Journal of Machine Learning Research 24 (2023) 1-43 Submitted 4/22; Revised 10/22; Published 3/23

An Analysis of Robustness of Non-Lipschitz Networks

Maria-Florina Balcan NINAMF@CS.CMU.EDU
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213, USA

Avrim Blum AVRIM@TTIC.EDU
Toyota Technological Institute at Chicago
6045 S Kenwood Ave, Chicago, IL 60637, USA

Dravyansh Sharma DRAVYANS@CS.CMU.EDU
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213, USA

Hongyang Zhang HONGYANG.ZHANG@UWATERLOO.CA
University of Waterloo
200 University Ave W, Waterloo, ON N2L 3G1, Canada

Editor: Moritz Hardt

Abstract

Despite significant advances, deep networks remain highly susceptible to adversarial attack. One fundamental challenge is that small input perturbations can often produce large movements in the network's final-layer feature space. In this paper, we define an attack model that abstracts this challenge, to help understand its intrinsic properties. In our model, the adversary may move data an arbitrary distance in feature space but only in random low-dimensional subspaces. We prove such adversaries can be quite powerful: defeating any algorithm that must classify any input it is given. However, by allowing the algorithm to abstain on unusual inputs, we show such adversaries can be overcome when classes are reasonably well-separated in feature space. We further provide strong theoretical guarantees for setting algorithm parameters to optimize over accuracy-abstention trade-offs using data-driven methods. Our results provide new robustness guarantees for nearest-neighbor style algorithms, and also have application to contrastive learning, where we empirically demonstrate the ability of such algorithms to obtain high robust accuracy with low abstention rates. Our model is also motivated by *strategic classification*, where entities being classified aim to manipulate their observable features to produce a preferred classification, and we provide new insights into that area as well.

Keywords: adversarial machine learning, abstention, nearest-neighbor algorithms, data-driven algorithm design, strategic classification

1. Introduction

A substantial body of work has shown that deep networks can be highly susceptible to adversarial attacks, in which minor changes to the input lead to incorrect, even bizarre classifications (Szegedy et al., 2014; Moosavi-Dezfooli et al., 2016; Madry et al., 2018; Su et al., 2019; Brendel et al., 2018). Much of this work has considered bounded ℓ_p-norm attacks, though many other forms of attack are considered as well (Brown et al., 2018; Engstrom et al., 2017; Gilmer et al., 2018; Xiao et al., 2018;

Alaifari et al., 2019). What these results have in common is that changes that either are imperceptible or should be irrelevant to the classification task can lead to drastically different network behaviors.

One key reason[1] for this vulnerability to attacks is the non-Lipschitzness of typical neural networks: small but adversarial movements in the input space can produce large perturbations in the feature space (Yang et al., 2020b; Szegedy et al., 2014; Goodfellow et al., 2014). This ability of an adversary to produce large movements in feature space appears to be at the heart of many of the successful attacks to date. If we assume that non-Lipschitzness is important for good performance on natural data, then it is crucial to understand to what extent this property makes a network intrinsically susceptible to attacks.

In this work, we propose and analyze an abstract attack model designed to focus on this question of the intrinsic vulnerability of non-Lipschitz networks, and what might help to make such networks robust. In particular, suppose an adversary, by making an imperceptible change to an input x, can cause its representation $F(x)$ in feature space (the final layer of the network) to move by an arbitrary amount: will such an adversary always win? Clearly if the adversary can modify $F(x)$ by an arbitrary amount *in an arbitrary direction*, then yes, because it can then move $F(x)$ into the classification region of any other class it wishes. But what if the adversary can modify $F(x)$ by an arbitrary amount but only in a *random* direction or within a random low-dimensional subspace (which it cannot control)? In this case, we show an interesting dichotomy: if the classifier must output a classification on any input it is given, then indeed the adversary will still win, no matter how well-separated the natural data points from different classes are in feature space and no matter what decision surface the classifier uses. Specifically, for any data distribution and any decision surface, there must exist at least one class such that the adversary wins with significant probability on random examples of that class. However, if we provide the classifier the ability to abstain, then we show it can defeat such an adversary (while maintaining a low abstention rate on natural data) with a nearest-neighbor style approach under fairly reasonable conditions on the distribution of natural data in feature space. Moreover, we show these conditions can often be achieved using contrastive learning. Our results also hold for generalizations of these models, such as directions that are not completely random. More broadly, our results provide a theoretical explanation for the importance of allowing abstaining, or selective classification, in the presence of adversarial attacks that exploit network non-Lipschitzness. Our results also provide new understanding of the robustness of nearest-neighbor algorithms.

A second motivation of our work comes from the area of *strategic classification*, where the concern is that entities being classified may try to manipulate their observable features to achieve a preferred outcome. Consider, for example, a public rating system used to classify companies into those that are good to work for and those that are not. Naturally, companies want to be viewed as being good to work for. So, they may try to modify any easy-to-manipulate features used by the system in order to achieve a positive classification, even if this does not change their true status. For example, perhaps the system uses the ratio of managers to associates, which the company can manipulate arbitrarily (from 0 to infinity) by changing employee titles, without actually changing pay or responsibilities. Suppose we assume agents (the companies) have a small number of parameters they can manipulate arbitrarily, and that there is an unknown linear function that maps changes in these parameters to movement in feature space. In this case, our results can provide some guidance. Our negative results imply that for any non-abstaining classifier, there must be at least one class such that for most examples from that class, manipulation in a random direction has a significant chance

[1] Additional explanations include the presence of brittle features that are human incomprehensible (Ilyas et al., 2019), and the location of the classification boundary relative to the submanifold of sampled data (Tanay and Griffin, 2016).

of being successful; whereas our positive results imply that by using the ability to abstain, we can be secure against manipulation in most low-dimensional subspaces. Note that this is very similar to the model used by Kleinberg and Raghavan (2019) (see also Alon et al. 2020; Shavit et al. 2020) who assume that the "effort conversion matrix" mapping changes in manipulable parameters to changes in observable features is *known* (or at least can be learned through experimentation, Shavit et al., 2020); our results provide insight into what can be done when it is *unknown*, and the classifier must be fielded before any manipulations are observed.

In addition to providing a formal separation between algorithms that can abstain and those that cannot, our work also yields an interesting trade-off between robustness and accuracy (Tsipras et al., 2019; Zhang et al., 2019; Raghunathan et al., 2020) for nearest-neighbor algorithms. By controlling a distance threshold determining the rate at which the nearest-neighbor algorithm abstains, we are able to trade off (robust) precision against recall, and we provide results for how to provably optimize for such a trade-off using a data-driven approach. We also perform experimental evaluation in the context of contrastive learning (He et al., 2020; Chen et al., 2020a; Khosla et al., 2020).

We acknowledge that our model is only an abstraction. Additionally, one can also consider relaxations of the Lipschitzness condition. We discuss some work along these lines in Section 1.2.

1.1 Our Contributions

Our main contributions are the following. Conceptually, we introduce a new *random feature subspace* threat model to abstract the effect of non-Lipschitzness in deep networks. Technically, we show the power of abstention and data-driven algorithm design in this setting, proving that classifiers with the ability to abstain are provably more powerful than those that cannot in this model, and giving provable guarantees for nearest-neighbor style algorithms and data-driven hyperparameter learning. Experimentally, we show that our algorithms perform well in this model on representations learned by supervised and self-supervised contrastive learning. More specifically,

- We introduce the *random feature subspace* threat model, an abstraction designed to focus on the impact of non-Lipschitzness on vulnerability to adversaries.
- We show for this threat model that *all* classifiers that partition the feature space into two or more classes—without an ability to abstain—are provably vulnerable to adversarial attacks. In particular, no matter how nice the data distribution is in feature space, for at least one class the adversary succeeds with significant probability.
- We show that in contrast, a classifier with the ability to abstain can overcome this vulnerability. We present a thresholded nearest-neighbor algorithm that is provably robust in this model when classes are sufficiently well separated, and characterize the conditions under which the algorithm does not abstain too often. This result can be viewed as providing new robustness guarantees for nearest-neighbor style algorithms as well as for *proof-carrying predictions*, where predictions are accompanied by certificates of confidence.
- We leverage and extend dispersion techniques from data-driven algorithm design, and present a novel data-driven method for learning data-specific hyperparameters in our defense algorithms to simultaneously obtain high robust accuracy and low abstention rates. Unlike typical hyperparameter tuning, our approach provably converges to a global optimum.
- Experimentally, we show that our proposed algorithm achieves *certified* adversarial robustness on representations learned by supervised and self-supervised contrastive learning. Our method significantly outperforms algorithms without the ability to abstain.

Our framework can be thought of as a kind of smoothed analysis (Spielman and Teng, 2004) in its combination of random and adversarial components. This is especially so for Section 4.3 where we broaden our guarantees to apply to arbitrary κ-bounded distributions. However, a key distinction is that in smoothed analysis, the adversary moves first, and randomness is added to its decision afterwards. In our model, in contrast, first a random restriction is applied to the space of perturbations the adversary may choose from, and then the adversary may move arbitrarily in that random subspace. Thus, the adversary in our setting has more power, because it can make its decision after the randomness has been applied.

1.2 Related Work

Large-magnitude adversarial perturbations. While most work on adversarial robustness considers small perturbations (for example, Szegedy et al. 2014; Madry et al. 2018; Zhang et al. 2019), there has also been significant work on other kinds of attacks such as adversarial rotations, translations, and deformations (Brown et al., 2018; Engstrom et al., 2017; Gilmer et al., 2018; Xiao et al., 2018; Alaifari et al., 2019). Perhaps most closely related to our negative results in Section 3 is work of Shamir et al. (2019). Shamir et al. (2019) consider an adversary that can make small ℓ_0 perturbations in the input space: that is, perturb a small number of input coordinates, but change them by an arbitrary amount. They present algorithms *for the adversary* giving targeted attacks against any learner that partitions space with a hyperplane partition using a limited number of hyperplanes in general position. Our negative results for non-abstaining classifiers are inspired by their work, though they are formally incomparable (our results are stronger in that they hold even if an adversary can just change one random linear combination and for an *arbitrary* partition of space, but weaker in that we consider an untargeted adversary, and different in that we assume randomness in the direction of adversarial power rather than in the space partition). Shamir et al. (2019) do not consider the use of abstention to combat this adversarial power. We discuss further connections to coordinate-wise perturbations in Section 3.1. Shafahi et al. (2019) look at the effect of dimensionality on robustness limits for ℓ_p-norm bounded attacks, but their negative results do not hold for abstentive classifiers.

Network Lipschitzness and relaxed notions. We model non-Lipschitzness of the network mapping in the context of robustness via large adversarial feature space movements corresponding to small input space perturbations. Several relaxations of the Lipschitz condition have been studied in the literature including Hölder smoothness (An and Gao, 2021), local Lipschitzness (Hein and Andriushchenko, 2017; Yang et al., 2020b) and probabilistic Lipschitzness (Urner and Ben-David, 2013). Typically satisfying these relaxed conditions leads to better performance than bounding the global Lipschitzness of the networks (Cisse et al., 2017).

Adversarial robustness with abstention options. Classification with abstention options (a.k.a. selective classification (Geifman and El-Yaniv, 2017)) is a relatively less explored direction in the adversarial machine learning literature. Hosseini et al. (2017) augmented the output class set with a NULL label and trained the classifier to reject the adversarial examples by classifying them as NULL; Stutz et al. (2020) and Laidlaw and Feizi (2019) obtained robustness by rejecting low-confidence adversarial examples according to confidence thresholding or predictions on the perturbations of adversarial examples. Another related line of research to our method is the detection of adversarial examples (Grosse et al., 2017; Li and Li, 2017; Carlini and Wagner, 2017; Meng and Chen, 2017; Metzen et al., 2017; Bhagoji et al., 2018; Xu et al., 2017; Hu et al., 2019; Liu et al., 2018; Deng et al., 2021). This direction also often involves thresholding a heuristic confidence score. For example, Ma et al. (2018) use a confidence metric based on k-nearest neighbors in the training sample, and Lee et al. (2018) fit class-wise Gaussian distributions and flag test points away from all distributions. These approaches have been studied empirically but typically lack formal guarantees. Goldwasser et al. (2020), on the other hand, gave provable guarantees for selective classification in a transductive setting in which performance was measured according to an adversarial test distribution from which unlabeled examples are provided to the learning algorithm in advance.

Data-driven algorithm design. Data-driven algorithm design refers to using machine learning for algorithm design, including choosing a good algorithm from a parameterized family of algorithms for given data. It is known as "hyperparameter tuning" to machine learning practitioners and typically involves a "grid search", "random search" (Bergstra and Bengio, 2012) or gradient-based search, with no guarantees of convergence to a global optimum.

Data-driven algorithm design was formally introduced to the theory of computing community by Gupta and Roughgarden (2017) as a learning paradigm, and was further extended in Balcan et al. (2017). The key idea is to model the problem of identifying a good algorithm from data as a statistical learning problem. The technique has found useful application in providing provably better algorithms for several problems of fundamental significance in machine learning including clustering (Balcan et al., 2020a, 2018c, 2021), semi-supervised learning (Balcan and Sharma, 2021), simulated annealing (Blum et al., 2021), regularized regression (Balcan et al., 2022b), mixed integer programming (Balcan et al., 2018a, 2022c), low rank approximation (Bartlett et al., 2022) and even beyond, providing guarantees like differential privacy and adaptive online learning (Balcan et al., 2018b, 2020c). See Balcan (2020) for further discussion on this rapidly growing body of research. For learning in an adversarial setting, we provide the first demonstration of the effectiveness of data-driven algorithm design in a defense method to optimize over the accuracy-abstention trade-off with strong theoretical guarantees.

Strategic classification. Strategic classification considers the case that entities being classified have a stake in the outcome, and will aim to manipulate their observable features to receive the classification they desire. Typically it is assumed these entities have some limited power to manipulate, and that this power is known to the classifier. Chen et al. (2020b); Ahmadi et al. (2021) consider entities that can manipulate inside a ball of some limited radius, whereas Kleinberg and Raghavan (2019); Alon et al. (2020); Shavit et al. (2020) consider agents that have "activities" they can engage in at some cost, which get converted into movement in feature space via an "effort conversion matrix". This latter work assumes the effort conversion matrices are known, or at least can be learned from experimentation. In contrast, our setting can be viewed as a case where the matrices are unknown and the classifier must be fielded before any manipulations are observed (and agents have an unlimited activity budget). Note, the work of Kleinberg and Raghavan (2019); Alon et al. (2020); Shavit et al. (2020) also considers the case that only certain activities correspond to "gaming" and others correspond to true self-improvement; we do not consider the self-improvement aspect here.

Adversarial defenses by non-parametric methods. Adversarial defenses by k-nearest neighbor classifier have received significant attention in recent years. In the setting of norm-bounded threat model without the ability to abstain, Wang et al. (2018) showed that the robustness properties of k-nearest neighbors depend critically on the value of k—the classifier may be inherently non-robust for small k, but its robustness approaches that of the Bayes Optimal classifier for fast-growing k. Yang et al. (2020a); Bhattacharjee and Chaudhuri (2020) provided and analyzed a general defense method, adversarial pruning, that works by preprocessing the data set to become well-separated and then running k-nearest neighbors. Theoretically, they derived an optimally robust classifier, which is analogous to the Bayes Optimal, and showed that adversarial pruning can be viewed as a finite sample approximation to this optimal classifier. In this work, we study the power of 1-nearest neighbors for adversarial robustness with the ability to abstain, under a random-subspace adversarial threat model.

Feature-space attacks. Different from most existing attacks that directly perturb input pixels, there are a few prior works that focus on perturbing abstract features as ours. More specifically, the subspaces of features typically characterize styles, which include interpretable styles such as vivid colors and sharp outlines, and uninterpretable ones (Xu et al., 2020). Ganeshan and Babu (2019) proposed a *feature disruptive attack* that generates an image perturbation that disrupts features at each layer of the network and causes deep-features to be highly corrupt. They showed that the attacks generate strong adversaries for image classification, even in the presence of various defense measures. Despite a large amount of empirical works on adversarial feature-space attack, many fundamental questions remain open, such as developing a *provable* defense against feature-space attacks.

Learning with noise. Classic work on learning with noise is a related line of work with theoretical guarantees (Kearns and Li, 1988; Bshouty et al., 2002; Awasthi et al., 2014). These models typically involve perturbations of input-space features of training points. Our nearest-neighbor based techniques for test-time feature-space attacks are different from the localization and disagreement-based approaches that are known to work for poisoning attacks (Awasthi et al., 2014, 2016; Balcan et al., 2022a). An interesting direction for future work is to determine how to adapt our techniques to handle noise in data.

2. Preliminaries and Threat Model

Notation. We will use *bold lower-case* letters such as $\mathbf{x}$ and $\mathbf{y}$ to represent vectors, *lower-case* letters such as x and y to represent scalars, and *calligraphic capital* letters such as $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{D}$ to represent distributions. Specifically, we denote by $\mathbf{x} \in \mathcal{X}$ a sample instance, and by $y \in \mathcal{Y}$ a label, where $\mathcal{X} \subseteq \mathbb{R}^{n_1}$ and $\mathcal{Y}$ indicate the image and label spaces, respectively. Let $F : \mathcal{X} \to \mathbb{R}^{n_2}$ be our given *feature embedding* (which we assume has already been learned) that maps an instance to a high-dimensional vector in the latent space $F(\mathcal{X})$. It can be parameterized, for example, by deep neural networks. We will frequently use $\mathbf{v} \in \mathbb{R}^{n_2}$ to represent an adversarial perturbation in the feature space. Denote by $\mathsf{dist}(\mathbf{z}, \mathbf{z}')$ the Euclidean distance between any two vectors $\mathbf{z}, \mathbf{z}'$ in the image or feature space, and let $\mathbb{B}(\mathbf{z}, \tau) = \{\mathbf{z}' : \mathsf{dist}(\mathbf{z}, \mathbf{z}') \leq \tau\}$ be the ball of radius τ about $\mathbf{z}$. We will use $\mathcal{D}_{\mathcal{X}}$ to denote the distribution of instances in the input space, $\mathcal{D}_{\mathcal{X}|y}$ the distribution of instances in the input space conditioned on the class y, $\mathcal{D}_{F(\mathcal{X})}$ the distribution of instances in feature space, and $\mathcal{D}_{F(\mathcal{X})|y}$ the distribution of instances in feature space conditioned on the class y. Finally, we will typically use $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_m, y_m)$ to denote a given set of m labeled training examples.

2.1 The Random Feature Subspace Threat Model

We now formally present the *random feature subspace* threat model, in which the adversary, by making small changes in the input space, is assumed to be able to create arbitrarily large movements in feature space, though only in random low-dimensional subspaces. Note that because this large modification in feature space is assumed to come from a small perturbation in input space, we always assume that the *true correct label y is the same for the modified point and the original point*.

