# Rethinking Clustering for Robustness

Motasem Alfarra*[1]
motasem.alfarra@kaust.edu.sa
Juan C. Pérez*[1, 2]
juan.perezsantamaria@kaust.edu.sa
Adel Bibi[3]
adel.bibi@eng.ox.ac.uk
Ali Thabet[4]
thabetak@fb.com
Pablo Arbeláez[2]
pa.arbelaez@uniandes.edu.co
Bernard Ghanem[1]
bernard.ghanem@kaust.edu.sa

[1] King Abdullah University of Science and Technology (KAUST) Saudi Arabia
[2] Universidad de los Andes Colombia
[3] University of Oxford United Kingdom
[4] Facebook Reality Labs (FRL) Zurich, Switzerland

**Abstract**

This paper studies how encouraging semantically-aligned features during deep neural network training can increase network robustness. Recent works observed that Adversarial Training leads to robust models, whose learnt features appear to correlate with human perception. Inspired by this connection from robustness to semantics, we study the complementary connection: from semantics to robustness. To do so, we provide a robustness certificate for distance-based classification models (clustering-based classifiers). Moreover, we show that this certificate is tight, and we leverage it to propose *ClusTR* (Clustering Training for Robustness), a clustering-based and adversary-free training framework to learn robust models. Interestingly, *ClusTR* outperforms adversarially-trained networks by up to 4% under strong PGD attacks. Our code for reproducing our results can be found at https://github.com/rethinking-clustering-for-robustness.

## 1 Introduction

Deep neural networks (DNNs) have demonstrated tremendous success in various fields, from computer vision [23, 24] and reinforcement learning [30, 31] to natural language processing [8, 44] and speech recognition [16]. Despite this breakthrough in performance, robustness is becoming a rising concern in DNNs. Specifically, DNNs have been shown to be vulnerable to imperceptible input perturbations [13, 41], known as adversarial attacks, which can entirely alter the DNN's output. This vulnerability has popularized a new line of research known as network robustness. Robust DNNs should not only be accurate, but also resistant against input perturbations. Given the importance of the problem, a plethora of network robustness approaches have been proposed, including those based on regularization [7, 19, 32, 49],


* Equal Contribution.

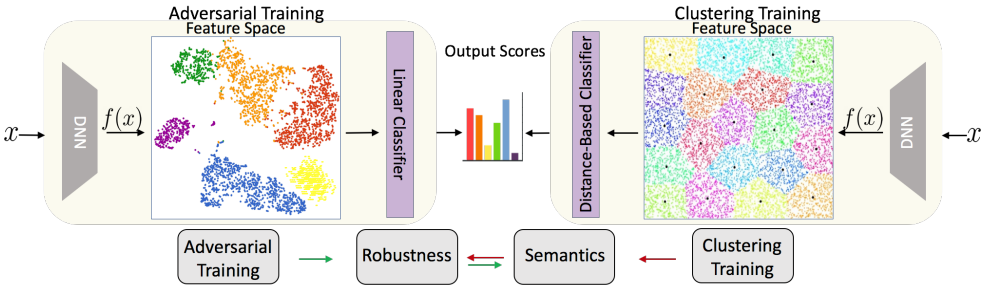


Figure 1: **Closing the loop on robustness and semantics**. Earlier work showed that adversarial training results in more semantically-aligned features, *i.e.* features of same-class instances tend to cluster together . We study the complementary path, *i.e.* the effect of learning more semantically-aligned features (via clustering) on network robustness (right figure).

distillation [35], and feature denoising [47], among many others. In this paper, we focus our attention on the popular and effective adversarial training approach [26].

Adversarial training explicitly trains DNNs on adversarial attacks generated on-the-fly through projected gradient descent (PGD). This technique has proven to significantly improve network robustness, and has become a standard for training robust networks. Interestingly, and as a byproduct, adversarially-trained networks seem to learn features that are more semantically aligned with human perception [9, 10], to such a degree that the learnt DNNs can be used for several image-synthesis tasks [37]. Learning more semantically-aligned features in DNNs remains an open problem. A promising direction for obtaining features with such properties is through Deep Metric Learning (DML) techniques. DML learns feature representations by preserving a notion of similarity between inputs and their feature representations [17, 34], and has achieved remarkable performance in face recognition [39], image retrieval [11], and zero-shot learning [12]. The preservation of similarity that DML seeks often involves clustering semantically-similar instances. Hence, recent clustering-based losses [17, 36] have been designed with this objective in mind, showing significant progress in learning semantic representations that are also competitive in performance with modern classification approaches.

Inspired by these developments, we theoretically show an intimate relation between semantics (through clustering approaches) and robustness, as illustrated in Figure 1. In particular, we show that, under certain continuity properties of the DNN, clustering-based classifiers enjoy a tight robustness radius against $\ell_2$-bounded input perturbations. Furthermore, we observe that this radius can be maximized by optimizing a *Clustering Loss*, *i.e.* a loss that encourages clustering of semantically-similar instances in feature space. Inspired by this observation, we show that training DNNs with such a loss results in high-performing classifiers that are also robust against PGD attacks. We enhance this clustering-based approach with standard techniques for DNN training, and dub this framework *Clustering Training for Robustness* (*ClusTR*). To validate the idea behind ClusTR, we experiment on several datasets and find that ClusTR can yield significant robustness gains. In summary, our **contributions** are three-fold: **(i)** We study the connection from semantics to PGD robustness by analyzing classifiers that employ clustering in representation space. We use this analysis to derive a tight $\ell_2$ robustness radius, under which all $\ell_2$ perturbations are unable to change the predictions. Moreover, we show that a deep metric learning approach for semantic clustering

