(\mathcal{H})\}. \tag{11}$$

We denote the VC dimension of the dual space as $VC^*(T(\mathcal{H})) := \text{VC}(\mathcal{G})$.

By Lemma section E.1,

$$\text{VC}(T(\mathcal{H})) = O\left((d_r + d_c)\log(d_r + d_c)\right). \tag{12}$$

By the classic result in [Ass83], the VC dimension of the dual space satisfies the following inequality:

$$VC^*(T(\mathcal{H})) < 2^{\text{VC}(T(\mathcal{H}))+1}. \tag{13}$$

Now, we can construct the compressed dataset $\hat{S}_{\mathcal{U}}$ as the following. For each $(x, x', y) \in S_{\mathcal{U}}$, $\{g_{(x,x',y)}(t)\}_{t\in T(\hat{\mathcal{H}})}$ gives a labeling. When ranging over $(x, x', y) \in S_{\mathcal{U}}$, the labeling may not be unique. So for each unique labeling, we choose a representative $(x, x', y) \in S_{\mathcal{U}}$, and let $\hat{S}_{\mathcal{U}}$ be the set of the representatives. That is:

$$\hat{S}_{\mathcal{U}} = \left\{(x, x', y) \in S_{\mathcal{U}} \;\middle|\; \{g_{(x,x',y)}(t)\}_{t\in T(\hat{\mathcal{H}})} \text{ provides a unique labeling}\right\}. \tag{14}$$

Intuitively, $\hat{S}_{\mathcal{U}}$ split the infinite size dataset $S_{\mathcal{U}}$ into finite size according to the labeling of $T(\hat{\mathcal{U}})$ on the dual space. Thus, $\hat{S}_{\mathcal{U}}$ is not necessarily unique but always exists. And $|\hat{S}_{\mathcal{U}}|$ equals the number of possible labeling for $T(\hat{\mathcal{H}})$.

Let $d_* := VC(\mathcal{G}) = \text{VC}^*(T(\mathcal{H}))$ denote the VC-dimension of $\mathcal{G}$, the dual hypothesis class of $T(\hat{\mathcal{H}})$ [Ass83]. By applying Sauer's Lemma, we obtain that for $|T(\hat{\mathcal{H}})| > d_*$,

$$|\hat{S}_{\mathcal{U}}| \leq \left(\frac{e|T(\hat{\mathcal{H}})|}{d_*}\right)^{d_*}. \tag{15}$$

Let $n = \Theta(\text{VC}(T(\mathcal{H})))$. For $m \geq n$, we have

$$|\hat{S}_{\mathcal{U}}| \leq \left(e|T(\hat{\mathcal{H}})|\right)^{d_*} \tag{16}$$

$$\leq \left(e|\hat{\mathcal{H}}|\right)^{d_*} \tag{17}$$

$$\leq \left(e\left(\frac{em}{n}\right)^n\right)^{d_*} \tag{18}$$

$$\leq \left(\frac{e^2 m}{n}\right)^{nd_*} \tag{19}$$

$$= \left(\frac{e^2 m}{\text{VC}(T(\mathcal{H}))}\right)^{\Theta(\text{VC}(T(\mathcal{H}))\cdot\text{VC}(T(\mathcal{H}^*)))}. \tag{20}$$

Now we have obtain the compression map: $\kappa(S) = \hat{S}_{\mathcal{U}}$.

Reconstruction Now, we want to reconstruct a hypothesis from $\hat{S}_{\mathcal{U}}$. First, suppose we have a data distribution over $\hat{S}_{\mathcal{U}}$, denoted as $\mathcal{P}$. This distribution $\mathcal{P}$ over samples will be later used in the $\alpha-$boosting procedure.

Then, we sample the set of n i.i.d. samples from $\mathcal{P}$ and obtain $S' \in \hat{S}_{\mathcal{U}}$. By classic PAC learning guarantee [BEHW89], for $n = \Theta(\mathrm{VC}(T(\mathcal{H}))) = \Theta(d_r + d_c)\log(d_r + d_c)$, we have with non-zero probability $\forall t \in T(\mathcal{H})$ with $\sum_{(x,x',y)\in S'} t(x,x',y) = 0$ implies $\mathbb{E}_{(x,x',y)\sim\mathcal{P}} t(x,x',y) < 1/9$. Let $L = \{(x,y) : (x,x',y) \in S'\} \subseteq S$, and $t_{\mathcal{P}} = T(\mathrm{RERM}_{\mathcal{H},\mathrm{rej}}(L))$. Since $\hat{\mathrm{R}}_{\mathcal{U},\mathrm{rej}}(\mathrm{RERM}_{\mathcal{H},\mathrm{rej}}(L); L) = 0$, $\forall (x,x',y) \in S', t_{\mathcal{P}}(x,x',y) = 0$. Thus, $\forall \mathcal{P}$ over $\hat{S}_{\mathcal{U}}$, there exists a weak learner $t_{\mathcal{P}} \in T(\hat{\mathcal{H}})$, s.t. $\mathbb{E}_{(x,x',y)\sim\mathcal{P}} t_{\mathcal{P}}(x,x',y) < 1/9$.

Now, we use $t_{\mathcal{P}}$ as a *weak hypothesis* in a boosting algorithm, specifically $\alpha-$boost algorithm from [SF12] with $\hat{S}_{\mathcal{U}}$ as the dataset and $\mathcal{P}_k$ generated at each round of the algorithm. Then with appropriate choice of α, running $\alpha-$boosting for $K = O(\log(|\hat{S}_{\mathcal{U}}|))$ rounds gives a sequence of hypothesis $h_1, \ldots, h_K \in \hat{\mathcal{H}}$ and the corresponding $t_i = T(h_i)$ such that $\forall (x,x',y) \in \hat{S}_{\mathcal{U}}$,

$$\frac{1}{K}\sum_{k=1}^{K} \mathbb{1}\{h_k(x) \neq y \vee h_k(x') \notin \{y, \perp\}\} \tag{21}$$

$$= \frac{1}{K}\sum_{k=1}^{K} t_k(x,x',y) \tag{22}$$

$$< \frac{2}{9} < \frac{1}{3}. \tag{23}$$

Since $\hat{S}_{\mathcal{U}}$ includes all the unique labellings, $\frac{1}{K}\sum_{k=1}^{K} t_k(x,x',y) < \frac{1}{3},\ \forall (x,x',y) \in \hat{S}_{\mathcal{U}}$ implies

$$\frac{1}{K}\sum_{k=1}^{K} t_k(x,x',y) < \frac{1}{3},\ \forall (x,x',y) \in S_{\mathcal{U}}. \tag{24}$$

Let $\bar{h} := \mathrm{Majority}(h_1, \ldots, h_K)$, i.e., $\bar{h}$ outputs the prediction in $\mathcal{Y} \cup \{\perp\}$ that receives the most votes from $\{h_1, \ldots, h_K\}$. Then $\forall (x,x',y) \in \hat{S}_{\mathcal{U}}$,

$$\mathbb{1}\{\bar{h}(x) \neq y \vee \bar{h}(x') \notin \{y, \perp\}\} = 0. \tag{25}$$

This is because: (1) on x, less than $1/3$ of h_i's do not output y, so $\bar{h}(x) = y$; (2) on x', less than $1/3$ of h_i's do not output y or $\perp$, so the majority vote must be in y or $\perp$, i.e., $\bar{h}(x) \in \{y, \perp\}$.

