On Adversarial Robustness through Clustering Training

Ankit Bhardwaj, Peter Fenteany, Sriram Ramesh

{ab9738, pf2184, sr5824}@nyu.edu

May 3, 2022

Abstract

An adversarially robust classifier should inherently force similar inputs to the same output. Naturally, this is the same paradigm guiding classical clustering algorithms, such as K-nearest neighbors. It is surprising, then, that existing literature in the field trying to link these two concepts entirely fails at strong guarantees of adversarial robustness. Namely, ClusTR, introduced by Alfarra et al. in 2020, achieves no adversarial robustness against the ensemble of attacks on RobustBench.

In this work, we investigate why this may be the case. We give a background on existing methods of both measuring adversarial robustness and learning concepts through clustering. We investigate why ClusTR fails at adversarial robustness, despite their strong claims. We then present new theoretical bounds on clustering techniques, guiding design principles on what should increase robustness. Finally, we begin an experimental study as to how we can improve adversarial robustness. In this study, we find a method that does achieve nonzero adversarial robustness against AutoAttack, though we admit that this requires further testing in order to achieve a usable (in terms of robustness and accuracy guarantees) neural network.

This work can be found at `https://github.com/SriramRamesh/fml-project`.

1 Introduction

Deep neural networks (DNNs) have proven themselves to be very useful for classification problems, from facial recognition [SLWT15] to disease detection [BFTSC19], natural language processing [LXLZ15], and content moderation [PMA17]. One desirable property of deep neural networks is *adversarial robustness* [SZS$^+$13], or resistance to misclassifying slightly tampered inputs. Due to the stakes inherent in many applications as well as the desire to subvert classification in many settings, strong robustness is needed even in the worst-case setting, where a potentially powerful adversary tries to misclassify inputs.

Perhaps unsurprisingly, not all DNNs are equally adept at adversarial robustness. In this work, we focus on one technique in classification: Clustering. Abstractly, clustering can be used to great effect in machine learning by allowing a classifier to separate the training data into more distinct classification groups. Then, testing reduces to finding the correct cluster to associate to the input point. Clustering classifiers may also be thought of as learning a distance metric [WS09], which can then be used to measure the output class of inputs. Specifically, in this work, we focus on the technique of robust clustering losses [APB$^+$20], of which perhaps the most well-known example is the Magnet Loss, introduced by Rippel et al. [RPDB16]

Alfarra et al. [APB$^+$20] showed that minimizing the Magnet Loss results in parameters that magnify the difference between the classifier's distance to a "correct" (read: of the point's true class) centroid and the classifier's distance to any "incorrect" (of any other class) centroid. In turn, this creates a natural robustness radius around each centroid, ensuring that each point within this radius will be drawn into the same classification. To this end, they propose ClusTR, which first runs a DNN with typical Cross Entropy loss training before augmenting that training with some additional training using the Magnet Loss. In their findings, they see this results in strong robustness against projected gradient descent attacks (see [APB$^+$20, Table 1]), as introduced by Madry et al. [MMS$^+$18]

Unfortunately, ClusTR does not seem to fare as well against more sophisticated attackers. RobustBench [CAS$^+$20], which uses AutoAttack [CH20b] to find adversarial examples, claims ClusTR has no robustness against AutoAttack, achieving zero percent accuracy on the adversarial points generated. Perhaps even stranger, this result is much worse than TRADES [ZYJ$^+$19], which ClusTR's experimental results augment and which manages to achieve 53.08 percent robustness accuracy on RobustBench. This result is surprising both experimentally — as AutoAttack seems to supercede claims of robustness in [APB$^+$20] — and theoretically — as AutoAttack seems to find examples that subvert the robustness radius inherent to clustering algorithms. This raises a natural question that will be the focus of this work:

Can we achieve stronger adversarial robustness using clustering-based learning algorithms?

1.1 Our Contribution

We give an overview of previous techniques, an investigation as to where they may fail, and some future directions to consider. Specifically, we note the following:

1. The work of Alfarra et al. [APB$^+$20] provides a good framework as to how clustering provides natural adversarial robustness, in particular observing that adversarial robustness and clustering both involve some notion of grouping semantically similar data points to the "same" class (See [APB$^+$20, Figure 1]). We investigate these results and posit other directions to improve clustering robustness classifiers.