Specifically, let $\mathbf{x}$ be an n_1-dimensional test input for classification. The input is embedded into an n_2-dimensional feature space using an abstract mapping F. Our threat model is that the adversary may corrupt $F(\mathbf{x})$ such that the modified feature vector is any point in a random n_3-dimensional affine subspace denoted by $\mathcal{S} + \{F(\mathbf{x})\}$. For example, if $n_3 = 1$ then $\mathcal{S} + \{F(\mathbf{x})\}$ is a random line through $F(\mathbf{x})$, and the adversary may select an arbitrary point on that line; if $n_3 = 2$ then $\mathcal{S}+\{F(\mathbf{x})\}$ is a random 2-dimensional plane through $F(\mathbf{x})$, and the adversary may select an arbitrary point in that plane. Conceptually, we are viewing F as "squashing" the adversarial ball about $\mathbf{x}$ in input space into a random infinitely thin and infinitely wide n_3-dimensional pancake in feature space. The adversary is given access to everything including the algorithm's classification function, F, $\mathbf{x}$, $\mathcal{S}$ and the true label of $\mathbf{x}$. Throughout the paper, we will use *adversary* and *adversarial example* to refer to this threat model.

2.2 Discussion and Examples

As noted above, we are viewing the network as conceptually squashing the ball about $\mathbf{x}$ in input space into a random infinitely wide n_3-dimensional pancake in feature space. Of course, in a real network there would be some limit on the magnitude of a perturbation in feature space, and the available directions wouldn't exactly form a subspace. However, we believe this is a clean theoretical model worthy of understanding for insight. Also, it is interesting to note that our negative results for non-abstaining classifiers, such as Theorem 1, apply even if $n_3 = 1$, whereas our positive results for classifiers that can abstain, such as Theorem 2, apply even if $n_3 \geq n_1$ so long as $n_3 \ll n_2$.

An example of a non-Lipschitz mapping. While our threat model is intended to be an abstraction, here is an example of a concrete non-Lipschitz mapping captured by our model. Let us say the support of the natural data distribution $\mathcal{D}$ only includes points with integer coefficients (data is in $\mathbb{R}^d$, but all natural points have integer coordinates). Assume the adversary can move points in the input space within an ℓ_2 ball of radius 1/4. Now, for a point x, let us define frac(x) to be its fractional part and int(x) to be its integral part. So if $d = 2$ and $x = (1.1, 2.3)$ then frac(x) $= (0.1, 0.3)$ and int(x) $= (1, 2)$. If $x = (0.9, 0.8)$ then frac(x) $= (-0.1, -0.2)$ and int(x) $= (1, 1)$. Now, let us say the network maps a point x to $F(x) = x + \langle w, \text{frac}(x)\rangle w$, where w is a large random vector (chosen independently at random for each lattice point int(x)). Then, all natural data will stay where they are (this is the identity mapping on natural data), but points in the adversarial ball can move very far in the direction of their w. So, if the true decision boundary is, say $x_1 \geq 1/2$, the adversary will not change the true label of any data point but (in the limit as $|w| \to \infty$) will be able to defeat any non-abstaining classifier in the feature space by Theorem 1.

Additional remarks. We wish to be clear that our intent is not to create a threat model against which one would design a new network architecture or training procedure. Instead, we are thinking of a network that has already been trained (say, using adversarial training or any of the other available methods that try to improve robustness). But, the designers are finding that the adversarial loss is unacceptably high, because for many test points, the adversary can still move those points a large distance in feature space and cross over their decision boundary (even if the natural data of different classes are well-separated in feature space). Our framework is aimed to consider this setting, and our results provide a practical suggestion: modify the final level to allow it to abstain if a test point is "too different" from the training data. The justification is that if the adversary can move large distances but not in every possible direction (if it can do that, then no defense will work) and indeed only do so in random lower-dimensional subspaces, then we can provide theoretical guarantees for this approach. Moreover, our lower bounds show that abstention is *necessary* no matter how nicely distributed the data may be. In fact, it is necessary even if the adversary can move points arbitrarily large distances in feature space even in just a single random direction.

3. Negative Results without an Ability to Abstain

We now present a hardness result showing that no matter how nicely data is distributed in feature space (for example, even if the network perfectly clusters data by label in feature space), any classifier that is not allowed to abstain will fail against our threat model even for an adversary that can perturb points in a single random direction ($n_3 = 1$).

Theorem 1 *For any classifier that partitions $\mathbb{R}^{n_2}$ into two or more classes, any data distribution $\mathcal{D}$, any $\delta > 0$ and any feature embedding F, there must exist at least one class y^*, such that for at least a $1 - \delta$ probability mass of examples $\mathbf{x}$ from class y^* (that is, $\mathbf{x}$ is drawn from $\mathcal{D}_{\mathcal{X}|y^*}$), for a random unit-length vector $\mathbf{v}$, with probability at least $1/2 - \delta$ for some $\Delta_0 > 0$, $F(\mathbf{x}) + \Delta_0\mathbf{v}$ is not labeled y^* by the classifier. In other words, there must be at least one class y^* such that for at least $1 - \delta$ probability mass of points $\mathbf{x}$ of class y^*, the adversary wins with probability at least $1/2 - \delta$.*

Proof Define r_δ to be a radius such that in the feature space, for every class y, at least a $1 - \delta$ probability mass of examples $F(\mathbf{x})$ of class y lie within distance r_δ of the origin. Define R such that for a ball of radius R, if we move the ball by a distance r_δ, at least a $1 - \delta$ fraction of the volume of the new ball is inside the intersection with the old ball. Now, let $\mathcal{B}$ be the ball of radius R centered at the origin in feature space. Let $\mathsf{vol}(\mathcal{B})$ denote the volume of $\mathcal{B}$ and let $\mathsf{vol}_y(\mathcal{B})$ denote the volume of the subset of $\mathcal{B}$ that is assigned label y by the classifier. Let y^* be any label such that $\mathsf{vol}_{y^*}(\mathcal{B})/\mathsf{vol}(\mathcal{B}) \leq 1/2$. Now by the definition of y^*, a point $\mathbf{z}$ picked uniformly at random from $\mathcal{B}$ has probability at least $1/2$ of being classified differently from y^*. This implies that, by the definition of R, if $F(\mathbf{x})$ is within distance r_δ of the origin, then a point $\mathbf{z}_x$ that is picked uniformly at random in the ball $\mathcal{B}_x$ of radius R centered at $F(\mathbf{x})$ has probability at least $1/2 - \delta$ of being classified differently from y^*. This immediately implies that if we choose a random unit-length vector $\mathbf{v}$, then with probability at least $1/2 - \delta$, there exists $\Delta_0 > 0$ such that $F(\mathbf{x}) + \Delta_0\mathbf{v}$ is classified differently from y^*, since we can think of choosing $\mathbf{v}$ by first sampling $\mathbf{z}_x$ from $\mathcal{B}_x$ and then defining $\mathbf{v} = (\mathbf{z}_x - F(\mathbf{x}))/\|\mathbf{z}_x - F(\mathbf{x})\|_2$. Moreover, since the classifier has no abstention region, being classified differently from y^* implies a win by the adversary. So, the theorem follows from the fact that, by the definition of r_δ, at least $1 - \delta$ probability mass of examples $F(\mathbf{x})$ from class y^* are within distance r_δ of the origin in feature space. ■

We remark that our lower bound applies to any classifier and exploits the fact that a classifier without abstention must label the entire feature space. For a simple linear decision boundary, a perturbation in any direction (except parallel to the boundary) can cross the boundary with an appropriate magnitude (Figure 1, mid). The left and right figures show that if we try to 'bend' the decision boundary to 'protect' one of the classes, the other class is still vulnerable. Our argument formalizes and generalizes this intuition, and shows that there must be at least one vulnerable class irrespective of how you may try to shape the class boundaries, where the adversary succeeds in a large fraction of directions.

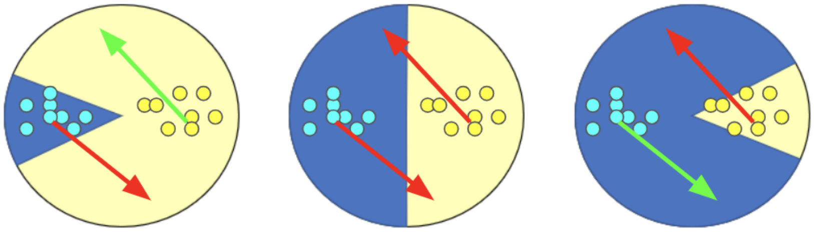

Figure 1: A simple example to illustrate Theorem 1. Bending the decision boundary to avoid adversarial examples for one class makes it harder to defend the other class.

3.1 Comparison to Coordinate-wise Perturbations

It is interesting to compare our model to one in which the adversary can make only *coordinate-wise* perturbations in the feature space. An adversary that can only make coordinate-wise changes would, in contrast, *not* be powerful enough to defeat any non-abstaining classifier. For example, consider data in $\mathbb{R}^3$ where all the positive examples are at location $(1, 1, 1)$ and all the negative examples are at location $(-1, -1, -1)$ in feature space. Then so long as the classifier partitions the space such that the lines $(1, 1, \cdot)$, $(1, \cdot, 1)$, and $(\cdot, 1, 1)$ are positive and $(-1, -1, \cdot)$, $(-1, \cdot, -1)$, and $(\cdot, -1, -1)$ are negative, the adversary will not be able to defeat it with a single coordinate-wise change. (We need to use $\mathbb{R}^3$ here rather than $\mathbb{R}^2$, because in $\mathbb{R}^2$ these lines would cross and so the classifier would not be well-defined). In contrast, by Theorem 1, an adversary that can perturb in a uniformly-random direction *will* defeat any non-abstaining classifier.

4. Positive Results with an Ability to Abstain

Theorem 1 gives a hardness result for robust classification without abstention. In this section, we give positive results for a nearest-neighbor style classifier that has the power to abstain.

Given a test instance $\mathbf{x} \sim \mathcal{D}_{\mathcal{X}}$, recall that the adversary is allowed to corrupt $F(\mathbf{x})$ with an arbitrarily large perturbation in a uniformly-distributed subspace S of dimension n_3. Consider the prediction rule that classifies the unseen example $F(\mathbf{x}) \in \mathbb{R}^{n_2}$ with the class of its nearest training example provided that the distance between them is at most τ; otherwise the algorithm outputs "don't know" (see Algorithm 1 when $\sigma = 0$). The threshold parameter τ trades off robustness against abstention rate; when $\tau \to \infty$, our algorithm is equivalent to the nearest-neighbor algorithm. Note that Algorithm 1 also contains a parameter σ to remove training points that are too close to other training points of a different class—we will consider non-zero values of this parameter later.

Denote by $\mathcal{E}_{\text{adv}}^{\mathbf{x}}(f) := \mathbb{E}_{S\sim\mathcal{S}}\mathbf{1}\{\exists \mathbf{e} \in S + F(\mathbf{x}) \text{ s.t. } f(\mathbf{e}) \neq \mathbf{y} \text{ and } f(\mathbf{e}) \text{ does not abstain}\}$ the robust error of a given classifier f for classifying instance $\mathbf{x}$. Our analysis leads to the following positive results on this algorithm. This theorem states that so long as the threshold τ is sufficiently small compared to the distance r between the test point $\mathbf{x}$ and the nearest training point $\mathbf{x}_i$ of a different class (see Figure 2), and the dimension n_2 of the ambient feature space is sufficiently large compared to the dimension n_3 of the adversarial subspace S, the algorithm will have low robust error on $\mathbf{x}$.

Algorithm 1 ROBUSTCLASSIFIER(τ, σ)

```
Input: A test example $F(\mathbf{x})$ (potentially an adversarial example), a set of training examples $F(\mathbf{x}_i)$ and their labels $y_i$, $i \in [m]$, a threshold parameter $\tau$, a separation parameter $\sigma$.
Preprocessing: Delete training examples $F(\mathbf{x}_i)$ if $\min_{j\in[m],y_i\neq y_j} \mathsf{dist}(F(\mathbf{x}_i), F(\mathbf{x}_j)) < \sigma$.
Output: A predicted label of $F(\mathbf{x})$, or "don't know".
if $\min_{i\in[m]} \mathsf{dist}(F(\mathbf{x}), F(\mathbf{x}_i)) < \tau$ then
    return $y_{\mathrm{argmin}_{i\in[m]} \mathsf{dist}(F(\mathbf{x}),F(\mathbf{x}_i))}$;
else
    return "don't know".
```

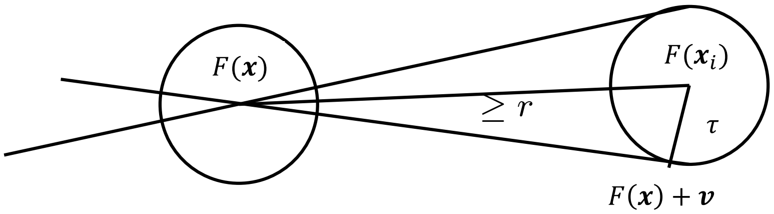

Figure 2: Adversarial misclassification for nearest-neighbor predictor. Here $F(\mathbf{x})$ is the test point and $F(\mathbf{x}_i)$ is a training point from a different class. For $n_3 = 1$, the adversary succeeds for the directions inside the depicted cone around $F(\mathbf{x}_i)$.

Theorem 2 *Let $\mathbf{x} \sim \mathcal{D}_\mathcal{X}$ be a test instance, m be the number of training examples and r be the shortest distance between $F(\mathbf{x})$ and $F(\mathbf{x}_i)$ where $\mathbf{x}_i$ is a training point from a different class. Suppose $\tau = o\left(r\sqrt{1-\frac{n_3}{n_2}}\right)$. The robust error of Algorithm 1, $\mathcal{E}_{\mathrm{adv}}^{\mathbf{x}}(\text{ROBUSTCLASSIFIER}(\tau, 0))$, is at most*

$$m\left(\frac{c\tau}{r\sqrt{1-\frac{n_3}{n_2}}}\right)^{n_2-n_3} + mc_0^{n_2-n_3},$$

where $c > 0$ and $0 < c_0 < 1$ are absolute constants. For the case $n_3 = 1$, the robust error is at most

$$m\left(\frac{\tau}{r}\right)^{n_2-1}.$$

Proof We begin our analysis with the case of $n_3 = 1$. Suppose we have a training example $\mathbf{x}'$ of another class, and suppose $F(\mathbf{x})$ and $F(\mathbf{x}')$ are at distance D in the feature space. That is, $\mathsf{dist}(F(\mathbf{x}), F(\mathbf{x}')) = D$. Because $\tau = o(D)$, the probability that the adversary can move $F(\mathbf{x})$ to within distance τ of $F(\mathbf{x}')$ is at most the ratio of the surface area of a sphere of radius τ to the surface area of a sphere of radius D, which is at most

$$\left(\frac{\tau}{D}\right)^{n_2-1} \leq \left(\frac{\tau}{r}\right)^{n_2-1}$$

if the feature space is n_2-dimensional. See Figure 2.

The analysis for the case of general values of n_3 follows from a peeling argument. For this, we need the following Random Projection Theorem (Dasgupta and Gupta, 2003; Vempala, 2005).

Lemma 3 (Random Projection Theorem) *Let $\mathbf{z}$ be a fixed unit length vector in d-dimensional space and $\widetilde{\mathbf{z}}$ be the projection of $\mathbf{z}$ onto a random k-dimensional subspace. For $0 < \delta < 1$,*

$$\Pr\left[\left|\|\widetilde{\mathbf{z}}\|_2^2 - \frac{k}{d}\right| \geq \delta\frac{k}{d}\right] \leq e^{-\frac{k(\delta^2-\delta^3)}{4}}.$$

Without loss of generality, we assume $F(\mathbf{x}) = \mathbf{0}$ in $\mathbb{R}^{n_2}$. Next, note that the random subspace in which the adversary vector is restricted to lie can be constructed by the following sampling scheme: we first sample a vector $\mathbf{v}_1$ uniformly at random from a unit sphere in the ambient space $\mathbb{R}^{n_2}$ centered at $\mathbf{0}$; fixing $\mathbf{v}_1$, we then sample a vector $\mathbf{v}_2$ uniformly at random from a unit sphere in the nullspace of $\mathsf{span}\{\mathbf{v}_1\}$; we repeat this procedure n_3 times and let $\mathsf{span}\{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_{n_3}\}$ be the desired subspace. Note that the sampling scheme satisfies the random adversary model. For the fixed nullspace $\mathsf{null}(\mathsf{span}\{\mathbf{v}_1, ..., \mathbf{v}_i\})$ of dimension $n_2 - i$, according to the analysis of the case $n_3 = 1$, if we condition on the distance D_i between $F(\mathbf{x})$ and $F(\mathbf{x}')$ when they are projected to $\mathsf{null}(\mathsf{span}\{\mathbf{v}_1, ..., \mathbf{v}_i\})$, the probability over the draw of $\mathbf{v}_{i+1}$ of failure with respect to $\mathbf{x}'$ is at most $(\mathcal{O}(\tau/D_i))^{n_2-i-1}$. We also note that $\mathsf{null}(\mathsf{span}\{\mathbf{v}_1, ..., \mathbf{v}_i\})$ is a random subspace of dimension $n_2 - i$. Thus by Lemma 3 (with constant δ), we have $D_i \geq Cr\sqrt{\frac{n_2-i}{n_2}}$ with probability at least $1 - e^{-c'(n_2-i)}$, where $C, c' > 0$ are absolute constants. Therefore, by the union bound over the choice of n_3 nullspaces and the failure probability of the event $D_i \geq Cr\sqrt{\frac{n_2-i}{n_2}}$, the failure probability of the algorithm over x' is at most

$$\sum_{i=1}^{n_3} e^{-c'(n_2-i)} + \sum_{i=1}^{n_3}\left(\mathcal{O}\left(\frac{\tau}{Cr\sqrt{\frac{n_2-i}{n_2}}}\right)\right)^{n_2-i} \leq c_0^{n_2-n_3} + \left(\frac{c\tau}{r\sqrt{\frac{n_2-n_3}{n_2}}}\right)^{n_2-n_3}.$$

By the union bound over all m training data points $\mathbf{x}'$ completes the proof. ■

Theorem 2 states that the robust error of Algorithm 1 on a test point $\mathbf{x}$ will be small so long as its distance r to its nearest training point $\mathbf{x}_i$ from a different class is sufficiently larger than τ, and so long as the number of labeled examples m is sub-exponential in $n_2 - n_3$. If m is so large that a sphere of radius r about point $\mathbf{x}$ can be covered by m balls of radius τ, then the adversary could indeed win, because any ray extending from $\mathbf{x}$ will pierce one of these balls. One simple way to address this would be that if size of the labeled sample really is exponential in $n_2 - n_3$, then to just use a sub-exponentially large random subsample of it.[2] We also prove the following asymptotic improvement over Theorem 2 for fixed n_3 and large n_2 via a tighter bound on the probability mass of the region of adversarial success.