In summary, given the same m training samples, we can simply find a $\bar{h}$ with 0 robust error on S:

$$\hat{\mathrm{R}}_{\mathcal{U},\mathrm{rej}}(\bar{h}; \mathcal{D}) = \sum_{i=1}^{m} \left[\sup_{z\in\mathcal{U}(x)} \mathbb{1}\{\bar{h}(x) \neq y \vee \bar{h}(z) \notin \{y, \perp\}\}\right] = 0. \tag{26}$$

Now we have the compression set with size:

$$nK = O(\mathrm{VC}(T(\mathcal{H}))\log(|\hat{S}_{\mathcal{U}}|)) = O(\mathrm{VC}(T(\mathcal{H}))^2\,\mathrm{VC}^*(T(\mathcal{H}))\log(m/\,\mathrm{VC}(T(\mathcal{H}))))$$

Then, we apply Lemma 11 of [MHS19] (Replacing $\mathrm{R}_{\mathcal{U}}$ with $\mathrm{R}_{\mathcal{U},\mathrm{rej}}$ still holds), we obtain for sufficiently large m, with probability at least $1-\delta$,

$$\mathrm{R}_{\mathcal{U},\mathrm{rej}}(\bar{h}; \mathcal{D}) \leq O\left(\mathrm{VC}(T(\mathcal{H}))^2\,\mathrm{VC}^*(T(\mathcal{H}))\frac{1}{m}\log(m/\,\mathrm{VC}(T(\mathcal{H})))\log(m) + \frac{1}{m}\log(1/\delta)\right). \tag{27}$$

We then can extend the sparsification procedure from [MY16, MHS19] to the rejection scenario. Since $t_1, \ldots, t_K \in T(\hat{\mathcal{H}})$, the classic uniform convergence results [SSBD14] implies that we can sample $N = O(\mathrm{VC}^*(T(\mathcal{H})))$ i.i.d. indices $i_1, \ldots, i_N \sim \mathrm{Uniform}([K])$ and obtain:

$$\sup_{(x,x',y)\in S_{\mathcal{U}}} \left|\frac{1}{N}\sum_{j=1}^{N} t_{i_j}(x,x',y) - \frac{1}{K}\sum_{i=1}^{T} t_i(x,x',y)\right| < \frac{1}{18} \tag{28}$$

And thus, we can combine Equation 21 with Equation 28 and obtain:

$$\forall (x, x', y) \in S_{\mathcal{U}}, \frac{1}{N}\sum_{j=1}^{N} t_{i_j}(x, x', y) \leq -\frac{1}{18} + \frac{1}{K}\sum_{i=1}^{K} t_k(x, x', y) < -\frac{1}{18} + \frac{4}{9} = \frac{1}{2}$$

we can further obtain an improved hypothesis $\bar{t}' := \text{Majority}(t_{i_1}, \dots t_{i_N})$ with

$$\bar{t}'(x, x', y) = 0, \forall (x, x', y) \in S_{\mathcal{U}}$$

Thus, the compression set has a reduced size:

$$nN = O(\text{VC}(T(\mathcal{H})) \cdot \text{VC}^*(T(\mathcal{H})))$$

Now, we apply Lemma 11 of [MHS19] and can obtain the following improved bound. Applying similar strategy from Equation 25, we can obtain

$$\bar{h}' := \text{Majority}(h_{i_1}, \dots h_{i_N}) = \rho(\hat{S}_{\mathcal{U}}) = \mathcal{A}(S) \tag{29}$$

which is our full reconstruction map.

Then, for large sample size $m \geq c\ \text{VC}(T(\mathcal{H}))\,\text{VC}^*(T(\mathcal{H}))$ (c is a sufficiently large constant), with probability at least $1 - \delta$,

$$\text{R}_{\mathcal{U},\text{rej}}(\bar{h}'; \mathcal{D}) \leq O\left(\text{VC}(T(\mathcal{H}))\,\text{VC}^*(\mathcal{H})\frac{1}{m}\log(m) + \frac{1}{m}\log(1/\delta)\right) \tag{30}$$

Plugging in Lemma section E.1 and solving for m gives

$$\mathcal{M}_{\text{RE}}(\epsilon, \delta; \mathcal{H}, \mathcal{U}) = 2^{O(\text{VC}(T(\mathcal{H})))}\frac{1}{\varepsilon}\log\left(\frac{1}{\varepsilon}\right) + O\left(\frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right)\right) \tag{31}$$

$$= 2^{O((d_r+d_c)\log(d_r+d_c))}\frac{1}{\varepsilon}\log\left(\frac{1}{\varepsilon}\right) + O\left(\frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right)\right) \tag{32}$$

□

Lemma [VC dimension of robust loss with rejection] Let $\text{VC}(\mathcal{H}_c) = d_c$, and $\text{VC}(\mathcal{H}_r) = d_r$. Then, $\text{VC}(T(\mathcal{H})) = O((d_r + d_c)\log(d_r + d_c))$.

Proof. Suppose $d > d_r + d_c$.

By definition of VC dimension, the max number of labeling of d points is 2^d on $h \in T(\mathcal{H})$. And since the label of h is a deterministic function of h_c and h_r, by Sauer's Lemma, the number of labeling of h is at most $O(d^{d_r}) \times O(d^{d_c}) = O(d^{d_r+d_c})$.

Thus, $2^d = O(d^{d_r+d_c})$. And $d = O((d_r + d_c)\log(d_r + d_c))$.

If $d < d_r + d_c$, $d = O(d_r + d_c)\log(d_r + d_c)$ by definition.

□

E.2 Inductive Agnostic Case

Now, we define notion of PAC learnability in the agnostic case under rejection setting as the follows:

Definition 3 (Robust PAC Learnability under Rejection)**.** *For $\mathcal{Y} = \{0, 1\}$, $\forall \epsilon, \delta \in (0, 1), \mathcal{H} = \mathcal{H}_c \times \mathcal{H}_r$, the sample complexity of robust (ϵ, δ) - PAC learning of $\mathcal{H}$ with respect to perturbation $\mathcal{U}$ under rejection, denoted as $\mathcal{M}_{AG}(\epsilon, \delta; \mathcal{H}, \mathcal{U})$, is defined as the smallest $m \in \mathbb{N} \cup \{0\}$ for which there exists a learning rule $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^m \longmapsto (\mathcal{Y} \cup \{\perp\})^{\mathcal{X}}$ s.t. for every data distribution $\mathcal{D}$ over $(\mathcal{X} \times \mathcal{Y})^m$,*

$$\text{R}_{\mathcal{U},rej}(\mathcal{A}(S); \mathcal{D}) \leq OPT_{\mathcal{U},rej} + \epsilon$$

with probability at least $1 - \delta$ over $S \sim \mathcal{D}^m$. If no such m exists, $\mathcal{M}_{AG}(\epsilon, \delta; \mathcal{H}, \mathcal{U}) = \infty$. We say that $\mathcal{H}$ is robustly PAC learnable under rejection if $\mathcal{M}_{AG}(\epsilon, \delta; \mathcal{H}, \mathcal{U})$ is finite for all $\epsilon, \delta \in (0, 1)$.

Lemma 1. *Let* $\mathcal{M}_{\text{RE}} = \mathcal{M}_{\text{RE}}(1/3, 1/3; \mathcal{H}, \mathcal{U})$. *Then,*

$$\mathcal{M}_{AG}(\epsilon, \delta; \mathcal{H}, \mathcal{U}) = O\left(\frac{\mathcal{M}_{\text{RE}}}{\epsilon^2}\log^2\left(\frac{\mathcal{M}_{\text{RE}}}{\epsilon}\right) + \frac{1}{\epsilon^2}\log\left(\frac{1}{\delta}\right)\right) \tag{33}$$

Proof. The proof detail follows exactly the same from the Proof of Theorem 8 from [MHS19] with the loss replaced. □

Theorem 4 (Sample Complexity for Agnostic Robust PAC Learning under Rejection)**.** *In the agnostic setting, for any* $\mathcal{H} = \mathcal{H}_c \times \mathcal{H}_r$ *and* $\mathcal{U}$*, and any* $\epsilon, \delta \in (0, 1/2)$*,*

$$\mathcal{M}_{\text{AG}}(\epsilon, \delta; \mathcal{H}, \mathcal{U}) = O\Big(\text{VC}(T(\mathcal{H}))\,\text{VC}^*(T(\mathcal{H}))\log\left(\text{VC}(T(\mathcal{H}))\,\text{VC}^*(T(\mathcal{H}))\right) \tag{34}$$

$$\frac{1}{\varepsilon^2}\log^2\left(\frac{\text{VC}(T(\mathcal{H}))\,\text{VC}^*(T(\mathcal{H}))}{\varepsilon}\right) + \frac{1}{\varepsilon^2}\log\left(\frac{1}{\delta}\right)\Big) \tag{35}$$

$$= 2^{O(\text{VC}(\mathcal{H}))}\frac{1}{\varepsilon^2}\log^2\left(\frac{1}{\varepsilon}\right) + O\left(\frac{1}{\varepsilon^2}\log\left(\frac{1}{\delta}\right)\right) \tag{36}$$

$$= 2^{O\left((d_r+d_c)\log(d_r+d_c)\right)}\frac{1}{\varepsilon^2}\log^2\left(\frac{1}{\varepsilon}\right) + O\left(\frac{1}{\varepsilon^2}\log\left(\frac{1}{\delta}\right)\right) \tag{37}$$

where $d_r = \text{VC}(\mathcal{H}_r), d_c = \text{VC}(\mathcal{H}_c)$.

Proof. Combining results from Lemma 1 and Theorem 3 gives the complexity result.