2. The strategy of ClusTR put forth by Alfarra et al. ultimately fails against the more sophisticated attacks within RobustBench because of underdeveloped training. Specifically, their amendment to TRADES [ZYJ$^+$19] actually strips the protocol of its robustness, resulting in a network with false robustness against sophisticated adversaries. Note that this doesn't necessarily contradict the promises of their theoretical work as the underdeveloped training results in poor results in regards to all terms in their bound.

3. Augmenting ClusTR by using more standard adversarial training results in a more adversarially robust network against the ensemble of attacks in AutoAttack. This shows there is still hope for clustering as a technique for robustness, despite the negative results on RobustBench.

4. We investigate possible directions that may further aid in studying clustering classifiers. We note in particular that relying on logit-based schemes (including TRADES) does not and should not result in optimal clustering classifiers, which instead rely on the learning of a distance metric. We conclude by giving a call-to-action on future directions.

2 Preliminaries

We represent the training set as a set of m pairs $\{x_i, y_i\}_{i=1,\ldots,m}$, where $x_i \in \mathcal{X}$ and y_i belongs to one of C classes. Let $f_\theta : \mathcal{X} \to \mathbb{R}^C$ be a (DNN) classifier parametrized by θ that assigns $x \in \mathcal{X}$ to a point in the feature space $\mathbb{R}^C$. Then, the feature space will be divided in a way such that $f_\theta(x)$ may be associated to some class $c \in [C]$.

We denote by $||\cdot||_2$ the ℓ_2-norm, by $||\cdot||_\infty$ the ℓ_∞-norm, and by $\{\cdot\}_+$ the hinge loss. We consider f_θ to be $\mathcal{L}_f$-Lipschitz continuous. That is, for all $x_1, x_2 \in \mathbb{R}^n$, we have $||f_\theta(x_1) - f_\theta(x_2)|| \leq \mathcal{L}_f||x_1 - x_2||$.

2.1 Adversarial Robustness

Adversarial robustness [SZS+13, BCM+13] captures the idea that a neural network should classify small, imperceptible perturbations in valid inputs the same as the original input. That is, for any adversary that changes an input x with classification $f_\theta(x)$ by some δ less than maximum tolerable shift δ_{max}, we would like adversarial robustness to capture the idea that the network would output $f_\theta(x+\delta) = f_\theta(x)$. Such a point $x + \delta$ that instead satisfies $f_\theta(x+\delta) \neq f_\theta(x)$ may be called an *adversarial example*.

Adversarial robustness has proven itself to be a difficult topic to define, and in fact it is trivially impossible to achieve perfect robustness for any classifier (as for any hypothesis, there will always be inputs on the boundary between hypothesis classifications). As such, much effort has been put in to understand achievable and useful notions of adversarial robustness. Perhaps the simplest measure of robustness is the following, as in Carlini et al. [CAP+19]:

$$\mathbb{E}_{(x,y)\sim\mathcal{X}}\left[\max_{x'|\mathcal{D}(x,x')<\delta} L(f(x'), y)\right],$$

where $(x, y) \sim \mathcal{X}$ is a sample from space $\mathcal{X}$ and $\mathcal{D}$ is some distance metric.

While this quantity is easy to answer in the average-case, though, it is usually infeasible to calculate exactly in the worst case — i.e., the setting where x' is adversarially chosen. This has been subverted by works restricting the ability of adversaries, such as by restricting their computational complexity [GJMM20].

Adversarial robustness can also be notioned about by testing a given network's robustness to known attacks. While this leaves open the possibility of stronger attacks in the future, this is still a helpful benchmark for comparing robustness of different techniques (if a model is not robust against a known attack, then it stands no chance at adversarial robustness already). Perhaps the most common of these attacks is the projected gradient descent (PGD) attack [MMS+18], which adversarially attempts to maximize the loss of f_θ on a point at most δ-far from a point x_0. This is a *white-box* attack, which at each step takes the projected gradient of f_θ on the previous iteration. At each stage, the PGD attack looks like so:

$$x^{(k+1)} = P_S(x^{(k)} + \alpha \ \mathrm{sgn}(\nabla_x L(f_\theta(x), y))),$$

where the loss in our setting will be the cross entropy loss and P_S is the projection onto some subset S.