Theorem 4 *If $\tau = o(r)$, the robust error $\mathcal{E}_{\text{adv}}^{\mathbf{x}}(f)$ of* ROBUSTCLASSIFIER$(\tau, 0)$ *in Algorithm 1 for classifying $\mathbf{x}$ is at most $\mathcal{O}\left(\frac{m}{n_2-n_3}\left(\frac{\tau}{r}\right)^{n_2-n_3}\frac{1}{B(n_3/2,(n_2-n_3)/2)}\right)$, where $B(\cdot,\cdot)$ is the Beta function. The Beta function is given by $B(r_1, r_2) = \int_0^1 t^{r_1-1}(1-t)^{r_2-1}dt$, for $r_1, r_2 \in \mathbb{R}^+$, and is closely related to binomial coefficients.*

[2] This observation shows that nearest-neighbor is not an optimal algorithm when the number of examples is exponential in the dimension. In the case of very large m, one could instead use an algorithm that estimated densities in each part of space.

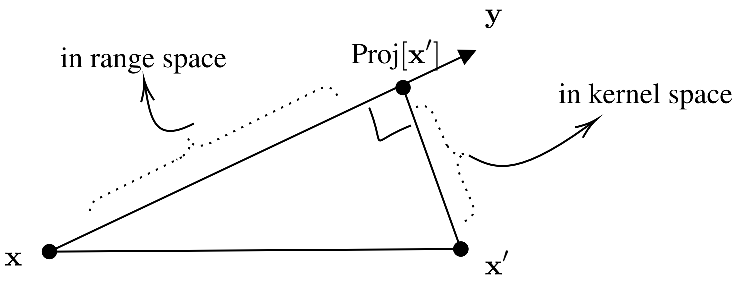

Figure 3: Rotational symmetry of adversarial subspaces. Let $\mathbf{y}$ be a random direction from test point $\mathbf{x}$, and Proj$[\mathbf{x}']$ be the projection of training point $\mathbf{x}'$ on to $\mathbf{xy}$. For any adversarial space with Proj$[\mathbf{x}']$ as the projection of $\mathbf{x}'$ on the space, we must have $\mathbf{xy}$ in the range space and $\mathbf{x}'$Proj$[\mathbf{x}']$ in the nullspace.

Proof We drop $F(\cdot)$ from the notation for simplicity. Let $\mathbf{x}$ be the origin. Let $\mathbf{x}'$ be a training point of another class, and R be a random n_3-dimensional linear subspace. Scale all distances by a factor of $\frac{1}{\mathsf{dist}(\mathbf{x},\mathbf{x}')}$. By rotational symmetry, we assume WLOG that R is given by $x_{n_3+1} = x_{n_3+2} = \cdots = x_{n_2} = 0$, and $\mathbf{x}'$ is the uniformly random unit vector $(z_1, \ldots, z_{n_2})$. Indeed, for a fixed direction from $\mathbf{x}$, the set of subspaces for which the projection of $\mathbf{x}'$ lies along that direction is constrained by one vector each in the range space and kernel space, and is therefore in bijection to the set of subspaces associated with another fixed direction (Figure 3).

The adversary can win only if the distance between $\mathbf{x}'$ with the closest vector Proj$[\mathbf{x}']$ in R, that is with $(z_1, \ldots, z_{n_3}, 0, \ldots, 0)$, is at most $\frac{\tau}{\mathsf{dist}(\mathbf{x},\mathbf{x}')} \leq \frac{\tau}{r}$. We can now apply Lemma 21 (Appendix A), which gives a bound on the fraction of the surface of the sphere at some fixed small distance from the orthogonal space, to get that the adversary succeeds by perturbing $\mathbf{x}$ to a point within $\mathcal{B}(\mathbf{x}', \tau)$ with probability at most

$$\frac{2(\tau/r)^{n_2-n_3}}{n_2-n_3} \cdot \frac{A(n_2-n_3-1)A(n_3-1)}{A(n_2-1)},$$

where $A(n)$ is the surface-area of the unit n-sphere embedded in $\mathbb{R}^{n+1}$. We have closed a form $A(n) = \frac{2\pi^{\frac{n+1}{2}}}{\Gamma\left(\frac{n+1}{2}\right)}$, where $\Gamma(z) = \int_{t=0}^{\infty} t^{z-1}e^{-t}dt$ is the gamma function.

Noting that $B(z_1, z_2) = \frac{\Gamma(z_1)\Gamma(z_2)}{\Gamma(z_1+z_2)}$, together with a union bound over all training points from a different class, gives the result. ■

4.1 Outlier Removal and Improved Upper Bound

The guarantees above are good when the test points are far from training points from other classes in the feature space. This empirically holds true for good data and perfect embeddings—a so-called neural collapse phenomenon that the trained network converges to representations such that all points of class k get embedded close to a single point μ_k in the feature space (Papyan et al., 2020). But for noisy data and good-but-not-perfect embeddings, the condition may not hold. In Theorem 24 (in Appendix B) we show that we obtain almost the same upper bound on failure probability as above by exploiting the outlier removal threshold σ. Intuitively, outlier removal artificially induces well-separateness in the feature space, by deleting training examples that are close to other examples with a different label.

4.2 Upper Bound on Abstention Rate on Natural Data

Of course, the statement that robust error is low just means the adversary has a low probability of being able to create an error. This is only half the picture: the other half is that we also want our algorithm to have a low probability of abstaining on natural data. This is what we address in the next two sections, and it will require assumptions on how natural data is distributed. In particular, we give two different sufficient conditions for having a low abstention rate on natural data: (1) that natural data is well-clustered in feature space (Section 4.2.1), and alternatively (2) that the natural data has low *doubling dimension* (Section 4.2.2). For these results, we assume our m training points $\mathbf{x}_1, \ldots, \mathbf{x}_m$ are i.i.d. draws from distribution $\mathcal{D}_{F(\mathcal{X})}$; if we also have additional training points used in the construction of F (which, therefore, cannot be treated as i.i.d. draws), this can only help.

4.2.1 Low Abstention Rate for Well-clustered Data

We show here that if natural data has the property that for every label class, one can cover most of the probability mass of the class with not too many (potentially overlapping) balls of at least some minimal probability mass, then our algorithm will have a low abstention rate.

Definition 5 *A distribution $\mathcal{D}$ is (δ, β, N)-coverable if at least a $1-\delta$ fraction of probability mass of the marginal distribution $\mathcal{D}_{F(\mathcal{X})}$ over $\mathbb{R}^{n_2}$ can be covered by N balls $\mathbb{B}_1$, $\mathbb{B}_2$, ... $\mathbb{B}_N$ of radius $\tau/2$ and of mass $\Pr_{\mathcal{D}_{F(\mathcal{X})}}[\mathbb{B}_k] \geq \beta$.*

Intuitively, if a set of balls cover (most of) the distribution and we sample enough points from the distribution, we should get at least one sample from each ball and our algorithm will not abstain on the covered points. Formally, we show the following guarantee on the abstention rate on distributions that are (δ, β, N)-coverable w.r.t. threshold τ.

Theorem 6 *Suppose that $F(\mathbf{x}_1), ..., F(\mathbf{x}_m)$ are m training instances i.i.d. sampled from marginal distribution $\mathcal{D}_{F(\mathcal{X})}$. If the distribution $\mathcal{D}$ is (δ, β, N)-coverable, for sufficiently large $m \geq \frac{1}{\beta} \ln \frac{N}{\gamma}$, with probability at least $1-\gamma$ over the sampling, we have $\Pr[\cup_{i=1}^m \mathbb{B}(F(\mathbf{x}_i), \tau)] \geq 1-\delta$.*

Proof Fix ball $\mathbb{B}_i$ in the cover from Definition 5. Let B_i denote the event that no point is drawn from ball $\mathbb{B}_i$ over the m samples. Since successive draws are independent, and by Definition 5 $\Pr_{\mathcal{D}_{F(\mathcal{X})}}[\mathbb{B}_i] \geq \beta$, we have that $\Pr[B_i] \leq (1-\beta)^m \leq \exp(-\beta m)$. Further, by a union bound over N balls $\Pr[\cup_i B_i] \leq N \exp(-\beta m) \leq \gamma$, for $m \geq \frac{1}{\beta} \ln \frac{N}{\gamma}$.

Therefore, with probability at least $1-\gamma$ for all $k \in [N]$ there is at least a sample $F(\mathbf{x}_{i_k}) \in \{F(\mathbf{x}_1), F(\mathbf{x}_2), ..., F(\mathbf{x}_m)\}$ such that $F(\mathbf{x}_{i_k}) \in \mathbb{B}_k$. This implies $\cup_{i=1}^m \mathbb{B}(F(\mathbf{x}_i), \tau) \supseteq \cup_{k=1}^N \mathbb{B}_k$, since $\mathbb{B}_k$ is a ball of radius $\tau/2$. So with probability at least $1-\gamma$ over the sampling, we have $\Pr[\cup_{i=1}^m \mathbb{B}(F(\mathbf{x}_i), \tau)] \geq \Pr[\cup_{k=1}^N \mathbb{B}_k] \geq 1-\delta$. ■

Note that in the special case that the N balls are disjoint and each has probability mass $\beta = 1/N$, then $m = \Omega(N \log N)$ samples are also necessary to get a point inside each ball, by a standard coupon-collector analysis.

Theorem 6 implies that if we have a covering with N balls, each with probability mass at least β and large enough sample size m, with probability at least $1-\gamma$ over the sampling, we have $\Pr[\cup_{i=1}^m \mathbb{B}(F(\mathbf{x}_i),\tau)] \geq 1-\delta$. Therefore, with high probability, the algorithm will output "don't know" only for a δ fraction of natural data. Below we give an example of a distribution where our algorithm will simultaneously achieve low robust error and low natural abstention rates.

Example distribution where Algorithm 1 is robust with low abstention rate. Our example will consist of well-separated data in the feature space. Suppose $\mathcal{D}_{F(\mathcal{X})|y}$ for each label class y consists of the uniform distribution over N_y n_2-balls of radius $\tau/2$ centered at axis-aligned unit vectors $\{\mathbf{e}_j \mid j \in S_y\}$, where $S_y \subset [n_2]$ is the set of axes with balls labeled by y, with $\tau < 1/3$ and $S_y \cap S_{y'} = \emptyset$ for $y \neq y'$. Further let $m = n_2 \log \frac{n_2}{\gamma}$ for some absolute constant $\gamma \in (0,1)$, so this distribution is (δ, β, N)-coverable with $\delta = 0$, $\beta = 1/N$ and $N = n_2$. If $n_3 = 1$, by Theorem 2, the robust error of Algorithm 1 is bounded by $O(n_2 \log n_2 \tau^{n_2-1}) = o(1)$. Thus, in this setting, our algorithm enjoys low robust error without abstaining too much (for sufficiently large n_2).

4.2.2 Controlling Abstention Rate via Doubling Dimension

Here, we give an alternative bound on the abstention rate on natural data based on the *doubling dimension* of the data distribution. Doubling dimension can be used to obtain sample complexity of generalization for learning problems (Bshouty et al., 2009). Bounded doubling dimension has also been used to give bounds on cluster quality for nearest-neighbor based extensions of clustering algorithms in the distributed learning setting (Dick et al., 2017).

Definition 7 (Doubling dimension) *A measure $\mathcal{D}_{F(\mathcal{X})}$ with support $F(\mathcal{X})$ is said to have a doubling dimension d, if for all points $F(\mathbf{x}) \in F(\mathcal{X})$ and all radii $\tau > 0$, $\mathcal{D}_{F(\mathcal{X})}(\mathcal{B}(F(\mathbf{x}), 2\tau)) \leq 2^d \mathcal{D}_{F(\mathcal{X})}(\mathcal{B}(F(\mathbf{x}), \tau))$.*

Given a sample S and point $\mathbf{x}$, let $NN_S(F(\mathbf{x}))$ denote $\mathbf{x}$'s nearest neighbor in S in feature space. We now have the following theorem, which implies low abstention under the assumption of bounded doubling dimension, which we show implies that $\mathcal{D}$ satisfies Definition 5 (for β, N that depend on the doubling dimension).

Theorem 8 *Suppose that the measure $\mathcal{D}_{F(\mathcal{X})}$ in the feature space has a doubling dimension d. Let D be the diameter of $F(\mathcal{X})$. For any $\tau > 0$ and any $\delta > 0$, if we draw an i.i.d. sample S of size $m \geq \left(\frac{2D}{\tau}\right)^d \left(d \log \frac{4D}{\tau} + \log \frac{1}{\delta}\right)$, then with probability at least $1-\delta$ over the draw of S, we have $\sup_{\mathbf{x}\in\mathcal{X}} d(F(\mathbf{x}), NN_S(F(\mathbf{x}))) \leq \tau$.*

Proof Lemma 10 (below) implies that there exists a covering of $F(\mathcal{X})$ of size $(4D/\tau)^d$ which consists of balls of radius $\tau/2$ around points $F(\mathbf{x}) \in F(\mathcal{X})$. Further Lemma 9 implies that for a ball B of radius $\tau/2$ around point $F(\mathbf{x}) \in F(\mathcal{X})$ we have $\mathcal{D}_{F(\mathcal{X})}(B) \geq \left(\frac{\tau}{2D}\right)^d$. Thus Definition 5 is satisfied with $N = (4D/\tau)^d$ and $\beta = \left(\frac{\tau}{2D}\right)^d$. Theorem 6 now implies the result. ■

Our proof of Theorem 8 relies on the following properties of a doubling measure. We first give a lower bound on the probability mass of a small ball in terms of the doubling dimension of the distribution.

Lemma 9 *Suppose that the measure $\mathcal{D}_{F(\mathcal{X})}$ has a doubling dimension d. Let D be the diameter of $F(\mathcal{X})$. Then for any point $F(\mathbf{x}) \in F(\mathcal{X})$ and any radius of the form $\tau = D/2^T$ for $T \in \mathbb{N}$, we have $\mathcal{D}_{F(\mathcal{X})}(\mathcal{B}(F(\mathbf{x}), \tau)) \geq (\tau/D)^d$.*

Proof Since D is the diameter of $F(\mathcal{X})$, we have $\mathcal{D}_{F(\mathcal{X})}(\mathcal{B}(F(\mathbf{x}), D)) = 1$. Therefore, we have

$$\begin{aligned}\mathcal{D}_{F(\mathcal{X})}(\mathcal{B}(F(\mathbf{x}), \tau)) &= \mathcal{D}_{F(\mathcal{X})}(\mathcal{B}(F(\mathbf{x}), D/2^T)) \\ &\geq 2^{-d} \cdot \mathcal{D}_{F(\mathcal{X})}(\mathcal{B}(F(\mathbf{x}), D/2^{T-1})) \\ &\geq \cdots \\ &\geq 2^{-Td} \cdot \mathcal{D}_{F(\mathcal{X})}(\mathcal{B}(F(\mathbf{x}), D)) \\ &= 2^{-Td} \\ &= (\tau/D)^d.\end{aligned}$$

■

This further lets us bound the covering number in terms of the doubling dimension as follows.

Lemma 10 (Relating doubling dimension to covering number) *Given any radius τ of the form $\tau = D/2^T$ for $T \in \mathbb{N}$, there is a covering of $F(\mathcal{X})$ using balls of radius τ around points $F(\mathbf{x}) \in F(\mathcal{X})$ of size no more than $(2D/\tau)^d$.*

Proof We construct the covering balls of $F(\mathcal{X})$ as follows: when there is a point $F(\mathbf{x}) \in F(\mathcal{X})$ which is not contained in any current covering ball of radius τ, we add the ball $\mathcal{B}(F(\mathbf{x}), \tau)$ to the cover. We follow this procedure until every point in $F(\mathcal{X})$ is covered by some covering balls. Denote by $\mathcal{C}$ the set of centers for the balls in the cover.

We now show that this procedure stops after adding at most $(2D/\tau)^d$ balls to the cover. We note that by our construction, the centers of the covering are at least distance τ from each other, implying that the collection of $\mathcal{B}(F(\mathbf{x}), \tau/2)$ for $F(\mathbf{x}) \in \mathcal{C}$ are disjoint. This yields

$$\begin{aligned}1 &\geq \mathcal{D}_{F(\mathcal{X})}\left(\cup_{F(\mathbf{x})\in\mathcal{C}} \mathcal{B}(F(\mathbf{x}), \tau/2)\right) \\ &= \sum_{F(\mathbf{x})\in\mathcal{C}} \mathcal{D}_{F(\mathcal{X})}(\mathcal{B}(F(\mathbf{x}), \tau/2)) \qquad \text{(since } \mathcal{B}(F(\mathbf{x}), \tau/2) \text{ are disjoint)} \\ &\geq \sum_{F(\mathbf{x})\in\mathcal{C}} \left(\frac{\tau}{2D}\right)^d \qquad \text{(by Lemma 9)} \\ &= |\mathcal{C}| \left(\frac{\tau}{2D}\right)^d.\end{aligned}$$

So we have $|\mathcal{C}| \leq (2D/\tau)^d$. ■

4.3 A More General Adversary with Bounded Density

We extend our results in Theorem 2 to a more general class of adversaries, which have a bounded density over the space of linear subspaces of a fixed dimension n_3 and the adversary can perturb a test feature vector arbitrarily in the sampled adversarial subspace. Specifically, a distribution is said to be *κ-bounded* if the corresponding probability density $f(x)$ satisfies, $\sup_x f(x) \leq \kappa$. For example, the standard normal distribution $\mathcal{N}(\mu, \sigma)$ is $\frac{1}{\sqrt{2\pi}\sigma}$-bounded.