Solving Equation 36 gives the following generalization result given in Table 2

$$\Pr_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}^n}\left[\text{R}_{\mathcal{U},\text{rej}}(\mathcal{A}(\boldsymbol{x},\boldsymbol{y});\mathcal{D}) \le \epsilon\right] \ge 1-\delta$$

where $\epsilon = O\left(\sqrt{\frac{2^{\text{VC}(T(\mathcal{H}))}+\log(1/\delta)}{n}}\right)$. □

E.3 Transductive Realizable Case

In general the set of optimally learned classifiers Δ is defined as follows [MHS21]:

$$\Delta^{\mathcal{U}}_{\mathcal{H}}(\boldsymbol{z},\boldsymbol{y},\tilde{\boldsymbol{z}}) = \begin{cases} \{h \in \mathcal{H} : \text{R}_{\mathcal{U}^{-1}}(h;\boldsymbol{z},\boldsymbol{y}) = 0 \wedge \text{R}_{\mathcal{U}^{-1}}(h;\tilde{\boldsymbol{z}}) = 0\} & \text{(Realizable Case)} \\ \arg\min_{h\in\mathcal{H}} \max\{\text{R}_{\mathcal{U}^{-1}}(h;\boldsymbol{z},\boldsymbol{y}), \text{R}_{\mathcal{U}^{-1}}(h;\tilde{\boldsymbol{z}})\} & \text{(Agnostic Case)} \end{cases}$$

where

$$\text{R}_{\mathcal{U}}(h;\boldsymbol{z},\boldsymbol{y}) = \sup_{\tilde{\boldsymbol{x}}\in\mathcal{U}(\boldsymbol{z})}\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}\{h(\tilde{\text{x}}_i) \ne y_i\}$$

and

$$\text{R}_{\mathcal{U}}(h;\boldsymbol{z}) = \text{R}_{\mathcal{U}}(h;\boldsymbol{z},h(\boldsymbol{z})).$$

Then, we define the *relaxed robust shattering dimension* following [MHS21]:

Definition 4 (Relaxed Robust Shattering Dimension)**.** *A sequence* $z_1, \ldots, z_k \in \mathcal{X}$ *is relaxed* $\mathcal{U}$*-robustly shattered by* $\mathcal{H}$*, if* $\forall y_1, \ldots, y_k \in \{\pm 1\}$*:* $\exists x_1^{y_1}, \ldots, x_k^{y_k} \in \mathcal{X}$ *and* $\exists h \in \mathcal{H}$ *such that* $z_i \in \mathcal{U}(x_i^{y_i})$ *and* $h(\mathcal{U}(x_i^{y_i})) = y_i$*,* $\forall 1 \le i \le k$*. The relaxed* $\mathcal{U}$*-robust shattering dimension* $rdim_{\mathcal{U}}(\mathcal{H})$ *is defined as the largest* k *for which there exist* k *points that are relaxed* $\mathcal{U}$*-robustly shattered by* $\mathcal{H}$*.*

Theorem 5. *For any* $n \in \mathbb{N}$*,* $\delta > 0$*, class* $\mathcal{H}$*, perturbation set* $\mathcal{U}$*, and distribution* $\mathcal{D}$ *over* $\mathcal{X}\times\mathcal{Y}$ *satisfying* $OPT_{\mathcal{U}^{-1}\mathcal{U}} = 0$*:*

$$\Pr_{\substack{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}^n \\ (\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\sim\mathcal{D}^n}}\left[\begin{array}{l} \forall\, z_0 \in \mathcal{U}(\boldsymbol{x}), \forall\, \tilde{z}_0 \in \mathcal{U}(\tilde{\boldsymbol{x}}), \forall\, z \in \mathcal{U}(z_0), \forall\, \tilde{z} \in \mathcal{U}(\tilde{z}_0), \\ \forall \hat{h} \in F\left(\Delta^{\mathcal{U}}_{\mathcal{H}}(z_0,\boldsymbol{y},\tilde{z}_0) \cap \Delta^{\mathcal{U}}_{\mathcal{H}}(z,\boldsymbol{y},\tilde{z})\right) : \text{err}^{rej}_{\tilde{\boldsymbol{x}},\tilde{z},\tilde{\boldsymbol{y}}}(\hat{h}) \le \epsilon \end{array}\right] \ge 1-\delta$$

where $\epsilon = \frac{rdim_{\mathcal{U}^{-1}}(\mathcal{H})\log(2n)+\log(1/\delta)}{n} \le \frac{\text{VC}(\mathcal{H})\log(2n)+\log(1/\delta)}{n}$.

Proof. We adapt the strategy of Tramèr's Theorem 5 for the rejection scenario.

By setting $z = z_0, \tilde{z} = \tilde{z}_0$ and applying Theorem 1 of [MHS21], we obtain the following

$$\Pr_{\substack{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}^n \\ (\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\sim\mathcal{D}^n}}\left[\forall\, z_0 \in \mathcal{U}(\boldsymbol{x}), \forall\, \tilde{z}_0 \in \mathcal{U}(\tilde{\boldsymbol{x}}), \forall h \in \Delta^{\mathcal{U}}_{\mathcal{H}}(z_0, \boldsymbol{y}, \tilde{z}_0) : \mathrm{err}_{\tilde{z}_0,\tilde{\boldsymbol{y}}}(h) \le \epsilon\right] \ge 1 - \delta \tag{38}$$

as $\mathrm{OPT}_{\mathcal{U}^{-1}(\mathcal{U})} = 0$.

Suppose $(\boldsymbol{x}, \boldsymbol{y}), (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) \sim \mathcal{D}^n$. Now, let $z_0 \in \mathcal{U}(\boldsymbol{x})$, $z \in \mathcal{U}(z_0)$, $\tilde{z}_0 \in \mathcal{U}(\tilde{\boldsymbol{x}})$, $\tilde{z} \in \mathcal{U}(\tilde{z}_0)$, and $\hat{\mathrm{h}} \in F\left(\Delta^{\mathcal{U}}_{\mathcal{H}}(z_0, \boldsymbol{y}, \tilde{z}_0) \cap \Delta^{\mathcal{U}}_{\mathcal{H}}(z, \boldsymbol{y}, \tilde{z})\right)$.

Write $\hat{\mathrm{h}} = F(h)$ for some $h \in \Delta^{\mathcal{U}}_{\mathcal{H}}(z_0, \boldsymbol{y}, \tilde{z}_0) \cap \Delta^{\mathcal{U}}_{\mathcal{H}}(z, \boldsymbol{y}, \tilde{z})$.

From Equation 38 (replacing z with z_0 and $\tilde{z}$ with $\tilde{z}_0$), it is enough to show that

$$\mathrm{err}^{\mathrm{rej}}_{\tilde{\boldsymbol{x}},\tilde{z},\tilde{\boldsymbol{y}}}(\hat{h}) \le \mathrm{err}_{\tilde{z}_0,\tilde{\boldsymbol{y}}}(h).$$

Suppose that $\hat{\mathrm{h}}$ incurs an error under rejection at point $\tilde{\mathrm{z}}_i$; it is enough to show that h incurs an error at $\tilde{\mathrm{z}}_{0_\mathrm{i}}$. Furthermore, note that because $h \in \Delta^{\mathcal{U}}_{\mathcal{H}}(z_0, \boldsymbol{y}, \tilde{z}_0)$, we have that $h(\mathcal{U}^{-1}(\tilde{\mathrm{z}}_{0_\mathrm{i}})) = \{h(\tilde{\mathrm{z}}_{0_\mathrm{i}})\}$ as $\tilde{\mathrm{z}}_{0_\mathrm{i}} \in \mathcal{U}^{-1}(\tilde{\mathrm{z}}_{0_\mathrm{i}})$. Write $h(\tilde{\mathrm{z}}_{0_\mathrm{i}}) = \hat{\mathrm{y}}_\mathrm{i}$. Hence, $\hat{\mathrm{h}}$ does not reject $\tilde{\mathrm{z}}_{0_\mathrm{i}}$ and $\hat{\mathrm{h}}(\tilde{\mathrm{z}}_{0_\mathrm{i}}) = \hat{\mathrm{y}}_\mathrm{i}$. Similarly, as $h \in \Delta^{\mathcal{U}}_{\mathcal{H}}(z, \boldsymbol{y}, \tilde{z})$, $\hat{\mathrm{h}}$ does not reject $\tilde{\mathrm{z}}_i$, and as $\tilde{\mathrm{z}}_i \in U(\tilde{\mathrm{z}}_{0_\mathrm{i}})$ and so $\tilde{\mathrm{z}}_{0_\mathrm{i}} \in U^{-1}(\tilde{\mathrm{z}}_i)$, we have $\hat{\mathrm{h}}(\tilde{\mathrm{z}}_i) = \hat{\mathrm{y}}_\mathrm{i} = h(\tilde{\mathrm{z}}_{0_\mathrm{i}})$.