RobustBench [CAS+20] call for a stronger notion of adversarial robustness than the PGD attack by augmenting and combining it with other attacks. It employs three strategies to do so, namely:

1. **Auto-PGD**, which augments the typical PGD attack by adaptively deciding at each step whether to decrease the step size, whether to move the neighborhood in which it searches for an adversarial example, and by converging more quickly on an adversarial example by "exploring" the space around an original input before "exploiting" the area in which an adversarial example is more likely. The Auto-PGD attack is considered both for the Cross Entropy and the Digits of Logits Ratio loss [CH20b] functions.

2. **Fast Adaptive Boundary attack** [CH20a], which on an input point x attempts to first find some shift of x that is classified differently, then tries to minimize the norm of this point to find an adversarial example.

3. **Square Attack** [ACFH19], a black-box attack that probes the model at randomized nearby locations to try and find an adversarial example.

This is of course a brief overview of the techniques — a full report can be found at [CH20b]. If a model fails any one of these attacks with high probability, it is trivially not adversarially robust, and we in fact know an attack it is weak to.

2.2 Clustering

The overall goal of a clustering-based classifier is to take points in the input space and classify them into clear clusters in the feature space, such that each cluster is related to a different class. In this way, clustering-based classifiers can be thought of as learning a distance metric such that the distance between entries of the same classification are closer than the distance between entries of different classes.

In essence, a clustering-based classifier takes the feature space in $\mathbb{R}^d$ and divides it into different clusters based on which target class. After training, for each class, the centroids can be found through any standard clustering algorithm, e.g. K-nearest neighbors. Concretely, this will result in a feature space divided into $C \cdot K$ clusters, and on running the neural network $f_\theta(x)$ for test input x, we will end up assigning x to the class of the cluster closest to $f_\theta(x)$.

Prior works discuss the design of clustering loss functions. Weinberger and Saul [WS09] present a pair of loss functions that in tandem can train a classifier to learn a distance metric. The first, denoted ϵ_{pull}, punishes transformations that spread nearby points in the training sample apart. The second, denoted ϵ_{push}, punishes transformations that place far-apart points in the training sample close together. In essence, then, minimizing $\epsilon_{pull} + \epsilon_{push}$ (perhaps with some stable weighting) results in learning a function that pulls points toward their (K) nearest neighbors and pushes points away from all others. As noted by Rippel et al. [RPDB16], though, this results in some short-sightedness. For example, because the techniques of [WS09] at each component only consider the relations between individual points, it fails to take the structure of the overall data set into account. This is in particular an issue with non-convex shapes.

To this end, Rippel et al. introduced the concept of the Magnet Loss:

$$\mathcal{L}_{magnet} = \frac{1}{m}\sum_{i=1}^{m}\left\{\alpha + \frac{1}{2\sigma^2}||f_\theta(x_i) - \mu_{y_i,\cdot}||_2^2 + \log\left(\sum_{c\neq y_i}\sum_{k=1}^{K} e^{-\frac{1}{2\sigma^2}||f_\theta(x_i)-\mu_{c,k}||_2^2}\right)\right\}_+, \tag{1}$$

where $\alpha \in \mathbb{R}$ is an appropriately chosen parameter, σ^2 is the variance of the distances between each data point and its respective centroid, $\mu_{y_i,\cdot}$ is the closest correct (for class y_i) cluster centroid to $f_\theta(x_i)$, and $\mu_{c,k}$ is the k-th centroid for the class c. In this way, the second term captures how far the training data are from their correct classifications, whereas the third term captures how close the training data are from incorrect classifications. In order to classify a point using a classifier trained with this loss, one just then must compute the class that results in the highest probability of the input point belonging to that class's cluster centroid, using the formula:

$$c_{magnet}(x) = \arg\max_{c\in[C]} \frac{\sum_{\mu_c} e^{-||f_\theta(x)-\mu_c||_2^2/2\sigma^2}}{\sum_{\mu} e^{-||f_\theta(x)-\mu||_2^2/2\sigma^2}}.$$

With respect to adversarial robustness, prior works consider the toy example of $C = 2$, $K = 1$. Here, the feature space is divided into only two subspaces, one associated to class $\mathcal{C}_1$ and the other to class $\mathcal{C}_2$. Let μ_1 be the centroid of $\mathcal{C}_1$ and μ_2 the centroid of $\mathcal{C}_2$. Then, we see x is assigned to class

$$\arg\min_{i \in \{1,2\}} ||f_\theta(x) - \mu_i||.$$

We observe from previous work that using clustering to map the feature space to classes implies a natural radius of adversarial robustness. We present Proposition 1 exactly as it appeared in [APB+20].