Theorem 11 *Consider the setting of Theorem 2, with an adversary having a κ-bounded distribution over the space of linear subspaces of a fixed dimension n_3 for perturbing the test point. If $\mathbf{E}(\tau, r)$ denotes the bound on error rate in Theorem 2 for* ROBUSTCLASSIFIER$(\tau, 0)$ *in Algorithm 1, then the error bound of the same algorithm against the κ-bounded adversary is $\mathcal{O}(\kappa\mathbf{E}(\tau, r))$.*

Proof To argue upper bounds on failure probability, we consider the set of adversarial subspaces which can allow the adversary to perturb the test point x close to a training point x'. Let $\mathcal{S}(x', \tau)$ denote the subset of linear subspaces of dimension n_3 such that for any $S \in \mathcal{S}(x', \tau)$ there exists $v \in S$ with $x + v \in \mathcal{B}(x', \tau)$. Note that we can upper bound the fraction of the total probability space occupied by $\mathcal{S}(x', \tau)$ by $\frac{1}{m}\mathbf{E}(\tau, r)$, where constants in n_2, n_3 have been suppressed. If we show that $\mathcal{S}(x', \tau)$ is a measurable set, we can use the κ-boundedness of the adversary distribution to claim that the failure probability for misclassifying as x' is upper bounded by $\kappa \mathsf{vol}(\mathcal{S})\frac{1}{m}\mathbf{E}(\tau, r) = \mathcal{O}\left(\frac{\kappa}{m}\mathbf{E}(\tau, r)\right)$, since the volume of the complete adversarial space $\mathcal{S}$ is a constant in n_2, n_3. In Lemma 22 (Appendix A), we make the stronger claim that $\mathcal{S}(x', \tau)$ is convex. We can then use a union bound on the training points to get a bound on the total failure probability as $\mathcal{O}\left(\kappa\mathbf{E}(\tau, r)\right)$. ■

5. Learning Data-Specific Optimal Thresholds

Given an embedding function F and a classifier f_τ which outputs either a predicted class if the nearest neighbor is within distance τ of a test point or abstains from predicting if not (see Algorithm 1), we want to evaluate the performance of f_τ on a test set $\mathcal{T}$ against an adversary which can perturb a test feature vector in a random n_3-dimensional subspace $S \sim \mathcal{S}$. To this end, we define

Definition 12 (Robust error.) *Let $\mathcal{E}_{\text{adv}}(\tau, S) := \frac{1}{|\mathcal{T}|}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{T}} \mathbf{1}\{\exists \mathbf{e} \in S + F(\mathbf{x}) \subseteq \mathbb{R}^{n_2}$ such that $f_\tau(\mathbf{e}) \neq \mathbf{y}$ and $f_\tau(\mathbf{e})$ does not abstain$\}$ denote the robust error on the test set $\mathcal{T}$, for n_3-dimensional perturbation subspace S and threshold setting τ in Algorithm 1. Also define average robust error as $\mathcal{E}_{\text{adv}}(\tau) := \mathbb{E}_{S\sim\mathcal{S}}\left[\mathcal{E}_{\text{adv}}(\tau, S)\right]$ for distribution $\mathcal{S}$ over n_3-dimensional subspaces (assumed to be the uniform distribution unless stated otherwise) and estimated robust error over a set $\hat{S}$ of subspaces as $\hat{\mathcal{E}}_{\text{adv}}(\tau, \hat{\mathcal{S}}) := \frac{1}{|\hat{\mathcal{S}}|}\sum_{S\in\hat{\mathcal{S}}} \mathcal{E}_{\text{adv}}(\tau, S)$. Let $\hat{\mathcal{S}}$ consist of multiple samples drawn from $\mathcal{S}$, and for conciseness, we will often denote $\hat{\mathcal{E}}_{\text{adv}}(\tau, \hat{\mathcal{S}})$ by $\hat{\mathcal{E}}_{\text{adv}}(\tau)$ and $\hat{\mathcal{S}}$ will be implicit from context.*

$\hat{\mathcal{E}}_{\text{adv}}(\tau)$ gives an easier-to-compute surrogate to $\mathcal{E}_{\text{adv}}(\tau)$, by drawing subspaces in $\hat{\mathcal{S}}$ according to $\mathcal{S}$ (Algorithm 3 gives the procedure to compute the attack perturbation given subspace S). For an abstentive classifier, the robust error can be trivially minimized by abstaining everywhere. We will therefore also need to control the abstention rate on unperturbed data.

Definition 13 (Natural abstention rate.) *Define $\mathcal{D}_{\text{nat}}(\tau) := \frac{1}{|\mathcal{T}|}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{T}} \mathbf{1}\{f_\tau(F(\mathbf{x}))$ abstains$\}$ as the abstention rate on the unperturbed test set $\mathcal{T}$.*

$\mathcal{E}_{\text{adv}}(\tau)$ and $\mathcal{D}_{\text{nat}}(\tau)$ are both monotonic in τ; while the former is non-decreasing, the latter is non-increasing (Lemma 14).

Lemma 14 *Robust error $\mathcal{E}_{\text{adv}}(\tau, S)$ is monotonically non-decreasing in τ for any S. Further, natural abstention rate $\mathcal{D}_{\text{nat}}(\tau)$ is monotonically non-increasing in τ.*

Proof Let $0 \leq \tau_1 \leq \tau_2 \leq \infty$. For any $(\mathbf{x}, \mathbf{y}) \in \mathcal{T}$, if there exists $\mathbf{e} \in S + F(\mathbf{x})$ for which the adversary succeeds for threshold τ_1, we have $f_{\tau_1}(\mathbf{e}) \neq \mathbf{y}$ and $f_{\tau_1}(\mathbf{e})$ does not abstain. Since f_{τ_2} does not abstain whenever f_{τ_1} does not abstain, we have in particular that $f_{\tau_2}(\mathbf{e})$ does not abstain. Moreover, conditioned on not abstaining, we have $f_{\tau_2}(\mathbf{e}) = f_{\tau_1}(\mathbf{e}) \neq \mathbf{y}$. Thus $\mathcal{E}_{\text{adv}}(\tau_2, S)$ incurs error for each test point $(\mathbf{x}, \mathbf{y})$ for which $\mathcal{E}_{\text{adv}}(\tau_2, S)$ incurs an error, implying monotonicity in τ. A similar argument for counting the abstention on any fixed test point for any pair of values of the threshold implies $\mathcal{D}_{\text{nat}}(\tau)$ is monotonically non-increasing. ■

Lemma 14 further implies that $\mathcal{E}_{\text{adv}}(\tau)$ and $\hat{\mathcal{E}}_{\text{adv}}(\tau)$ are also monotonic non-decreasing in τ. The robust error $\mathcal{E}_{\text{adv}}(\tau)$ is optimal at $\tau = 0$, but this implies that we abstain from prediction all the time (that is, $\mathcal{D}_{\text{nat}}(0) = 1$). Conversely, we can minimize the abstention rate by not abstaining, that is, $\mathcal{D}_{\text{nat}}(\infty) = 0$ corresponding to vanilla nearest-neighbor, but this maximizes the robust error. This motivates us to consider the following objective function which combines robust error and natural abstention rate.

Definition 15 (Robust Chow's objective.) *Define $l(\tau) := \mathcal{E}_{\text{adv}}(\tau) + c\mathcal{D}_{\text{nat}}(\tau)$ as the robust Chow's objective, where c is a positive constant and denotes the cost of abstention. Further define $\hat{l}(\tau) := \hat{\mathcal{E}}_{\text{adv}}(\tau) + c\mathcal{D}_{\text{nat}}(\tau)$ as the estimated robust Chow's objective.*

Definition 15 may be viewed as an adversarial version of Chow's objective for abstentive classifiers (Chow, 1970), which uses natural risk instead of adversarial risk. If, for example, we are willing to take a one percent increase of the abstention rate for a two percent drop in the error rate, we could set c to $\frac{1}{2}$. For a single test set $\mathcal{T}$, the abstention rate $\mathcal{D}_{\text{nat}}(\tau)$ can change at (at most) $|\mathcal{T}|$ 'critical' values of τ corresponding to nearest neighbor distances. Given oracle access to $\mathcal{E}_{\text{adv}}(\tau)$, we can minimize $l(\tau)$ over the given test sample with at most $|\mathcal{T}|$ evaluations. Suppose, however, the test data arrives sequentially in batches of size b, potentially from related tasks with different data distributions, and we need to figure out how to set the threshold τ for unseen tasks. As we will show, techniques from data-driven algorithm design (Balcan et al., 2018b, 2021) can help approach this multi-task robustness setting.

Formally, we define our online learning setting as follows. Consider a game consisting of T rounds. In each round $t = 1, \ldots, T$, the learner is presented with a new test batch $\mathcal{T}_t$ of size b. In Theorem 17, we show no regret can be achieved for online learning of the threshold τ using test batches of size b (consisting of unperturbed points) on which the learner chooses abstention threshold τ_t, that is, predicting using classifier f_{τ_t}. Let l_t (resp. $\hat{l}_t$) be the (resp. estimated) robust Chow's objective on the test set $\mathcal{T}_t$. The learner suffers loss $l_t(\tau_t)$ and observes $l_t(\tau)$. The goal of the learner is to minimize total expected regret, defined as $R_T := \mathbb{E}\left[\sum_{t=1}^{T} l_t(\tau_t) - \min_\tau \sum_{t=1}^{T} l_t(\tau)\right]$, where the expectation is over the randomness of the loss functions as well as learner's internal randomness.

Our main result is the following theorem (Theorem 17) in the above setting. Our proof strategy is to show that the sequence of loss functions $l_t(\tau)$ is (w, k)-*dispersed* in the sense of Balcan et al. (2018b). We present a simplified definition of *dispersion* for real-valued functions.

Definition 16 (Dispersion, Balcan et al. (2018b)) *Let $u_1, \ldots, u_T : \mathbb{R} \to [0, 1]$ be a collection of functions where u_i is piecewise Lipschitz over a partition P_i of $\mathbb{R}$. We say that P_i splits a set A if A*

Algorithm 2 Exponential Forecaster Algorithm (Balcan et al., 2018b)

```
Input: step size parameter $\lambda \in (0, 1]$.
Output: thresholds $\tau_t$ for times $t = 1, 2, \ldots, T$.
Set $w_1(\tau) = 1$ for all $\tau \in [0, \tau_{\max}]$.
for $t = 1, 2, \ldots, T$ do
    $W_t := \int_{[0,\tau_{\max}]} w_t(\tau) d\tau$.
    Sample $\tau$ with probability proportional to $w_t(\tau)$, that is, with probability $p_t(\tau) = \frac{w_t(\tau)}{W_t}$.
    Output the sampled $\tau$ as $\tau_t$.
    Observe $l_t(\cdot)$. Set $u_t(\tau) := 1 - \frac{l_t(\tau)}{1+c}$.
    For each $\tau \in [0, \tau_{\max}]$, set $w_{t+1}(\tau) = e^{\lambda u_t(\tau)} w_t(\tau)$.
```

intersects with at least two sets in P_i. The collection of functions is (w, k)-dispersed if every interval of length w is split by at most k of the partitions $P_1, \ldots, P_T$.

Intuitively, if a sequence of functions is piecewise-Lipschitz except for a finite number of breakpoints (or points of discontinuity), it is said to be dispersed if the discontinuities do not concentrate in a small region of the domain space over time. Finally, we will employ known results about no-regret learning of (w, k)-dispersed functions using Algorithm 2 a continuous version of Exponential Weights algorithm for finite experts (Balcan et al., 2018b). Proofs of the technical lemmas needed for proving Theorem 17 can be found in Appendix C.

Theorem 17 *Consider the online learning setting described above. Assume $\tau \in [0, \tau_{\max}]$ with $\tau_{\max} = o(r)$, $r > 1$, and each test batch $\mathcal{T}_t$ is sampled from a data distribution $\mathcal{D}$ that has κ-bounded density. If τ_t is set using a continuous version of the multiplicative updates algorithm, Algorithm 2, for T rounds of the online game, then with probability at least $1 - \delta$, the total expected regret of the learner for the loss sequence $l_t(\tau)$ is bounded by $O\left(\sqrt{n_2 T \log\left(\frac{\kappa m b \tau_{\max} T}{\delta}\right)}\right)$, where b is the batch size, s is the number of sample subspaces used to estimate the robust Chow's objective $\hat{l}(\cdot)$ and r is the smallest distance between points of different labels.*

Proof We show the sequence of loss functions $l_t(\tau)$ is (w, k)-*dispersed* (Definition 16) in two steps. We first argue that the robust error part of the loss $l(\tau)$ is Lipschitz, and we further show that the natural abstention rate is piecewise constant with dispersed discontinuities.

A key challenge is to analyze the adversary success probability and show that $\mathcal{E}_{\text{adv}}(\tau)$ is Lipschitz for sufficiently small τ. In Lemma 26 (see Appendix C for a proof), we show that $\mathcal{E}_{\text{adv}}(\tau)$ is L-Lipschitz, where $L = O\left(m\tau_{\max}^{n_2-n_3-1}/r^{n_2-n_3}\right)$. Intuitively, for any test point the probability the adversary succeeds by perturbing to within a distance τ and $\tau + \Delta$ of a fixed training point can be upper bounded using arguments similar to our proof of bounds on robust error in Section 4. A union bound over training points then gives the bound on L. Note that $\mathcal{D}_{\text{nat}}(\tau)$ is piecewise constant. This is because, for any set $\mathcal{T}_t$ of test points, we have at most $|\mathcal{T}_t|$ points corresponding to distances of the test points to the nearest training point, where the function value decreases by $\frac{1}{|\mathcal{T}_t|}$. Together with L-Lipschitzness of $\mathcal{E}_{\text{adv}}(\tau)$, this implies $l(\tau)$ is piecewise L-Lipschitz.

In Lemma 29 we show that, for batch size b, $\mathcal{D}_{\text{nat}}(\tau)$ has $O(\kappa b m w \tau_{\max}^{n_2-1})$ discontinuities in expectation (over the data distribution) in any interval of width w. Note that if a discontinuity occurs within the interval $I = [\tau, \tau + w]$, then there must exist a test point $\mathbf{x}$ in the test set $\mathcal{T}$ for which the nearest-neighbor training point is at distance $\tau' \in I$. That is, the training point lies within $\mathcal{B}(\mathbf{x}, \tau + w) \setminus \mathcal{B}(\mathbf{x}, \tau)$. The proof involves bounding the fraction of points at distance $d \in [\tau, \tau + w]$ for any test point, using smoothness of the data distribution, and using a union bound over the b test points. See Appendix C for a formal argument. Since $\mathcal{E}_{\text{adv}}(\tau)$ is Lipschitz continuous, $l(\tau)$ has at most $O\left(\kappa bmw\tau_{\max}^{n_2-1}\right)$ discontinuities in expectation in any w-interval.

Using a standard argument based on the VC-dimension of 1D intervals (for example, Theorem 7 in Balcan et al. (2020b)), the maximum number of discontinuities in any interval of width w is $k = O\left(\kappa bmw\tau_{\max}^{n_2-1}T + \sqrt{T \log \frac{b}{\gamma}}\right)$ with high probability $1-\gamma$. In other words, $l(\tau)$ is (w, k)-Lipschitz with high probability over the data distribution. This allows us to use a continuous version of standard Exponential Weights update introduced by Balcan et al. (2018b) as our online algorithm (which we include as Algorithm 2 for completeness), for which they show an $O\left(\sqrt{T \log \frac{R}{w}} + k + wLT\right)$ bound on the expected regret if the sequence of loss functions is (w, k)-dispersed with L-Lipschitz pieces, where R is a bound on the diameter of the continuous domain ($R = \tau_{\max}$ in our setting). Formally, we can apply Theorem 30 with $w = \frac{1}{\kappa bm\tau_{\max}^{n_2-1}\sqrt{T}}$ to get the desired regret bound.

$$\begin{aligned}
R_T &= O\left(\sqrt{T \log \frac{R}{w}} + k + wLT\right) \\
&\leq O\left(\sqrt{T \log \frac{\tau_{\max}}{(\kappa bm\tau_{\max}^{n_2-1}\sqrt{T})^{-1}}} + O\left(\sqrt{T} + \sqrt{T \log \frac{b}{\delta}}\right) + \frac{O(m\tau_{\max}^{n_2-n_3-1}/r^{n_2-n_3})}{\kappa bm\tau_{\max}^{n_2-1}\sqrt{T}} \cdot T\right) \\
&\leq O\left(\sqrt{T \log \left(\frac{\kappa mb\tau_{\max}^{n_2}T}{\delta}\right)}\right),
\end{aligned}$$

where the first inequality holds with probability at least $1 - \delta$. ■

A similar no-regret learning guarantee can also be given for the estimated robust Chow's objective $\hat{l}(\tau)$. In practice $l(\tau)$ can be hard to compute, but as discussed above the learner can more easily estimate this loss by computing $\hat{l}(\tau)$. The key difference in the proof is that the estimated robust error $\hat{\mathcal{E}}_{\text{adv}}(\tau)$ is piecewise constant, while $\mathcal{E}_{\text{adv}}(\tau)$ was shown to be Lipschitz for small τ. Roughly speaking, we will use smoothness of the adversary distribution to argue that location of discontinuities of $\hat{\mathcal{E}}_{\text{adv}}(\tau)$ cannot concentrate in a small interval. Formally, we show that

Theorem 18 *Consider the online learning setting described above. Assume $\tau \in [0, \tau_{\max}]$ with $\tau_{\max} = o(r)$, $r > 1$ and each test batch $\mathcal{T}_t$ is sampled from a data distribution $\mathcal{D}$ that has κ-bounded density. If τ_t is set using a continuous version of the multiplicative updates algorithm, Algorithm 2, for T rounds of the online game, then with probability at least $1 - \delta$, the total expected regret of the learner for the loss sequence $\hat{l}_t(\tau)$ is bounded by $O\left(\sqrt{n_2 T \log\left(\frac{\kappa msb\tau_{\max}T}{\delta}\right)}\right)$, where b is the batch size, s is the number of sample subspaces used to estimate the robust Chow's objective $\hat{l}(\cdot)$ and r is the smallest distance between points of different labels.*

Proof Lipschitzness of $\mathcal{E}_{\text{adv}}(\tau)$ also implies that the breakpoints of $\hat{\mathcal{E}}_{\text{adv}}(\tau)$ are smoothly distributed, in particular in any interval of width w, we have at most $O\left(bmw\tau_{\max}^{n_2-n_3-1}/r^{n_2-n_3}\right)$ discontinuities

(Corollary 28), in expectation over the draw of the adversarial subspace. The rest of the argument is very similar to part (i) above. ■

In this work we restrict our attention to the *full information* setting where entire function $l_t(\tau)$ is available to the learner after the prediction in round t. It is an interesting future question to model the adversary with bandit feedback where only $l_t(\tau_t)$ is revealed to the learner. The test sets $\mathcal{T}_t$ may be adversarial as long as they are generated by smooth but possibly different data distributions (in the sense of Theorem 17). Our experiments in Section 6 indicate Algorithm 1 can be made more effective by tuning both parameters τ and σ together. Effective tuning of data-driven algorithms with multiple parameters is an interesting research direction (Balcan et al., 2022d). Finally, we perform the analysis for tuning our relatively simple thresholded nearest-neighbor approach, but data-driven algorithm design may prove useful for selecting the best data-specific robust approach from candidate algorithms more generally.

Remark 19 *A simple goal for setting τ is to fix an upper limit d^* on $\mathcal{D}_{\text{nat}}(\tau)$, corresponding to a maximum abstention rate allowed on the natural data. It is straightforward to search for an optimal τ^* such that $\mathcal{D}_{\text{nat}}(\tau^*) = \max_{\tau, \mathcal{D}_{\text{nat}}(\tau) \leq d^*} \mathcal{D}_{\text{nat}}(\tau)$—simply use the nearest neighbor distances (to training examples) for the test points to compute the abstention rate at any τ, and do a binary search for d^*. For $\tau < \tau^*$ we have a higher abstention rate, and when $\tau > \tau^*$ we have a higher robust error rate. For more sophisticated goals, for example minimizing objectives that depend on both $\mathcal{E}_{\text{adv}}(\tau)$ and $\mathcal{D}_{\text{nat}}(\tau)$, we may not be able to perform a binary search, though a linear search would still suffice. Here we have considered a setting where we have multiple test sets, conceptually coming from different but related tasks in some domain, and rather than separately performing this parameter tuning on each task, we want instead to learn a common value of τ that works well across all the tasks.*

5.1 A simple intuitive example with exact calculation demonstrating significance of data-driven algorithm design

The significance of data-driven design in this setting is underlined by the following two observations. Firstly, as noted above, optimization for τ across problem instances is difficult due to the non-Lipschitz nature of $\mathcal{D}_{\text{nat}}(\tau)$ and the intractability of characterizing the objective function $l(\tau)$ exactly due to $\mathcal{E}_{\text{adv}}(\tau)$. Secondly, the optimal value of τ can be a complex function of the data geometry and sampling rate. We illustrate this by exact computation of optimal τ for a simple intuitive setting: consider a binary classification problem where the features lie uniformly on two one-dimensional manifolds embedded in two-dimensions (that is, $n_2 = 2$, see Figure 4). Assume that the adversary perturbs in a uniformly random direction ($n_3 = 1$). Further assume that our training set consists of $2m$ examples, m from each class. In this toy setting, we show that the optimal threshold varies with data-specific factors.