We have one of the following:

1. $\hat{\mathrm{h}}(\tilde{\mathrm{z}}_i) \neq \tilde{\mathrm{y}}_i$ and $\tilde{\mathrm{z}}_i = \tilde{\mathrm{x}}_i$
2. $\hat{\mathrm{h}}(\tilde{\mathrm{z}}_i) \notin \{\tilde{\mathrm{y}}_i, \perp\}$ and $\tilde{\mathrm{z}}_i \neq \tilde{\mathrm{x}}_i$

In either case, we have that $h(\tilde{\mathrm{z}}_{0_\mathrm{i}}) = \hat{\mathrm{h}}(\tilde{\mathrm{z}}_i) \neq \tilde{\mathrm{y}}_i$ and so h makes an error at $\tilde{\mathrm{z}}_{0_\mathrm{i}}$.

Consider Case 2 further. If we assume that $\mathcal{U} = \mathcal{U}^{-1}$ as well, then as $\tilde{z} \in \mathcal{U}(\tilde{z}_0)$ we have $\tilde{z} \in U^{-1}(\tilde{z}_0)$, so $h(\tilde{\mathrm{z}}_i) = h(\tilde{\mathrm{z}}_{0_\mathrm{i}})$ since $\hat{\mathrm{h}}(\tilde{\mathrm{z}}_i) \neq \perp$, and so we must have $\hat{\mathrm{h}}(\tilde{\mathrm{z}}_i) = h(\tilde{\mathrm{z}}_i) = h(\tilde{\mathrm{z}}_{0_\mathrm{i}})$, so h makes an error at $\tilde{\mathrm{z}}_{0_\mathrm{i}}$. Hence, only case (1) requires $h(\mathcal{U}^{-1}(\tilde{\mathrm{z}}_i)) = \{h(\tilde{\mathrm{z}}_i)\}$, so the requirement $h \in \Delta^{\mathcal{U}}_{\mathcal{H}}(z_0, \boldsymbol{y}, \tilde{z}_0) \cap \Delta^{\mathcal{U}}_{\mathcal{H}}(z, \boldsymbol{y}, \tilde{z}')$ (where $\tilde{z}'$ is the subset of $\tilde{z}$ where $z_i = \boldsymbol{x}_i$) rather than $h \in \Delta^{\mathcal{U}}_{\mathcal{H}}(z_0, \boldsymbol{y}, \tilde{z}_0) \cap \Delta^{\mathcal{U}}_{\mathcal{H}}(z, \boldsymbol{y}, \tilde{z})$ would be sufficient, as would $h \in \Delta^{\mathcal{U}}_{\mathcal{H}}(z_0, \boldsymbol{y}, \tilde{z}_0) \cap \Delta^{\mathcal{U}}_{\mathcal{H}}(\boldsymbol{x}, \boldsymbol{y}, \tilde{\boldsymbol{x}})$. □

Remark: The OPT requirement is only needed to apply the result from [MHS21]; it does also show that there exists a consistent hypothesis, but this is not required to show the result. To see this, note that we can assume without loss of generality that $\Delta^{\mathcal{U}}_{\mathcal{H}}(z_0, \boldsymbol{y}, \tilde{z}_0) \cap \Delta^{\mathcal{U}}_{\mathcal{H}}(z, \boldsymbol{y}, \tilde{z})$ is nonempty, as if it were empty, there would be no $\hat{h} \in F\left(\Delta^{\mathcal{U}}_{\mathcal{H}}(z_0, \boldsymbol{y}, \tilde{z}_0) \cap \Delta^{\mathcal{U}}_{\mathcal{H}}(z, \boldsymbol{y}, \tilde{z})\right)$ with $\mathrm{err}^{\mathrm{rej}}_{\tilde{\boldsymbol{x}},\tilde{z},\tilde{\boldsymbol{y}}}(\hat{h}) > \epsilon$.

Sample Complexity Given ϵ and δ, we need

$$\frac{\mathrm{rdim}_{\mathcal{U}^{-1}}(\mathcal{H})\log(2n) + \log(1/\delta)}{n} \le \epsilon$$

for the result to hold.

Now, noting that $\log(2n) = 1 + \log n \le 1 + \sqrt{n}$ for $n \ge 16$; hence we need to solve for the n such that

$$\frac{\mathrm{rdim}_{\mathcal{U}^{-1}}(\mathcal{H})(1 + \sqrt{n}) + \log(1/\delta)}{n} = \epsilon$$

or, equivalently

$$\frac{\mathrm{rdim}_{\mathcal{U}^{-1}}(\mathcal{H}) + \log(\frac{1}{\delta}) + \sqrt{n}}{n} = \epsilon$$

or

$$\sqrt{n} = n\epsilon - \mathrm{rdim}_{\mathcal{U}^{-1}}(\mathcal{H}) - \log(\frac{1}{\delta})$$

or

$$n = n^2\epsilon^2 - 2\epsilon\left(\mathrm{rdim}_{\mathcal{U}^{-1}}(\mathcal{H}) + \log(\frac{1}{\delta})\right)n + \left(\mathrm{rdim}_{\mathcal{U}^{-1}}(\mathcal{H}) + \log(\frac{1}{\delta})\right)^2$$

or

$$n^2\epsilon^2 - \left(2\epsilon\left(\mathrm{rdim}_{\mathcal{U}^{-1}}(\mathcal{H}) + \log(\frac{1}{\delta})\right) + 1\right)n + \left(\mathrm{rdim}_{\mathcal{U}^{-1}}(\mathcal{H}) + \log(\frac{1}{\delta})\right)^2 = 0.$$

Solving, the result holds if

$$n \geq \frac{2\epsilon\left(\mathrm{rdim}_{\mathcal{U}^{-1}}(\mathcal{H}) + \log(\frac{1}{\delta})\right) + 1 + \sqrt{(2\epsilon\left(\mathrm{rdim}_{\mathcal{U}^{-1}}(\mathcal{H}) + \log(\frac{1}{\delta})\right) + 1)^2 - 4\left(\mathrm{rdim}_{\mathcal{U}^{-1}}(\mathcal{H}) + \log(\frac{1}{\delta})\right)^2\epsilon^2}}{2\epsilon^2}$$

$$= O\left(\frac{\mathrm{rdim}_{\mathcal{U}^{-1}}(\mathcal{H}) + \log(\frac{1}{\delta})}{\epsilon} + \frac{\sqrt{\mathrm{rdim}_{\mathcal{U}^{-1}}(\mathcal{H}) + \log(\frac{1}{\delta})}}{\epsilon^{\frac{3}{2}}}\right)$$

and, similarly, using

$$\frac{\mathrm{rdim}_{\mathcal{U}^{-1}}(\mathcal{H})\log(2n) + \log(1/\delta)}{n} \leq \frac{\mathrm{VC}(\mathcal{H})\log(2n) + \log(1/\delta)}{n}$$

we have the result if

$$n = O\left(\frac{\mathrm{VC}(\mathcal{H}) + \log(\frac{1}{\delta})}{\epsilon} + \frac{\sqrt{\mathrm{VC}(\mathcal{H}) + \log(\frac{1}{\delta})}}{\epsilon^{\frac{3}{2}}}\right)$$

Simplified Result To obtain a bound which does not involve an intermediate perturbation step, we may let

$$\Delta^{\mathcal{U}}_{\mathrm{rej},\mathcal{H}}(z, \boldsymbol{y}, \tilde{z}) = \bigcap_{z' \in \mathcal{U}^{-1}(z)\cup\{z\},\, \tilde{z}' \in \mathcal{U}^{-1}(\tilde{z})\cup\{\tilde{z}\}} \Delta^{\mathcal{U}}_{\mathcal{H}}(z', \boldsymbol{y}, \tilde{z}')$$

Note that for common classes of perturbations, we can simplify the definition of Δ_{rej}. Note that the conditions of the theorem hold for perturbations defined via ϵ-balls in a metric.