Proposition 2.1 (Proposition 1, [APB+20])**.** *Consider the clustering-based binary classifier that classifies x as class $\mathcal{C}_1$, i.e. $||f_\theta(x) - \mu_1|| < ||f_\theta(x) - \mu_2||$, with $\mathcal{L}_f$-Lipschitz f_θ. The classifier's output for the perturbed input $(x + \delta)$ will not differ from x, i.e. $||f_\theta(x+\delta) - \mu_1|| < ||f_\theta(x+\delta) - \mu_2||$, for all perturbations δ that satisfy:*

$$||\delta||_2 < \frac{||f_\theta(x) - \mu_2||_2^2 - ||f_\theta(x) - \mu_1||_2^2}{2\mathcal{L}_f ||\mu_2 - \mu_1||_2}. \tag{2}$$

In their work, Alfarra et al. attempted to increase this radius of robustness by finding and tuning a classifier that maximizes the numerator, that is, the difference between the distance from $f_\theta(x)$ to the incorrect cluster and the distance from $f_\theta(x)$ to the correct cluster. It is clear how a notion like optimizing based on the Magnet Loss may help. To this end, they proposed a system dubbed ClusTR, which consists of a ”warm start” period (using, in their example, a modified version of the TRADES classifier [ZYJ+19]) to reach decent performance quickly, and then removing the last linear layer and replacing it with a clustering classifier (in their case, the Magnet Loss).

While proving a robustness radius in an ℓ_p-norm may not be sufficient for achieving all the desired notions for adversarial robustness [CAP+19], it does suffice for all the attacks given in AutoAttack, which each only try to find adversarial examples within a given ϵ-radius with respect to some ℓ_p-norm — in this work, $p = \infty$. It is surprising, then, that we see that ClusTR achieves no robustness against adversarial examples generated by AutoAttack. We investigate further in Section 4.

3 Estimating Clustering Robustness

Unfortunately, as noted by Carlini et al. [CAP+19], protecting against ℓ_p-norm shifts is not sufficient for achieving all notions of adversarial robustness we desire. However, studying adversarial robustness in this way can still provide insight as to the properties that aid or detract from robustness in the general setting. In addition, the attacks detailed in RobustBench are all based on ℓ_∞-constrained shifts, so studying this setting is sufficient for our purposes.

In this section, we present new bounds on the adversarial robustness guarantees of clustering classifiers, extending the results of Proposition 2.1. We also present new generalizations of robustness guarantees as a result.

Denote by δ_{max} the upper bound for $||\delta||$ given in Equation 2. As stated before, Alfarra et al. [APB+20] focused on maximizing the robustness radius by maximizing the numerator. Here, we consider the natural other technique — minimizing the denominator $||\mu_2 - \mu_1||$.

In order to minimize this value, one way forward may be to increase the number of cluster centroids used in for each classification. Especially when assuming the data obeys some notion of well-spreadedness, this seems promising. ClusTR only considered each class as having two centroids.