that optimizes a *Clustering Loss* is directly related to maximizing the derived robustness radius. **(ii)** Motivated by our theoretical findings, we propose the ClusTR framework, which employs a popular *Clustering Loss* (the *Magnet Loss* [36]), to learn robust models against PGD attacks without generating adversaries during training. We validate the theory behind ClusTR through extensive experiments and find that ClusTR results in a significant boost in robustness against PGD attacks without relying on adversarial training. Specifically, we observe that classifiers learnt using ClusTR outperform (in robustness) adversarially-trained classifiers [40] by 3% and 4% under strong $^8/_{255}$ PGD attacks on the CIFAR10 [22] and SVHN [33] datasets, respectively.**(iii)** Equipping ClusTR with a quick and cheap version of adversarial training can increase robustness against $^8/_{255}$ attacks on several benchmarks by significant margin.

## 2 Related Work

**Metric Learning.** The idea of encouraging learnt features to be more semantically meaningful to the human visual system has been extensively studied in the metric learning community, where the goal is to learn a similarity measure in feature space that correlates with a similarity measure between inputs [17, 20, 45, 48, 50]. In such a setting, semantically-similar inputs (*i.e*. those belonging to the same class) are expected to be clustered together. This paradigm has shown remarkable performance in several tasks [11, 29, 39]. Closely related to our work, the approach of [27] used the *Triplet Loss* [39] to regularize learnt features and enhance network robustness. We complement the previous art with a theoretical justification on the intimate relation between robustness and the general family of metric-learning classifiers that subsumes the *Triplet Loss* as a special case. Namely, we find a connection between the *Magnet Loss* [36] and theoretical guarantees of network robustness.

**Adversarial Robustness.** The existence of adversarial perturbations has dramatically increased security concerns in DNNs. Consequently, there has been a surge of research aiming at learning adversarially-robust models [4, 7, 25]. Despite its high computational cost, adversarial training [26] remains one of the most popular, successful and reliable techniques for attaining adversarial robustness. Furthermore, adversarial training was regularized by enforcing similarity between logits of both natural and adversarial pairs [19]. This work was further developed in TRADES [49]. Moreover, regularization also studied the data-complexity perspective, demonstrating an inherent sample complexity barrier on robust learning [38], and that pre-training or learning from unlabeled data can vastly improve robustness of adversarially-trained networks [6, 15].

**Robust Features.** Recent work demonstrated that networks trained adversarially enjoy an unexpected benefit: the learnt features tend to align with salient data characteristics and human perception [43]. Moreover, the learnt features, commonly referred to as robust features [18], seem to be clustered in feature space, while being perceptually aligned [10]. Based on these findings, the power of such semantically-aligned features was harnessed to perform image synthesis tasks with a single robust classifier [37]. In this paper, we take an orthogonal direction to robustness, in which we encourage robustness by training DNNs to specifically learn more semantically-aligned features via clustering.

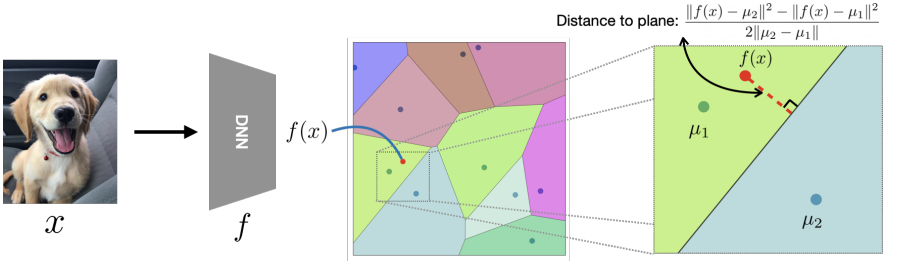


Figure 2: **Illustration of Proposition 1**. For a classifier $f$ trained with a *Clustering Loss*, an instance $x$ is classified by assigning it to the class of the closest cluster to its feature representation $f(x)$. The resulting decision boundaries form a Voronoi diagram in feature space. As a consequence, the robustness radius in Proposition 1 is proportional to the distance to the decision boundary separating the two closest clusters to $f(x)$.

# 3 From Robustness to Clustering Loss

Recent work has shown that adversarially-trained DNNs, while robust, also tend to learn more semantically-aligned features [10, 43]. Inspired by these findings, we are interested in studying the converse implication, *i.e.* whether DNNs trained to learn such features enjoy robustness properties. To this end, we start by studying the robustness of a common family of classifiers used in deep metric learning [17, 36], namely classifiers that are based on clustering semantically-similar inputs.