Lemma 2. *In the realizable case, if* $\mathcal{U} = \mathcal{U}^{-1}$,

$$\Delta^{\mathcal{U}}_{rej,\mathcal{H}}(z, \boldsymbol{y}, \tilde{z}) = \Delta^{\mathcal{U}^2}_{\mathcal{H}}(z, \boldsymbol{y}, \tilde{z})$$

Proof. As $x \in U(x)$ for all x, note that $\mathcal{U}^{-1}(z) \cup \{z\} = \mathcal{U}^{-1}(z)$ and $\mathcal{U}^{-1}(\tilde{z}) \cup \{\tilde{z}\} = \mathcal{U}^{-1}(\tilde{z})$; we will use this simplification of the definition of Δ_{rej} below.

Suppose $h \in \Delta^{\mathcal{U}}_{\mathrm{rej},\mathcal{H}}(z, \boldsymbol{y}, \tilde{z})$. Then by the definitions of Δ_{rej} and Δ, for any $z' \in \mathcal{U}^{-1}(z)$, $\tilde{z}' \in \mathcal{U}^{-1}(\tilde{z})$, we have that, for any $\boldsymbol{x} \in \mathcal{U}^{-1}(z')$ and $\tilde{\boldsymbol{x}} \in \mathcal{U}^{-1}(\tilde{z}')$, $h(x_i) = h(z'_i)$ and $h(\tilde{\mathrm{x}}_i) = h(\tilde{\mathrm{z}}'_i)$. But since $\mathcal{U} = \mathcal{U}^{-1}$, $z \in \mathcal{U}^{-1}(z')$ and $\tilde{z} \in \mathcal{U}^{-1}(\tilde{z}')$, so $h(z'_i) = h(z_i)$ and $h(\tilde{z}'_i) = h(\tilde{z}_i)$ for all $z' \in \mathcal{U}^{-1}(z)$, $\tilde{z}' \in \mathcal{U}^{-1}(\tilde{z})$. Hence, for any $\boldsymbol{x} \in \mathcal{U}^{-2}(z)$ and $\tilde{\boldsymbol{x}} \in \mathcal{U}^{-2}(\tilde{z})$, we have that $h(x_i) = h(z_i)$ and $h(\tilde{\mathrm{x}}_i) = h(\tilde{\mathrm{z}}_i)$, and so

$$\Delta^{\mathcal{U}}_{\mathrm{rej},\mathcal{H}}(z, \boldsymbol{y}, \tilde{z}) \subseteq \Delta^{\mathcal{U}^2}_{\mathcal{H}}(z, \boldsymbol{y}, \tilde{z})$$

Now, if $h \in \Delta^{\mathcal{U}^2}_{\mathcal{H}}(z, \boldsymbol{y}, \tilde{z})$, we have that, for any $\boldsymbol{x} \in \mathcal{U}^{-2}(z)$ and $\tilde{\boldsymbol{x}} \in \mathcal{U}^{-2}(\tilde{z})$, $h(x_i) = h(z_i)$ and $h(\tilde{\mathrm{x}}_i) = h(\tilde{\mathrm{z}}_i)$. Now, suppose $z' \in \mathcal{U}^{-1}(z)$, $\tilde{z}' \in \mathcal{U}^{-1}(\tilde{z})$. Since $x \in \mathcal{U}(x)$ for all x, $z' \in \mathcal{U}^{-2}(z)$, $\tilde{z}' \in \mathcal{U}^{-2}(\tilde{z})$ as well. Hence, $h(z'_i) = h(z_i)$ and $h(\tilde{z}'_i) = h(\tilde{z}_i)$. Now, if $\boldsymbol{x} \in \mathcal{U}^{-1}(z')$ and $\tilde{\boldsymbol{x}} \in \mathcal{U}^{-1}(\tilde{z}')$, we have $\boldsymbol{x} \in \mathcal{U}^{-2}(z)$ and $\tilde{\boldsymbol{x}} \in \mathcal{U}^{-2}(\tilde{z})$ and so $h(x_i) = h(z_i)$ and $h(\tilde{\mathrm{x}}_i) = h(\tilde{z}_i)$. But then $h(x_i) = h(z'_i)$ and $h(\tilde{\mathrm{x}}_i) = h(\tilde{z}'_i)$. Hence, we have that

$$\Delta^{\mathcal{U}^2}_{\mathcal{H}}(z, \boldsymbol{y}, \tilde{z}) \subseteq \Delta^{\mathcal{U}}_{\mathrm{rej},\mathcal{H}}(z, \boldsymbol{y}, \tilde{z})$$

and the result follows. □

Now, by the above and from Theorem 5 we may immediately derive Theorem 2 by noting that if $\mathcal{U} = \mathcal{U}^{-1}$, $\mathcal{U}^{-1}\mathcal{U} = \mathcal{U}^2$, and if $\hat{\mathrm{h}} \in F(\Delta^{\mathcal{U}^2}_{\mathcal{H}}(z, \boldsymbol{y}, \tilde{z})) = F(\Delta^{\mathcal{U}}_{\mathrm{rej},\mathcal{H}}(z, \boldsymbol{y}, \tilde{z}))$ then we have $\hat{\mathrm{h}} \in F\left(\Delta^{\mathcal{U}}_{\mathcal{H}}(z_0, \boldsymbol{y}, \tilde{z}_0) \cap \Delta^{\mathcal{U}}_{\mathcal{H}}(z, \boldsymbol{y}, \tilde{z})\right)$ for some $z_0 \in \mathcal{U}(\boldsymbol{x}) \cap \mathcal{U}^{-1}(z)$ and $\tilde{z}_0 \in \mathcal{U}(\tilde{\boldsymbol{x}}) \cap \mathcal{U}^{-1}(\tilde{z})$.

E.4 Transductive Agnostic Case

While the constant factors on $OPT_{\mathcal{U}^2}$ do increase compared to the result without rejection [MHS21], note that, if $\mathcal{U}$ can be decomposed into a form $\mathcal{U} = (\mathcal{U}^{1/2})^2$ where $\mathcal{U}^{1/2} = \mathcal{U}^{-1/2}$ (as with standard perturbations in l_p), we obtain a bound which depends on $\mathrm{OPT}_{\mathcal{U}}$ rather than $\mathrm{OPT}_{\mathcal{U}^2}$, enabling much stronger guarantees if $\mathrm{OPT}_{\mathcal{U}} << \mathrm{OPT}_{\mathcal{U}^2}$. Note that as $\forall x\, x \in \mathcal{U}(x)$, $\forall x\, \mathcal{U}(x) \subseteq \mathcal{U}^2(x)$, and so $\mathrm{OPT}_{\mathcal{U}} \leq \mathrm{OPT}_{\mathcal{U}^2}$.

Theorem 6. *For any $n \in \mathbb{N}$, $\delta > 0$, class $\mathcal{H}$, perturbation set $\mathcal{U}$, and distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$:*

$$\Pr_{\substack{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}^n \\ (\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\sim\mathcal{D}^n}}\left[\begin{array}{l} \forall\, z_0 \in \mathcal{U}(\boldsymbol{x}), \forall\, \tilde{z}_0 \in \mathcal{U}(\tilde{\boldsymbol{x}}), \forall\, z \in \mathcal{U}(z_0), \forall\, \tilde{z} \in \mathcal{U}(\tilde{z}_0), \\ \forall \hat{h} \in F(\Delta^{\mathcal{U}}_{\mathcal{H}}(z_0, \boldsymbol{y}, \tilde{z}_0) \cap \Delta^{\mathcal{U}}_{\mathcal{H}}(z, \boldsymbol{y}, \tilde{z})) : \mathrm{err}^{rej}_{\tilde{\boldsymbol{x}},\tilde{z},\tilde{\boldsymbol{y}}}(\hat{h}) \leq \epsilon \end{array}\right] \geq 1 - \delta$$

where $\epsilon = \min\left\{4\mathrm{OPT}_{\mathcal{U}^{-1}\mathcal{U}} + O\left(\sqrt{\frac{\mathrm{VC}(\mathcal{H})+\log(1/\delta)}{n}}\right), 5OPT_{\mathcal{U}^{-1}\mathcal{U}} + O\left(\sqrt{\frac{\mathrm{rdim}\,\mathcal{U}(\mathcal{H})\ln(2n)+\ln(1/\delta)}{n}}\right)\right\}$.

Proof. Suppose $(\boldsymbol{x}, \boldsymbol{y}), (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) \sim \mathcal{D}^n$. Now, let $z_0 \in \mathcal{U}(\boldsymbol{x})$, $z \in \mathcal{U}(z_0)$, $\tilde{z}_0 \in \mathcal{U}(\tilde{\boldsymbol{x}})$, $\tilde{z} \in \mathcal{U}(\tilde{z}_0)$, and $\hat{\mathrm{h}} \in F(\Delta^{\mathcal{U}}_{\mathcal{H}}(z_0, \boldsymbol{y}, \tilde{z}_0))$.