However, attempting to do so has unintended consequences theoretically. For a given point x, consider the trio of points as in Proposition 2.1: $f_\theta(x)$, the mapping of x onto the feature space; μ_c, the centroid of class c that is closest to $f_\theta(x)$; and $\mu_{c'}$, the centroid of class $c' \neq c$ that is second-closest to $f_\theta(x)$. Note that, even in the case with $C > 2, K > 1$, there will be some two clusters of different classes which are the closest to $f_\theta(x)$, so this setting is sufficient for understanding the multi-class, multiple cluster setting. Then, we witness the following bound:

$$\frac{||f_\theta(x) - \mu_{c'}|| - ||f_\theta(x) - \mu_c||}{2\mathcal{L}_f} \leq \delta_{max} \leq \frac{||f_\theta(x) - \mu_{c'}|| + ||f_\theta(x) - \mu_c||}{2\mathcal{L}_f}. \tag{3}$$

Proof. Consider the trio of points in $\mathbb{R}^d$ $f_\theta(x), \mu_c, \mu_{c'}$. We will apply the triangle inequality with respect to these three points to the right-hand side of Equation 2, finding

$$\begin{aligned}
\delta_{max} = \frac{||f_\theta(x) - \mu_{c'}||^2 - ||f_\theta(x) - \mu_c||^2}{2\mathcal{L}_f||\mu_{c'} - \mu_c||} &= \frac{(||f_\theta(x) - \mu_{c'}|| - ||f_\theta(x) - \mu_c||)(||f_\theta(x) - \mu_{c'}|| + ||f_\theta(x) - \mu_c||)}{2\mathcal{L}_f||\mu_{c'} - \mu_c||} \\
&\leq \frac{(||f_\theta(x) - \mu_{c'}|| - ||f_\theta(x) - \mu_c||)(||f_\theta(x) - \mu_{c'}|| + ||f_\theta(x) - \mu_c||)}{2\mathcal{L}_f(||f_\theta(x) - \mu_{c'}|| - ||f_\theta(x) - \mu_c||)} \\
&= \frac{||f_\theta(x) - \mu_{c'}|| + ||f_\theta(x) - \mu_c||}{2\mathcal{L}_f}
\end{aligned}$$

as an upper bound. Similarly, for the lower bound, we have

$$\begin{aligned}
\delta_{max} = \frac{||f_\theta(x) - \mu_{c'}||^2 - ||f_\theta(x) - \mu_c||^2}{2\mathcal{L}_f||\mu_{c'} - \mu_c||} &= \frac{(||f_\theta(x) - \mu_{c'}|| - ||f_\theta(x) - \mu_c||)(||f_\theta(x) - \mu_{c'}|| + ||f_\theta(x) - \mu_c||)}{2\mathcal{L}_f||\mu_{c'} - \mu_c||} \\
&\geq \frac{(||f_\theta(x) - \mu_{c'}|| - ||f_\theta(x) - \mu_c||)(||f_\theta(x) - \mu_{c'}|| + ||f_\theta(x) - \mu_c||)}{2\mathcal{L}_f(||f_\theta(x) - \mu_{c'}|| + ||f_\theta(x) - \mu_c||)} \\
&= \frac{||f_\theta(x) - \mu_{c'}|| - ||f_\theta(x) - \mu_c||}{2\mathcal{L}_f}
\end{aligned}$$

as a lower bound. Note that these results hold hold if and only if $||f_\theta(x) - \mu_{c'}|| \neq \pm||f_\theta(x) - \mu_c||$. However, for any nontrivial data set, these are true with overwhelming probability, as either case would require two or more of $f_\theta(x), \mu_c, \mu_{c'}$ to coincide. □

We see this gives us a tight maximum robustness range, where δ_{max} is in a $||f_\theta(x) - \mu_c||/2\mathcal{L}_f$-radius around $||f_\theta(x) - \mu_{c'}||/2\mathcal{L}_f$. We would like and expect this to be fairly tight, as this would give us our best-case robustness radius. To achieve this, our goal should be to minimize the distance $||f_\theta(x) - \mu_c||$. From this perspective, one approach is clear — increasing the number of clusters should in theory decrease the distance from $f_\theta(x)$ to its nearest centroid, given some assumptions of well-spreadedness. However, as our bounds show, this may be a double edged sword. As the distance between $f_\theta(x)$ and its nearest centroid decreases, we may also see that the distance between $f_\theta(x)$ and its second-nearest centroid will also decrease, thereby decreasing the bounds we showed. Therefore, it is unclear what an optimal number of centroids per class is, and it is highly likely this amount is dependent on the classifier used and the underlying data properties. In the following sections, we detail our efforts in researching this further for the case of the CIFAR-10 dataset.