## 3.1 Robustness

**Clustering-based classifiers.** Consider a training set consisting of input-label pairs $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$, where $x_i \in \mathbb{R}^n$ belongs to one of $L$ classes, and a parameterized function $f_\theta : \mathbb{R}^n \to \mathbb{R}^d$, which can be a DNN. A clustering-based classifier learns parameters $\theta$ such that $f_\theta$ clusters semantically-similar inputs $x_i$ (inputs with similar labels $y_i$) in feature space $\mathbb{R}^d$. That is, $f_\theta$ clusters each of the $L$ classes into $K$ different clusters (where $K$ may vary across classes). Hence, an input $x_i$ is assigned a label $c$, if and only if, $f_\theta(x_i)$ is closest, under some notion of distance, to one of the $K$ clusters representing class $c$. To analyze the robustness of such classifiers, and without loss of generality, we consider a binary classification problem, where inputs belong to one of two classes, $\mathcal{C}_1$ or $\mathcal{C}_2$, and each class is represented with a single cluster center, *i.e.* $L = 2$ and $K = 1$. Let the cluster centers of $\mathcal{C}_1$ and $\mathcal{C}_2$ be $\mu_1$ and $\mu_2$, respectively, in $\mathbb{R}^d$. Thus, $x_i$ is classified as $\mathcal{C}_1$, if and only if, $\|f_\theta(x_i) - \mu_1\| < \|f_\theta(x_i) - \mu_2\|$, and as $\mathcal{C}_2$ otherwise. Throughout this paper, we assume that $f_\theta$ is $\mathcal{L}_f$-Lipschitz continuous [7], *i.e.* $\|f_\theta(x) - f_\theta(y)\| \leq \mathcal{L}_f \|x - y\| \; \forall x, y$, where $\|\cdot\|$ denotes the $\ell_2$ norm.

We are interested in the maximum norm of an input perturbation $\delta$ such that the clustering-based binary classifier assigns the same class to both $x$ and $(x+\delta)$. The following proposition provides a bound on such a $\delta$, denoted as the robustness radius.

**Proposition 1** *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 $f_\theta$. 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* $\delta$ *that satisfy:*

$$\|\delta\| < \frac{\|f_\theta(x)-\mu_2\|^2 - \|f_\theta(x)-\mu_1\|^2}{2\mathcal{L}_f\|\mu_2-\mu_1\|}. \tag{1}$$

*Proof Sketch.* It suffices to observe that the clustering-based classifier is equivalent to a linear classifier, operating in representation space, defined by the hyperplane $(\mu_1-\mu_2)^\top(f_\theta(x)-(\mu_2+\mu_1)/2)=0$. The result is deduced from the Cauchy-Schwarz inequality and the Lipschitz continuity property of $f_\theta$, where the bound is proportional to the $\ell_2$ distance to the hyperplane, as illustrated in Figure 2.

**Generalization to the Multi-Class Multi-Cluster Setting.** We first consider the multi-class single-cluster case, *i.e.* $L \geq 2$, $K=1$, where each class is represented by a single cluster center $\mu_i$, as depicted in Figure 2. Analyzing the robustness around an input $x$ in this case is equivalent to analyzing the previously discussed binary classification case with respect to the two closest cluster centers *i.e.* $\mu_1=\mu_{i^*}=\arg\min_{i\in\{1,\dots,L\}}\|f_\theta(x)-\mu_i\|$ and $\mu_2=\mu_{j^*}=\arg\min_{i\in\{1,\dots,L\}/\{i^*\}}\|f_\theta(x)-\mu_i\|$. We leave the rest of the details for the **appendix**.

## 3.2 Clustering Loss as a Robustness Regularizer

Proposition 1 provides a tight robustness radius for each input. To attain both accurate and robust models, one can train DNNs to achieve accuracy, while simultaneously maximizing the robustness radius in Proposition 1 for every training input $x$. Several observations can be made about the robustness radius. First, it is inversely proportional to the DNN's Lipschitz constant $\mathcal{L}_f$, *i.e.* networks with smaller $\mathcal{L}_f$ tend to enjoy better robustness. This is consistent with previous work that exploited this observation to enhance network robustness [7]. In this paper, we focus on the term $\|f_\theta(x)-\mu_2\|^2-\|f_\theta(x)-\mu_1\|^2$, and on learning parameters $\theta$ to maximize it, *i.e.* to push features far from cluster centers of different classes ($\mu_2$) and to pull features closer to cluster centers of their class ($\mu_1$). As such, a general class of robustness-based clustering losses can be formulated as follows:

$$\mathcal{L}^{\text{Robust}}_{\text{Clustering}} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{H}\left(\mathcal{F}\left(f_\theta(x_i),\{\mu_{c_i,j}\}_{j=1}^K\right),\mathcal{G}\left(f_\theta(x_i),\{\mu_{v\neq c_i,j}\}_{j=1}^K\right)\right), \tag{2}$$

where $c_i=\mathcal{C}(x_i)$ is the class of $x_i$ and $\mu_{i,j}$ denotes the $j^{\text{th}}$ cluster of class $i$. The function $\mathcal{F}$ measures the separation between the feature representation of $x_i$, *i.e.* $f_\theta(x_i)$, and the cluster centers of its class. Similarly, $\mathcal{G}$ measures the separation between $f_\theta(x_i)$ and the cluster centers of all other classes. The function $\mathcal{H}$ combines the two measurements in an overall stable loss, so that minimization of the loss incites larger values for the numerator in Proposition 1. Note that iterative optimization of this loss requires updating $\theta$. Hence, after every update, cluster centers $\mu_{i,j}$ can be recomputed by any clustering algorithm, *e.g. K-means*. Moreover, many losses commonly used in the deep metric learning literature [28] conform with Equation (2) as special cases, one of which is the popular *Magnet Loss* [36], defined as:

$$\mathcal{L}^{\text{Magnet}}_{\text{Clustering}} = \frac{1}{N}\sum_{i=1}^{N}\left\{\alpha+\frac{1}{2\sigma^2}\|f_\theta(x_i)-\mu_{c_i,v^*}\|^2+\log\left(\sum_{j=1}^{K}\sum_{v\neq c}e^{-\frac{1}{2\sigma^2}\|f_\theta(x_i)-\mu_{v,j}\|^2}\right)\right\}_+ \tag{3}$$

where $\{x\}_+=\max(x,0)$, $\sigma^2=\frac{1}{N-1}\sum_{i=1}^{N}\|f(x_i)-\mu_{c_i,v^*}\|^2$, $\alpha\geq 0$, and $v^*=\text{argmin}_v\|f_\theta(x_i)-\mu_{c_i,v}\|$. Note that the *Magnet Loss* is a special case of the previously formulated general