Formal setting: We set the feature and adversary dimensions as $n_2 = 2, n_3 = 1$. Examples of class A are all located on the segment $S_A = [(0,0),(D,0)]$, similarly instances of class B are located on $S_B = [(D+r,0),(2D+r,0)]$ (where $[\mathbf{a},\mathbf{b}] := \{\alpha\mathbf{a} + (1-\alpha)\mathbf{b} \mid \alpha \in [0,1]\}$). The data distribution returns an even number of samples, $2m$, with $m > 0$ points each drawn uniformly from S_A and S_B. For this setting, we show that the optimal value of the threshold is a function of both the geometry

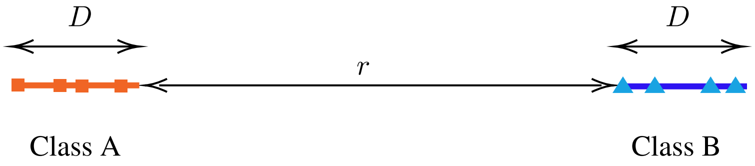

Figure 4: A simple example where we compute the optimal value of the abstention threshold exactly. Classes A and B are both distributed respectively on segments of length D, embedded collinear and at distance r in $\mathbb{R}^2$.

(D, r) and the sampling rate (m). Proof of lemmas needed to prove the following result appear in Appendix E.

Theorem 20 *Let $\tau^* := \operatorname{argmin}_{\tau\in\mathbb{R}^+} l(\tau)$. For the setting considered above, if we further assume $D = o(r)$ and $m = \omega\left(\log\left(\frac{2\pi cr}{D}\right)\right)$, then there is a unique value of τ^* in $[0, D/2)$. Further,*

$$\tau^* = \begin{cases} \Theta\left(\frac{D\log((\pi crm)/D)}{m}\right), & \text{if } \frac{1}{m} < \frac{\pi cr}{D}; \\ 0, & \text{if } \frac{\pi cr}{D} \le \frac{1}{m}. \end{cases}$$

Proof We compute the robust error $\mathcal{E}_{\text{adv}}(\tau)$ and abstention rate $\mathcal{D}_{\text{nat}}(\tau)$ as functions of τ. Even with $D = o(r)$, the exact computation of the robust error as a simple closed form is difficult without further assuming $\tau = o(r)$ as well. Fortunately, by Lemma 32, we only need to consider $\tau \le D$. For this case, indeed $\tau = o(r)$. We compute the abstention and robust error rates in Lemmas 33 and 34, respectively. This gives us, for $\tau \le D$,

$$\begin{aligned} l(\tau) =& \frac{\tau}{\pi r}\left(1 - \frac{m+3}{m+1}\cdot\Theta\left(\frac{D}{r}\right)\right) - \Theta\left(\left(\frac{\tau}{r}\right)^3\right) \\ &+ \frac{c}{m+1}\left[2\left(1-\frac{\tau}{D}\right)^{m+1} + (m-1)\mathbb{I}_{\tau\le D/2}\left(1-\frac{2\tau}{D}\right)^{m+1}\right]. \end{aligned}$$

For $\tau \le D/2$,

$$\begin{aligned} l'(\tau) =& \frac{1}{\pi r}\left(1 - \frac{m+3}{m+1}\cdot\Theta\left(\frac{D}{r}\right)\right) - \Theta\left(\frac{1}{r}\left(\frac{\tau}{r}\right)^2\right) \\ &- \frac{2c}{D}\left[\left(1-\frac{\tau}{D}\right)^{m} + (m-1)\left(1-\frac{2\tau}{D}\right)^{m}\right]. \end{aligned}$$

We need to consider two cases.
Case 1. $\frac{\pi cr}{D} \le \frac{1}{m}$. In this case $l'(0) = \frac{1}{\pi r} - \frac{2cm}{D} \ge 0$. Since $l''(\tau) \ge 0$, so we must have the only minimum at $\tau = 0$.

Case 2. $\frac{1}{m} < \frac{\pi cr}{D}$. $l'(0) = \frac{1}{\pi r} - \frac{2cm}{D} < 0$. Also $l'(D/2) = \frac{1}{\pi r} - \frac{2c}{D2^m} > 0$ since $m > \log\left(\frac{2\pi cr}{D}\right)$. But $l''(\tau) \ge 0$, so we must have a unique local minimum in $(0, D/2)$, which is the global minimum. Further, define y as $\tau = \frac{D}{m}\log y$. Now if $y = 2^{o(m)}$, we have $\frac{\tau}{D} = o(1)$, or

$$\left(1-\frac{\tau}{D}\right)^m = \exp\left(m\log\left(1-\frac{\tau}{D}\right)\right) = y^{-1-o(1)}.$$

If $y > 1$, for $y = \frac{2\pi crm}{D}$,

$$\begin{aligned} l'(\tau) &= \frac{1}{\pi r} - \frac{2c}{D}\left[\left(\frac{D}{2\pi crm}\right)^{1+o(1)} + (m-1)\left(\frac{D}{2\pi crm}\right)^{2+o(1)}\right] \\ &> \frac{1}{\pi r} - \frac{2c}{D}\left[\left(\frac{D}{2\pi crm}\right)^{1} + (m-1)\left(\frac{D}{2\pi crm}\right)^{1}\right] \\ &= \frac{1}{\pi r} - \frac{2c}{D}\left[\frac{D}{2\pi cr}\right] = 0, \end{aligned}$$

and for $y = \left(\frac{2\pi cr(m-1)}{D}\right)^{1/4}$,

$$\begin{aligned} l'(\tau) &= \frac{1}{\pi r} - \frac{2c}{D}\left[\left(\frac{D}{2\pi cr(m-1)}\right)^{\frac{1}{4}+o(1)} + (m-1)\left(\frac{D}{2\pi crm}\right)^{\frac{1}{2}+o(1)}\right] \\ &< \frac{1}{\pi r} - \frac{2c}{D}\left[\left(\frac{D}{2\pi cr(m-1)}\right)^{1} + (m-1)\left(\frac{D}{2\pi cr(m-1)}\right)^{1}\right] \\ &= \frac{-1}{\pi r(m-1)} < 0. \end{aligned}$$

Together, we get that $\tau^* = \Theta\left(\frac{D\log((\pi crm)/D)}{m}\right)$ in this case. ■

6. Experiments on Contrastive Learning

Contrastive learning has received significant attention due to the recent popularity of self-supervised learning: many recent studies (Wu et al., 2018; Oord et al., 2018; Hjelm et al., 2018; Zhuang et al., 2019; Hénaff et al., 2020; Tian et al., 2019; Bachman et al., 2019) present promising results of unsupervised representation learning against their supervised counterparts. Representative self-supervised contrastive learning includes MoCo(v2) (He et al., 2020) and SimCLR (Chen et al., 2020a). In ImageNet classification task, both methods almost match the accuracy of their supervised counterparts; in 7 detection/segmentation tasks on PASCAL VOC, COCO, and other data sets, MoCo (He et al., 2020) can outperform its supervised pre-training counterpart sometimes by large margins. A more recent work of Khosla et al. (2020) proposed *supervised contrastive learning*.

Theorem 2 sheds light on how to design algorithms for robust learning of feature embedding F. In order to preserve robustness against adversarial examples regarding a given test point $\mathbf{x}$, in the feature space the theorem suggests minimizing τ—the closest distance between $F(\mathbf{x})$ and any training example $F(\mathbf{x}_i)$ with the same label, and maximizing r—the closest distance between $F(\mathbf{x})$ and any training example $F(\mathbf{x}_i)$ with a different label. This is conceptually consistent with the spirit of the nearest-neighbor algorithm. Indeed, contrastive loss can be seen as nearest-neighbor loss (in the feature space) with the *max* operator replaced by a *softmax* operator for differentiable training:

$$\min_F -\frac{1}{m}\sum_{i\in[m]} \log\left(\frac{\sum_{j\in[m],j\neq i,y_i=y_j} e^{-\frac{\|F(\mathbf{x}_i)-F(\mathbf{x}_j)\|^2}{T}}}{\sum_{k\in[m],k\neq i} e^{-\frac{\|F(\mathbf{x}_i)-F(\mathbf{x}_k)\|^2}{T}}}\right), \tag{1}$$

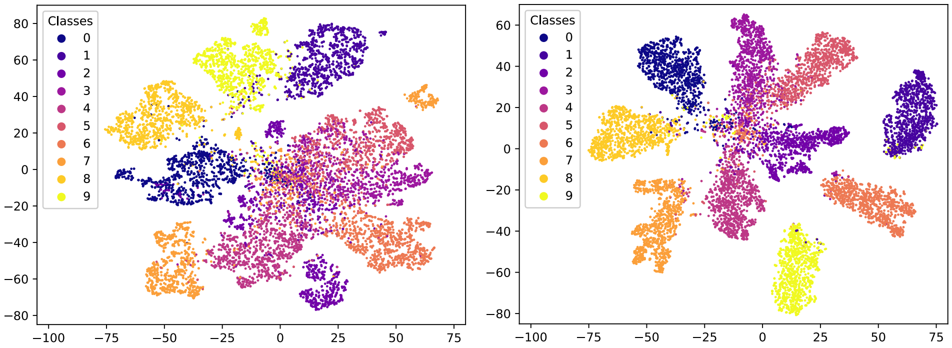

Figure 5: Two-dimensional t-SNE visualization of 512-dimensional embedding by contrastive learning on the CIFAR10 test data set. **Left Figure:** Self-supervised contrastive learning. **Right Figure:** Supervised contrastive learning.

where $T > 0$ is the temperature parameter. Loss (1) is also known as the soft-nearest-neighbor loss in the context of supervised learning (Frosst et al., 2019), or the InfoNCE loss in the setting of self-supervised learning (He et al., 2020).

We will now describe an implementation of the attack and empirically measure the performance of our algorithm in the context of supervised and self-supervised contrastive learning[3].

6.1 Visualization of Representations of Contrastive Learning

Figure 5 shows the two-dimensional t-SNE visualization of 10,000 features by minimizing loss (1) on the CIFAR10 test data set. It shows that $\tau_x \ll r_x$ for most of data, where we define $\tau_x := \min_{i:y=y_i} \mathsf{dist}(F(\mathbf{x}), F(\mathbf{x}_i))$, $r_x := \min_{i:y\neq y_i} \mathsf{dist}(F(\mathbf{x}), F(\mathbf{x}_i))$, and $\{\mathbf{x}_i\}_{i=1}^m$ is a set of training example with labels y_i.

To have a closer look at τ_x vs. r_x, we plot the frequency of τ_x/r_x in Figure 6. For self-supervised contrastive learning, there is 84.5% data which has τ_x/r_x smaller than 1.0, while for supervised setting, there is 94.3% data which has τ_x/r_x smaller than 1.0.

6.2 Certified Adversarial Robustness against Exact Computation of Attacks

We verify the robustness of Algorithm 1 when the representations are learned by contrastive learning. Given a embedding function F and a classifier f which outputs either a predicted class or abstains from predicting, recall that we define the natural and robust errors, respectively, as $\mathcal{E}_{\text{nat}}(f) := \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}\mathbf{1}\{f(F(\mathbf{x})) \neq \mathbf{y} \text{ and } f(F(\mathbf{x})) \text{ does not abstain}\}$, and $\mathcal{E}_{\text{adv}}(f) := \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D},S\sim\mathcal{S}}\mathbf{1}\{\exists\mathbf{e} \in S + F(\mathbf{x}) \subseteq \mathbb{R}^{n_2} \text{ s.t. } f(\mathbf{e}) \neq \mathbf{y} \text{ and } f(\mathbf{e}) \text{ does not abstain}\}$, where $S \sim \mathcal{S}$ is a random adversarial subspace of $\mathbb{R}^{n_2}$ with dimension n_3. $\mathcal{D}_{\text{nat}}(f) := \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}\mathbf{1}\{f(F(\mathbf{x})) \text{ abstains}\}$ is the abstention rate on the natural examples. Note that the robust error is always at least as large as the natural error.

Self-supervised contrastive learning setup. Our experimental setup follows that of SimCLR (Chen et al., 2020a). We use the ResNet-18 architecture (He et al., 2016) for representation learning with a

[3] Code used in the experiments may be found at the following github link: `https://github.com/dravyanshsharma/adversarial-contrastive`

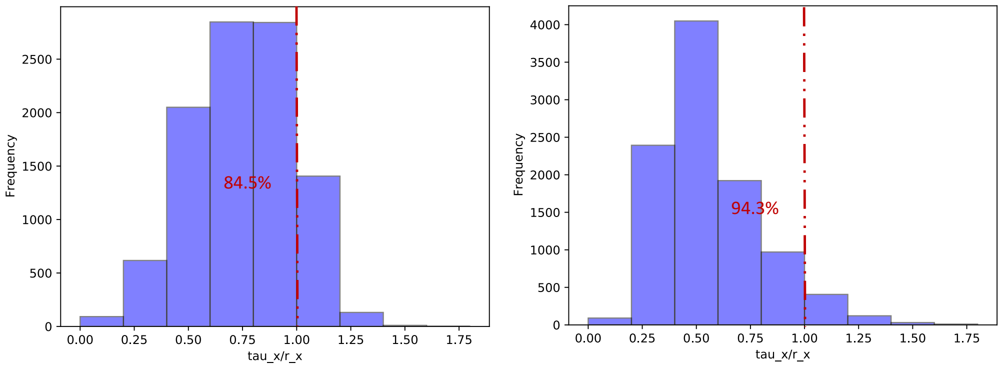

Figure 6: Frequency of τ_x/r_x by contrastive learning on the CIFAR10 data set, where τ_x represents the closest distance between the test embedding and any training embedding of the same label, and r_x stands for the closest distance between the test embedding and any training embedding of different labels. **Left Figure:** Self-supervised contrastive learning. **Right Figure:** Supervised contrastive learning.

Table 1: Natural error $\mathcal{E}_{\text{nat}}$ and robust error $\mathcal{E}_{\text{adv}}$ on the CIFAR-10 data set (Szegedy et al., 2015) when $n_3 = 1$ and the 512-dimensional representations are learned by contrastive learning, where $\mathcal{D}_{\text{nat}}$ represents the fraction of each algorithm's output of "don't know" on the natural data. We report values for $\sigma \approx \tau$ as they tend to give a good abstention-error trade-off w.r.t. σ. Bold values correspond to parameter settings that minimize $\mathcal{E}_{\text{adv}} + \mathcal{D}_{\text{nat}}$ over the grid.

	Contrastive	Linear Protocol		Ours ($\tau = 3.0$)			Ours ($\tau = 2.0$)		
		$\mathcal{E}_{\text{nat}}$	$\mathcal{E}_{\text{adv}}$	$\mathcal{E}_{\text{nat}}$	$\mathcal{E}_{\text{adv}}$	$\mathcal{D}_{\text{nat}}$	$\mathcal{E}_{\text{nat}}$	$\mathcal{E}_{\text{adv}}$	$\mathcal{D}_{\text{nat}}$
$(\sigma = 0)$	Self-supervised	8.9%	100.0%	15.4%	40.7%	2.2%	14.3%	26.2%	28.7%
	Supervised	5.6%	100.0%	5.7%	60.5%	0.0%	5.7%	33.4%	0.0%
$(\sigma = 0.9\tau)$	Self-supervised	8.9%	100.0%	**7.2%**	**9.4%**	**12.9%**	10.0%	17.7%	29.9%
	Supervised	5.6%	100.0%	6.2%	18.9%	0.0%	5.6%	22.0%	0.1%
$(\sigma = \tau)$	Self-supervised	8.9%	100.0%	1.1%	1.2%	33.4%	2.1%	3.1%	49.9%
	Supervised	5.6%	100.0%	1.9%	2.8%	10.6%	**4.1%**	**4.8%**	**3.3%**

two-layer projection head of width 128. The dimension of the representations is 512. We set batch size 512, temperature $T = 0.5$, and initial learning rate 0.5 which is followed by cosine learning rate decay. We sequentially apply four simple augmentations: random cropping followed by resizing back to the original size, random flipping, random color distortions, and randomly converting image to grayscale with a probability of 0.2. In the linear evaluation protocol, we set batch size 512 and learning rate 1.0 to learn a linear classifier in the feature space by empirical risk minimization. All experiments are run on two GeForce RTX 2080 GPUs.

Supervised contrastive learning setup. Our experimental setup follows that of Khosla et al. (2020). We use the ResNet-18 architecture for representation learning with a two-layer projection head of width 128. The dimension of the representations is 512. We set batch size 512, temperature $T = 0.1$, and initial learning rate 0.5 which is followed by cosine learning rate decay. We sequentially apply four simple augmentations: random cropping followed by resize back to the original size, random flipping, random color distortions, and randomly converting image to grayscale with a probability of 0.2. In the linear evaluation protocol, we set batch size 512 and learning rate 5.0 to learn a linear classifier in the feature space by empirical risk minimization.

Algorithm for exact implementation of the attack. In both self-supervised and supervised setups, we compare the robustness of the linear protocol with that of our defense protocol in Algorithm 1 under exact computation of adversarial examples using a convex optimization program in n_3 dimensions and m constraints. Algorithm 3 provides an efficient implementation of the attack.