4 Experimental Results

We have experimented extensively with the ClusTR [APB+20] implementation and have tested their models in several settings. We have also modified their implementation to experiment with other models not included in their work. Our codebase can be found at `https://github.com/SriramRamesh/fml-project`.

We use modified ResNet-18 backbone model with warm start used by [APB$^+$20] for different experiments. We have used the random seed 99 for all of our experiments. We now present our experiments and inferences evaluating the robustness of different models.

4.1 QTRADES

From the implementation of Alfarra et al. [APB$^+$20], we begin the ClusTR + QTRADES model, defined using the following loss function:

$$\mathcal{L}_{Total} = \mathcal{L}^{Magnet}_{Clustering} + \lambda * \mathcal{L}_{CE}(p(f_\theta(x)), p(f_\theta(x_{adv})))$$

Here, $f_\theta(x)$ represents the embeddings obtained by the model with x as input, and $p(f_\theta(x))$ represents the probabilities for different classes based on the embeddings (equivalent to logits).

$$x_{adv} = \Pi_S(x' + \eta \ \text{sgn}(\nabla_{x'} \mathcal{L}_{CE}(p(f_\theta(x')), p(f_\theta(x)))))$$

Here, S represents the sample and η is the step-size, and x' refers to a uniformly randomly perturbed image generated with x as input. According to RobustBench (`https://robustbench.github.io/`), this model should have (approximately) no robustness against AutoAttack. To thoroughly evaluate this method's adversarial training, we consider the following variants of QTRADES for our experiments:

$$\text{ClusTR+QTRADES\_MSE} = \begin{cases} L^{Magnet}_{Clustering} + \lambda * \mathcal{L}_{MSE}(p(f_\theta(x)), p(f_\theta(x_{adv}))) \\ \\ x_{adv} = \Pi_S(x' + \eta \ \text{sgn}(\nabla_{x'} \mathcal{L}_{MSE}(p(f_\theta(x')), p(f_\theta(x))))) \end{cases}$$

$$\text{ClusTR+CE+QTRADES} = \begin{cases} L^{Magnet}_{Clustering} + \lambda * \mathcal{L}_{CE}(p(f_\theta(x)), p(f_\theta(x_{adv}))) + \lambda' \mathcal{L}_{CE}(p(f_\theta(x)), y) \\ \\ x_{adv} = \Pi_S(x' + \eta \ \text{sgn}(\nabla_{x'} \mathcal{L}_{CE}(p(f_\theta(x')), p(f_\theta(x))))) \end{cases}$$

$$\text{ClusTR+CE+QTRADES\_MSE} = \begin{cases} L^{Magnet}_{Clustering} + \lambda * \mathcal{L}_{MSE}(p(f_\theta(x)), p(f_\theta(x_{adv}))) + \lambda' \mathcal{L}_{CE}(p(f_\theta(x)), y) \\ \\ x_{adv} = \Pi_S(x' + \eta \ \text{sgn}(\nabla_{x'} \mathcal{L}_{MSE}(p(f_\theta(x')), p(f_\theta(x))))) \end{cases}$$

We obtain the following results for the above experiments after using $\lambda = 8$ and $\lambda' = 2$. As in Alfarra et al., we use $K = 2$ clusters per class, $L = 10$ nearest neighbors for normalization, and $M = 12$ subsampled clusters.

Experiment	Clean Accuracy	Robust Accuracy
ClusTR+QTRADES	90.88	0.11
ClusTR+QTRADES_MSE	90.48	0.41
ClusTR+CE+QTRADES	91.34	0.27
ClusTR+CE+QTRADES_MSE	90.57	0.21

Table 1: Performance for different variations of QTRADES.

From these results, it is clear that QTRADES does *not* provide robustness against sophisticated attacks like AutoAttack and is not sufficient for guaranteeing robustness of our clustering classifier.