Robust Clustering loss that incentivizes the increase in the numerator of Proposition 1. That is to say, the feature representations $f_\theta(x_i)\ \forall i$ are pulled closer to clusters representing the correct class and pushed away from clusters of other classes. While the *Magnet Loss* was introduced to address performance issues in metric learning algorithms, our objective of learning more semantically-aligned features and our subsequent analysis of Proposition 1 suggest that this loss *inherently* encourages robustness.

Regarding inference, DNNs trained with *Magnet Loss* predict the class of a test input by computing a soft probability over the features produced by $f_\theta$, as follows:

$$Pr(\mathcal{C}(x_i)=c)=p_c(f_\theta(x_i))=\frac{\sum_{j=1}^{K} e^{-\frac{1}{2\sigma^2}\|f_\theta(x_i)-\mu_{c,j}\|^2}}{\sum_{j=1}^{K}\sum_{v=1}^{L} e^{-\frac{1}{2\sigma^2}\|f_\theta(x_i)-\mu_{v,j}\|^2}}. \tag{4}$$

Hence, $x_i$ is assigned to class $\text{argmax}_c\, p_c(f_\theta(x_i))$. We refer the reader to [36] for more details.

### 3.3 ClusTR: Clustering Training for Robustness

Our theoretical study finds an intrinsic connection between clustering and robustness: clustering-based classifiers intrinsically possess a robustness radius. As such, optimizing a loss designed for clustering tends to maximize this robustness radius. We also observe that a *Clustering Loss* such as Equation (2), which is designed to induce robustness according to Proposition 1, can be reduced to the *Magnet Loss* of Equation (3) as a special case. Based on these observations, we propose Clustering Training for Robustness (ClusTR): a simple and theoretically-motivated framework for inducing robustness during DNN training without the need to generate adversaries. ClusTR exploits our theoretical findings by combining a *Clustering Loss* with simple DNN-training techniques.

For the *Clustering Loss*, ClusTR incorporates the well-studied *Magnet Loss* to induce semantic clustering of instances in feature space. Although effective in its task, this loss suffers from slow convergence [36]. ClusTR mitigates this issue by introducing a simple *warm start* initialization. For a given model and dataset, ClusTR first conducts nominal training, *i.e.* standard Cross Entropy training, until reasonable performance is achieved. Then, it removes the last linear layer and fine-tunes the resulting DNN by applying the *Magnet Loss* on the output of the penultimate layer. The *Magnet Loss* in ClusTR aims at optimizing the robustness radius of Proposition 1, while using a warm start initialization to increase convergence speed without hindering test set accuracy. In this work, we choose the *Magnet Loss* to be the *Clustering Loss* in ClusTR. However, we remark that Proposition 1 is agnostic to this choice, so we expect our results to extend to other choices of a *Clustering Loss*.

## 4 Experiments

In this section, we conduct several experiments on synthetic and real datasets to validate the idea behind ClusTR. Specifically, we study (a) the effect of a *warm start* on convergence speed and robustness, (b) how ClusTR-trained DNNs compare to their adversarially-trained counterparts, and (c) how ClusTR can be equipped with a quick version of adversarial training to further enhance robustness.

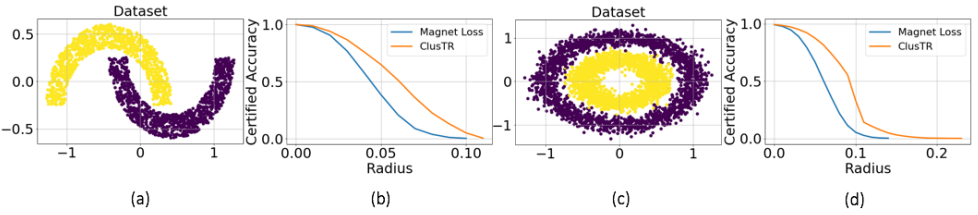


Figure 3: **Effect of *warm start* on certified accuracy**. Figures (a)-(c) show the synthetic datasets, while Figures (b)-(d) show the effect of *warm start* in ClusTR on certified accuracy. In both datasets, *warm start* induces a larger robustness radius than random initialization.

## 4.1 Effect of Warm Start Initialization in ClusTR

**Convergence.** We assess the training convergence and the overall test accuracy performance for our proposed ClusTR-training of ResNet18 on CIFAR10 and SVHN. In CIFAR10, we observe that training without warm start (*i.e. Magnet Loss* only) requires 106 minutes to fully train, while introducing the warm start reduces the required training time to 83 minutes.