Algorithm 3 Exact computation of attacks under threat model 2.1 against Algorithm 1

```
 1: Input: A randomly-sampled adversarial subspace $S$ of dimension $n_3$, a test example $F(\mathbf{x})$ and
    its label $y$, a set of training examples $F(\mathbf{x}_i)$ and their labels $y_i$, $i \in [m]$, a threshold parameter $\tau$.

 2: Output: A misclassified adversarial feature $F(\mathbf{x}) + \mathbf{v}$, $\mathbf{v} \in S$ if it exists; otherwise, output "no
    adversarial example".
 3: $F_{\text{center}}(\mathbf{x}_i) \leftarrow F(\mathbf{x}_i) - F(\mathbf{x})$ for $i \in [m]$.
 4: for $i = 1, ..., m$ do
 5:   if $y_i \neq y$ then
 6:     $\mathbf{u}_i = \text{argmin}_{\mathbf{u} \in S}\, d(\mathbf{u}, F_{\text{center}}(\mathbf{x}_i))$;          (candidate adversarial perturbation)
 7:     $C \leftarrow \{\mathbf{x}_j \mid y_j = y\}$;
 8:     if $\exists \mathbf{w} \in C \mid \mathsf{dist}(\mathbf{u}_i, F_{\text{center}}(\mathbf{w})) < \mathsf{dist}(\mathbf{u}_i, F_{\text{center}}(\mathbf{x}_i))$ then
 9:       $H_j \leftarrow \{\mathbf{z} \mid \mathsf{dist}(F_{\text{center}}(\mathbf{x}_i), \mathbf{z}) \leq \mathsf{dist}(\mathbf{w}_j, \mathbf{z}), \mathbf{w}_j \in C\}$;
10:       $H \leftarrow \cap_i H_i$;
11:       $A \leftarrow H \cap S$;
12:       if $A = \{\}$ then
13:         continue;
14:       $\mathbf{z}_i = \text{argmin}_{\mathbf{z} \in A}\, \mathsf{dist}(\mathbf{z}, F_{\text{center}}(\mathbf{x}_i))$;          (candidate adversarial perturbation)
15:     else
16:       $\mathbf{z}_i \leftarrow \mathbf{u}_i$;
17:     if $\mathsf{dist}(\mathbf{z}_i, F_{\text{center}}(\mathbf{x}_i)) < \tau$ then
18:       return $F(\mathbf{x}) + \mathbf{z}_i$.
19: return "no adversarial example".
```

Overview of Algorithm 3. If the point $\mathbf{u}_i$ closest to the training point $\mathbf{x}_i$ of different label than test point $\mathbf{x}$ in the adversarial subspace S (slight abuse of notation to refer to $\mathbf{x} + S$ as S) is closer to $\mathbf{x}_i$ than any training point $\mathbf{w}_j$ with the same label as $\mathbf{x}$ and within the threshold τ of $\mathbf{x}_i$, it will be misclassified as $\mathbf{x}_i$ (or potentially another point of an incorrect label). If however $\mathbf{u}_i$ is closer to some $\mathbf{w}_j$, we look at the points closer to $\mathbf{x}_i$ than all $\mathbf{w}_j$ in the subspace S, and consider the closest point $\mathbf{z}_i$ to $\mathbf{x}_i$ (if it is within threshold τ) which should be misclassified. This can be computed using a convex optimization program (Line 14 of Algorithm 3) in n_3 dimensions. We claim it is sufficient to look at these two points for each training example $\mathbf{x}_i$.

Proof of correctness. To argue correctness of Algorithm 3, suppose an adversary wins by perturbing to some point $\mathbf{v}$. Then $\mathbf{v}$ must be closer to some point $\mathbf{x}_i$ than all $\mathbf{w}_j \in C$ (the set of

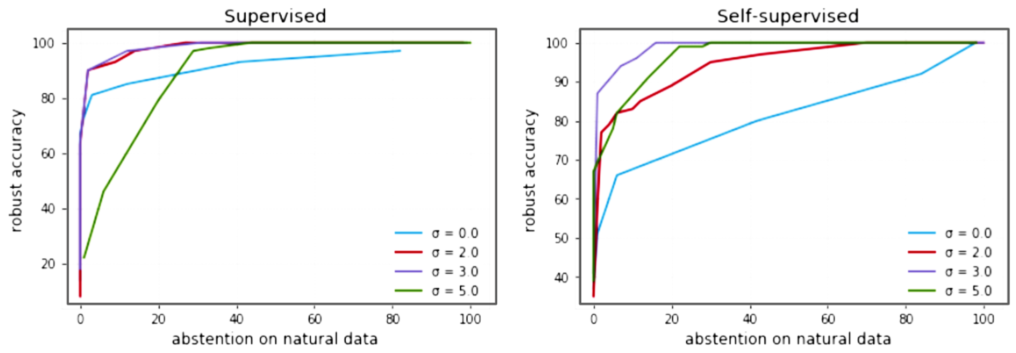

Figure 7: Adversarial accuracy (that is, rate of adversary failure) vs. abstention rate as threshold τ varies for $n_3 = 1$ and different outlier removal thresholds σ. Each colored line corresponds to a fixed σ, as τ is varied from 0 (always abstain) to infinity (vanilla nearest-neighbor).

training points with same label as $\mathbf{x}$) and within τ of $\mathbf{x}_i$. If $\mathbf{u}_i$ is closer to $\mathbf{x}_i$ than all $\mathbf{w}_j \in C$ then, it must be at least as close as $\mathbf{v}$ (since $\mathbf{v}$ is in the adversarial subspace S) and therefore within τ of $\mathbf{x}_i$.

Otherwise there is some $\mathbf{w}_j$ closer to $\mathbf{u}_i$ than $\mathbf{x}_i$. Let H be the convex polytope of points closer to $\mathbf{x}_i$ than $\mathbf{w}_j$'s in C. Consider the intersection A of H with S. All points in A are misclassified by our algorithm, if within the threshold τ. $\mathbf{v}$ must lie within A since it is closer to $\mathbf{x}_i$. $\mathbf{u}_i$ must lie outside of A in this case. If $\mathbf{v}$ is within τ of $\mathbf{x}_i$, so is $\mathbf{u}_i$ and therefore also the line joining the two. If this line intersects A at point $\mathbf{v}$, then $\mathbf{v}$ is a valid adversarial point and so is point closest to $\mathbf{x}_i$ in A. This proves completeness of the algorithm, soundness is more straightforward to verify.

Experimental results. We summarize our results in Table 1. Comparing with a linear protocol, our algorithms have much lower robust error. Note that even if abstention is added based on distance from the linear boundary, sufficiently large perturbations will ensure the adversary can always succeed. For an approximate adversary which can be efficiently implemented for large n_3, see Appendix F.1.

6.3 Robustness-abstention Trade-off

The threshold parameter τ captures the trade-off between the robust accuracy $\mathcal{A}_{\text{adv}} := 1 - \mathcal{E}_{\text{adv}}$ and the abstention rate $\mathcal{D}_{\text{nat}}$ on the natural data. We report both metrics for different values of τ for supervised and self-supervised contrastive learning. The supervised setting enjoys higher adversarial accuracy and a smaller abstention rate for fixed τ's due to the use of extra label information. We plot $\mathcal{A}_{\text{adv}}$ against $\mathcal{D}_{\text{nat}}$ for Algorithm 1 as hyperparameters vary. For small τ, both accuracy and abstention rate approach 1.0. As the threshold increases, the abstention rate decreases rapidly and our algorithm enjoys good accuracy even with small abstention rates. For $\tau \to \infty$ (that is the nearest neighbor search), the abstention rate on the natural data $\mathcal{D}_{\text{nat}}$ is 0% but the robust accuracy is also roughly 0%. Increasing σ (for small σ) gives us higher robust accuracy for the same abstention rate. Too large σ may also lead to degraded performance (Figure 7).

7. Discussion and Conclusion

We propose a model to study robustness of non-Lipschitz networks, against an adversary whose perturbations modify the features in a random low-dimensional subspace. Our first result is that in our model if the learner does not use any abstention, then the adversary will succeed for any data distribution. To complement our lower bound, we present a threshold-equipped nearest-neighbor classifier that simultaneously achieves low robust error as well as low abstention rate on natural data. Our robust error guarantee is independent of the distribution, and is small as long as the label classes are well-separated in the feature space. Our bounds for abstention rate scale with the covering number of the distribution, and hold for sufficiently large training set size m. Our positive results indicate a trade-off between the robust error and abstention rate. We further show how one may tune the threshold to minimize a combination of robust error and abstention rate using techniques from data-driven algorithm design. We also validate our positive results empirically for contrastive learning based deep networks.

Adversarial robustness is an important challenge for the practical deployment of deep networks. We believe we should analyze different types of adversaries beyond classic ones (ℓ_∞, ℓ_2, ℓ_1 bounded-norm perturbations) which have largely been the focus in our community. We view our contribution as defining and analyzing a new and interesting type of adversary designed to help in studying the robustness of non-Lipschitz networks. It is an interesting open question to provide new families of adversaries as well as defenses for them, since bounded-norm models are limited in their ability to capture all possible realistic attacks.

Acknowledgments

This work was supported in part by the National Science Foundation under grants CCF-1815011, CCF-1535967, IIS-1901403, CCF-1910321, SES-1919453, the Defense Advanced Research Projects Agency under cooperative agreement HR00112020003, the NSERC Discovery Grant RGPIN-2022-03215, DGECR-2022-00357, an AWS Machine Learning Research Award, a Microsoft Research Faculty Fellowship, and a Bloomberg Research Grant. The views expressed in this work do not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred. Approved for public release; distribution is unlimited.

Appendix A. Technical Lemmas Needed for Results in Section 4

The following lemma gives a bound on the fraction of the surface of the sphere at some fixed small distance from the subspace in Theorem 4. The bound involves a geometric calculation of a surface element of a sphere in $\mathbb{R}^n$.

Lemma 21 *The fraction of the surface of the unit $(n-1)$-sphere at a distance at most small $\varepsilon = o(1)$ from a fixed $(n-k)$-hyperplane through its center is at most $\frac{2\varepsilon^k}{k} \cdot \frac{A(k-1)A(n-k-1)}{A(n-1)}$, where $A(m)$ is the surface-area of the unit m-sphere embedded in $m+1$ dimensions.*

Proof Let the fixed hyperplane be $x_1 = x_2 = \cdots = x_k = 0$. We change the coordinates to a product of spherical coordinates (ρ is the distance from the hyperplane, r is the orthogonal component of the radius vector).

$$x_j = \begin{cases} \rho S_{j-1} \cos \phi_j, & \text{if } j < k; \\ \rho S_{j-1}, & \text{if } j = k; \\ r T_{j-k-1} \cos \alpha_{j-k}, & \text{if } k < j < n; \\ r T_{j-k-1}, & \text{if } j = n. \end{cases}$$

where $S_l = \prod_{i=1}^{l} \sin \phi_i$ and $T_l = \prod_{i=1}^{l} \sin \alpha_i$. The desired surface area is easier to compute in the new coordinate system.

The new coordinates are $(y_1, \ldots, y_n) = (\rho, \phi_1, \phi_2, \ldots, \phi_{k-1}, r, \alpha_1, \ldots, \alpha_{n-k-1})$. Let $z = \sqrt{r^2 + \rho^2} = \sqrt{\sum_{i=1}^n x_i^2}$ denote the usual radial spherical coordinate. Volume element in this new coordinate system is given by

$$dV = |\det(J)|\, d\rho\, d\phi_1 \ldots d\phi_{k-1} dr\, d\alpha_1 \ldots d\alpha_{n-k-1},$$

where J is the Jacobian matrix, $J_{ij} = \frac{\partial x_i}{\partial y_j}$. We can write

$$J = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix},$$

where $A_{ij} = \frac{\partial x_i}{\partial y_j}$ for $1 \le i, j \le k$ and $B_{ij} = \frac{\partial x_{i+k}}{\partial y_{j+k}}$ for $1 \le i, j \le n-k$.

By Leibniz formula for determinants, it is easy to see

$$\begin{aligned} \det(J) &= \det(A) \cdot \det(B) \\ &= \rho^{k-1} \left(\prod_{i=1}^{k-2} \sin^{k-i-1} \phi_i \right) \cdot r^{n-k-1} \left(\prod_{i=1}^{n-k-2} \sin^{n-k-i-1} \alpha_i \right) \\ &= \rho^{k-1} r^{n-k-1} \left(\prod_{i=1}^{k-2} \sin^{k-i-1} \phi_i \right) \left(\prod_{i=1}^{n-k-2} \sin^{n-k-i-1} \alpha_i \right). \end{aligned}$$

Now the surface element is given by

$$dS = \frac{1}{z^{n-1}} \frac{dV}{dz} = \frac{1}{z^{n-1}} \left(\frac{dV}{dr} \frac{\partial r}{\partial z} + \frac{dV}{d\rho} \frac{\partial \rho}{\partial z} \right) = \frac{1}{r z^{n-2}} \frac{dV}{dr} + \frac{1}{\rho z^{n-2}} \frac{dV}{d\rho}.$$

Plugging in our computation for dV,

$$dS = \left(\frac{\rho^{k-1}r^{n-k-2}}{z^{n-2}}\,d\rho + \frac{\rho^{k-2}r^{n-k-1}}{z^{n-2}}\,dr\right)\left(\prod_{i=1}^{k-2}\sin^{k-i-1}\phi_i d\phi_i\right)\left(\prod_{i=1}^{n-k-2}\sin^{n-k-i-1}\alpha_i d\alpha_1\right).$$

We care about $z = 1$ and $\rho \le \varepsilon$ (or $r \ge \sqrt{1-\varepsilon^2}$). Notice

$$\int_{\sqrt{1-\varepsilon^2}}^{1} \frac{\rho^{k-2}r^{n-k-1}}{z^{n-2}}\,dr = \int_{\varepsilon}^{0} \rho^{k-2}r^{n-k-1}\frac{-\rho d\rho}{r} = \int_0^{\varepsilon} \rho^{k-1}r^{n-k-2}d\rho.$$

Thus, using the surface element in the new coordinates and integrating, we get

$$\text{Area of } \varepsilon\text{-close points} = A(k-1)A(n-k-1)\cdot 2\int_0^{\varepsilon} \rho^{k-1}r^{n-k-2}d\rho \le A(k-1)A(n-k-1)\cdot\frac{2\varepsilon^k}{k}$$

which gives the desired fraction. ■

The following lemma establishes a useful convexity property for the adversarial linear subspaces.

Lemma 22 *Let* $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^{n_2}, \tau \in \mathbb{R}^+$ *and* $\mathcal{S}(x', \tau)$ *denote the subset of linear subspaces of dimension* n_3 *such that for any* $S \in \mathcal{S}(x', \tau)$ *there exists* $v \in S$ *with* $x + v \in \mathcal{B}(x', \tau)$*. The set* $\mathcal{S}(x', \tau)$ *is convex.*

Proof Let $S, S' \in \mathcal{S}(x', \tau)$. Then we have $v \in S, v' \in S'$ such that $x + v, x + v' \in \mathcal{B}(x', \tau)$. Let $S^* = \alpha S + (1-\alpha)S'$, $\alpha \in [0, 1]$. Pick $v^* = \alpha v + (1-\alpha)v' \in S^*$. $x + v^*$ must lie in $\mathcal{B}(x', \tau)$ by convexity of $\mathcal{B}(x', \tau)$. ■

Appendix B. Error Upper Bound with Outlier Removal

Our results will be good for distributions for which the induced distribution $\mathcal{D}_\sigma$ after the preprocessing step of Algorithm 1 satisfies the following property with small $N = \sum_y |\mathcal{B}^y|$.

Definition 23 *A distribution* $\mathcal{D}$ *is* σ*-separably* $\{\mathcal{B}^y\}$*-coverable if all points in the support of the marginal distribution* $\mathcal{D}_{F(\mathcal{X})|y}$ *over* $\mathbb{R}^{n_2}$ *can be covered by balls in the set* $\mathcal{B}^y = \{\mathbb{B}_1^y, \dots, \mathbb{B}_{N_y}^y\}$*, of radius* $\tau/2$ *such that*

$$\min_{\substack{F(\mathbf{x})\in\mathbb{B}_i^y, F(\mathbf{x}')\in\mathbb{B}_j^{y'},\\ y\neq y'}} \mathsf{dist}(F(\mathbf{x}), F(\mathbf{x}')) \ge \sigma.$$

In addition, we will assume that a test point $(\mathbf{x}, y)$ from the natural distribution $\mathcal{D}$ has the property that $\mathbf{x}$ is covered by some ball in $\mathcal{B}^y$ with high probability.

Theorem 24 *Suppose the distribution* $\mathcal{D}_\sigma$ *induced by the preprocessing step of Algorithm 1 is* σ*-separably* $\{\mathcal{B}^y\}$*-coverable with finite* $N = \sum_y |\mathcal{B}^y|$*. Let* $\Pr_{\mathbf{x},y\sim\mathcal{D}}[\mathbf{x} \in \cup_{\mathbb{B}_i\in\mathcal{B}^y}\mathbb{B}_i] \ge 1-\gamma$*. If* $\tau = o(\sigma)$*, the robust error of Algorithm 1 on any test point* $\mathbf{x} \sim \mathcal{D}_{F(\mathcal{X})}$ *is at most*

$$\mathcal{O}\left(N\left(\frac{c\tau}{(\sigma+\tau/2)\sqrt{1-\frac{n_3}{n_2}}}\right)^{n_2-n_3} + Nc_0^{n_2-n_3} + \gamma\right),$$

where $c > 0$ *and* $0 < c_0 < 1$ *are absolute constants.*

Proof Let $\mathbf{x}, y \sim \mathcal{D}$. We will bound the probability the adversary succeeds for a test point $\mathbf{x}$ covered by $\cup_y \mathcal{B}^y$, that is $\Pr[\text{adversary succeeds on } \mathbf{x} \mid \mathbf{x} \in \cup_{\mathbb{B}_i \in \mathcal{B}^y} \mathbb{B}_i]$. Let $\mathbf{x}_i$ be a training point that survives the preprocessing step of Algorithm 1, and belongs to a different class than $\mathbf{x}$. By the covering assumption, $\mathbf{x}_i \in \mathbb{B}_j^{y'}$ for some $y' \neq y$ and $\mathbb{B}_j^{y'} \in \mathcal{B}^{y'}$. Let $\mathbf{c}$ denote the center of $\mathbb{B}_j^{y'}$. By the σ-separable property, we have $\mathsf{dist}(\mathbf{x}, \mathbf{c}) \geq \sigma + \tau/2$. Moreover, to succeed by perturbing close to any training point in $\mathbb{B}_j^{y'}$, the adversary must perturb to a point at distance at most $\tau + \tau/2 = 3\tau/2$ from $\mathbf{c}$ (by triangle inequality).

Using the same argument as in Theorem 2, the adversary succeeds in causing misclassification by perturbing $\mathbf{x}$ close to a point in $\mathbb{B}_j^{y'}$ with probability at most

$$\left(\frac{c\tau}{(\sigma + \tau/2)\sqrt{1 - \frac{n_3}{n_2}}}\right)^{n_2 - n_3} + c_0^{n_2 - n_3}$$

over the randomness of the adversarial subspace, for absolute constants $c > 0$ and $0 < c_0 < 1$. By a union bound, the adversary's success probability is at most N times the above quantity, conditioned on $\mathbf{x} \in \cup_{\mathbb{B}_i \in \mathcal{B}^y} \mathbb{B}_i$. Finally by assumption $\Pr_{\mathbf{x}, y \sim \mathcal{D}}[\mathbf{x} \notin \cup_{\mathbb{B}_i \in \mathcal{B}^y} \mathbb{B}_i] \leq \gamma$, and using the law of total probability we get the desired upper bound. ■

Appendix C. New Lemmas and Results from Prior Work needed to prove Theorem 17

We begin with an observation, which allows us to focus on small τ. In particular we note that the nearest-neighbor distance for most points is $O(m^{-1/n_2})$, and therefore searching for threshold in the range $[0, \tau_{\max}]$ with $\tau_{\max} = O(m^{-1/n_2})$ is sufficient for almost no abstention. This can provide a useful guide in setting $\tau_{\max}$ in Theorem 17. To simplify our results, we will treat n_2, n_3 as constants in the following.