Write $\hat{\mathrm{h}} = F(h)$ for some $h \in \Delta^{\mathcal{U}}_{\mathcal{H}}(z_0, \boldsymbol{y}, \tilde{z}_0)$. First, by a proof similar to that in Theorem 5 we can show that

$$\mathrm{err}^{\mathrm{rej}}_{\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}},\tilde{z}}(\hat{h}) \leq \mathrm{err}_{\tilde{z}_0,\tilde{\boldsymbol{y}}}(h) + \mathrm{R}_{\mathcal{U}}(h; \tilde{z}_0) + \mathrm{R}_{\mathcal{U}}(h; \tilde{\boldsymbol{x}}).$$

To see this, note that the same argument as before holds for i such that $\hat{\mathrm{h}}$ does not reject $\tilde{\mathrm{z}}_{0_i}$ or $\tilde{\mathrm{z}}_i$ such that $\tilde{\mathrm{z}}_i = x_i$, so we have that

$$\mathrm{err}^{\mathrm{rej}}_{\tilde{\boldsymbol{x}}',\tilde{\boldsymbol{y}}',\tilde{\boldsymbol{z}}'}(\hat{h}) \leq \mathrm{err}_{\tilde{\boldsymbol{z}}'_0,\tilde{\boldsymbol{y}}'}(h)$$

where $\tilde{\boldsymbol{x}}', \tilde{\boldsymbol{y}}', \tilde{\boldsymbol{z}}', \ldots$ correspond to the points not rejected; now, as $\mathrm{R}_{\mathcal{U}}(h; \tilde{\mathrm{z}}_0)$ is the fraction of $\tilde{\mathrm{z}}_{0_i}$ rejected by h and similarly for $\mathrm{R}_{\mathcal{U}}(h; \tilde{\boldsymbol{x}})$, the result follows as the indicator for error is ≤ 1.

Now, we consider two intermediate results from the proof of Theorem 2 of [MHS21].

VC Dimension Bound We have

$$\Pr_{\substack{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}^n \\ (\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\sim\mathcal{D}^n}}\left[\begin{array}{l} \forall\, z \in \mathcal{U}(\boldsymbol{x}), \forall\, \tilde{z} \in \mathcal{U}(\tilde{\boldsymbol{x}}), \forall h \in \Delta^{\mathcal{U}}_{\mathcal{H}}(z, \boldsymbol{y}, \tilde{z}) : \\ \max\{\mathrm{R}_{\mathcal{U}}(h; z, \boldsymbol{y}), \mathrm{R}_{\mathcal{U}}(h; \tilde{z})\} \leq \mathrm{OPT}_{\mathcal{U}^{-1}\mathcal{U}} + \epsilon \wedge \left|\mathrm{err}_{\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}}(h) - \mathrm{err}_{\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}}(h)\right| \leq \epsilon \end{array}\right] \geq 1 - 2\delta$$

where

$$\epsilon = O\left(\sqrt{\frac{\mathrm{VC}(\mathcal{H}) + \log(1/\delta)}{n}}\right)$$

from which it is shown that

$$\Pr_{\substack{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}^n \\ (\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\sim\mathcal{D}^n}}\left[\begin{array}{l} \forall\, z \in \mathcal{U}(\boldsymbol{x}), \forall\, \tilde{z} \in \mathcal{U}(\tilde{\boldsymbol{x}}), \forall h \in \Delta^{\mathcal{U}}_{\mathcal{H}}(z, \boldsymbol{y}, \tilde{z}) : \\ \max\{\mathrm{R}_{\mathcal{U}}(h; z, \boldsymbol{y}), \mathrm{R}_{\mathcal{U}}(h; \tilde{z})\} \leq \mathrm{OPT}_{\mathcal{U}^{-1}\mathcal{U}} + \epsilon \wedge \mathrm{err}_{\tilde{z},\tilde{\boldsymbol{y}}}(h) \leq 2\mathrm{OPT}_{\mathcal{U}^{-1}\mathcal{U}} + 3\epsilon \end{array}\right] \geq 1 - 2\delta$$

Now, recall that, by the above, for any $(\boldsymbol{x}, \boldsymbol{y}), (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) \sim \mathcal{D}^n$, $z_0 \in \mathcal{U}(\boldsymbol{x})$, $z \in \mathcal{U}(z_0)$, $\tilde{z}_0 \in \mathcal{U}(\tilde{\boldsymbol{x}})$, $\tilde{z} \in \mathcal{U}(\tilde{z}_0)$, and $\hat{\mathrm{h}} = F(h) \in F(\Delta^{\mathcal{U}}_{\mathcal{H}}(z_0, \boldsymbol{y}, \tilde{z}_0))$, we have

$$\mathrm{err}^{\mathrm{rej}}_{\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}},\tilde{z}}(\hat{h}) \leq \mathrm{err}_{\tilde{z}_0,\tilde{\boldsymbol{y}}}(h) + \mathrm{R}_{\mathcal{U}}(h; \tilde{z}_0) + \mathrm{R}_{\mathcal{U}}(h; \tilde{\boldsymbol{x}})$$

and so, if $\mathrm{R}_{\mathcal{U}}(h; \tilde{z}_0) \leq \mathrm{OPT}_{\mathcal{U}^{-1}\mathcal{U}} + \epsilon$, $\mathrm{R}_{\mathcal{U}}(h; \tilde{\boldsymbol{x}}) \leq \mathrm{OPT}_{\mathcal{U}^{-1}\mathcal{U}} + \epsilon$, and $\mathrm{err}_{\tilde{z}_0,\tilde{\boldsymbol{y}}}(h) \leq 2\mathrm{OPT}_{\mathcal{U}^{-1}\mathcal{U}} + 3\epsilon$ we have

$\mathrm{err}^{\mathrm{rej}}_{\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}},\tilde{z}}(\hat{h}) \leq \mathrm{err}_{\tilde{z}_0,\tilde{\boldsymbol{y}}}(h) + \mathrm{R}_{\mathcal{U}}(h; \tilde{z}_0) + \mathrm{R}_{\mathcal{U}}(h; \tilde{\boldsymbol{x}}) \leq \mathrm{OPT}_{\mathcal{U}^{-1}\mathcal{U}} + \epsilon + \mathrm{OPT}_{\mathcal{U}^{-1}\mathcal{U}} + \epsilon + 2\mathrm{OPT}_{\mathcal{U}^{-1}\mathcal{U}} + 3\epsilon = 4\mathrm{OPT}_{\mathcal{U}^{-1}\mathcal{U}} + 5\epsilon$

and hence

$$\Pr_{\substack{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}^n \\ (\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\sim\mathcal{D}^n}}\left[\begin{array}{l} \forall\, z_0 \in \mathcal{U}(\boldsymbol{x}), \forall\, \tilde{z}_0 \in \mathcal{U}(\tilde{\boldsymbol{x}}), \forall\, z \in \mathcal{U}(z_0), \forall\, \tilde{z} \in \mathcal{U}(\tilde{z}_0), \forall \hat{h} \in F(\Delta^{\mathcal{U}}_{\mathcal{H}}(z_0, \boldsymbol{y}, \tilde{z}_0)) : \\ \mathrm{err}^{\mathrm{rej}}_{\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}},\tilde{z}}(\hat{h}) \leq 4\mathrm{OPT}_{\mathcal{U}^{-1}\mathcal{U}} + 5\epsilon \end{array}\right] \geq 1 - 2\delta.$$

noting that $\tilde{\boldsymbol{x}} \in \mathcal{U}(\tilde{\boldsymbol{x}})$ by assumption.