**Robustness.** We study the effect of the warm start initialization on robustness by conducting controlled synthetic experiments and computing *exact* robustness radii by computing a tight estimate of the Lipschitz constant. We train a 3-layered neural network with 20 hidden units on the synthetic binary classification datasets depicted in Figures 3(a) and (c). On both datasets, we train (1) *Magnet Loss* with random initialization and (2) ClusTR. For simplicity, each class is represented with a single cluster, *i.e.* $K = 1$. Upon convergence, both models achieve 99% accuracy. Given model predictions, we compute the robustness radius for each instance and report certified accuracy under various radii $r$ in Figures 3(b) and (d). This is in line with common practice in the network certification literature [8]. Note that certified accuracy at radius $r$ is defined as the percentage of instances that are both correctly classified and have a robustness radius larger than $r$, as given by Proposition 1. We find that the ClusTR-trained DNNs, while accurate, also enjoy a larger robustness radius than DNNs trained with *Magnet Loss* without the warm start.

## 4.2 ClusTR Robustness against PGD

**Setup and Implementation Details.** In this section, we conduct experiments with ResNet18 on the CIFAR10, CIFAR100, and SVHN datasets. We train models using our proposed ClusTR framework. Specifically, we first conduct nominal training until we get a reasonable performance[1]. We then remove the last linear layer and fine-tune the network by applying the *Magnet Loss* on the output feature of the resulting DNN. Fine-tuning is done for 30 epochs on CIFAR10 and SVHN, and 60 epochs on CIFAR100. Following [36], we use $k$-means$_{++}$ [1] to update cluster centers after each training epoch. To assess model robustness, we follow prior work and perform projected gradient descent (PGD) [26] attacks with $\varepsilon$-$\ell_\infty$-bounded perturbations that take the following form:

$$x^{k+1} = \prod_{S} \left( x^k + \eta \, \text{sign} \left( \nabla_{x^k} \mathcal{L}_{\text{ce}}(p(f_\theta(x^k)), y) \right) \right), \tag{5}$$

[1] Models with test accuracies of 90%, 75%, 90% on CIFAR10, CIFAR100 and SVHN, respectively.

Table 1: **Adversarial accuracy comparison on CIFAR10 and SVHN**. We compare ClusTR and ClusTR+QTRADES against Magnet Loss, Free Adversarial Training (Free AT), AT with ImageNet pre-training, TRADES, and QTRADES under $\varepsilon = 8/255$ PGD attacks. ClusTR+QTRADES outperforms the adversarially-trained models by a large margin.

| | CIFAR10 | | | SVHN | | |
|---|---|---|---|---|---|---|
| | Natural | $PGD^{20}$ | $PGD^{100}$ | Natural | $PGD^{20}$ | $PGD^{100}$ |
| Nominal Training | **95.01** | 0.00 | 0.00 | **98.38** | 0.00 | 0.00 |
| Free AT [40] | 85.96 | 46.33 | 46.19 | 86.98 | 46.52 | 46.06 |
| AT + Pre-Training [15] | 87.30 | 57.40 | 57.20 | 85.12 | 47.18 | 46.72 |
| TRADES [49] | 84.92 | 56.61 | 56.43 | 91.63 | 57.45 | 55.28 |
| Magnet Loss [36] | 83.14 | 23.71 | 22.54 | 91.95 | 40.73 | 38.59 |
| ClusTR | 87.34 | 49.04 | 47.76 | 94.28 | 50.78 | 50.77 |
| QTRADES | 81.07 | 44.18 | 43.42 | 86.36 | 43.05 | 42.24 |
| ClusTR + QTRADES | 91.03 | **74.44** | **74.04** | 95.06 | **84.76** | **84.75** |

where $\prod_{\mathcal{S}}(x+\delta)$ denotes the projection of the perturbed input onto the set $\mathcal{S} = \{(x+\delta) \in [0,1]^n, \|\delta\|_\infty \leq \varepsilon\}$, $p(f_\theta(x^k))$ is the probability prediction vector computed through Equation (4), and $\mathcal{L}_{\text{ce}}$ is the Cross Entropy loss. In all experiments, we perform PGD attacks with 10 random restarts around each input for 20 and 100 iterations, denoted as $PGD^{20}$ and $PGD^{100}$, respectively. Following common practice in the literature [40, 46], we set the PGD step size to $\eta = 2/255$. We report the attacks with an attack strength of $\varepsilon = 8/255$, and leave experiments with other choices of $\varepsilon$ for the **appendix**.

**Experiments on CIFAR10 and SVHN.** We evaluate the PGD robustness of nominal training (as baseline), the Magnet Loss (*i.e.* ClusTR without warm start), and ClusTR, and we compare against several approaches that provide PGD robustness in this experimental setup, namely *Free adversarial training* (Free AT) [40] with its reported best setting of 8 minibatch-replays that outperforms vanilla adversarial training [26], *Adversarial Training with ImageNet pre-training* (AT + PreTraining) that leverages external data to improve robustness, and *TRADES* [49]. Note that all the robustness methods in this comparison employ various forms of adversarial training. We report both natural accuracy, *i.e.* test set accuracy on clean images, and PGD test accuracy. Table 1 reports these results. First, we observe that training with Magnet Loss *only* on clean images results in substantial gains in robustness compared to nominal training. In fact, this choice of loss function increases $PGD^{20}$ accuracy from 0% to 23.71%, while natural accuracy drops from 95.01% to 83.14%. This result constitutes empirical evidence of the theoretical robustness properties we presented for clustering-based classifiers. Furthermore, training with ClusTR consistently outperforms Free AT in both natural and PGD accuracy for both CIFAR10 and SVHN. Specifically, ClusTR outperforms Free AT in $PGD^{20}$ accuracy by 3% and 4% on CIFAR10 and SVHN, respectively, even though the former *only* trains with clean images. We note that ClusTR's robustness gains over adversarial training are not accompanied with lower natural accuracy. In fact, the natural accuracy of ClusTR is 1% more in CIFAR10 and 7% more in SVHN. These results show that the design of ClusTR inherently provides robustness properties without introducing adversaries during training. We complement this finding by studying the following question: Can equipping ClusTR with some form of adversarial training provide even larger PGD robustness gains? We equip ClusTR with a TRADES loss term, where the total loss becomes:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}^{\text{Magnet}}_{\text{Clustering}} + \lambda \mathcal{L}_{\text{ce}}(p(f_\theta(x_{\text{adv}})), p(f_\theta(x))). \tag{6}$$