Lemma 25 *Let Φ be a distribution defined on a compact convex subset C of $\mathbb{R}^n$ whose density function ϕ is continuous and strictly positive on C (that is $\phi(\mathbf{x}) > 0$ for $\mathbf{x} \in C$), and has bounded partial derivatives throughout C. If m samples $B = \{\beta_1, \ldots, \beta_m\}$ are drawn from Φ, for any β_i the probability that the distance d_i to its nearest neighbor in B is not $O(m^{-1/n})$ is $o(1)$.*

Proof We use the asymptotic moments of nearest neighbor distance distribution due to Evans et al. (2002) together with a concentration inequality to complete the proof. Indeed, the asymptotic mean nearest neighbor distance is shown to be $O(m^{-1/n})$, and the variance is $O(m^{-2/n})$. By Chebyshev's inequality, the probability that d_i is outside $\omega(1)$ standard deviations is $o(1)$. ■

We will need the following lemma about Lipschitzness of $\mathcal{E}_{\text{adv}}(\tau)$. The argument can also be adapted to bounded density adversary (Corollary 27), and to show a bound on the breakpoints in $\hat{\mathcal{E}}_{\text{adv}}(\tau)$ (Corollary 28).

Lemma 26 *If $\tau \leq \tau_{\max} = o(r)$, $\mathcal{E}_{\text{adv}}(\tau)$ is $O\left(m\tau_{\max}^{n_2 - n_3 - 1}/r^{n_2 - n_3}\right)$-Lipschitz.*

Proof Consider the probability that the adversary is able to succeed in misclassifying a test point x as a fixed training point x' (of different label) only when the threshold increases from τ to $\tau + d\tau$. Scale all distances by a factor of $\frac{1}{\mathsf{dist}(x,x')} =: \frac{1}{r'}$. WLOG, let x be the origin and the adversarial subspace S be given by $x_{n_3+1} = x_{n_3+2} = \cdots = x_{n_2} = 0$, and x' is the uniformly random unit vector $(z_1, \ldots, z_{n_2})$. The adversary can win only if the distance Δ of x' from S is at most $\frac{\tau}{r'}$. Therefore a threshold change of τ to $\tau + d\tau$ corresponds to $\Delta \in \left(\frac{\tau}{r'}, \frac{\tau+d\tau}{r'}\right)$. We observe from the proof of Lemma 21 that

$$\begin{aligned}\Pr\left[\Delta \in \left(\frac{\tau}{r'}, \frac{\tau+d\tau}{r'}\right)\right] &= C(n_2,n_3)\cdot\int_{\tau/r'}^{(\tau+d\tau)/r'} \rho^{n_2-n_3-1}\left(\sqrt{1-\rho^2}\right)^{n_3-2} d\rho \\ &\leq C(n_2,n_3)\cdot\frac{\tau^{n_2-n_3-1}d\tau}{r'^{n_2-n_3}},\end{aligned}$$

where $C(n_2, n_3) = 2A(n_3-1)A(n_2-n_3-1)$ is a constant for fixed dimensions n_2, n_3. This holds for any test point $\mathbf{x} \in \mathcal{T}$, and in particular, in average over the test points. Using a union bound over training points we conclude,

$$\mathcal{E}_{\text{adv}}(\tau + d\tau) - \mathcal{E}_{\text{adv}}(\tau) \leq mC(n_2,n_3)\frac{\tau^{n_2-n_3-1}d\tau}{r'^{n_2-n_3}}.$$

The slope bound increases with τ, substituting $\tau \leq \tau_{\max}$ and $r' \geq r$ gives the desired bound on Lipschitzness. ■

Corollary 27 *For a $\tilde{\kappa}$-bounded adversary distribution $\mathcal{S}$ in Lemma 26, we have that $\mathcal{E}_{\text{adv}}(\tau)$ is $O\left(\tilde{\kappa}m\tau_{\max}^{n_2-n_3-1}/r^{n_2-n_3}\right)$-Lipshcitz.*

Proof The proof follows using the same arguments in the proof of Theorem 11 applied to Lemma 26 (instead of our upper bounds on the robust error). ■

Corollary 28 *For S drawn from a $\tilde{\kappa}$-bounded adversary distribution $\mathcal{S}$, the expected number of discontinuities of $\mathcal{E}_{\text{adv}}(\tau, S)$ in any τ-interval of length w is at most $O\left(\tilde{\kappa}bmw\tau_{\max}^{n_2-n_3-1}/r^{n_2-n_3}\right)$.*

Proof Consider the interval $[\tau, \tau + w]$. We are interested in bounding the probability that for a given test point $\mathbf{x}$, the smallest threshold τ' for which the adversary succeeds when perturbing along S (over the draw $S \sim \mathcal{S}$) lies in the interval $[\tau, \tau + w]$.

For a fixed training point $\mathbf{x}_i$, the probability of adversarial success on any $\mathbf{x} \in \mathcal{T}$ by perturbing to a point at distance $\tau' \in [\tau, \tau + w]$ from $\mathbf{x}_i$ is bounded by $O\left(\tilde{\kappa}w\tau_{\max}^{n_2-n_3-1}/r^{n_2-n_3}\right)$ as argued above (Lemma 26). Taking a union bound over training points $\mathbf{x}_i$ implies the adversary succeeds with probability at most $O\left(\tilde{\kappa}mw\tau_{\max}^{n_2-n_3-1}/r^{n_2-n_3}\right)$ by perturbing to within $[\tau, \tau + w]$ of some training point. Thus, for b test points the expected number of breakpoints is at most $O\left(\tilde{\kappa}bmw\tau_{\max}^{n_2-n_3-1}/r^{n_2-n_3}\right)$. ■

The following lemma gives a bound on the expected number of breakpoints in $\mathcal{D}_{\text{nat}}(\tau)$, a piecewise constant function in τ, in a small interval of width w.

Algorithm 4 Robust classifier in the feature space with point-specific threshold $\tau_i^{\mathcal{A}}$ of “don't know”

```
Input: A test example $F(\mathbf{x})$ (potentially an adversarial example), a set $\mathcal{A}$ of training examples $F(\mathbf{x}_i^{\mathcal{A}})$ and their labels $y_i^{\mathcal{A}}$, $i \in [m_{\mathcal{A}}]$, a set $\mathcal{B}$ of training examples $F(\mathbf{x}_i^{\mathcal{B}})$ and their labels $y_i^{\mathcal{B}}$, $i \in [m_{\mathcal{B}}]$.
Output: A predicted label of $F(\mathbf{x})$, or “don't know”.
$\tau_i^{\mathcal{A}} \leftarrow \min_{j:\, y_i^{\mathcal{A}} \neq y_j^{\mathcal{B}}} \mathsf{dist}(F(\mathbf{x}_i^{\mathcal{A}}), F(\mathbf{x}_j^{\mathcal{B}}))$ for all $i \in [m_{\mathcal{A}}]$.
$i_{\min} \leftarrow \mathrm{argmin}_{i\in[m]} \mathsf{dist}(F(\mathbf{x}), F(\mathbf{x}_i^{\mathcal{A}}))$.
if $\mathsf{dist}(F(\mathbf{x}), F(\mathbf{x}_{i_{\min}}^{\mathcal{A}})) < \tau_{i_{\min}}^{\mathcal{A}}$ then
    return $y_{i_{\min}}^{\mathcal{A}}$;
else
    return “don't know”.
```

Lemma 29 *Suppose that the data distribution satisfies the assumptions in Lemma 25, and further is κ-bounded. The expected number of discontinuities in $\mathcal{D}_{\mathrm{nat}}(\tau)$ in any interval of width w for $\tau \leq \tau_{\max}$ is $O(\kappa b m w \tau_{\max}^{n_2-1})$.*

Proof Note that the discontinuities of $\mathcal{D}_{\mathrm{nat}}(\tau)$ in an interval $(\tau, \tau + w)$ corresponds to points $(\mathbf{x}, \mathbf{y}) \in T$ such that nearest neighbor distance of $\mathbf{x}$ is in that interval.

$$\begin{aligned}
E[\text{number of discontinuities in } (\tau,\tau+w)] &= b\Pr[\text{nearest neighbor of a test point } \in (\tau,\tau+w)] \\
&\leq b\Pr[\text{some neighbor of a test point } \in (\tau,\tau+w)] \\
&\leq \kappa b m \mathsf{vol}(\text{spherical shell of radius } \tau \text{ and width } w) \\
&= \kappa b m O(\tau_{\max}^{n_2-1} w) \\
&= O(\kappa b m w \tau_{\max}^{n_2-1}).
\end{aligned}$$

■

For the full proof of Theorem 17, we will need a low-regret bound for dispersed functions due to Balcan et al. (2018b). If the sequence of functions is dispersed (Definition 16), we can bound the regret of a simple exponential forecaster algorithm (Algorithm 2) by the following theorem.

Theorem 30 (Balcan et al. (2018b)) *Let $u_1, \ldots, u_T : C \to [0,1]$ be any sequence of piecewise L-Lipschitz functions that are (w,k)-dispersed. Suppose $C \subset \mathbb{R}^d$ is contained in a ball of radius R and $B(\rho^*, w) \subset C$, where $\rho^* = \mathrm{argmax}_{\rho\in C} \sum_{i=1}^{T} u_i(\rho)$. The exponentially weighted forecaster with $\lambda = \sqrt{d\ln(R/w)/T}$ has expected regret bounded by $O\left(\sqrt{Td\log(R/w)} + k + TLw\right)$.*

Appendix D. Estimating Point-Specific Threshold of “Don't Know”

Algorithm 4 gives an alternative to our algorithm where instead of using a fixed threshold for each point, we use a variable point-specific threshold learned from the data. For this algorithm, we have the following guarantee.

Theorem 31 *Suppose that the sets $\mathcal{A}$ and $\mathcal{B}$ are two independent samples from $F(\mathcal{X})$ of size $m_{\mathcal{A}}$ and $m_{\mathcal{B}}$, respectively. Let $m_{\mathcal{B}} = \frac{m_{\mathcal{A}}}{\epsilon\delta}$. Then with probability at least $1-\delta$ over the draw of $\mathcal{A}$, for a* *new sample $F(\mathbf{x})$, the probability that "there exists $F(\mathbf{x}^{\mathcal{A}}) \in \mathcal{A}$ such that $F(\mathbf{x})$ is closer to $F(\mathbf{x}^{\mathcal{A}})$ than any point in $\mathcal{B}$ of different labels than $F(\mathbf{x}^{\mathcal{A}})$, and $F(\mathbf{x})$ has a different label than $F(\mathbf{x}^{\mathcal{A}})$" is at most ϵ, where the probability is taken over the draw of $F(\mathbf{x})$ and the draw of $\mathcal{B}$.*

Proof Fixing the draw of set $\mathcal{A}$, we can think of picking a random set $\mathcal{S}$ of size $m_{\mathcal{B}} + 1$ and randomly choosing one of the points in it to be $F(\mathbf{x})$ and the rest to be $\mathcal{B}$. Let $F(\mathbf{x}^{\mathcal{A}})$ be an arbitrary point in $\mathcal{A}$. Assuming $\mathcal{S}$ has at least one point in it of a different label than $F(\mathbf{x}^{\mathcal{A}})$, then there is exactly a $\frac{1}{m_{\mathcal{B}}+1}$ probability that we choose $F(\mathbf{x})$ to be the closest point in $\mathcal{S}$ to $F(\mathbf{x}^{\mathcal{A}})$ of a different label than $F(\mathbf{x}^{\mathcal{A}})$; if $\mathcal{S}$ has all points of the same label as $\mathbf{x}^{\mathcal{A}}$, then the probability is 0. Now we can apply the union bound over all $F(\mathbf{x}^{\mathcal{A}})$ in $\mathcal{A}$ to get a total probability of failure at most $\frac{m_{\mathcal{A}}}{m_{\mathcal{B}}+1} < \epsilon\delta$.

The above analysis gives an expected failure probability over the draw of set $\mathcal{A}$. Applying the Markov inequality gives a high-probability bound. ■

Appendix E. Technical Lemmas for Proof of Theorem 20

Lemma 32 *In the setting of Theorem 20, $l(\tau)$ is monotonically non-decreasing for $\tau > D$.*

Proof Note that $\mathcal{D}_{\text{nat}}(\tau) = 0$ for $\tau > D$ as long as $m > 0$, since any test point of a class must be within D of every training point of that class. Hence, it suffices to note that $\mathcal{E}_{\text{adv}}(\tau)$ is monotonically non-decreasing in τ (increasing the threshold can only increase the ability of the adversary to successfully perturb to the opposite class). ■

Lemma 33 *In the setting of Theorem 20, the abstention rate is given by*

$$\mathcal{D}_{\text{nat}}(\tau) = \frac{1}{m+1}\left[2\mathbb{I}_{\tau\le D}\left(1-\frac{\tau}{D}\right)^{m+1} + (m-1)\mathbb{I}_{\tau\le D/2}\left(1-\frac{2\tau}{D}\right)^{m+1}\right].$$

Proof Note that for $\tau \ge D$, if $m > 0$, we never abstain on any test point. So we will assume $\tau \le D$ in the following. Consider a test point $\mathbf{x} = (x, 0)$ sampled from class A (class B is symmetric, so the overall abstention rate is the same is that of points drawn from class A). Let $\text{nbd}_{\mathbf{x}}(\tau)$ denote the intersection of a ball of radius τ around x with S_A. For x to be classified as 'don't know', we must have no training point sampled from $\text{nbd}_{\mathbf{x}}(\tau)$. This happens with probability $\left(1 - \frac{|\text{nbd}_{\mathbf{x}}(\tau)|}{D}\right)^m$, where $|\text{nbd}_{\mathbf{x}}(\tau)|$ is the size of $\text{nbd}_{\mathbf{x}}(\tau)$ and is given by

$$|\text{nbd}_{\mathbf{x}}(\tau)| = \begin{cases} \min\{x+\tau, D\}, & x < \tau; \\ \min\{2\tau, D\}, & \tau \le x \le D-\tau; \\ \min\{D-x+\tau, D\}, & x > D-\tau. \end{cases}$$

Averaging over the distribution of test points $\mathbf{x}$, we get

$$\begin{aligned}\mathcal{D}_{\text{nat}}(\tau) &= \frac{1}{D}\int_0^D \left(1-\frac{|\text{nbd}_{\mathbf{x}}(\tau)|}{D}\right)^m dx \\ &= \frac{1}{m+1}\left[2\left(1-\frac{\tau}{D}\right)^{m+1} + (m-1)\mathbb{I}_{\tau\le D/2}\left(1-\frac{2\tau}{D}\right)^{m+1}\right].\end{aligned}$$

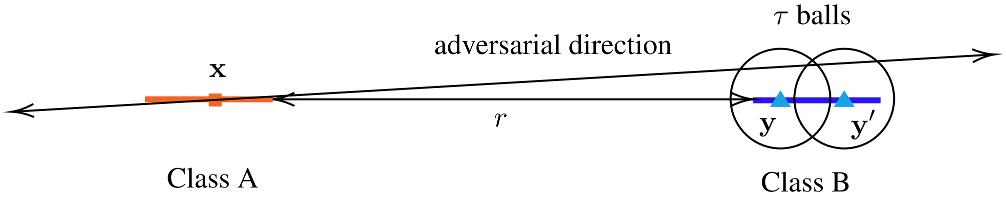

Figure 8: It suffices to consider the nearest point of the opposite class for adversarial perturbation.

■

Lemma 34 *In the setting of Theorem 20, the robust accuracy rate for $\tau \leq D$ is given by*

$$\mathcal{A}_{\text{adv}}(\tau) = 1 - \frac{\tau}{\pi r}\left(1 - \frac{m+3}{m+1} \cdot \Theta\left(\frac{D}{r}\right)\right) - \Theta\left(\left(\frac{\tau}{r}\right)^3\right).$$

Proof Consider a test point $\mathbf{x} = (x, 0)$ from S_A. Let $\mathbf{y} = (y, 0)$ denote the nearest point in S_B. In the given geometry, it is easy to see that if $\mathbf{x}$ can be perturbed into the τ neighborhood of some point $\mathbf{y}' \in S_B$ when moved along a fixed direction, then it must be possible to perturb it into the τ neighborhood of $\mathbf{y}$ (Figure 8). Therefore it suffices to consider directions where perturbation to the τ-ball around $\mathbf{y}$ is possible.

Therefore the probability of adversary's success for $\mathbf{x}$, given $\mathbf{y}$ is the nearest point of the opposite class, is

$$\text{err}_{\mathbf{x}|\mathbf{y}}(\tau) = \frac{1}{\pi}\arcsin\left(\frac{\tau}{y-x}\right) = \frac{1}{\pi}\arcsin\left(\frac{\tau}{r+d}\right),$$

where $d = y - x - r \in [0, 2D]$. Now since $\tau \leq D = o(r)$, we have

$$\text{err}_{\mathbf{x}|\mathbf{y}}(\tau) = \frac{\tau}{\pi(r+d)} + \Theta\left(\left(\frac{\tau}{r}\right)^3\right) = \frac{\tau}{\pi r}\left(1 - \Theta\left(\frac{d}{r}\right)\right) + \Theta\left(\left(\frac{\tau}{r}\right)^3\right).$$

We can now compute the average error using the probability distributions for $\mathbf{x}$ and $\mathbf{y}$, $\mathbf{x}$ is a uniform distribution over S_A, while $\mathbf{y}$ is a nearest-neighbor distribution.

$$p(x) = \frac{1}{D},\ \ p(y) = \frac{m}{D}\left(1 - \frac{y-r-D}{D}\right)^{m-1}.$$

The average value of d is

$$\bar{d} = \int_0^D \int_0^D (y' + x')\frac{m}{D}\left(1 - \frac{y'}{D}\right)^{m-1} dy' \frac{dx'}{D} = \frac{D(m+3)}{2(m+1)}.$$

Using this to compute the average of $\text{err}_{\mathbf{x}|\mathbf{y}}(\tau)$ gives the result. ■

Appendix F. Additional Experiments

Algorithm 5 Approximate computation of attacks under threat model 2.1 against Algorithm 1

```
Input: A randomly-sampled adversarial subspace $S$ of dimension $n_3$, a test example $F(\mathbf{x})$ and
  its label $y$, a set of training examples $F(\mathbf{x}_i)$ and their labels $y_i$, $i \in [m]$, a threshold parameter $\tau$.
Output: A misclassified adversarial example $F(\mathbf{x}) + \mathbf{v}$, $\mathbf{v} \in S$ if it exists; otherwise, output
  "no adversarial example found".
$F_{\text{center}}(\mathbf{x}_i) \leftarrow F(\mathbf{x}_i) - F(\mathbf{x})$ for $i \in [m]$.
  (The $F_{\text{proj}}(\mathbf{x}_i)$'s in the next step are candidate adversarial examples.)
Project $F_{\text{center}}(\mathbf{x}_i)$, $i \in [m]$ onto $S$ and obtain $F_{\text{proj}}(\mathbf{x}_i)$ for $i \in [m]$.
for $i = 1, ..., m$ do
    Run the nearest-neighbor algorithm to predict the label of $F_{\text{proj}}(\mathbf{x}_i)$ with the training set
      $\{(F_{\text{center}}(\mathbf{x}_j), y_j) : j = 1, ..., m\}$.
    if the output of the nearest-neighbor algorithm is NOT $y$ and the closest distance is smaller
      than $\tau$ then
        return $F(\mathbf{x}) + F_{\text{proj}}(\mathbf{x}_i)$.
return "no adversarial example found".
```

F.1 Approximating Robust Accuracy for Large n_3

The experiments in Section 6 consider an adversary which is difficult to compute in practice for large adversarial space, that is large n_3. In this section we present a 'greedy' adversary (Algorithm 5) which provides a good approximation to the exact adversary for small τ, which can be easily run even for large n_3: we can generate the adversarial examples of $F(\mathbf{x})$ by projecting each training example onto the affine subspace $F(\mathbf{x}) + S$ and pick the one with the closest distance to $F(\mathbf{x})$. We denote the accuracy against this algorithm as $\hat{\mathcal{A}}_{\text{adv}}$. The averaged results of multiple runs are in Table 2: we report the *natural accuracy* ($\mathcal{A}_{\text{nat}} = 1 - \mathcal{E}_{\text{nat}}$), the *adversarial accuracy*, and the *abstention rate*, where the *abstention rate* represents the fraction of algorithm's output of "don't know" among the misclassified data by the nearest-neighbor classifier.