Relaxed Robust Shattering Dimension Bound We have

$$\Pr_{\substack{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}^n\\(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\sim\mathcal{D}^n}}\left[\begin{array}{l}\forall\, \boldsymbol{z}\in\mathcal{U}(\boldsymbol{x}),\forall\,\tilde{\boldsymbol{z}}\in\mathcal{U}(\tilde{\boldsymbol{x}}),\forall h\in\Delta^{\mathcal{U}}_{\mathcal{H}}(\boldsymbol{z},\boldsymbol{y},\tilde{\boldsymbol{z}}):\\ \max\left\{\mathrm{R}_{\mathcal{U}}(h;\boldsymbol{z},\boldsymbol{y}),\mathrm{R}_{\mathcal{U}}(h;\tilde{\boldsymbol{z}})\right\}\leq \mathrm{OPT}_{\mathcal{U}^{-1}\mathcal{U}}+\tilde{\epsilon}\wedge\left|\mathrm{err}_{\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}}(h)-\mathrm{err}_{\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}}(h)\right|\leq\tilde{\epsilon}\end{array}\right]\geq 1-\delta$$

where

$$\tilde{\epsilon}\leq OPT_{\mathcal{U}^{-1}\mathcal{U}}+\epsilon_0+\sqrt{\frac{\mathrm{rdim}\,\mathcal{U}(\mathcal{H})\ln(2n)+\ln(1/\delta)}{n}}$$

and

$$\epsilon_0=\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}$$

from which it is shown that

$$\Pr_{\substack{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}^n\\(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\sim\mathcal{D}^n}}\left[\begin{array}{l}\forall\, \boldsymbol{z}\in\mathcal{U}(\boldsymbol{x}),\forall\,\tilde{\boldsymbol{z}}\in\mathcal{U}(\tilde{\boldsymbol{x}}),\forall h\in\Delta^{\mathcal{U}}_{\mathcal{H}}(\boldsymbol{z},\boldsymbol{y},\tilde{\boldsymbol{z}}):\\ \max\left\{\mathrm{R}_{\mathcal{U}}(h;\boldsymbol{z},\boldsymbol{y}),\mathrm{R}_{\mathcal{U}}(h;\tilde{\boldsymbol{z}})\right\}\leq \mathrm{OPT}_{\mathcal{U}^{-1}\mathcal{U}}+\tilde{\epsilon}\wedge\mathrm{err}_{\tilde{\boldsymbol{z}},\tilde{\boldsymbol{y}}}(h)\leq 3\mathrm{OPT}_{\mathcal{U}^{-1}\mathcal{U}}+3\epsilon_0+\sqrt{\frac{\mathrm{rdim}\,\mathcal{U}(\mathcal{H})\ln(2n)+\ln(1/\delta)}{n}}\end{array}\right]\geq 1-\delta$$

and so we have (by an argument similar to that for the VC bound)

$$\Pr_{\substack{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}^n\\(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\sim\mathcal{D}^n}}\left[\begin{array}{l}\forall\, \boldsymbol{z}_0\in\mathcal{U}(\boldsymbol{x}),\forall\,\tilde{\boldsymbol{z}}_0\in\mathcal{U}(\tilde{\boldsymbol{x}}),\forall\,\boldsymbol{z}\in\mathcal{U}(\boldsymbol{z}_0),\forall\,\tilde{\boldsymbol{z}}\in\mathcal{U}(\tilde{\boldsymbol{z}}_0),\forall\hat{h}\in F(\Delta^{\mathcal{U}}_{\mathcal{H}}(\boldsymbol{z}_0,\boldsymbol{y},\tilde{\boldsymbol{z}}_0)):\\ \mathrm{err}^{\mathrm{rej}}_{\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}},\tilde{\boldsymbol{z}}}(\hat{h})\leq 5\mathrm{OPT}_{\mathcal{U}^{-1}\mathcal{U}}+5\epsilon_0+2\sqrt{\frac{\mathrm{rdim}\,\mathcal{U}(\mathcal{H})\ln(2n)+\ln(1/\delta)}{n}}\end{array}\right]\geq 1-\delta.$$

rBy combining these results and simplifying, we are done. □

As in the realizable case, we can immediately derive the following corollary. However, we cannot simplify the definition of Δ_{rej} as before; see Lemma 3.

Corollary 1. *For any $n\in\mathbb{N}$, $\delta>0$, class $\mathcal{H}$, perturbation set $\mathcal{U}$ where $\mathcal{U}=\mathcal{U}^{-1}$, and distribution $\mathcal{D}$ over $\mathcal{X}\times\mathcal{Y}$:*

$$\Pr_{\substack{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}^n\\(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\sim\mathcal{D}^n}}\left[\forall\,\boldsymbol{z}\in\mathcal{U}^2(\boldsymbol{x}),\forall\tilde{\boldsymbol{z}}\in\mathcal{U}^2(\tilde{\boldsymbol{x}}),\forall\hat{h}\in F(\Delta^{\mathcal{U}}_{rej,\mathcal{H}}(\boldsymbol{z},\boldsymbol{y},\tilde{\boldsymbol{z}})):\mathrm{err}^{rej}_{\tilde{\boldsymbol{x}},\tilde{\boldsymbol{z}},\tilde{\boldsymbol{y}}}(\hat{h})\leq\epsilon\right]\geq 1-\delta$$

where $\epsilon=\min\left\{4\mathrm{OPT}_{\mathcal{U}^2}+O\left(\sqrt{\frac{\mathrm{VC}(\mathcal{H})+\log(1/\delta)}{n}}\right),5OPT_{\mathcal{U}^2}+O\left(\sqrt{\frac{\mathrm{rdim}\,\mathcal{U}(\mathcal{H})\ln(2n)+\ln(1/\delta)}{n}}\right)\right\}$.

Lemma 3. *In the agnostic case, we have that if $\mathcal{U}=\mathcal{U}^{-1}$,*

$$\Delta^{\mathcal{U}}_{rej,\mathcal{H}}(\boldsymbol{z},\boldsymbol{y},\tilde{\boldsymbol{z}})\subseteq\Delta^{\mathcal{U}^2}_{\mathcal{H}}(\boldsymbol{z},\boldsymbol{y},\tilde{\boldsymbol{z}})$$

Proof. As in Lemma 2, since $x\in U(x)$ for all x we have that $\mathcal{U}^{-1}(\boldsymbol{z})\cup\{\boldsymbol{z}\}=\mathcal{U}^{-1}(\boldsymbol{z})$ and $\mathcal{U}^{-1}(\tilde{\boldsymbol{z}})\cup\{\tilde{\boldsymbol{z}}\}=\mathcal{U}^{-1}(\tilde{\boldsymbol{z}})$, and we will use this simplification of the definition of Δ_{rej} below.

Recalling the following definitions:

$$\mathrm{R}_{U^{-2}}(h;\boldsymbol{z},\boldsymbol{y})=\max_{\boldsymbol{z}'\in\mathcal{U}^{-1}(\boldsymbol{z})}\mathrm{R}_{U^{-1}}(h;\boldsymbol{z}',\boldsymbol{y})$$

and

$$\mathrm{R}_{U^{-2}}(h;\tilde{\boldsymbol{z}})=\max_{\tilde{\boldsymbol{z}}'\in\mathcal{U}^{-1}(\tilde{\boldsymbol{z}})}\mathrm{R}_{U^{-1}}(h;\tilde{\boldsymbol{z}}'),$$

note that

$$\begin{aligned}\mathrm{R}_{\mathcal{U}^{-2}}(h;\tilde{\boldsymbol{z}})=\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}\left\{\exists\tilde{\boldsymbol{x}}_i\in\mathcal{U}^{-2}\left(\tilde{\boldsymbol{z}}_i\right):h\left(\tilde{\boldsymbol{x}}_i\right)\neq h\left(\tilde{\boldsymbol{z}}_i\right)\right\}&=\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}\left\{\exists\tilde{\boldsymbol{z}}'_i\in\mathcal{U}^{-1}\left(\tilde{\boldsymbol{z}}_i\right)\exists\tilde{\boldsymbol{x}}_i\in\mathcal{U}^{-1}\left(\tilde{\boldsymbol{z}}'_i\right):h\left(\tilde{\boldsymbol{x}}_i\right)\neq h\left(\tilde{\boldsymbol{z}}_i\right)\right\}\\&=\max_{\tilde{\boldsymbol{z}}'_i\in\mathcal{U}^{-1}(\tilde{\boldsymbol{z}}_i)}\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}\left\{\exists\tilde{\boldsymbol{x}}_i\in\mathcal{U}^{-1}\left(\tilde{\boldsymbol{z}}'_i\right):h\left(\tilde{\boldsymbol{x}}_i\right)\neq h\left(\tilde{\boldsymbol{z}}_i\right)\right\}\\&=\max_{\tilde{\boldsymbol{z}}'_i\in\mathcal{U}^{-1}(\tilde{\boldsymbol{z}}_i)}\mathrm{R}_{\mathcal{U}^{-1}}(h;\tilde{\boldsymbol{z}}')\end{aligned}$$

where the last equality holds as $\tilde{z} \in \mathcal{U}^{-1}(\tilde{z})$ and as $\mathcal{U} = \mathcal{U}^{-1}$, which together show that if for some $\tilde{z}_i$ and $\tilde{z}_i' \in \mathcal{U}^{-1}(\tilde{z}_i)$ we have that $h(\tilde{z}_i') \neq h(\tilde{z}_i)$, that (since we must have $\{\tilde{z}_i, \tilde{z}_i'\} \in \mathcal{U}^{-1}(\tilde{z}_i')$) both $\exists \tilde{x}_i \in \mathcal{U}^{-1}(\tilde{z}_i') : h(\tilde{x}_i) \neq h(\tilde{z}_i)$ and $\exists \tilde{x}_i \in \mathcal{U}^{-1}(\tilde{z}_i') : h(\tilde{x}_i) \neq h(\tilde{z}_i')$.