Table 2: **Adversarial accuracy on CIFAR100**. We compare ClusTR+QTRADES against Free AT, AT+Pre-Training, and TRADES under $\varepsilon = 8/255$ PGD attacks. Our proposed ClusTR+QTRADES framework surpasses all competition by a large margin.

| | CIFAR100 | | |
|---|---|---|---|
| | Natural | $PGD^{20}$ | $PGD^{100}$ |
| Nominal Training | **78.84** | 0.00 | 0.00 |
| Free AT [40] | 62.13 | 25.88 | 25.58 |
| AT+Pre-Training [15] | 59.23 | 34.22 | 33.91 |
| TRADES [49] | 55.36 | 28.11 | 27.96 |
| ClusTR+QTRADES | 69.25 | **52.47** | **52.40** |

Note that the Cross Entropy-based TRADES formulation [49] is similar to Equation (6), but with the first term replaced with $\mathcal{L}_{\text{ce}}(p(f_\theta(x)), y)$, where $p(f_\theta(x))$ is the output logits of the last linear layer and $y$ is the true label. In order to keep the framework simple and computationally efficient, we compute a *quick* estimate of the adversary $x_{\text{adv}}$ in Equation (6). Namely, we start from a random uniform initialization and perform a *single* PGD step as opposed to TRADES' multiple iterations. We refer to this setup as QTRADES[2]. Formally, for an input $x$, we construct an adversary by perturbing $x$ with uniform noise, *i.e.* $x' = x + \mathcal{U}[-\varepsilon, \varepsilon]$, and then generate $x_{\text{adv}}$ by:

$$x_{\text{adv}} = \prod_{\mathcal{S}} \left( x' + \eta \operatorname{sign}\left( \nabla_{x'} \mathcal{L}_{\text{ce}}(p(f_\theta(x')), p(f_\theta(x))) \right) \right).$$

We report results for this experiments in Table 1. While QTRADES alone only achieves slightly lower natural accuracy and adversarial robustness (when compared to Free AT), our results show that equipping ClusTR with QTRADES enhances PGD robustness results on both datasets, outperforming all other methods. In particular, we observe that ClusTR+QTRADES achieves the highest natural accuracy among all methods with 91.03% and 95.06%, on CIFAR10 and SVHN, respectively, thus improving upon the best competitor by 4% on both datasets. Also, ClusTR+QTRADES surpasses other baselines by sizable margins: 16.84% and 29.47% under strong PGD attacks on CIFAR10 and SVHN, respectively.

**Experiments on CIFAR100.** We extend our analysis of ClusTR+QTRADES to CIFAR100, and assess PGD robustness with $\varepsilon = 8/255$ attacks. We report the results of this setup in Table 2, which shows that ClusTR+QTRADES outperforms the strongest competitor by 18.25% under strong PGD attacks. We note that these large gains in PGD robustness also come with a substantial 7% increase in natural accuracy. For CIFAR100, the total number of clusters is 100 (classes) $\times$ 2 (clusters per class) = 200. Following how [36] tackles the large-cluster-number regime, in this case we compute predictions for ClusTR+QTRADES without considering all clusters, as in Equation (4), but only the $D$ nearest clusters. While we take $D = 20$ in this experiment, we find that the choice of $D$ around this value has a marginal impact on robustness. We leave an ablation of $D$ for the **appendix**.

**Adaptive Attacks**. While going against the current paradigm in the network robustness literature, it has been argued that common attacks may be insufficient to demonstrate network robustness. Specifically, recent work shows that many defenses can be broken with carefully-crafted attacks [2], now dubbed *adaptive attacks*, tailored to break the underlying defense [42]. Following this principle, we construct a potential powerful attack tailored to our trained

[2]The rest of the implementation details of QTRADES are left for the **appendix**.

networks. Namely, we construct adversaries that maximize the $\mathcal{L}^{\text{Magnet}}_{\text{Clustering}}$ loss, as opposed to the standard Cross Entropy loss in the PGD formulation. Similar to previous experiments, the attacks are performed with 10 random restarts for 100 iterations and $\varepsilon = 8/255$. Note that this attack precisely targets the objective, with which our models are trained, thus, the attack is expected to be stronger. Indeed, running this adaptive attack lowers the robustness accuracy from 74.04% to 66.52% on CIFAR10, and from 84.75% to 78.79% on SVHN. Despite this drop, our ClusTR+QTRADES approach still outperforms other methods by substantial margins. It is essential to note here that this drop in robustness is considered to be rather marginal, as other defenses, when subjected to such tailored attacks, have their robustness drop close to 0, or at least to lower-than-baseline robust models [2, 42].