We observe that as the dimension of adversarial subspaces n_3 increases, the adversarial accuracy $\hat{\mathcal{A}}_{\text{adv}}$ decreases while the abstention rate tends to increase, which verifies an intrinsic trade-off

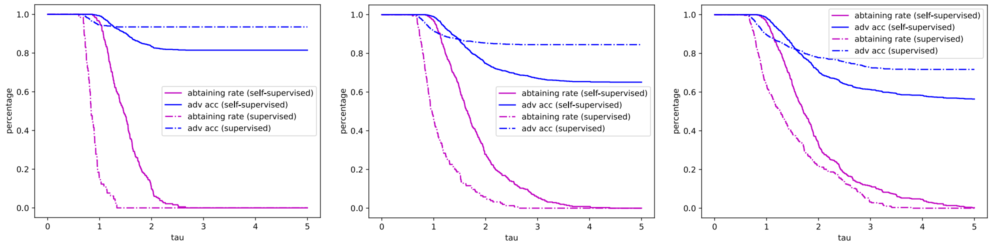

Figure 9: Sensitivity of model success rate (estimated by $\hat{\mathcal{A}}_{\text{adv}}$) and abstention rate on the parameter τ, where *abstain* represents the fraction of algorithm's output of "don't know" among the misclassified data by ours ($\tau \to \infty$, a.k.a. the nearest-neighbor classifier). **Left Figure:** $n_3 = 1$. **Middle Figure:** $n_3 = 25$. **Right Figure:** $n_3 = 50$.

Table 2: Natural accuracy $\mathcal{A}_{\text{nat}}$ and adversarial accuracy $\hat{\mathcal{A}}_{\text{adv}}$ on the CIFAR-10 data set when the 512-dimensional representations are learned by contrastive learning, where *abstain* represents the fraction of each algorithm's output of "don't know" among the misclassified data by ours ($\tau \to \infty$, a.k.a. the nearest-neighbor classifier).

	Contrastive	Linear Protocol		Ours ($\tau \to \infty$)		Ours ($\tau = 1.0$)		Ours ($\tau = 0.8$)	
		$\mathcal{A}_{\text{nat}}$	$\hat{\mathcal{A}}_{\text{adv}}$	$\mathcal{A}_{\text{nat}}$	$\hat{\mathcal{A}}_{\text{adv}}$	$\mathcal{A}_{\text{nat}}$/abstain	$\hat{\mathcal{A}}_{\text{adv}}$/abstain	$\mathcal{A}_{\text{nat}}$/abstain	$\hat{\mathcal{A}}_{\text{adv}}$/abstain
$n_3 = 1$	Self-supervised	91.1%	0.0%	84.5%	81.5%	99.3%/95.5%	99.2%/95.7%	100.0%/100.0%	100.0%/100.0%
	Supervised	94.4%	0.0%	94.3%	93.5%	95.0%/12.3%	94.5%/15.4%	97.7%/59.6%	97.7%/64.6%
$n_3 = 25$	Self-supervised	91.1%	0.0%	84.5%	65.1%	99.3%/95.5%	98.8%/96.6%	100.0%/100.0%	100.0%/100.0%
	Supervised	94.4%	0.0%	94.3%	84.5%	95.0%/12.3%	91.6%/45.8%	97.7%/59.6%	96.8%/79.4%
$n_3 = 50$	Self-supervised	91.1%	0.0%	84.5%	56.3%	99.3%/95.5%	98.3%/96.1%	100.0%/100.0%	100.0%/100.0%
	Supervised	94.4%	0.0%	94.3%	71.7%	95.0%/12.3%	89.7%/63.6%	97.7%/59.6%	95.5%/84.1%
$n_3 = 100$	Self-supervised	91.1%	0.0%	84.5%	31.1%	99.3%/95.5%	96.7%/95.2%	100.0%/100.0%	99.7%/99.6%
	Supervised	94.4%	0.0%	94.3%	35.0%	95.0%/12.3%	86.3%/78.9%	97.7%/59.6%	93.0%/89.2%
$n_3 = 200$	Self-supervised	91.1%	0.0%	84.5%	1.2%	99.3%/95.5%	91.1%/91.0%	100.0%/100.0%	98.6%/98.6%
	Supervised	94.4%	0.0%	94.3%	0.7%	95.0%/12.3%	74.7%/74.5%	97.7%/59.6%	85.8%/85.7%

between robustness and abstention rate. Recall that our algorithm abstains if and only if the closest distance in feature space between the given test example and any training example is larger than a threshold τ. As the threshold parameter τ decreases, the adversarial accuracy $\hat{\mathcal{A}}_{\text{adv}}$ increases while the algorithm abstains from predicting the class of more data.

F.1.1 SENSITIVITY OF THRESHOLD PARAMETER τ

The threshold parameter τ is an important hyperparameter in our proposed method. It captures the trade-off between the accuracy and the abstention rate. We show how the threshold parameter affects the performance of our robust classifiers by numerical experiments on the CIFAR-10 data set. We first train a embedding function F by following the setups in Section 6.2. We then fix F and run our evaluation protocol by varying τ from 0.0 to 5.0 with step size 0.001. We summarize our results in Figure 9 which plots the adversarial accuracy $\hat{\mathcal{A}}_{\text{adv}}$ and the abstention rate for three representative dimension of adversarial subspace. Compared with self-supervised contrastive learning (the solid line), supervised contrastive learning (the dashed line) enjoys higher adversarial accuracy (the blue curve) and smaller abstention rate (the red curve) for fixed τ's due to the use of extra label information. For both setups, the adversarial accuracy is not very sensitive to the choice of τ.

References

Saba Ahmadi, Hedyeh Beyhaghi, Avrim Blum, and Keziah Naggita. The strategic perceptron. In *ACM Conference on Economics and Computation (EC)*, pages 6–25, 2021.

Rima Alaifari, Giovanni S Alberti, and Tandri Gauksson. ADef: an iterative algorithm to construct adversarial deformations. In *International Conference on Learning Representations (ICLR)*, 2019.

Tal Alon, Magdalen Dobson, Ariel Procaccia, Inbal Talgam-Cohen, and Jamie Tucker-Foltz. Multiagent evaluation mechanisms. In *AAAI Conference on Artificial Intelligence*, pages 1774–1781, 2020.

Yang An and Rui Gao. Generalization bounds for (Wasserstein) robust optimization. *Advances in Neural Information Processing Systems*, 34:10382–10392, 2021.

Pranjal Awasthi, Maria Florina Balcan, and Philip M Long. The power of localization for efficiently learning linear separators with noise. In *ACM Symposium on Theory of Computing (STOC)*, pages 449–458, 2014.

Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Hongyang Zhang. Learning and 1-bit compressed sensing under asymmetric noise. In *Conference on Learning Theory (COLT)*, pages 152–192, 2016.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15535–15545, 2019.

Maria-Florina Balcan. Book chapter Data-Driven Algorithm Design. In *Beyond Worst Case Analysis of Algorithms, T. Roughgarden (Ed)*. Cambridge University Press, 2020.

Maria-Florina Balcan and Dravyansh Sharma. Data driven semi-supervised learning. *Advances in Neural Information Processing Systems*, 34:14782–14794, 2021.

Maria-Florina Balcan, Vaishnavh Nagarajan, Ellen Vitercik, and Colin White. Learning-theoretic foundations of algorithm configuration for combinatorial partitioning problems. In *Conference on Learning Theory (COLT)*, pages 213–274, 2017.

Maria-Florina Balcan, Travis Dick, Tuomas Sandholm, and Ellen Vitercik. Learning to branch. In *International Conference on Machine Learning (ICML)*, pages 344–353. PMLR, 2018a.

Maria-Florina Balcan, Travis Dick, and Ellen Vitercik. Dispersion for data-driven algorithm design, online learning, and private optimization. In *Annual Symposium on Foundations of Computer Science*, pages 603–614. IEEE, 2018b.

Maria-Florina Balcan, Travis Dick, and Colin White. Data-driven clustering via parameterized Lloyd's families. *Advances in Neural Information Processing Systems*, 31, 2018c.

Maria-Florina Balcan, Travis Dick, and Manuel Lang. Learning to link. In *International Conference on Learning Representations (ICLR)*, 2020a.

Maria-Florina Balcan, Travis Dick, and Wesley Pegden. Semi-bandit optimization in the dispersed setting. In *Uncertainty in Artificial Intelligence (UAI)*, 2020b.

Maria-Florina Balcan, Travis Dick, and Dravyansh Sharma. Learning piecewise Lipschitz functions in changing environments. In *Artificial Intelligence and Statistics (AISTATS)*, pages 3567–3577, 2020c.

Maria-Florina Balcan, Mikhail Khodak, Dravyansh Sharma, and Ameet Talwalkar. Learning-to-learn non-convex piecewise-Lipschitz functions. *Advances in Neural Information Processing Systems*, 34:15056–15069, 2021.

Maria-Florina Balcan, Avrim Blum, Steve Hanneke, and Dravyansh Sharma. Robustly-reliable learners under poisoning attacks. *Conference on Learning Theory (COLT)*, 2022a.

Maria-Florina Balcan, Mikhail Khodak, Dravyansh Sharma, and Ameet Talwalkar. Provably tuning the ElasticNet across instances. *Advances in Neural Information Processing Systems*, 2022b.

Maria-Florina Balcan, Siddharth Prasad, Tuomas Sandholm, and Ellen Vitercik. Structural analysis of branch-and-cut and the learnability of gomory mixed integer cuts. In *Advances in Neural Information Processing Systems*, 2022c.

Maria-Florina Balcan, Christopher Seiler, and Dravyansh Sharma. Faster algorithms for learning to link, align sequences, and price two-part tariffs. *arXiv preprint arXiv:2204.03569*, 2022d.

Peter Bartlett, Piotr Indyk, and Tal Wagner. Generalization bounds for data-driven numerical linear algebra. In *Conference on Learning Theory (COLT)*, pages 2013–2040. PMLR, 2022.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research (JMLR)*, 13(1):281–305, 2012.

Arjun Nitin Bhagoji, Daniel Cullina, Chawin Sitawarin, and Prateek Mittal. Enhancing robustness of machine learning systems via data transformations. In *Annual Conference on Information Sciences and Systems*, pages 1–5, 2018.

Robi Bhattacharjee and Kamalika Chaudhuri. When are non-parametric methods robust? In *International Conference on Machine Learning (ICML)*, 2020.

Avrim Blum, Chen Dan, and Saeed Seddighin. Learning complexity of simulated annealing. In *Artificial Intelligence and Statistics (AISTATS)*, pages 1540–1548. PMLR, 2021.

Wieland Brendel, Jonas Rauber, Alexey Kurakin, Nicolas Papernot, Behar Veliqi, Marcel Salathé, Sharada P Mohanty, and Matthias Bethge. Adversarial vision challenge. *arXiv preprint arXiv:1808.01976*, 2018.

Tom B Brown, Nicholas Carlini, Chiyuan Zhang, Catherine Olsson, Paul Christiano, and Ian Goodfellow. Unrestricted adversarial examples. *arXiv preprint arXiv:1809.08352*, 2018.

Nader H Bshouty, Nadav Eiron, and Eyal Kushilevitz. PAC learning with nasty noise. *Theoretical Computer Science (TCS)*, 288(2):255–275, 2002.

Nader H Bshouty, Yi Li, and Philip M Long. Using the doubling dimension to analyze the generalization of learning algorithms. *Journal of Computer and System Sciences*, 75(6):323–335, 2009.

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020a.

Yiling Chen, Yang Liu, and Chara Podimata. Learning strategy-aware linear classifiers. *Advances in Neural Information Processing Systems*, 33:15265–15276, 2020b.

CK Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning (ICML)*, pages 854–863, 2017.

Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

Zhijie Deng, Xiao Yang, Shizhen Xu, Hang Su, and Jun Zhu. LiBRe: A practical Bayesian approach to adversarial detection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 972–982, June 2021.

Travis Dick, Mu Li, Venkata Krishna Pillutla, Colin White, Maria-Florina Balcan, and Alex Smola. Data driven resource allocation for distributed learning. In *Artificial Intelligence and Statistics (AISTATS)*, pages 662–671, 2017.

Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. A rotation and a translation suffice: Fooling CNNs with simple transformations. *arXiv preprint arXiv:1712.02779*, 2017.

Dafydd Evans, Antonia J Jones, and Wolfgang M Schmidt. Asymptotic moments of near–neighbour distance distributions. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 458(2028):2839–2849, 2002.

Nicholas Frosst, Nicolas Papernot, and Geoffrey Hinton. Analyzing and improving representations with the soft nearest neighbor loss. In *International Conference on Machine Learning (ICML)*, 2019.

Aditya Ganeshan and R Venkatesh Babu. FDA: Feature disruptive attack. In *International Conference on Computer Vision (ICCV)*, pages 8069–8079, 2019.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, pages 4878–4887, 2017.

Justin Gilmer, Ryan P Adams, Ian Goodfellow, David Andersen, and George E Dahl. Motivating the rules of the game for adversarial example research. *arXiv preprint arXiv:1807.06732*, 2018.

Shafi Goldwasser, Adam Tauman Kalai, Yael Tauman Kalai, and Omar Montasser. Beyond perturbations: Learning guarantees with arbitrary adversarial test examples. In *Advances in Neural Information Processing Systems*, 2020.

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

Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017.

Rishi Gupta and Tim Roughgarden. A PAC approach to application-specific algorithm selection. *SIAM Journal on Computing*, 46(3):992–1017, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Computer Vision and Pattern Recognition (CVPR)*, pages 9729–9738, 2020.

Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Ali Eslami, and Aaron Van Den Oord. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning (ICML)*, pages 4182–4192, 2020.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

Hossein Hosseini, Yize Chen, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. Blocking transferability of adversarial examples in black-box learning systems. *arXiv preprint arXiv:1703.04318*, 2017.

Shengyuan Hu, Tao Yu, Chuan Guo, Wei-Lun Chao, and Kilian Q Weinberger. A new defense against adversarial images: Turning a weakness into a strength. In *Advances in Neural Information Processing Systems*, pages 1635–1646, 2019.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in Neural Information Processing Systems*, 32, 2019.

Michael Kearns and Ming Li. Learning in the presence of malicious errors. In *ACM Symposium on Theory of Computing (STOC)*, pages 267–280, 1988.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

Jon Kleinberg and Manish Raghavan. How do classifiers induce agents to invest effort strategically? In *ACM Conference on Economics and Computation (EC)*, page 825–844, New York, NY, USA, 2019. Association for Computing Machinery.

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

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in Neural Information Processing Systems*, 31, 2018.

Xin Li and Fuxin Li. Adversarial examples detection in deep networks with convolutional filter statistics. In *International Conference on Computer Vision (ICCV)*, pages 5764–5772, 2017.

Ninghao Liu, Hongxia Yang, and Xia Hu. Adversarial detection with model interpretation. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1803–1811, 2018.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations (ICLR)*, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

Dongyu Meng and Hao Chen. MagNet: a two-pronged defense against adversarial examples. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147, 2017.

Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. In *International Conference on Learning Representations (ICLR)*, 2017.

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

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2020.

Ali Shafahi, W Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? In *International Conference on Learning Representations (ICLR)*, 2019.

Adi Shamir, Itay Safran, Eyal Ronen, and Orr Dunkelman. A simple explanation for the existence of adversarial examples with small hamming distance. *arXiv preprint arXiv:1901.10861*, 2019.

Yonadav Shavit, Benjamin Edelman, and Brian Axelrod. Causal strategic linear regression. In *International Conference on Machine Learning (ICML)*, pages 8676–8686, 2020.

Daniel A Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004.

David Stutz, Matthias Hein, and Bernt Schiele. Confidence-calibrated adversarial training: Generalizing to unseen attacks. In *International Conference on Machine Learning (ICML)*, 2020.

Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

Thomas Tanay and Lewis Griffin. A boundary tilting perspective on the phenomenon of adversarial examples. *arXiv preprint arXiv:1608.07690*, 2016.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European Conference on Computer Vision (ECCV)*, 2019.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations (ICLR)*, 2019.

Ruth Urner and Shai Ben-David. Probabilistic Lipschitzness: a niceness assumption for deterministic labels. In *Learning Faster from Easy Data-Workshop@ NIPS*, volume 2, page 1, 2013.

Santosh S Vempala. *The Random Projection Method*, volume 65. American Mathematical Soc., 2005.

Yizhen Wang, Somesh Jha, and Kamalika Chaudhuri. Analyzing the robustness of nearest neighbors to adversarial examples. In *International Conference on Machine Learning (ICML)*, pages 5133–5142, 2018.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3733–3742, 2018.

Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2018.

Qiuling Xu, Guanhong Tao, Siyuan Cheng, Lin Tan, and Xiangyu Zhang. Towards feature space adversarial attack. *arXiv preprint arXiv:2004.12385*, 2020.

Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *Network and Distributed Systems Security Symposium*, 2017.

Yao-Yuan Yang, Cyrus Rashtchian, Yizhen Wang, and Kamalika Chaudhuri. Robustness for non-parametric classification: A generic attack and defense. In *Artificial Intelligence and Statistics (AISTATS)*, pages 941–951, 2020a.

Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Russ R Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. *Advances in Neural Information Processing Systems*, 33:8588–8601, 2020b.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2019.

Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *International Conference on Computer Vision (ICCV)*, pages 6002–6012, 2019.