4.2 Extended QTRADES

Because QTRADES does not suffice for robustness against AutoAttack, we attempt to improve upon the method by sacrificing the efficiency optimizations that the method incorporates. Our first attempt extends the QTRADES method to take multiple steps along the gradients rather than taking a single step. The experiment settings we use are as such:

$$\text{ClusTR+E-QTRADES} = \begin{cases} L^{Magnet}_{Clustering} + \lambda * \mathcal{L}_{CE}(p(f_\theta(x)), p(f_\theta(x_{adv}))) \\ \\ x_{adv} = \Pi_S I^{10}_{i=1}(x' + \eta \ \text{sgn}(\nabla_{x'} \mathcal{L}_{CE}(p(f_\theta(x')), p(f_\theta(x))))) \end{cases}$$

$$\text{ClusTR+E-QTRADES\_MSE} = \begin{cases} L^{Magnet}_{Clustering} + \lambda * \mathcal{L}_{MSE}(p(f_\theta(x)), p(f_\theta(x_{adv}))) \\ \\ x_{adv} = \Pi_S I^{10}_{i=1}(x' + \eta \ \text{sgn}(\nabla_{x'} \mathcal{L}_{MSE}(p(f_\theta(x')), p(f_\theta(x))))) \end{cases}$$

Here, $I^{10}_{i=1}$ refers to 10 iterations where x' is updated in each iteration. We use $\lambda = 6$, and $K = 1$ cluster per class for these experiments, and L and M are both set to 10.

Experiment	Clean Accuracy	Robust Accuracy
ClusTR+E-QTRADES	89.91	0.78
ClusTR+E-QTRADES_MSE	90.95	0.17

Table 2: Performance for different variations of E-QTRADES.

This is a surprising result. While we have already discussed the upper-limit of clustering approach, we would reasonably expect better results than what we obtain here. We believe this may be due to the static formulae for converting embeddings to logits in our implementation and so, we propose a learnable conversion from embeddings to logits as a direction for further investigation.

4.3 TRADES

One natural extension after looking at the poor performance of QTRADES is to look at the "tested and proven" method of adversarial training and origin of QTRADES — TRADES [ZYJ+19]. We hoped that with TRADES, we would be able to achieve better robust accuracy. We used two implementations of TRADES after looking at the implementations used by others, one of which was used in the original paper.

$$\text{ClusTR+TRADES\_1} = \begin{cases} L^{Magnet}_{Clustering} + \lambda * \mathcal{L}_{CE}(p(f_\theta(x)), p(f_\theta(x_{adv}))) \\ \\ x_{adv} = \Pi_S I^{10}_{i=1}(x' + \eta \ \text{sgn}(\nabla_{x'} \mathcal{L}_{KLDiv}(p(f_\theta(x')), p(f_\theta(x))))) \end{cases}$$

$$\text{ClusTR+TRADES\_2} = \begin{cases} L^{Magnet}_{Clustering} + \lambda * \mathcal{L}_{KLDiv}(p(f_\theta(x)), p(f_\theta(x_{adv}))) \\ \\ x_{adv} = \Pi_S I^{10}_{i=1}(x' + \eta \ \text{sgn}(\nabla_{x'} \mathcal{L}_{KLDiv}(p(f_\theta(x')), p(f_\theta(x))))) \end{cases}$$

Here, we witness another surprising result. We found that we were not able to train our model properly for the above given loss functions. We could achieve the highest training accuracy of 42.18%, which resulted in clean accuracy of 38.35% and robust accuracy of 0.97%. It should be noted that these results only tell us that TRADES loss doesn't play well with magnet clustering loss, and training an adversarially robust model using this method would require further adaptation of the method for clustering based classifiers.

4.4 Ablations and Baseline

As part of our ablation study, we vary the values for our clustering parameters like number of clusters per class K, number of nearest neighbors for score normalization L and number of subsampled clusters at each step M. Varying these parameters gave us similar values of clean accuracy with all values lying in the interval of $90 \pm 2\%$. The effects on robust accuracy were even more marginal, with robust accuracy being less than 1% in all cases.

For baseline, we did direct adversarial training using magnet loss. The experiment setting can be written as:

$$\text{ClusTR+ADV} = \begin{cases} L^{Magnet}_{Clustering}(x, y) + \lambda * L^{Magnet}_{Clustering}(p(f_\theta(x_{adv})), y) \\ \\ x_{adv} = \Pi_S(x' + \eta \text{ sgn}(\nabla_{x'} L^{Magnet}_{Clustering}(p(f_\theta(x')), p(f_\theta(x))))) \end{cases}$$

We found here that magnet clustering adversarial training performs the best on robustness benchmark of AutoAttack. We used $\lambda = 8$ for the following results.