It is worthwhile to mention that our choice of QTRADES, out of the many adversarial training schemes with which ClusTR can be equipped, is motivated by *(i)* the theoretical support behind TRADES [49] and *(ii)* QTRADES' low computational cost. We also emphasize here that PGD robustness could possibly be improved further by incorporating another adversarial training technique with ClusTR instead of QTRADES. We leave the search for this optimal choice to future work.

### 4.3 Discussion

Evaluating adversarial robustness is a complex task with frequent methodological changes [5]. Empirically evaluating defense mechanisms requires selecting a threat model and an adversary that aims at exploiting such threat. In this work, we theoretically characterized an existing connection between semantics, achieved through clustering, and adversarial PGD robustness. Given our theoretical insights, we then set out to conduct proof-of-concept experiments to test our findings. For this purpose, we choose the threat model of $\ell_\infty$ attacks of norm smaller than a given $\varepsilon$, and the well-studied Cross Entropy-based PGD adversary to conduct attacks. Our experiments show that networks trained with our approach are notably robust against this adversary. These results provide empirical evidence that our theoretical findings correspond with practical applications. However, we abstain from claiming that our approach provides adversarial robustness in the most generic sense beyond PGD attacks: other attacks may be able to find vulnerabilities in our defense.

**Acknowledgments.** This work was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. OSR-CRG2019-4033.

## References

[1] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*, 2015.

[4] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2018.

[5] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

[6] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[7] 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)*, 2017.

[8] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning (ICML)*, 2019.

[9] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations. *arXiv:1906.00945*, 2019.

[10] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Learning perceptually-aligned representations via adversarial robustness. *arXiv:1906.00945*, 2019.

[11] Andrea Frome, Yoram Singer, Fei Sha, and Jitendra Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *International Conference on Computer Vision (ICCV)*, 2007.

[12] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

[13] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations (ICLR)*, 2014.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[15] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. *International Conference on Machine Learning (ICML)*, 2019.

[16] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 2012.

[17] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, 2015.

[18] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[19] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv:1803.06373*, 2018.

[20] Mahmut Kaya and H.s Bilge. Deep metric learning: A survey. *Symmetry*, 2019.

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. In *University of Toronto, Canada*, 2009.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

[24] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

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

[26] 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.

[27] Chengzhi Mao, Ziyuan Zhong, Junfeng Yang, Carl Vondrick, and Baishakhi Ray. Metric learning for adversarial robustness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[28] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE transactions on pattern analysis and machine intelligence*, 2013.

[29] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

[30] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

[31] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 2015.

[32] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[33] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2011.

[34] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. *International Conference on Learning Representations (ICLR)*, 2014.

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

[36] Oren Rippel, Manohar Paluri, Piotr Dollar, and Lubomir Bourdev. Metric learning with adaptive density discrimination. *International Conference on Learning Representations (ICLR)*, 2016.

[37] Shibani Santurkar, Andrew Ilyas, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Image synthesis with a single (robust) classifier. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[38] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[39] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[40] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[41] 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.

[42] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *arXiv:2002.08347*, 2020.

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

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[45] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research (JMLR)*, 2009.

[46] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations (ICLR)*, 2020.

[47] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[48] Eric P Xing, Michael I Jordan, Stuart J Russell, and Andrew Y Ng. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2003.

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

[50] Qiuyu Zhu, Pengju Zhang, and Xin Ye. A new loss function for cnn classifier based on pre-defined evenly-distributed class centroids, 2019.

# A Implementation Details

We describe the implementation details of ClusTR, along with details regarding QTRADES.

**Architecture.** We use a ResNet18 [14] modified to accept $32 \times 32$ input images. The size of the output of the network in the penultimate layer, *i.e.* the feature dimension, is set to 512 for all experiments.

**Optimization.** For the warm start stage of training ClusTR, we use the Adam optimizer [21] for 90 epochs with learning rate of $10^{-2}$ that is multiplied by $10^{-1}$ at epochs 30 and 60 with cross entropy loss. After that, we fine-tune the DNN with the Magnet Loss with a learning rate of $10^{-4}$ for another 30 epochs for CIFAR10 and 60 epochs for CIFAR100 and SVHN.

**Pre-processing.** Images are normalized by their channel-wise mean and standard deviation. For CIFAR10 and CIFAR100. We apply standard data augmentation of random $32 \times 32$ crops with a padding of 4. For SVHN, we do not employ any data augmentation.

**Magnet Loss.** Following Rippel *et al.* [36], we compute a stochastic approximation of the *Magnet Loss*. Hence, Magnet Loss training requires sampling neighborhoods of points in representation space, rather than independent samples. These neighborhoods are defined by a number of clusters and a number of samples per cluster. This sampling procedure does not guarantee that every instance will be sampled, nor that an instance shall be sampled only once. Therefore, we define an epoch as passing as many instances as there are available in the dataset, regardless if some instances were repeated or some instances were seen more than once. We use $K = 2$ as the number of clusters per class for our experiments. For sampling, we set the total number of sampled clusters to 12, and the number of samples per cluster to 20. Hence, the total amount of samples in each batch of each batch is $12 \times 20 = 240$. Cluster assignments are recomputed at the end of every epoch with the *K-means* clustering algorithm with the *K-means*$_{++}$ initialization. We run grid search for optimizing the $\alpha$ parameter in the Magnet Loss. We set $\alpha$ to 12.5 for ClusTR and ClusTR+QTRADES on CIFAR10; to 13 for ClusTR and to 10 for ClusTR+QTRADES on SVHN; to 8.5 for ClusTR+QTRADES on CIFAR100.