We can derive a result for $\mathrm{R}_{U^{-2}}(h; \boldsymbol{z}, \boldsymbol{y})$ similarly.

Suppose $h \in \Delta^{\mathcal{U}}_{\mathrm{rej},\mathcal{H}}(\boldsymbol{z}, \boldsymbol{y}, \tilde{\boldsymbol{z}})$. Then, h minimizes $\max\{\mathrm{R}_{\mathcal{U}^{-1}}(h; \boldsymbol{z}', \boldsymbol{y}), \mathrm{R}_{\mathcal{U}^{-1}}(h; \tilde{\boldsymbol{z}}')\}$ for all $\boldsymbol{z}' \in \mathcal{U}^{-1}(\boldsymbol{z})$, $\tilde{\boldsymbol{z}}' \in \mathcal{U}^{-1}(\tilde{\boldsymbol{z}})$, so by the above, h must also minimize

$$\begin{aligned}\max_{\boldsymbol{z}' \in \mathcal{U}^{-1}(\boldsymbol{z}), \tilde{\boldsymbol{z}}' \in \mathcal{U}^{-1}(\tilde{\boldsymbol{z}})} \max\{\mathrm{R}_{\mathcal{U}^{-1}}(h; \boldsymbol{z}', \boldsymbol{y}), \mathrm{R}_{\mathcal{U}^{-1}}(h; \tilde{\boldsymbol{z}}')\} &= \max\left\{\max_{\boldsymbol{z}' \in \mathcal{U}^{-1}(\boldsymbol{z})} \mathrm{R}_{\mathcal{U}^{-1}}(h; \boldsymbol{z}', \boldsymbol{y}), \max_{\tilde{\boldsymbol{z}}' \in \mathcal{U}^{-1}(\tilde{\boldsymbol{z}})} \mathrm{R}_{\mathcal{U}^{-1}}(h; \tilde{\boldsymbol{z}}')\right\} \\ &= \max\{\mathrm{R}_{U^{-2}}(h; \tilde{\boldsymbol{z}}), \mathrm{R}_{U^{-2}}(h; \boldsymbol{z}, \boldsymbol{y})\}\end{aligned}$$

and so $h \in \Delta^{\mathcal{U}^{-1}\mathcal{U}}_{\mathcal{H}}(\boldsymbol{z}, \boldsymbol{y}, \tilde{\boldsymbol{z}})$.

However, minimizing $\max_{\boldsymbol{z}' \in \mathcal{U}^{-1}(\boldsymbol{z}), \tilde{\boldsymbol{z}}' \in \mathcal{U}^{-1}(\tilde{\boldsymbol{z}})} \max\{\mathrm{R}_{\mathcal{U}^{-1}}(h; \boldsymbol{z}', \boldsymbol{y}), \mathrm{R}_{\mathcal{U}^{-1}}(h; \tilde{\boldsymbol{z}}')\}$ does not necessarily imply that h minimizes $\max\{\mathrm{R}_{\mathcal{U}^{-1}}(h; \boldsymbol{z}', \boldsymbol{y}), \mathrm{R}_{\mathcal{U}^{-1}}(h; \tilde{\boldsymbol{z}}')\}$ for all $\boldsymbol{z}' \in \mathcal{U}^{-1}(\boldsymbol{z}), \tilde{\boldsymbol{z}}' \in \mathcal{U}^{-1}(\tilde{\boldsymbol{z}})$, so the reverse may not hold. □

F Extension to Unbalanced Data

We provide a sketch of a proof that allows extending Theorem 1 of [MHS21] to unbalanced training and test sets; however, for simplicity, we will work with the original form. The assumptions are the same, except that we have n training points and m test points.

The proof is exactly as before up to the "Finite robust labelings" portion (which points are and are not labelled don't matter up to then and the symmetry arguments still apply). The basic idea of determining the probability of zero loss on the training and test sets and error $> \epsilon$ on the test examples with permutation still applies. Let $E_{\sigma,\boldsymbol{x}}$ be the event that there exists a labelling $\hat{\mathrm{h}}(\boldsymbol{x}_{\sigma(1:n+m)})$ in the allowable set where this occurs.1

We have

$$Pr_{\sigma}\left[E_{\sigma,\boldsymbol{x}}\right] \leq \Pr_{\sigma}\left[\exists \hat{\mathrm{h}} \in \Pi^{\mathcal{U}}_{\mathcal{H}}(x_1, \ldots, x_{n+m}) : err_{\boldsymbol{x}_{\sigma(1:n)}, \boldsymbol{y}_{\sigma(1:n)}}(\hat{\mathrm{h}}) = 0 \wedge err_{\boldsymbol{x}_{\sigma(n:n+m)}, \boldsymbol{y}_{\sigma(n:n+m)}}(\hat{\mathrm{h}}) > \epsilon\right]$$

and, as in [MHS21], note the probability of choosing such a perturbation σ for a fixed $\hat{\mathrm{h}}$ is at most

$$\left(\frac{\mathrm{m}}{\mathrm{n}+\mathrm{m}}\right)^{s} \leq \left(\frac{\mathrm{m}}{\mathrm{n}+\mathrm{m}}\right)^{\lceil \epsilon m \rceil} = \left(\frac{\mathrm{n}+\mathrm{m}}{\mathrm{m}}\right)^{-\lceil \epsilon m \rceil} \leq \left(\frac{\mathrm{n}+\mathrm{m}}{\mathrm{m}}\right)^{\lceil -\epsilon m \rceil}$$

if we assume the number of total errors $s \geq \lceil \epsilon m \rceil$ without loss of generality (otherwise, $err > \epsilon$ would be impossible).

Hence, by a union bound,

$$Pr_{\sigma}\left[E_{\sigma,\boldsymbol{x}}\right] \leq \left|\Pi^{\mathcal{U}}_{\mathcal{H}}(x_1, \ldots, x_{n+m})\right| \left(\frac{\mathrm{n}+\mathrm{m}}{\mathrm{m}}\right)^{\lceil -\epsilon m \rceil}$$

and so

$$Pr_{\sigma}\left[E_{\sigma,\boldsymbol{x}}\right] \leq (n+m)^{rdim_{\mathcal{U}^{-1}}(\mathcal{H})} \left(\frac{\mathrm{n}+\mathrm{m}}{\mathrm{m}}\right)^{\lceil -\epsilon m \rceil}$$

by Sauer's Lemma (in the form of Lemma 3 of [MHS21]).

Now, we to bound the probability by δ, we need

$$(n+m)^{rdim_{\mathcal{U}^{-1}}(\mathcal{H})} \left(\frac{\mathrm{n}+\mathrm{m}}{\mathrm{m}}\right)^{\lceil -\epsilon m \rceil} \leq \delta$$

which, solving, gives us

$$\epsilon \geq \frac{rdim_{\mathcal{U}^{-1}}(\mathcal{H}) \log_{\frac{n+m}{m}}(n+m) + \log_{\frac{n+m}{m}} \frac{1}{\delta}}{m} = \frac{rdim_{\mathcal{U}^{-1}}(\mathcal{H}) \log(n+m) + \log \frac{1}{\delta}}{m \log\left(1 + \frac{m}{n}\right)}$$

Which reduces to the original result if $n = m$ (note that the logarithms are base-2).

Corollary If we fix $n + m$, $\mathcal{H}$, and δ, the guarantee is strongest (i.e. we minimize ϵ) when $n = m$. To see this, consider the denominator. Write $\alpha = \frac{m}{n}$. Then, we wish to maximize $n\alpha \log(1 + \alpha)$ (or equivalently $f(\alpha) = \alpha \log(1 + \alpha)$ subject to $\alpha \geq 0$. Now, note that $f'(\alpha) = \log(1 + \alpha) - 1 = 0$ when $\alpha = 1$, i.e. when $m = n$.

Also, we can see from the result above, that if we fix m and δ, then the minimum value of ϵ tends towards ∞ as $n \to \infty$, so there does not necessarily exist a labelled training set sampled from $\mathcal{D}$ which provides a guarantee with high probability of arbitrarily low error on a fixed test set.