Experiment	**Clean Accuracy**	**Robust Accuracy**
ClusTR+ADV	77.63	**6.23**

Table 3: Performance for baseline adversarial training.

It should be noted that we could achieve marginally better numbers by choosing higher values for λ and training for higher number of epochs. The baseline performance gives us hope that clustering based classifiers can be made adversarially robust, although we might need to solve the problem from scratch in context of clustering based classifiers, as other logit-based methods are not readily applicable.

5 Conclusion

While optimizing clustering still has a long way to go in reaching the state of the art in adversarial robustness, its inherent link to avoiding adversarial examples makes it very appealing from a theoretical standpoint. Prior work showed that this does give positive results for certain restricted adversaries; we showed that we can achieve some robustness even for more sophisticated ones. Clustering based classifiers inherently carry meaningful distance metrics that can be used to understand adversarial robustness better. Hence, the problem of optimizing adversarially robust clustering shouldn't only be studied for the sake of improving state of art in adversarial robustness, but also for getting deeper insights into the problem of robustness. With this train of thought, we mention the following open problems for further study.

5.1 Further Directions

This work represents an early investigation into the domain of clustering as a robust classification technique, and we are confident that further improvements can be made. While we began the search into an optimal number of clusters, our results are by no means complete, theoretically or experimentally. Other clustering techniques, such as fuzzy c-means [BEF84], may also have better results. While we used the Magnet Loss, optimizing for other losses may also result in better-clustered data. These results may also depend on the type of classification problem.

Because the robust clustering functions as a fortification for existing techniques, more work is needed to understand if there are certain robustness techniques that "play nice" with clustering. We propose looking into better methods of transforming embeddings into logits as a possible direction for this. While TRADES was used because it is a powerful existing classifier, other algorithms may give better or more efficient experimental results. It is also possible a better loss may exist tailor-made to achieve clustering robustness. We also propose looking into L2 robustness of clustering based classifiers as a worthwhile direction because clustering methods inherently have some L2 robustness guarantees. Replacing warm start by adversarial pretraining, might also give us surprising results.

References

[ACFH19] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square Attack: a query-efficient black-box adversarial attack via random search. *arXiv e-prints*, page arXiv:1912.00049, November 2019.

[APB+20] Motasem Alfarra, Juan C. Pérez, Adel Bibi, Ali K. Thabet, Pablo Arbeláez, and Bernard Ghanem. Clustr: Clustering training for robustness. *CoRR*, abs/2006.07682, 2020.

[BCM+13] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 387–402, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[BEF84] James C Bezdek, Robert Ehrlich, and William Full. Fcm: The fuzzy c-means clustering algorithm. *Computers & geosciences*, 10(2-3):191–203, 1984.

[BFTSC19] Justine Boulent, Samuel Foucher, Jérôme Théau, and Pierre-Luc St-Charles. Convolutional neural networks for the automatic identification of plant diseases. *Frontiers in Plant Science*, 10, 2019.

[CAP+19] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness, 2019.

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

[CH20a] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *ICML*, 2020.

[CH20b] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[GJMM20] Sanjam Garg, Somesh Jha, Saeed Mahloujifar, and Mohammad Mahmoody. Adversarially robust learning could leverage computational hardness. In *ALT*, 2020.

[LXLZ15] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2267–2273. AAAI Press, 2015.

[MMS$^+$18] 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*, 2018.

[PMA17] John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. Deeper attention to abusive user content moderation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1125–1135, Copenhagen, Denmark, sep 2017. Association for Computational Linguistics.

[RPDB16] Oren Rippel, Manohar Paluri, Piotr Dollár, and Lubomir D. Bourdev. Metric learning with adaptive density discrimination. *CoRR*, abs/1511.05939, 2016.

[SLWT15] Yi Sun, Ding Liang, Xiaogang Wang, and Xiaoou Tang. Deepid3: Face recognition with very deep neural networks. *CoRR*, abs/1502.00873, 2015.

[SZS$^+$13] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2013.

[WS09] Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(9):207–244, 2009.

[ZYJ$^+$19] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482. PMLR, 09–15 Jun 2019.