**QTRADES.** We initialize the adversary by adding uniform noise in $[-\varepsilon, \varepsilon]$ to the original instance, computing Cross Entropy between the original and adversarial instances and following one step of gradient ascent for Cross Entropy. The result of gradient ascent is always clipped so that the adversarial instances lies in image space, *i.e.* $[0,1]^n$. The total loss with which the network is trained is a weighted sum of the Clustering Loss and the Cross Entropy between the original and adversarial instances. We cross validate over the regularization term $\lambda$ balancing the two terms in Equation (6). We set $\lambda$ to 8 on CIFAR10, to 9.7 on SVHN, and to 2 on CIFAR100.

# B Additional Experiments

## B.1 Combining CE with Distance-Based Classifier

The robustness radius in Proposition 1 holds for any clustering-based classifier of features produced by a Lipschitz-continuous function $f_\theta$. Therefore, we start by addressing the following question: if robustness is the aim, can one replace the last layer of a nominally-trained DNN with a clustering-based classifier to achieve robustness? Addressing this question is essential to establish the necessity of enforcing clustering during training, *i.e.* training with ClusTR. To answer this question, we study a nominally-trained ResNet18 on CIFAR10,

Table 3: **Performance of ClusTR+QTRADES on CIFAR10, CIFAR100 and SVHN**. We report the PGD Accuracy of ClusTR+QTRADES on more $\varepsilon$ Values where we show that the robustness of the resultant model is agnostic from the choice of $\varepsilon$.

| $\varepsilon$ | CIFAR10 | | SVHN | | CIFAR100 | |
|---|---|---|---|---|---|---|
| | $PGD^{20}$ | $PGD^{100}$ | $PGD^{20}$ | $PGD^{100}$ | $PGD^{20}$ | $PGD^{100}$ |
| $2/255$ | 81.99 | 81.54 | 87.48 | 87.47 | 60.15 | 59.77 |
| $16/255$ | 57.67 | 57.05 | 80.04 | 80.00 | 33.32 | 33.25 |
| $25.5/255$ | 35.88 | 34.98 | 71.56 | 71.45 | 17.76 | 17.65 |

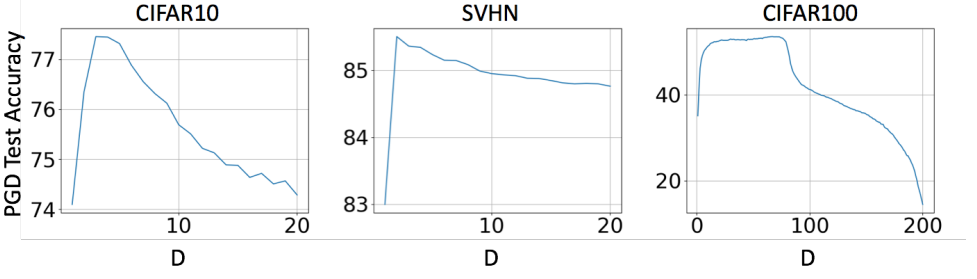


Figure 4: **Effect of $D$ on $8/255-PGD^{20}$ Test Accuracy.** Note that with $D = 1$, *i.e.* the assumption in our theoretical analysis, our methods outperforms the state-of-the-art. Moreover, it can be seen that considering only about 20% of the total number of clusters yields the best performance.

which achieves an accuracy of 95.0%. We observe that directly applying *K-means* on the representations of the penultimate layer, and performing classification according to Equation (4) achieves an accuracy of 21.6%, *i.e.* a performance drop of over 70%. As adversaries will aim at changing the classifier's predictions, the highest adversarial accuracy that this classifier can attain is upper bounded by 21.6%. This result demonstrates that features learnt through nominal training are not spatially configured for clustering-based classification. Hence, this result establishes that exploiting the benefits of clustering-based classification requires to explicitly enforce clustering during DNN training.

## B.2 Results of PGD Attacks with Other $\varepsilon$ Values.

Table 3 reports the adversarial accuracies ClusTR + QTRADES under PGD attacks with $\varepsilon \in \{2/255, 16/255, 0.1\}$ since we reported the results and comparisons for $\varepsilon = 8/255$ in the main patper. Note that the robustness of our model is not limited to a specific value of $\varepsilon$.

## B.3 Ablation on $D$

ClusTR predicts the class of an input as a soft nearest cluster through Equation (4). The probabilities can also be computed by only considering the $D$ nearest clusters, as reported in the Experiments Section. Next, we report the effect of varying $D$ in terms of the natural and adversarial accuracies.

Figure 4 depicts the behavior of clean and adversarial accuracies with varying $D$ on CIFAR10. We observe that the effect of varying $D$ on both CIFAR10 and SVHN is negligible ($\sim 3\%$). The best PGD accuracy for both CIFAR10 and SVHN under the strong $^{8}/_{255}-\text{PGD}^{20}$ attack was 77.04% and 85.33%, respectively (corresponding to $D = 4$). On the other hand, this effect seems to be stronger on CIFAR100. It is worthwhile to mention that more than 50% of the choices of $D$ yields better robustness than the state of the art. Moreover, with $D = 1$ which is exact setup of our theoretical result in Proposition 1, ClusTR+QTRADES surpasses the state of the art on all of the datasets by a significant margin. Finally, the best $^{8}/_{255}-\text{PGD}^{20}$ accuracy on CIFAR100 is 53.25% with $D = 60$.